Terrain Synthesis for Treadmill Exergaming in Virtual Reality (Supplementary Material)

Wanwan Li
University of South Florida

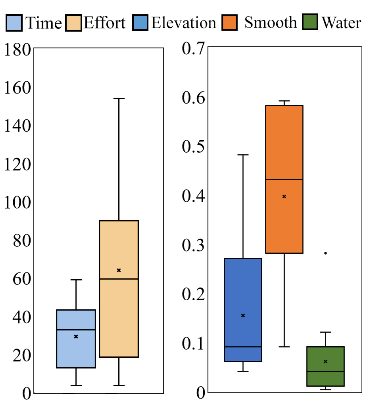
(a) Treadmill Manuel Setup. (b) VR walk experience in *City on Hill*. (c) VR walk experience in *City on Ground* .

Figure 1: User Study 2: Experimental Process Details.

1 User Study 1: Terrain Manual Design.

We recruited 15 undergraduate students to participate in this terrain manual design experiment. Every participant is asked to design a terrain using the default terrain editor embedded in Unity3D, a terrain brush that can raise or lower the terrain using custom shapes. We developed an easy-to-use plugin to help users visualize the terrain errors during their terrain design process. Those design errors are path elevation errors, smoothness errors, and water errors which are calculated in the same way as our proposed optimization approach. After explaining each type of error to the users, they will be given at most one hour to design their own terrain to match the target path as much as possible where the target path is specified as an orange curve and the generated path is a cyan curve lying on their designed terrain. During the study, we track the number of error analyses (called efforts) and the time needed to complete the design task. The assigned task is the same as the first row in Figure 8 (Main Paper) Seattle terrain case. After the study, we ask users' general feedback about their manual terrain design experiences.

Statistics. Figure 2 shows the box plot of the experimental results of User Study 1. Figure 2(a) shows the time and efforts users made during the terrain manual design process. According to the statistical analysis, the time (mins) spent by users is (AVG: 31.33; STD: 17.11; MD: 36) and the effort (number of error analyses) spent by users is (AVG: 68.26; STD: 45.26; MD: 60). Figure 2(b) shows the design errors made by the users. The elevation errors made by users are (AVG: 0.154; STD: 0.13; MD: 0.09), the smooth errors made by users are (AVG: 0.39; STD: 0.17; MD: 0.43), and the water errors made by users are (AVG: 0.05; STD: 0.07; MD: 0.04). Compared to the errors made by our optimization approach as shown in Figure 8 (Main Paper)'s first row of the Seattle terrain case which is (Elevation: 0.05; Smooth: 0.07; Water: 0.03), our automatic synthesized terrain overperforms the user's manual design significantly. At the same time, our approach only takes 2 mins to optimize the terrain in 500 iterations, which is much faster than users' average design time (about 30 mins). Please refer to more details about users' general feedback about the manual terrain design experiences in Section 4.



(a) Time and effort. (b) Design errors.

Figure 2: Statistics of Study 1.

User	Time	Effort	Elevation	Smooth	Water
1	41	60	0.1	0.5	0.01
2	18	12	0.29	0.59	0.12
3	44	92	0.09	0.09	0.06
4	39	52	0.27	0.39	0.01
5	59	154	0.05	0.58	0.01
6	13	81	0.06	0.28	0.04
7	10	87	0.48	0.13	0.01
8	30	29	0.06	0.45	0.09
9	12	15	0.08	0.43	0.09
10	36	59	0.1	0.5	0.003
11	13	5	0.29	0.59	0.12
12	59	91	0.07	0.14	0.28
13	39	52	0.27	0.39	0.01
14	44	154	0.04	0.59	0.01
15	13	81	0.06	0.28	0.04

Table 1: Detailed Result of Study 1.

2 User Study 2: Experimental Process Details

Before User Study 2, we manually set up the workout profile on the treadmill device called *The True*, the treadmill can automatically adjust its speed and inclines according to the workout profile settings as shown in Figure 1 (a). We build a VR auto-navigation program on Occulus Quest 2 VR headset. When the treadmill program and this VR program are started at the same time, the motion of the treadmill will be automatically synchronized with the virtual navigation. For example, when the treadmill's incline goes up, players automatically go up in virtual scene simultaneously. During User Study 2, we let participants try two VR programs after the treadmill device is set up

