Elliptical4VR: An Interactive Exergame Authoring Tool for Personalized Elliptical Workout Experience in VR

Wanwan Li
University of South Florida
Tampa, Florida, USA

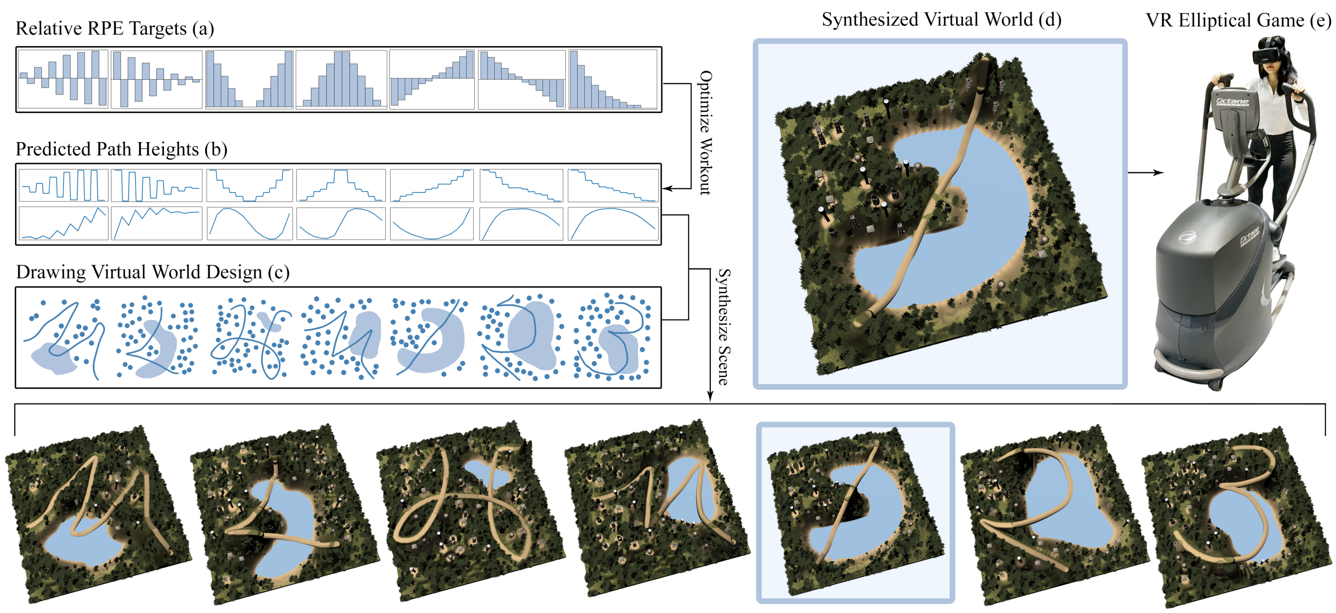

Figure 1: Given relative Rate of Perceived Exertion (RPE) scales as targets (a), our approach automatically optimizes the elliptical workout plan (e.g. resistance level sequences) and predicts the navigation path elevations (b). After employing the user's simple free-hand sketch-based drawing designs of the virtual environments (c), our approach procedurally generates virtual environments shown in the bottom part of this figure. As shown in (d) and (e), to give such optimized workout plans, we implemented the VR exergames by mapping the players' motion on an elliptical machine to the navigation in a virtual world.

ABSTRACT

In this paper, we propose Elliptical4VR, an interactive exergame authoring tool that synthesizes virtual environments and training programs for personalized elliptical workout experiences in Virtual Reality (VR) by exploring the possibility to combine functionality, aesthetics, and human activities as multi-dimensional interactions between humans, computers, and elliptical machines. Considering several key features of elliptical training such as Relative Rate of Perceived Exertion (Relative RPE) scales, target energy expenditure, total walking distance, etc., we optimize the elliptical workout profile to deliver expected training effects while at the same time considering the aesthetics of the virtual environment and the immersiveness delivered to the players by providing with the compatibility between the visual experience and haptic feedback during the gym activities enhanced by VR. In the end, we validate the effectiveness of our approach through a series of numerical experiments and preliminary user studies.

