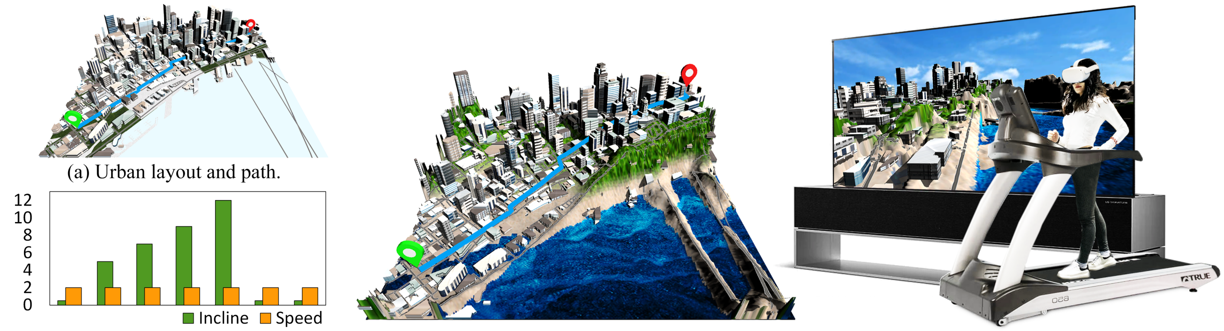
Terrain Synthesis for Treadmill Exergaming in Virtual Reality

Wanwan Li
University of South Florida



(a) Urban layout and path.

(b) Treadmill workout profile.

(c) Terrain synthesized with our approach.

(d) Treadmill exertion game in Virtual Reality (VR).

Figure 1: Given a user-specified urban layout with a navigation path (a) and given a user-specified treadmill workout profile (b), we automatically generate the terrain (c) for treadmill exertion games in Virtual Reality (VR) such that the haptic feedback from the treadmill device matches with the visual content in the VR headset. After synchronizing the VR exergame with the treadmill device, the player can have an immersive virtual exercise experience (d).

Abstract

In order to connect haptic feedback with visual content in treadmill VR exergames for getting more immersive user experiences, we present a novel optimization-based approach to automatically generate terrains for treadmill exergames that can match well with the user-specified treadmill workout profile. In order to validate the effectiveness of our approach, a series of numerical experiments are conducted to investigate the visual effects of the generated terrain, the virtual walking experience of the treadmill exergame, and the comparison between the manual terrain creation approach and the terrain synthesis approach.

Keywords: terrain systhesis, haptic device, treadmill, exergame, virtual reality, stochastic optimization

1 Introduction

As more people realize the importance of living healthy lives, exertion games (i.e. exergames), which is a special genre of video games, have become a trend in modern video games. In recent years, VR technologies that match virtuality with reality [9,20] through immersive haptic devices [27,28,33] are widely studied and claimed by researchers that exergaming provides positive results by enhancing social well-being, reducing loneliness, and increasing social connection [21]. Due to the potential of exergaming to improve mental and physical health is promising, the game industry has a strong incentive to explore advanced computer graphics technologies that are able to create an immersive, realistic virtual experience that considers players' physical activity and exercise health. Especially, for better visual effects, many advanced interactive computer graphics technologies such as realistic rendering, geometric modeling, and terrain procedural authoring, have been widely studied to develop those immersive exertion games in virtual reality.

Mere visual realism without haptic feedback is not immersive for players living in a physical world. On one hand, in order to get better body training effects with more engaging user experiences by filling such a gap between the virtual synthesized world and the real physical world, growing interests from researchers are switching toward those novel technologies that connect haptic feedback with virtual content in VR headsets. On the other hand, the traditional approach which is based on manually creating or modifying the virtual content to match the haptic feedback through trials and errors can be tedious and time-consuming work for exergame designers.

Since synthesizing and editing procedural terrain with advanced technologies has been a popular research topic for decades, different terrain editing and modeling approaches, such as GPU-based [7,26], patch example-based [36], ATGS-based [31], sketch-based [6], physics-based [29], volumetric-based [5], layered depth normal images-based [11], voxel cubes-based [25], global-to-local control-based [34], etc., have been studied systematically. Recent research works also show growing interest in applying inverse procedural modeling on terrains synthesis and editing by employing data-driven approaches such as the Conditional GANs-based [16], multi-theme-based [35], Points-of-Interests-based [30], etc. Procedural terrain synthesis technologies have also been employed in game level design such as building virtual worlds [32], landscape automata [3], personalizing exercise therapy [24], etc. As another research work closely related to ours, Li et al. [22] propose a novel approach to automatically generate a path on arbitrary terrain that can deliver users with desired exertion effects. However, their approach suffers failures to optimize a path when the terrain input is not qualified. Also, there are hardware constraints behind their approach such as a custom-built VR bike programmed with Adriano. Therefore, for overcoming those limitations, we propose a novel optimization approach that synthesizes terrain in VR according to arbitrary hardware settings (i.e., workout profile) for a general-purposed treadmill that users can buy everywhere. Our approach can help the exergame designers automatically generate virtual terrains that are compatible with user-specified treadmill workout profiles without demanding manual effort. Major contributions of our proposed work include:

- We preprocess a terrain heightmap dataset and use this dataset to train a RaLSGAN (a variation of GAN, please refer to Section 3 for more details), for synthesizing high-resolution photorealistic terrain heightmaps in real-time.
- We synthesize realistic virtual environments by combining the terrain generated from the RaLSGAN with realistic urban layouts extracted from OpenStreetMap data.