MLHMI '23, March 24–26, 2023, Singapore

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Computer Graphics**; • **Interactive Systems**; • **Virtual Reality**;

KEYWORDS

Exertion Game Design, Game Authoring Tool, Sketch-basd Interface

ACM Reference Format:

Wanwan Li. 2023. Elliptical4VR: An Interactive Exergame Authoring Tool for Personalized Elliptical Workout Experience in VR. In *Proceedings of International Conference on Machine Learning and Human-Computer Interaction (MLHMI '23)*. ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

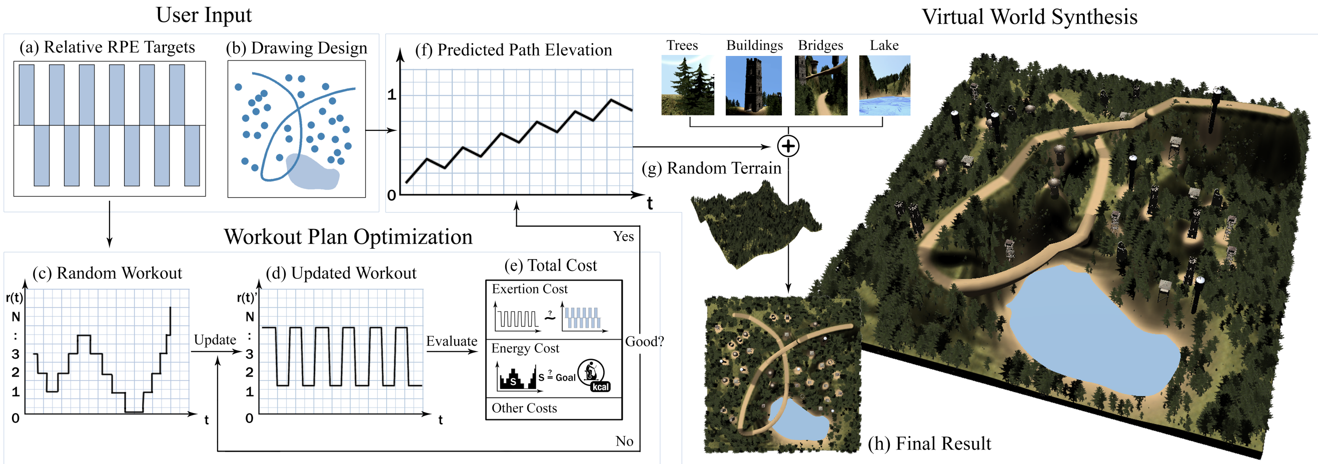

Figure 2: Overview of our approach.

1 INTRODUCTION

As human-computer interaction-based computational design technologies are gaining popularity in recent decades, more tentative explorations are applied to combining functionality, aesthetics, and human activities through multi-dimensional interactions between humans and computers. According to the standard criteria that are commonly adopted in the fitness centers, body training on the elliptical machine mainly considers four parameters: time intervals, resistance levels, incline levels, and running speed. In general, time intervals and inclines are regularly set as constant numbers for specific training purposes. However, more interesting observations are conducted around the concepts of the relative Rate of Perceived Exertion (RPE) scale, running speed (50-150) in the unit of Strides Per Minute (SPM), and the resistance levels (1-15). Motivated by this observation, our project focuses on how to fill the gap between the virtual synthesized world, the real gym activities of elliptical workouts, and the aesthetic considerations from the exergame designing perspective so as to make it not only possible for the player to feel the realistic physical feedback from the elliptical machine while navigating our synthesized virtual scenes, but also possible for the player to navigate the virtual world designed by themselves.Our proposed interface called Elliptical4VR achieves such a goal by combining sketch-based interface, virtual world procedural modeling, and stochastic optimization-based computational design.

There are lots of related works that are focusing on building virtual scenes using sketch-based interfaces. For example, in 2006, Zeleznik et al. [18] proposed an interface for sketching 3D scenes called SKETCH. In 2013, Xu et al. [17] proposed a sketch-based interface called Sketch2Scene for sketch-based co-retrieval and co-placement of 3D models. Nie et al. [12] proposed a sketch-based interface called SHREC'18 Track for 2D scene sketch-based 3D scene retrieval. In 2020, Gao et al. [3] proposed an image generation interface called Sketchycoco which can generate image scenes from freehand scene sketches. In 2021, Ribeiro et al. [14] proposed an unified model for scene search and synthesis from sketch called Scene Designer. In 2022, Ham et al. [5] proposed Cogs which is a sketch-based interface for controllable generation and search from sketch and style. On other hand, there are lots of related works that are focusing on building virtual scenes for exergame experience such as the Walking-In-Place (WIP) techniques and Immersive Virtual Environments (IVEs) techniques. In 2008, Feasel et al. [2] proposed an interface called LLCM-WIP for low-latency and continuous-motion walking-in-place. In 2009, Kulkarni et al. [7] developed a wind display device for locomotion interface in a virtual environment. In 2010, Steinicke [15] explored the gradual transitions and their effects on presence and distance estimation. In 2014, Nilsson et al. [13] established the range of perceptually natural visual walking speeds for virtual walking-in-place locomotion. In 2015, Kulkarni et al. [8] proposed a full body steerable wind display for a locomotion interface. In 2018, Ludwig et al. [11] studied the The influence of visual flow and perceptual load on locomotion speed. Tran et al. [16] studied the perceived changes in moving speed through a comparison between directly and indirectly powering the locomotion in virtual environments. In 2020, Li et al. [10] proposed a novel approach for exertion-aware path generation. In 2023, Li et al. [9] proposed a novel approach for synthesizing terrains for treadmill VR exergames. However, none of these mentioned works have shown efforts in combining sketch-based interfaces for elliptical exergame designs. Therefore, contributions of our work include:

- We implement an easy-to-use drag-and-drop interface for the game designer to specify the relative Rate of Perceived Exertion (RPE) scales and a low-learning-curve sketch-based interface that help the game designers manually, interactively, or randomly generate virtual scenes.
- We propose a novel approach for optimizing the resistance levels and speed levels for the elliptical workout plan through which the players can achieve their exercise goals.
- We propose a mathematical model, which is validated through a preliminary user study, to connect elliptical device's haptic feedback with the virtual world generated through our proposed interface via the motion simulation in VR.

2 OVERVIEW

As shown in Figure 2, our interactive framework takes input from the user-specified personalized workout settings including relative RPE score (a) complied with some other exertion targets such as overall energy expenditure, our approach automatically optimization a workout plan that satisfies such user-specified exercise training goal. During the workout plan optimization process, we initialize a workout plan (c) and randomly update it (d) according to the total cost function (e). If the total cost is high, we repeat updating the workout plan with the Monte Carlo Markov Chain process; Otherwise, we calculate the predicted path elevation (f) according to the optimized workout plan. With such path elevation function, we deform arbitrary randomly generated terrain heightmap (g) according to the user's creative virtual world design (b) created via an easy-to-use sketching canvas tool we developed to help the user easily sketch, click, or randomly generate a virtual world design with paths, landmarks, and lakes. (h) shows the final result of the virtual world synthesized with our approach that can both deliver the player's expected exertion goals while complying with the user's artistic design of the visual scene.

3 TECHNICAL APPROACH

3.1 Elliptical Workout Plan

In overall, we optimize a workout plan for the personalized elliptical training according to three criteria: (1) the exercise experience matches the relative RPE score manually designers by the exergame designer. (2) the overall energy expenditure matches the target energy expenditure and (3) the total running distance matches the target running distance. We use three different cost functions to represent encode these three criteria, they are relative RPE cost C_r, energy expenditure cost C_e, and total distance cost C_d respectively. There corresponding weights are w_r, w_e, and w_d , then the total cost function C_total is defined as:

$$C_\text{total}(R,S) = w_\text{r}C_\text{r}(R,S) + w_\text{e}C_\text{e}(R,S) + w_\text{d}C_\text{d}(R,S) \tag{1}$$

where list R is the resistance levels (1-15) and S is the list of running speeds (50-150 SPM). Both list of R and S are the parameters of workout plan that are optimized during the optimization process.

Relative RPE Cost. As the Rate of Perceived Exertion (RPE) scales measure physical activity intensity level, RPE generally tells the perceived exertion is how hard the exergame player's body is working. It is based on the physical sensations a person experiences during physical activity, including increased heart rate, increased respiration or breathing rate, increased sweating, and muscle fatigue. Intuitively, a higher resistance level or higher speed will result in a higher RPE score. Therefore, the relative RPE score is proportional to resistance level and speed. Without loss of generality, assume there are N workout intervals corresponding to the N rows of the workout table. Let $R = \{r_i\}$ and $S = \{s_i\}$ denotes the resistance levels and speeds for i^th interval respectively, where $i = 1, 2, ..., N$. Therefore, given the exergame designers' input of relative RPE scales targets $Z = \{\zeta_i\}$, we define the relative RPE cost as the accumulative errors between K, R, and S according to the relative RPE score's defination $RPE(R,S) = RS$, we have:

$$C_\text{r}(R,S) = 1 - \exp\left(-\frac{1}{\sigma_\text{r}}\frac{1}{N}\sum_{i=1}^{N}\left\|\tilde{\zeta_i} - \tilde{r_i s_i}\right\|\right), \tag{2}$$

where we empirically set $\sigma_\text{r} = 0.25$ and define the normalization operator $\tilde{x_i}$, where $\forall x_i \in X$ and $\tilde{x_i}$ is calculated as:

$$\tilde{x_i} = \begin{cases} \frac{x_i - \min(X)}{\max(X) - \min(X)} & \max(X) \neq \min(X) \\ 0 & \max(X) = \min(X) \end{cases} \tag{3}$$