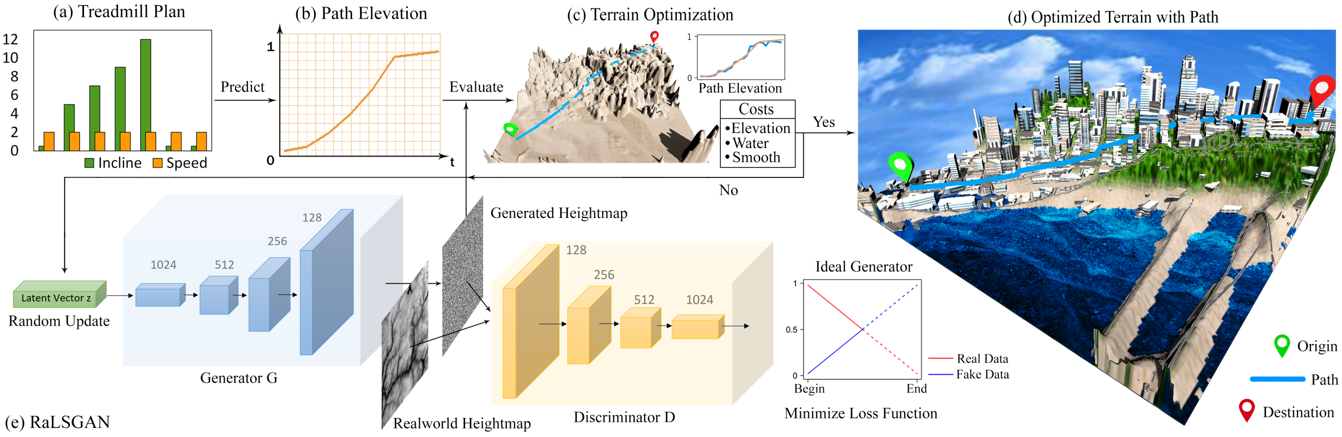


Figure 2: Overview of our approach.

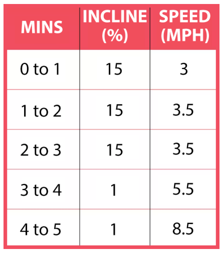
- We propose a novel technical approach for optimizing the synthesized terrain with target inclines (elevation angles) as specified by the treadmill workout profile, resulting in immersive user experiences for VR exergames.
- We conducted a series of numerical experimental and user studies to show that our approach can deliver the desired visual effects with less manual effort and higher accuracy to enhance the VR treadmill exergaming experience.

2 Overview

In order to generate realistic terrain that can deliver users with their expected exercise training effects compatible with their visual effects during their gaming experiences in VR display, we propose a novel technical approach includes three main steps: (1) Training the RaLSGAN for efficient terrain inverse procedural modeling controlled by latent vectors. (2) We optimize the latent vector until the terrain output from the RaLSGAN matches well with the user-specified treadmill workout profile. (3) Procedurally embeds the given urban layout into the synthesized terrain to result in an immersive virtual environment. Figure 2 shows the overview of our approach. (a) Given the treadmill workout profile that specifies the speed and inclines for each time interval, we calculate the path elevation as shown in (b). Then we optimize the high-resolution realistic terrain (c) synthesized from a RaLSGAN trained with real-world terrain heightmap data as shown in (e). During the terrain optimization process as shown in (c), we evaluate the total cost function which considers the path elevation, water area, and smoothness surrounding the path. If the total cost is small enough, we output the final terrain as shown in (d); Otherwise, we update latent vector Z and synthesize terrain through generator G again as shown in (e).

3 Technical Approach

3.1 Navigation Path



MINS	INCLINE (%)	SPEED (MPH)
0 to 1	15	3
1 to 2	15	3.5
2 to 3	15	3.5
3 to 4	1	5.5
4 to 5	1	8.5

Figure 3: Profile.

Workout Profile. Our approach is flexible and based on arbitrary interval workout devices such as True Fitness Treadmill devices for TRUE CS Series [1]. This type of treadmill is integrated with a powerful deck and motor combination that makes it easy to adjust speed and incline with custom workout profile settings on the touch screens. Our approach doesn't require specific functions to control programmed custom-built treadmill devices as it can be achieved by setting up the treadmill with a workout profile and starting the treadmill program with our VR exergame program at the same. Therefore, the only task that our approach needs to handle is to align the user's navigation path with an arbitrarily specified workout profile. As shown in Figure 3, a standard exercise plan for treadmill workout [4] is a table with three columns: (1) Time interval, (2) Incline, and (3) Speed. A treadmill workout plan is appended with several time intervals for different training stages, which are corresponding to the rows of the table. For example, the first row can be a starting stage with about 1 min duration, a low average speed (3 MPH), and a high incline (15%). Incline specifies the angle of the treadmill. Mathematically, assume there are N different workout intervals corresponding to the N rows of the table. Let Δt_i, θ_i, and v_i denotes the duration, incline and speed for i^{th} interval respectively, where $i = 1, 2, ..., N$. 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 target elevation of the players' navigation path is calculated as:

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

where $v_i(t) = v_i$ when $t_i \le t < t_{i+1}$, otherwise is 0. Similarly, $\theta_i(t) = \theta_i$ when $t_i \le t < t_{i+1}$, otherwise is 0. Noted that this representation is similar to the B-Spline Curve defined by construction by means of the Cox–de Boor recursion formula [12] to represent discrete values as a continuous function with time t.

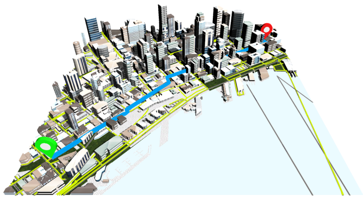
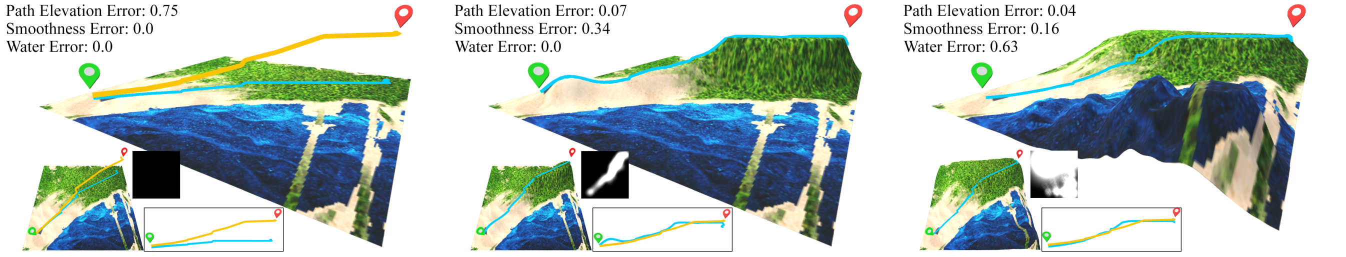


Figure 5: Urban Layout.

Urban Layout. After user specifies a place in the real world, our approach automatically downloads realistic urban street layouts from the OpenStreetMap (OSM) database [23] using OSM API. As shown in Figure 5, in this case, the user chooses a region near Seattle, Washington, USA. Then, by using the Floyd-Warshall algorithm [13], the navigation graph is constructed according to all pairs' shortest paths (as specified as yellow curves). Then, by specifying the start location as the origin (green pin), and the end location as the destination (red pin), the shortest path connecting these two places on the map will be automatically calculated (blue curve). Without loss of generality, the user can specify an arbitrary urban layout and an arbitrary path R on that layout using the approach mentioned above. Noted that users-specified path curve R is $\mathbf{r}(s) = (x(s), z(s))$ on a 2D texture domain on XZ-plane, where s is the displacement along R, $s \in [0, |R|]$ and $|R|$ is the total length of R. We will augmented the 2D path R with elevation $h(t)$ calculated through Equation 1. Then 2D curve R on the xz-plane can be mapped to the 3D path P on the surface of the 3D terrain (assume this terrain exists) using path elevation function $h(t)$. Now it is important to reveal the relation between time t in $h(t)$ and the spacial displacement s in $\mathbf{r}(s)$. As the displacement s along XZ-plane is projected from the 3D path onto the XZ-plane, therefore, speed $v_i(t)$ is projected through the cosine



(a) Missing elevation cost.

(b) Missing smoothness cost.

(c) Missing water cost.

Figure 4: Missing costs. This figure shows the possible poor solutions that are caused by missing costs. Subfigure (a), (b), and (c) shows the result terrain for missing elevation cost, smoothness cost, and water cost respectively.

of incline angle $\theta_i(t)$, relation between time t and displacement s is:

$$s(t) = \sum_{i=1}^{N} \int_0^t v_i(t)\cos\theta_i(t)\mathrm{d}t \tag{2}$$

Therefore, combine path elevation $h(t)$ from Equation 1 on y axis and path curve function $\mathbf{r}(s(t)) = (x(s(t)), z(s(t)))$ on x and z axis, we have the parameterization of the expected 3D navigation path curve P as $\mathbf{p}(t) = (x(s(t)), h(t), z(s(t)))$.

3.2 Terrain Synthesis

High-Resolution Heightmap. In this work, we use a data-driven approach to synthesize high-resolution terrain heightmaps in real-time using a variation of the Standard Generative Adversarial Network (SGAN) called Relativistic Average Least Square Generative Adversarial Network (RaLSGAN). As proposed by Jolicoeur et al. [17], RaLSGAN extends the SGAN (SGAN was first proposed by Goodfellow [15] in 2014) with a relativistic discriminator. The main difference between SGANs and RGANs has resided in their goals: SGANs' are hoping to make both fake data and real data look real in the end. Instead, RGANs make their goal even harder to achieve, that is hoping to make fake data look real but real data look fake at the end, which means fake data look "more real" than real data. So, given these assumptions, in RaLSGAN, the Mean Square Error (MSE) loss functions for discriminator D and generator G are:

$$L_D = \left|D(G(z)) - \left(\overline{D}(x) - 1\right)\right|^2 + \left|D(x) - \left(\overline{D}(G(z)) + 1\right)\right|^2 \tag{3}$$

$$L_G = \left|D(G(z)) - \left(\overline{D}(x) + 1\right)\right|^2 + \left|D(x) - \left(\overline{D}(G(z)) - 1\right)\right|^2 \tag{4}$$

For more details and explanations of the RaLSGAN implementations and training process proposed in our approach, please refer to the Supplementary Material (Section 7).

Heightmap Image Processing. Synthesized terrain with a coarse surface is not suitable for placing the urban layout from the OpenStreetMap due to the fact that the areas beneath the buildings or the roads need to be flat or smooth. Therefore, we proposed an efficient approach to processing the heightmap synthesized from the RaLSGAN so as to guarantee the areas beneath the buildings and roads are flattened on the heightmap. As the heightmap is a gray image, we apply an image processing method to flatten the building areas and road areas on the heightmap. Figure 6 shows our heightmap processing approach: given an arbitrary heightmap synthesized with a RaLSGAN (a), we first flatten the building areas by looping for each building, consider the bounding area of each building as a polygon, calculate the center color of the heightmap in that polygon and fill that polygon shape onto the heightmap image using that center color. After this processing step, we got the heightmap with the building areas flattened (b). The next step is to smooth the road area on the heightmap image. In this step, we loop for each road line in the road geometry from OpenStreetMap data and draw a line onto the heightmap image with the linear color transition using two different colors that are corresponding to the colors of the road's two endpoints on the heightmap image. In this way, the road will be smoothed according to the elevation of the two endpoints of that road. (c) shows the heightmap with both the building and road areas flattened. After these heightmap image processing steps, the terrain with the final heightmap is suitable for placing the buildings and the roads in the urban layout from the OpenStreetMap.

(a) Given heightmap.

(b) Add building area.

(c) Add road area.

Figure 6: Heightmap image processing. (a) shows the terrain heightmap synthesized with a RaLSGAN. (b) shows the heightmap with the building areas flattened. (c) shows the heightmap with the building and road areas flattened.