Total Energy Cost. As another important aspect of personalized elliptical training, the target energy expenditure is important for the exergame players to achieve a specific exercise effect. Thereof, we measure the difference between the energy expenditure of a proposed workout plan and the expected energy expenditure target ρ_e set by the exergame level designer. We propose the energy expenditure cost function which is calculated as:

$$C_\text{e}(R,S) = 1 - \exp\left(-\frac{1}{\sigma_\text{e}}\left\|\kappa\sum_{i=1}^{N}\psi(r_i, s_i) - \rho_\text{e}\right\|\right), \tag{4}$$

where we empirically set $\sigma_\text{e} = 0.25$, set the calory coefficient $\kappa = 0.0875W$ and the average weight W is set to 90 kg by default, and define the MET function $\psi(r_i, s_i)$ which is calculated as:

$$\psi(r_i, s_i) = \frac{1}{50}(4.4 + 0.02r_i^2)s_i \tag{5}$$

Total Distance Cost. Given the same amount of training time, adjusting the total distance target ρ_d will affect the speed of the elliptical training process. Therefore, as another parameter to consider, we minimize the difference between the total distance demanded by the proposed workout plan and the total distance target as total distance cost function which is calculated as:

$$C_\text{d}(R,S) = 1 - \exp\left(-\frac{1}{\sigma_\text{d}}\left\|\lambda\sum_{i=1}^{N}s_i\Delta t - \rho_\text{d}\right\|\right), \tag{6}$$

where we empirically set $\sigma_\text{d} = 0.25$, set the time duration for each traning interval $\Delta t = 5$ (mins), and stride length $\lambda = 0.5$ (meter).

3.2 Optimization Process

In order to quickly find out an optimized workout plan whose resistance levels R and speeds S to minimize the total costs, we need to sample solutions within the solution space efficiently. However, a tiny solution search step to search each element in the array is necessary for guaranteeing an accurate solution. In this case, both R and S have integer values, therefore, the minimum search step for both R and S is one. However, as the time intervals of the workout plan increase, the solution space's dimension increases exponentially. Therefore, we are trying to solve this problem by reducing its time complexity using a stochastic search algorithm to find a local optimum as the approximate solution. Duirng the initialization stage, the resistance levels are set as $R_0 = \{i|i = 1, 2, ..., N\}$ and speeds are set as constants $S_0 = \{100, 100, 100, ...\}$. Then, we formulate the optimization problem as an integer programming problem by employing the Markov chain Monte Carlo method [4] to search for a solution that minimizes the total cost function. Given any randomly updated solution R and S, a new solution R' and S' are randomly proposed within the solution space through the following step, i.e. randomly pick an element in R or S and increase or decrease that element by one. Our approach uses the Metropolis criterion of simulated annealing technique [1, 6] to determine the acceptance probability $\Pr(R', S'|R, S)$ for accepting R' and S' as:

$$\Pr(R', S'|R, S) = \min(1, \frac{f(R', S')}{f(R, S)}), \tag{7}$$

where $f(R,S)$ is a Boltzmann-like objective function encoding the total cost function which calculated from:

$$f(R,S) = \exp(-\frac{1}{t}C_\text{total}(R,S)), \tag{8}$$

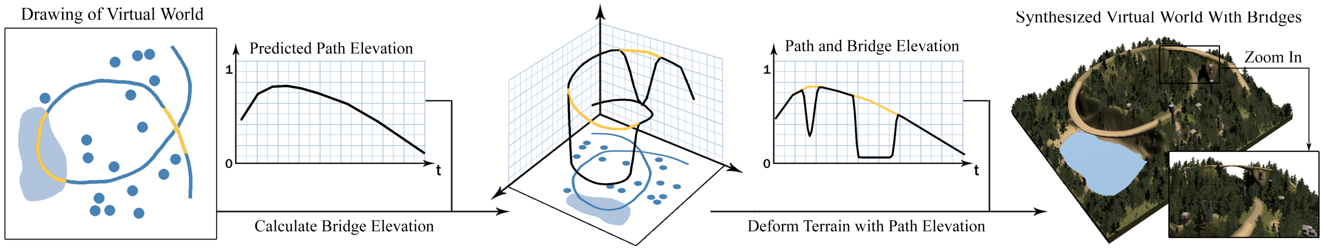

Figure 3: Bridge Structure Generation.

where t is the temperature parameter of simulated annealing, which decreases gradually throughout the optimization. As the temperature t decreases over iterations, the optimizer becomes less aggressive and more greedy. By the end, as the temperature drops to a low value near zero, the optimizer tends to accept better solutions only. We empirically use temperature $t = 1.0$ at the beginning of the optimization and decrease it by 0.2 every 100 iterations until it reaches zero. The optimization process is terminated if the total cost change is smaller than 3% over the past 50 iterations.

3.3 Virtual Environment Synthesis

Path Elevation. Given the optimized elliptical workout plan as resistance levels $R = \{r_i\}$ and speeds $S = \{s_i\}$ where $i = 1, 2, ..., N$ and given the resistance force table $\Phi = \{\phi_i\}$ where empirically $\Phi = \{0, 10, 14, 18, 25, 40, 70, 80, 95, 115, 124, 150, 175, 200, 240, 255\}$ (Unit in Newton), we can predict the path elevation $h(t)$ at time t. According to the physics rule, a player navigating a virtual path will experience a resistance force $F = \{f_i | f_i = \Phi(r_i)\}$ for each interval i. Mathematically, f_i is composed of the gravitational force and the frictional force with respect to path elevation angle θ_i through the decomposition rule of forces which is calculated as:

$$f_i = \mathrm{mg}(\sin\theta_i + \mu\cos\theta_i), \tag{9}$$

where μ is the frictional coefficient. Using the sine function to represent the resistance force model f_i by assuming that:

$$f_i = \mathrm{mg}R\sin(\theta_i + \alpha) \tag{10}$$

According to the angle addition formula of sine, there is:

$$f_i = \mathrm{mg}(R\sin\theta_i\cos\alpha + R\cos\theta_i\sin\alpha) \tag{11}$$

Then, compared with Equation 9, we have $R\cos\alpha = 1$ and $R\sin\alpha = \mu$. Them, we have $R^2\cos^2\alpha = 1$ and $R^2\sin^2\alpha = \mu^2$. Therefore, $R^2(\sin^2\alpha + \cos^2\alpha) = 1 + \mu^2$. Since $\sin^2\alpha + \cos^2\alpha = 1$ leads to $R = \sqrt{1+\mu^2}$, so there is: $\sin\alpha = \mu/\sqrt{1+\mu^2}$ and $\cos\alpha = 1/\sqrt{1+\mu^2}$. As $\tan\alpha = \sin\alpha/\cos\alpha = \mu$, we have $\alpha = \tan^{-1}\mu$. So we have:

$$f_i = \mathrm{mg}\sqrt{1+\mu^2}\sin(\theta_i + \tan^{-1}\mu) \tag{12}$$

Therefore, the elevation angle θ_i can be calculated as:

$$\theta_i = \sin^{-1}\left(\frac{f_i}{\mathrm{mg}\sqrt{1+\mu^2}}\right) - \tan^{-1}(\mu) \tag{13}$$

Empirically, we set the frictional coefficient $\mu = 0.5$. Let t_i denotes the accumulative time for duration Δt_i where $t_0 = 0$ and $t_i = t_{i-1} + \Delta t_i$, then the path elevation $h(t)$ is calculated as:

$$h(t) = \sum_{i=1}^{N}\int_0^t s_i(t)\sin\theta_i(t)\mathrm{d}t \tag{14}$$

where speed $s_i(t) = s_i$ when $t_i \le t < t_{i+1}$, otherwise is 0. Similarly, elevation angle $\theta_i(t) = \theta_i$ when $t_i \le t < t_{i+1}$, otherwise is 0.

Cosine Decay Function. In order to smoothly embed the predicted path onto arbitrarily given terrain, we use the cosine decay function to generate the α-map for the texture and the height-map of the path. The cosine decay function is denoted as $\alpha_w^r(x)$, where w is the user-specified width of the path, and r is the path impact range, representing the path has the following features: (1) if the α-map pixel's distance to the center the path is less than the half of the path width then the pixel's alpha value should be 1. (2) if the pixel's distance to the center the path is greater than the half of the impact range then the pixel's alpha value should be 0. 3) otherwise the alpha value decays smoothly from 1 to 0 as the distance increases. Given these features, we defined the cosine decay function $\varrho_\kappa(x)$:

$$\varrho_\kappa(x) = \begin{cases} 0.5(1+\cos(|x|\pi/\kappa) & |x| \le \kappa, \\ 0 & |x| > \kappa. \end{cases} \tag{15}$$

where the alpha decay function $\alpha_w^r(x)$is defined as:

$$\alpha_w^r(x) = \begin{cases} 1 & |x| \le w, \\ \varrho_{r-w}(|x|-w) & |x| > w. \end{cases} \tag{16}$$

Bridge Structure. In order to handle the situations where the input path has self-intersections or intersections with the lake, we generate bridge structures to solve this problem. When the path curve $p(s)$ has self-intersections, it means there is height ambiguity happening to the intersecting points, in another word, there can be two different heights on one single point on the heightmap. However, heightmap is a one-to-one mapping function, it cannot be able to support two different points. So one possible solution is to build a bridge so that one point on the heightmap can be mapped to two spatial points among which one is on the terrain while another is on the bridge above the terrain. Let the set of path curve $p(t)$'s self-intersection points denoted as I, then there is:

$$I = \{\forall(t_0, t_1) | p(t_0) = p(t_1) \wedge h(t_0) \ne h(t_1)\} \tag{17}$$