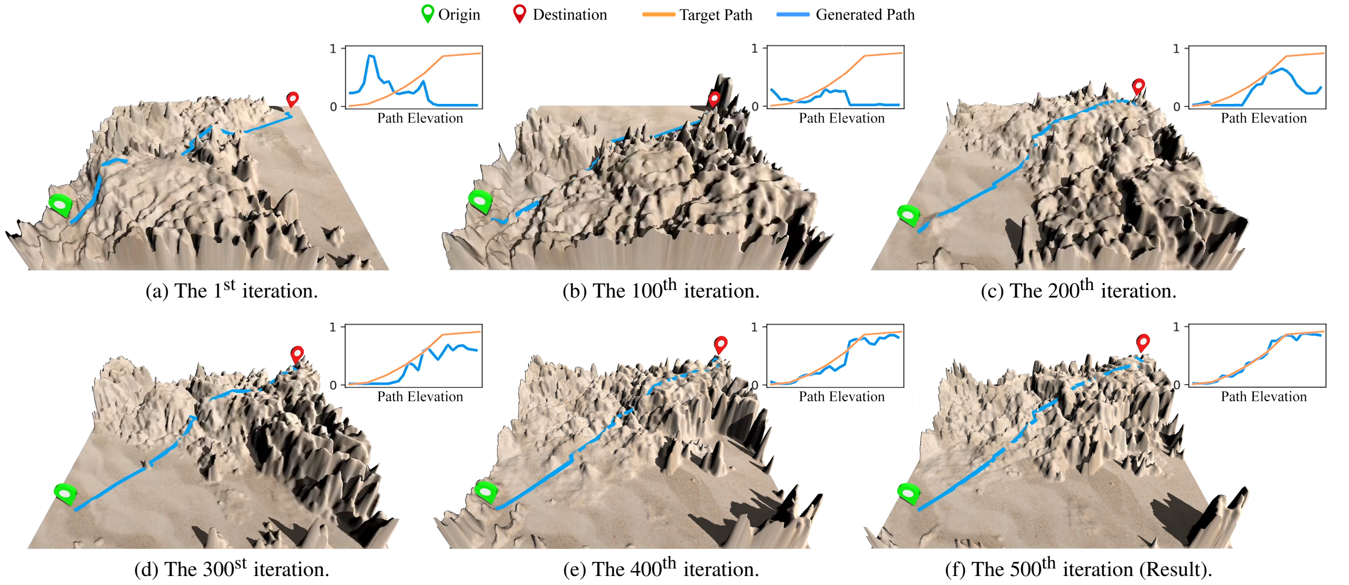
3.3 Cost Functions

Total Cost. In our approach, a terrain's heightmap is generated from a RaLSGAN using the random input latent vector z, the terrain heightmap output is $H(u,v) = G(z)$, where (u,v) is the texture coordinates of the heightmap image. Then there are three terms to evaluate a generated terrain H including Elevation Cost C_e measures the difference between the elevation of the path along the generated terrain $H(u(t), v(t))$ and the target path elevation $h(t)$ calculated from user's workout profile; Smoothness Cost C_S measures how bumpy is the nearby region on generated terrain where the path is passing through; Water Cost C_W measures whether there is any mismatch between the generated terrain and the specified urban layout where there are water areas. Total cost $C_\mathrm{total}(H)$ is:

$$C_\mathrm{total}(H) = w_\mathrm{e}C_\mathrm{e}(H) + w_\mathrm{S}C_\mathrm{S}(H) + w_\mathrm{W}C_\mathrm{W}(H), \tag{5}$$

where the w_e, w_S, and w_W represent the respective blending weights of the elevation cost, smoothness cost, and water cost respectively. Empirically, we set $w_\mathrm{e} = 0.4$, $w_\mathrm{S} = 0.3$, and $w_\mathrm{W} = 0.3$.

Elevation Cost. In order to generate a terrain that can match the user's physical experience in real life with the visual experience in the virtual world, the key issue is to match the elevation angle of the virtual path on the generated terrain and the incline angles of the treadmill device where the user is running. From Equation 1 we can calculate the path elevation from a pre-set workout profile, therefore, we need to measure the difference between the expected path elevation function $h(t)$ and the path elevation function calculated from the generated terrain's heightmap $H(u(t), v(t))$, then minimize that difference as path elevation error $E_h(H)$. Assume the total duration



(a) The 1st iteration.

(b) The 100th iteration.

(c) The 200th iteration.

(d) The 300st iteration.

(e) The 400th iteration.

(f) The 500th iteration (Result).

Figure 7: Optimization process. The blue curve is the navigation path lying on the synthesized terrain while the orange curve is the target navigation path calculated from the treadmill workout profile. (a) The navigation path is initialized with a terrain randomly generated by RaLSGAN. Throughout the iterations, the latent vector is randomly updated. Figure (b-f) shows the intermediate results generated through the optimization process. Figure (f) shows the result of the final synthesized terrain that has the blue path matched with the orange path.

of the treadmill exercise is t_N as defined in the exercise plan table, then the path elevation error $E_e(t)$ is:

$$E_e(H) = \int_0^{t_N} |H(u(t), v(t)) - h(t)| \mathrm{d}t, \tag{6}$$

where $u(t) = x(s(t))$ and $v(t) = z(s(t))$ according to the 2D curve function $\mathbf{r}(s(t)) = (x(s(t)), z(s(t)))$ as explained in Equation 2. Then the elevation cost function $C_\mathrm{e}(H)$ is a Gaussian-like function to smooth the path elevation error $E_e(H)$ and $C_\mathrm{e}(H)$ is:

$$C_\mathrm{e}(H) = 1 - \exp(-\left(\frac{E_e(H)}{\sigma_\mathrm{e}}\right)^2), \tag{7}$$

where we empirically set $\sigma_\mathrm{e} = 0.25$. As shown in Figure 4(a), missing elevation cost might generate a terrain like ”plane” and the target path (orange curve) will mismatch with the path on the generated terrain (blue curve).