Then for each intersection pairs $\forall(t_0, t_1) \in I$ swap (t_0, t_1) if $h(t_0) > h(t_1)$. Then, for new intersection pairs, the former one is always under the later one. Then, all the points on bridge are B:

$$B = \{\forall p(t) | L_b > 2|p(t) - p(t_1)| \wedge |t - t_1| < |t - t_0|\}, \tag{18}$$

where the L_b is the predefined bridge length. Once we calculated the points on bridge B, there are another two steps left to generate the bridge's structure: (1) generate the 3D geometry of the bridge points using the B. (2) reset the path height of the points under the bridges using the B. The new path height function is:

$$h^*(p(t)|p(t) \in B) = h(t_0)\lambda_{L_b} + h(t)(1 - \lambda_{L_b}), \tag{19}$$

where blending factor λ_{L_b} is calculated via cosine decay function:

$$\lambda_{L_b} = \varrho_{\frac{L_b}{2}}(p(t) - p(t_0)) \tag{20}$$

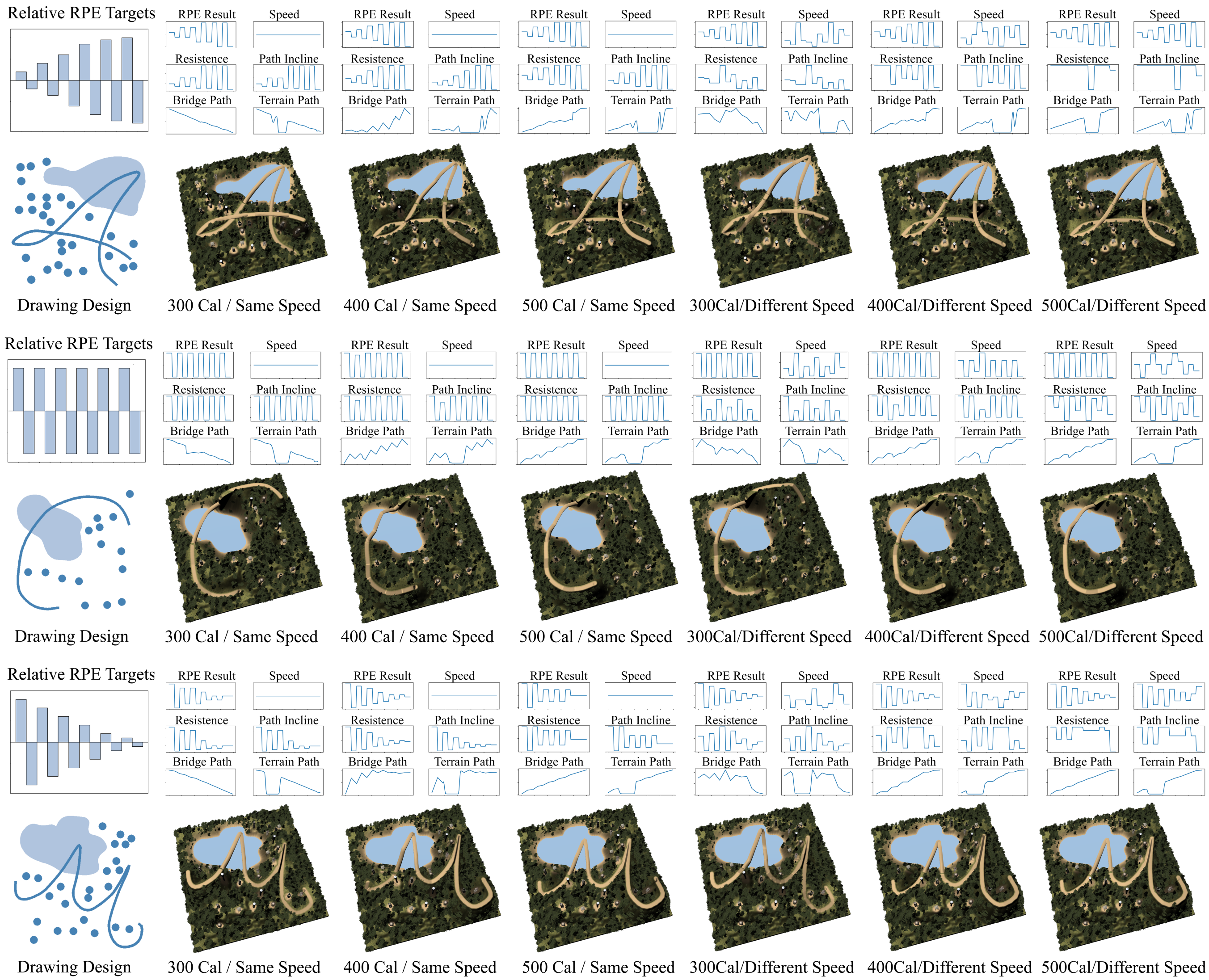

Figure 4: Numerical Experiment Results.

Terrain Synthesis. We render the terrain texture at any given pixel (u, v) by blending the path texture $C_p(u, v)$ with another different texture for the terrain background $C_b(u, v)$ using the alpha decay function $\alpha_w^r(x)$ defined above. In this way, the path can be separated from the whole terrain realistically. At the same time, the heightmap of the path can also be calculated using this above alpha blending approach. Mathematically, let $d(u, v) = \min(|p(t) - (u, v)|)$ denotes the minimal distance from any given pixel (u, v) on the α-map to the path curve $p(t)$, $t \in [0, 1]$, then the texture color at this pixel $C(u, v)$ can be calculated using this formula:

$$C(u, v) = \alpha_w^r(d(u, v))C_p(u, v) + (1 - \alpha_w^r(d(u, v)))C_b(u, v) \quad (21)$$

However, we want to make the path level and parallel to the horizon from the player's view, so we should make the height of the terrain near the path equal to to the height of the center of the path. So the heightmap $h(u, v)$ can be calculated using this formula:

$$h(u, v) = \alpha_w^r(d(u, v))h(p(t^*)) + (1 - \alpha_w^r(d(u, v))h(u, v), \quad (22)$$

where t^* is the solution of t for the equation: $|p(t) - (u, v)| = d(u, v)$.

4 EXPERIMENT RESULTS

Figure 4 shows the numerical experiment results of testing our proposed interface. All experiments are under the setting of the same distance target of 2000 m and time interval as 5 mins. Three rows show three different Relative RPE targets along with three different drawing virtual world designs. The first relative RPE target shows an alternating effort with amplifying amplitude, the second relative RPE target shows an alternating effort with constant amplitude, and the third relative RPE target shows an alternating effort with decaying amplitude. Those three different drawing virtual world designs contain paths in the shapes of letters 'A', 'C', and 'M' respectively. For the subfigure in each row, there are seven columns. The first column shows the user input of relative RPE targets and drawing virtual world designs. The second column to the fourth column shows the optimizations with three different energy expenditure settings of 300 Cal, 400 Cal, and 500 Cal under the same speed constraint setting as 100 SPM. The fifth column to the seventh column shows the optimizations under varying speeds condition.

User Study. Through the motion simulation in VR, we conduct a preliminary user study by connecting the elliptical machine's interaction with the navigation in the virtual world generated through our proposed computational design interface. Figure 5 shows the screenshot of this study where the left subfigure shows the user's view in the Oculus Quest 2 VR headset while the right subfigure shows the user's gym activities on an elliptical machine. During this experiment, the user's controllers are bounded on the two handles of the elliptical machine. As the user is walking on the elliptical machine, the user's controllers will move forward or backward in the meanwhile. Therefore, according to the motion captured from the user's controllers, the navigation speed can be accurately simulated in the virtual scene generated with our proposed approach.

At the beginning of the study, we ask the user wearing a VR headset to keep looking ahead for Δt time. Let $\mathbf{f}(t)$ denote the VR headset's forward orientation at time t, Then, we estimate the forward orientation of the elliptical machine $\mathbf{d}$ as:

$$\mathbf{d} \approx \frac{1}{\Delta t} \int_0^{\Delta t} \mathbf{f}(t) \mathrm{d}t, \tag{23}$$

where we empirically set $\Delta t = 3$ (sec). Let $\mathbf{p}_{l/r}(t)$ denote the left/right VR controller's position at time t, then the user's walking velocity in the virtual environment is $\mathbf{v}(t)$:

$$\mathbf{v}(t) = k \left(\max(0, \dot{\mathbf{p}_l}(t) \cdot \hat{\mathbf{d}}) + \max(0, \dot{\mathbf{p}_r}(t) \cdot \hat{\mathbf{d}}) \right) \hat{\mathbf{d}} \tag{24}$$

where k is the speed up setting which is empirically set as 2.