Smoothness Cost. In order to avoid the rapid changes that happen to the synthesized terrain in the nearby area where a path is passing through, we introduced the Laplace operators (∇^2) to calculate the sharpness of the heightmap H near a path. By minimizing the integration of the divergence ($\nabla\cdot$) of the gradient (∇) of the heightmap within the nearby region where path $\mathbf{r}(s(t)) = (x(s(t)), z(s(t)))$ enters, we need the heightmap in that region as smooth as possible. The smoothness error E_S is represented as:

$$E_\mathrm{S}(H) = \int_0^{t_N} \nabla^2 H(u(t), v(t)) \mathrm{d}t \tag{8}$$

where $u(t) = x(s(t))$ and $v(t) = z(s(t))$. The Laplace operator (∇^2) for two dimensions coordinates (u, v) is given by the sum of the second partial derivatives for both u and v. which is defined as:

$$\nabla^2 H(u, v) = \frac{\partial^2 H(u, v)}{\partial u^2} + \frac{\partial^2 H(u, v)}{\partial v^2} \tag{9}$$

where (u, v) are the 2D coordinate of the heightmap's texture space. By minimizing this term, we can make the heightmap's local regions where the path is passing nearby as smooth as possible. Similarly, the smoothness cost function $C_\mathrm{S}(H)$ is defined as:

$$C_\mathrm{S}(H) = 1 - \exp(-\left(\frac{E_s(H)}{\sigma_\mathrm{S}}\right)^2), \tag{10}$$

where we empirically set $\sigma_\mathrm{S} = 0.25$. As shown in Figure 4(b), missing smoothness cost results in a terrain has a sharp ”edge” near the path (blue curve). This can result in a trivial solution that only elevates the surrounding area of the terrain where the path is passing through to match target path but result in an unrealistic result.

Water Cost. In order to avoid the generated terrain not matching with the user-specified urban layout where there is a water area, this term is used to ”tune” the terrain to be flat where it is supposed to be in the water. Therefore, we penalize the elevation of the terrain where that area belongs to the water region W evaluated through the water error function $E_\mathrm{W}(H)$ which is defined as:

$$E_\mathrm{W}(H) = \frac{1}{|W|} \iint_{(u,v)\in W} H(u, v) \mathrm{d}u\mathrm{d}v \tag{11}$$

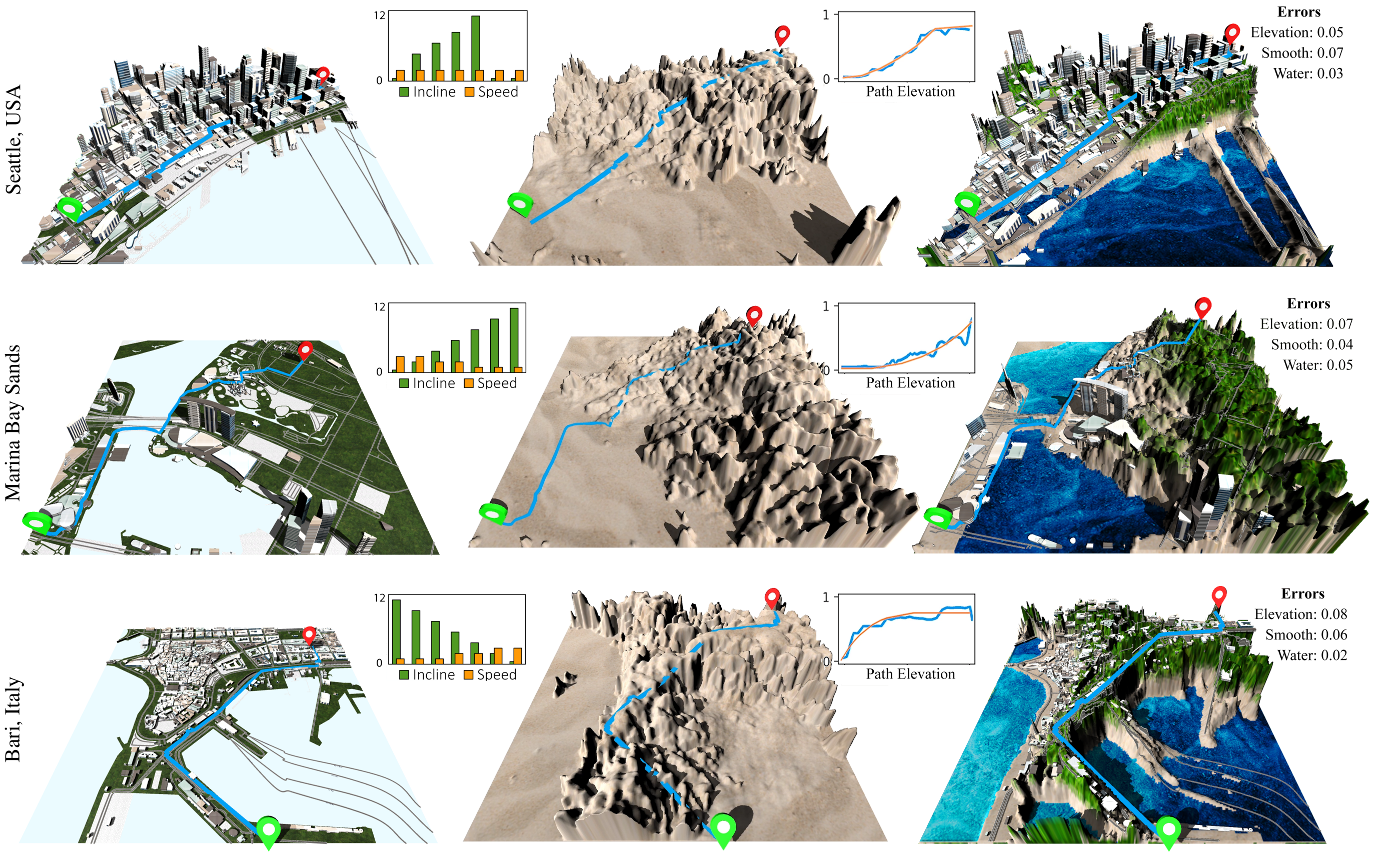
Then the water cost function $C_\mathrm{W}(H)$ is represented as:

$$C_\mathrm{W}(H) = 1 - \exp(-\left(\frac{1 - E_\mathrm{W}(H)}{\sigma_\mathrm{W}}\right)^2), \tag{12}$$

where we empirically set $\sigma_\mathrm{W} = 0.25$. As shown in Figure 4(c), missing water cost can generate a terrain that has ”mountain” above the sea region (blue texture)) which is unrealistic.

3.4 Terrain Optimization

Figure 7 shows the terrain optimization process proposed by our approach. Given the workout profile as the exercise table, we calculate the path elevation as specified in Equation 1. We use a pre-trained RaLSGAN to synthesize a realistic terrain heightmap. During the terrain optimization process, we evaluate the total cost function according to the synthesized terrain, if the total cost is small enough,



(a) User-specified urban layout and path.

(b) Terrains synthesized with our approach.

(c) Urban scene on the synthesized terrains.

Figure 8: Numerical experimental results. This figure shows the synthesized terrains with different settings. Different rows show different locations including Seattle, Marina Bay Sands, and Bari. Different columns are: (a) input of urban layouts, paths, and workout profiles; (b) terrains synthesized with our approach given the input from (a); (c) virtual urban environments generated on the synthesized terrains in (b).

we output the result as the final terrain; Otherwise, we update the latent vector z into z' and synthesize the terrain through the generator G again. We formulate the optimization problem as a searching problem by employing the Markov chain Monte Carlo method [14] to search for a solution that minimizes the total cost function. Given any randomly chosen latent vector z in the current stage, the proposed new latent vector z' is randomly searched in the solution space with three types of move:

- *Increase a Random Value*: a random value in z is increased by Δz to create a proposed latent vector z'.
- *Decrease a Random Value*: a random value in z is decreased by Δz to create a proposed latent vector z'.
- *Swap Two Random Values*: two random values in z are swapped with each other to create a proposed latent vector z'.

Empirically, we set random moving step $\Delta z \sim N(\mu = 0, \sigma^2 = 0.25)$. For a proposed update of the latent vector z', the new heightmap synthesized from the generator G is $H' = G(z')$. According to the formulation of the method by Kirkpatrick et al. [18], the acceptance probability function $\Pr(H'|H)$ is defined as:

$$\Pr(H'|H) = \min(1, \frac{f(H')}{f(H)}), \tag{13}$$

where $f(H)$ is a Boltzmann-like objective function related to a Metropolis-Hastings state searching step [10]:

$$f(H) = \exp(-\frac{1}{t}C_{\text{total}}(H)) \tag{14}$$

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, the temperature drops to a low value near zero, and 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 or terminated if the total cost change is smaller than 3% over the past 50 iterations.

4 Numerical Experiments

We have tested our proposed approach by synthesizing terrains with different settings. As shown in Figure 8, terrains are synthesized in different places with different workout profiles. The first row shows the terrain synthesized in Seattle, USA with a workout profile where users are running at a constant speed of 3 MPH. The inclines are increasing as time collapse, in the last two minutes, the inclines decrease to 0 for a recovery period. As we can see from the synthesized terrain, the terrain errors (Elevation error is 0.05, Smooth error is 0.07, Water error is 0.03) are relatively low and the overall trend of the elevation of the path matches well with the workout profile. The second row shows the terrain synthesized in Marina Bay Sands, Singapore, with a workout profile that starts will a faster speed while keeping a low incline at the beginning. As time goes by, the incline increases while the speed is getting lower in the end. The target path elevation mimics a quadratic equation curve like $y = x^2$. As we can see, our proposed approach is still quite good at solving this type of workout profile as the terrain errors are quite low (Elevation error is

0.07, Smooth error is 0.04, and Water error is 0.05). The third row in Figure 8 shows the terrain synthesized in Bari, Italy with a workout profile that starts will a slower speed while keeping a higher incline. As time goes by, the incline decreases while the speed is getting faster. This target path elevation is aimed at mimicking a square-root equation curve like $y = \sqrt{x}$. As shown in the result, our proposed approach is robust enough to solve this type of workout profile as the elevation error is low enough to be acceptable (Elevation error is 0.08, Smooth error is 0.06, and Water error is 0.02). As we can see, the synthesized terrain successfully reflects such a pattern that the terrain raised up immediately at the beginning and after then it elevates higher slowly. Besides, the water area almost exactly matches the urban layout input. In general, our approach solved these proposed numerical problems successfully with acceptable terrain errors. So, users are able to experience an immersive VR exertion game while virtually navigating this scene, as the treadmill's haptic feedback matches their visual effects in virtual scene.

5 User Study Experiments

In order to address two questions: (1) whether our workout profile-driven terrain synthesis approach significantly improves the exergame terrain design efficiency and (2) whether the match between visual feedback in VR display and the haptic feedback from treadmill significantly improves the user's workout experience, we have conducted two users studies to answer these two questions respectively. User Study 1: Terrain Manual Design. We compare our approach with a manual approach for treadmill workout profile-driven terrain design; User Study 2: Treadmill Walking in VR. We compare the user's workout experiences in the scene on a flat terrain that has no elevation changes with the workout experiences in a scene generated on our synthesized terrain that has elevation changes.

Experiment Process. For the experiment process of User Study 1, please refer to Supplementary Material (Section 1). For User Study 2, we recruited 10 undergraduate students for this treadmill VR walking experiment. With the help of 5 students as the user study organizers, 10 student participants are running through two VR navigation programs on the treadmill. One VR program is walking within an urban environment on a flat terrain called *City on Ground*. Another one is walking within an urban environment on the terrain synthesized with our approach called *City on Hill*. These two virtual environments are shown in the first row of Figure 8 (a) and (c) respectively. Note that the order to play these two programs is randomly decided. During the study, each participant is asked to put two hands on the two handles on the treadmill to keep the process safe. The speed of the treadmill is set to 0.3 MPH which is slow enough to ensure safety while wearing a VR headset. After the study, we ask questions about users' exergaming experiences with these two sessions of VR programs and rated them with Likert scores for level of enjoyment.

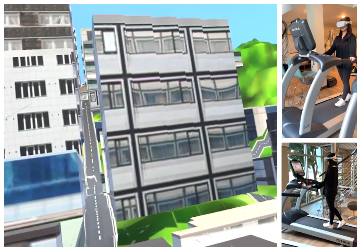
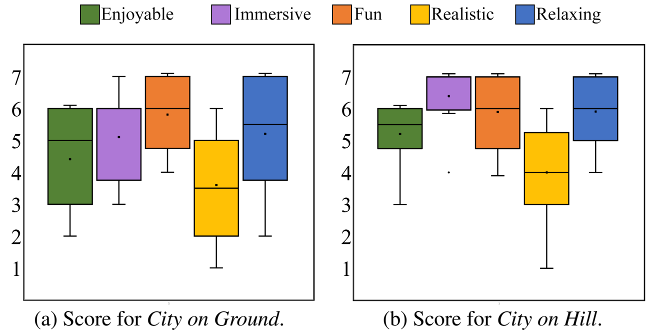


Figure 9: Treadmill walking.

As shown in Figure 9, during User Study 2, we tested the treadmill virtual walking experience on an Oculus Quest 2 VR headset. We build a VR navigation program that can be automatically synchronized with the treadmill program by making both programs set up with the same exercise profile. As shown in the Supplementary Video, a manual presetting of the workout profile can automatically adjust the inclines of the True treadmill device after this workout profile is started. At the same time, through a VR program that is implemented by us using Unity3D, a virtual-navigation script automatically pushes the player's VR camera to move forward, left, right, or up (at the same speed as the treadmill device) in the virtual scene, by setting up this VR program with the same workout profile for the treadmill presetting. During the study, one organizer starts the VR program at the same time as another organizer finished setting up the treadmill workout plan. Therefore, as the treadmill program and the VR program are started at the same time, the motion of the treadmill will be automatically synchronized with the virtual navigation. For example, at any time when the treadmill's incline goes up, players go up simultaneously in the VR scene. For more details, please refer to Supplementary Material (Section 2).



(a) Score for *City on Ground*.
(b) Score for *City on Hill*.

Figure 10: Perception score of Study 2 (Likert scale 1 to 7).

6 Results and Discussions

For experimental results of User Study 1, please refer to Supplementary Material (Section 1, 4, 6). For User Study 2, according to the standard questionnaire testing users' perceived enjoyment level [2, 19], every question is asking about one perceptive adjective (in our study, they are enjoyable, immersive, fun, realistic, and relaxing) to rate their VR walking experience. (E.g., How much extent do you agree this is fun?) According to these two VR program settings for *City on Hill* and for *City on Ground*, the Likert perception score between 1 and 7 evaluated by the users are shown in Figure 10. According to the statics, there is a higher perception score for the *City on Hill* than the *City on Ground* according to the AVG scores for the *City on Ground* game which are (4.7, 5.1, 5.8, 3.6, 5.2) and those for *City on Hill* which are (5.2, 6.4, 5.9, 4, 5.9). Furthermore, we applied single factor ANOVA tests [8] for users' Likert perception scores for these questions. Using $\alpha = 0.05$ (95% confidence interval), we obtain the ANOVA test results showing that among these two groups, $p = 0.0343 < 0.05$. Therefore, with a 95% confidence level, we claim that, due to the consideration of the inclines matches afforded by our terrain synthesis approach, VR walking experience in *City on Hill* is much more enjoyable, immersive, fun, realistic, and relaxing than VR walking experience in *City on Ground*. For users' general feedback about VR walking, please refer to Supplementary Material (Section 5).

7 Conclusion

In this paper, we propose a novel optimization approach to synthesize terrains for treadmill exergaming in virtual reality. Unlike other existing approaches that adjust gym device's haptic feedback according to the elevation angles of virtual paths on a fixed terrain, our approach takes the treadmill workout profile as the fixed input, then, the elevation angles of virtual paths are adjusted during a terrain optimization process so as to match the fixed target workout profile. Furthermore, we conduct two user studies that are aiming at claiming that first, our approach can efficiently improve the treadmill exergame design process; and second, the compatibility between virtual contents and haptic feedback improves the player's treadmill VR exergaming experience. However, for safety considerations, our experiment can only show that the user's low-speed walking experience on treadmill in improved by our approach. Whether our approach is suitable for high-speed intensive treadmill VR exercise, it still needs more experiments to support. In future work, other advanced technologies for synthesizing terrains connecting with gym activities need to be explored. For example, generating a terrain with rivers to simulate the rowing machine that mimics the boating activities. Or simulating the virtual climbing experience while introducing the arm workouts experience in the gym. We believe our work can inspire more research work on devising computational approaches to extend gym activities with immersive VR experiences.

References

[1] Treadmills. True Fitness. 2022 TRUE. All Rights Reserved. `https://shop.truefitness.com/products-landing/treadmills/`, 2022.

[2] J. Ábrahám, A. Velenczei, and A. Szabo. Perceived determinants of well-being and enjoyment level of leisure activities. *Leisure Sciences*, 34(3):199–216, 2012.

[3] D. Ashlock and C. McGuinness. Landscape automata for search based procedural content generation. In *2013 IEEE Conference on Computational Inteligence in Games (CIG)*, pp. 1–8. IEEE, 2013.