According to the user's feedback about the navigation experience in virtual reality, the walking experience on the elliptical machine is immersively mapping to the virtual environment which is generated through our proposed approach. This validates the correctness and accuracy of our proposed simulation approach for mapping the elliptical walking experience into the VR navigation experience. The video recording for this preliminary user study can be find through this URL link at here https://youtu.be/iZGVSpzPZWs

5 CONCLUSION

In this paper, we propose Elliptical4VR, an interactive exergame authoring tool that synthesizes virtual environments and training programs for personalized elliptical workout experiences in virtual reality. The goals of our project are mainly focused on two aspects: (1) Optimize the elliptical workout plan so that the player can achieve the proposed exercise goals including the overall intensiveness of the game (e.g. how many calories are burned, how long the running distance is, etc.) and also match the user-specified relative RPE patterns during the game which are reflecting the experience during the exergame. (2) The visual perception of the force of users walking on the path in the virtual world need to match with the haptic feedback from the elliptical machine, so as to create a high-level immersiveness through such compatibility between visual feedback and haptic feedback. Both numerical experiments and the preliminary user study validate our interface w.r.t these two goals. In future work, we will focus on extending the user study scale so as to test the efficacy and accuracy of our proposed interface with a wider range of demands from the exergame designers.

Figure 5: Preliminary User Study Result.

REFERENCES

[1] Siddhartha Chib and Edward Greenberg. 1995. Understanding the metropolis-hastings algorithm. *The american statistician* 49, 4 (1995), 327–335.

[2] Jeff Feasel, Mary C Whitton, and Jeremy D Wendt. 2008. LLCM-WIP: Low-latency, continuous-motion walking-in-place. In *2008 IEEE symposium on 3D user interfaces*. IEEE, 97–104.

[3] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. 2020. Sketchycoco: Image generation from freehand scene sketches. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5174–5183.

[4] Charles J Geyer. 1992. Practical markov chain monte carlo. *Statistical science* (1992), 473–483.

[5] Cusuh Ham, Gemma Canet Tarrés, Tu Bui, James Hays, Zhe Lin, and John Collomosse. 2022. Cogs: Controllable generation and search from sketch and style. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*. Springer, 632–650.

[6] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science* 220, 4598 (1983), 671–680.

[7] Sandip Kulkarni, Charles Fisher, Eric Pardyjak, Mark Minor, and John Hollerbach. 2009. Wind display device for locomotion interface in a virtual environment. In *World Haptics 2009-Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE, 184–189.

[8] Sandip D Kulkarni, Charles J Fisher, Price Lefler, Aditya Desai, Shanthanu Chakravarthy, Eric R Pardyjak, Mark A Minor, and John M Hollerbach. 2015. A full body steerable wind display for a locomotion interface. *IEEE Transactions on Visualization and Computer Graphics* 21, 10 (2015), 1146–1159.

[9] Wanwan Li. 2023. Terrain synthesis for treadmill exergaming in virtual reality. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 1–7.

[10] Wanwan Li, Biao Xie, Yongqi Zhang, Walter Meiss, Haikun Huang, and Lap-Fai Yu. 2020. Exertion-aware path generation. *ACM Trans. Graph.* 39, 4 (2020), 115.

[11] Casimir JH Ludwig, Nicholas Alexander, Kate L Howard, Alicja A Jedrzejewska, Isha Mundkur, and David Redmill. 2018. The influence of visual flow and perceptual load on locomotion speed. *Attention, Perception, & Psychophysics* 80 (2018), 69–81.

[12] Weizhi Nie, Van-Tu Ninh, Yuting Su, Vinh Ton-That, Minh-Triet Tran, Shu Xiang, Heyu Zhou, Yang Zhou, and Zhichao Zhou. 2018. SHREC'18 track: 2D scene sketch-based 3D scene retrieval. *Training* 18 (2018), 70.

[13] Niels Christian Nilsson, Stefania Serafin, and Rolf Nordahl. 2014. Establishing the range of perceptually natural visual walking speeds for virtual walking-in-place locomotion. *IEEE transactions on visualization and computer graphics* 20, 4 (2014), 569–578.

[14] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. 2021. Scene designer: a unified model for scene search and synthesis from sketch. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2424–2433.

[15] Frank Steinicke, Gerd Bruder, Klaus Hinrichs, and Anthony Steed. 2010. Gradual transitions and their effects on presence and distance estimation. *Computers & Graphics* 34, 1 (2010), 26–33.

[16] Tanh Quang Tran, Thanh Dat Ngoc Tran, Tam Duy Nguyen, Holger Regenbrecht, and Minh-Triet Tran. 2018. Can we perceive changes in our moving speed: a comparison between directly and indirectly powering the locomotion in virtual environments. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. 1–10.

[17] Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. 2013. Sketch2Scene: Sketch-based co-retrieval and co-placement of 3D models. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–15.

[18] Robert C Zeleznik, Kenneth P Herndon, and John F Hughes. 2006. SKETCH: An interface for sketching 3D scenes. In *ACM SIGGRAPH 2006 Courses*. 9–es.