[4] J. Autuori-Dedic. The Ultimate Treadmill Interval Workout for Every Fitness Level. `https://www.shape.com/fitness/cardio/treadmill-interval-workouts`, September 2019.

[5] M. Becher, M. Krone, G. Reina, and T. Ertl. Feature-based volumetric terrain generation and decoration. *IEEE Transactions on Visualization and Computer Graphics*, 25(2):1283–1296, 2017.

[6] A. Bernhardt, A. Maximo, L. Velho, H. Hnaidi, and M.-P. Cani. Real-time terrain modeling using cpu-gpu coupled computation. In *2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 64–71. IEEE, 2011.

[7] S. Bhattacharjee, S. Patidar, and P. Narayanan. Real-time rendering and manipulation of large terrains. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 551–559. IEEE, 2008.

[8] G. Casella. *Statistical design*. Springer Science & Business Media, 2008.

[9] M. Centorrino, J. Condemi, L. D. Paola, and C. Ferrigno. From virtual reality to augmented reality: Devices, bodies, places and relationships. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 1–7, 2021. doi: 10.1109/ISMAR-Adjunct54149.2021.00011

[10] S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

[11] G. Cordonnier, M.-P. Cani, B. Benes, J. Braun, and E. Galin. Sculpting mountains: Interactive terrain modeling based on subsurface geology. *IEEE transactions on visualization and computer graphics*, 24(5):1756–1769, 2017.

[12] C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor. *A practical guide to splines*, vol. 27. springer-verlag New York, 1978.

[13] R. W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[14] C. J. Geyer. Practical markov chain monte carlo. *Statistical science*, pp. 473–483, 1992.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[16] É. Guérin, J. Digne, E. Galin, A. Peytavie, C. Wolf, B. Benes, and B. Martinez. Interactive example-based terrain authoring with conditional generative adversarial networks. *Acm Transactions on Graphics (TOG)*, 36(6):1–13, 2017.

[17] A. Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

[18] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[19] H.-G. Lee, S. Chung, and W.-H. Lee. Presence in virtual golf simulators: The effects of presence on perceived enjoyment, perceived value, and behavioral intention. *New media & society*, 15(6):930–946, 2013.

[20] B. Li, A. Nijholt, and D. Xu. A novel approach based on pcnns template for fingerprint image thinning. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 115–119. IEEE Computer Society, Los Alamitos, CA, USA, sep 2009. doi: 10.1109/ICIS.2009.132

[21] J. Li, M. Erdt, L. Chen, Y. Cao, S.-Q. Lee, and Y.-L. Theng. The social effects of exergames on older adults: systematic review and metric analysis. *Journal of medical Internet research*, 20(6):e10486, 2018.

[22] W. Li, B. Xie, Y. Zhang, W. Meiss, H. Huang, and L.-F. Yu. Exertion-aware path generation. *ACM Trans. Graph.*, 39(4), jul 2020. doi: 10.1145/3386569.3392393

[23] P. Mooney, M. Minghini, et al. A review of openstreetmap data. 2017.

[24] J. E. Muñoz, S. Cao, and J. Boger. Kinematically adaptive exergames: personalizing exercise therapy through closed-loop systems. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 118–1187. IEEE, 2019.

[25] A. Petrovas and R. Baušys. Automated digital terrain elevation modification by procedural generation approach. In *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pp. 1–5. IEEE, 2019.

[26] J. Schneider, T. Boldte, and R. Westermann. Real-time editing, synthesis, and rendering of infinite landscapes on gpus. In *Vision, modeling and visualization*, vol. 2006, pp. 145–152, 2006.

[27] L. Shapira, J. Amores, and X. Benavides. Tactilevr: Integrating physical toys into learn and play virtual reality experiences. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 100–106. IEEE Computer Society, Los Alamitos, CA, USA, sep 2016. doi: 10.1109/ISMAR.2016.25

[28] D. Shor, B. Zaaijer, L. Ahsmann, S. Immerzeel, M. Weetzel, D. Eikelenboom, J. Hartcher-O'Brien, and D. Aschenbrenner. Designing haptics: Comparing two virtual reality gloves with respect to realism, performance and comfort. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 318–323, 2018. doi: 10.1109/ISMAR-Adjunct.2018.00095

[29] J. Vanek, B. Benes, A. Herout, and O. Stava. Large-scale physics-based terrain editing using adaptive tiles on the gpu. *IEEE computer graphics and applications*, 31(6):35–44, 2011.

[30] G. Voulgaris, I. Mademlis, and I. Pitas. Procedural terrain generation using generative adversarial networks. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 686–690. IEEE, 2021.

[31] P. Walsh and P. Gade. Terrain generation using an interactive genetic algorithm. In *IEEE Congress on Evolutionary Computation*, pp. 1–7. IEEE, 2010.

[32] J. Wang, O. Leach, and R. W. Lindeman. Diy world builder: an immersive level-editing system. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 195–196. IEEE, 2013.

[33] K. Yeom, J. Kwon, J. Maeng, and B. You. [poster] haptic ring interface enabling air-writing in virtual reality environment. In *2015 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 124–127. IEEE Computer Society, Los Alamitos, CA, USA, oct 2015. doi: 10.1109/ISMAR.2015.37

[34] J. Zhang, C. Li, P. Zhou, C. Wang, G. He, and H. Qin. Authoring multi-style terrain with global-to-local control. *Graphical Models*, 119:101122, 2022.

[35] Y. Zhao, H. Liu, I. Borovikov, A. Beirami, M. Sanjabi, and K. Zaman. Multi-theme generative adversarial terrain amplification. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019.

[36] H. Zhou, J. Sun, G. Turk, and J. M. Rehg. Terrain synthesis from digital elevation models. *IEEE transactions on visualization and computer graphics*, 13(4):834–848, 2007.