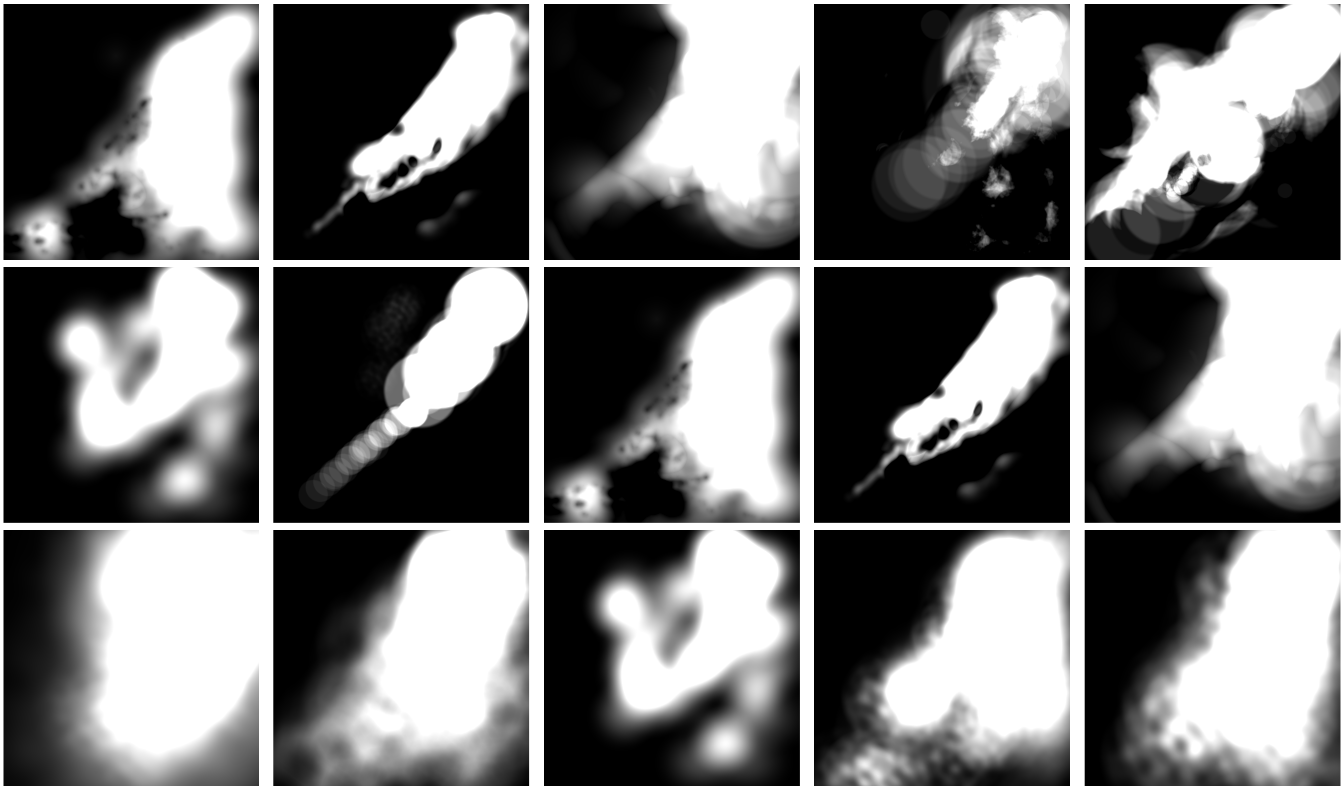


Figure 3: User Study 1: Terrain designs. Grayscale images (512x512 pixels) are the high-resolution heightmaps designed by 15 participants. It's challenging and time-consuming for users to manually design terrains that satisfy the workout profile path constraints.

User	Enjoyable	Immersive	Fun	Realistic	Relaxing
1	3	6	4	2	6
2	6	3	5	5	7
3	5	6	7	2	7
4	5	3	6	1	6
5	5	6	6	5	5
6	3	4	4	3	4
7	2	4	5	4	3
8	6	7	7	6	7
9	3	6	7	5	2
10	6	6	7	3	5

Table 2: Perception Score of Study 2: *City on Ground*.

User	Enjoyable	Immersive	Fun	Realistic	Relaxing
1	5	6	5	1	6
2	4	7	7	6	7
3	5	7	6	3	7
4	6	7	6	6	6
5	6	6	6	5	5
6	3	4	4	3	4
7	6	6	4	3	6
8	6	7	7	5	7
9	5	7	7	5	6
10	6	7	7	3	5

Table 3: Perception Score of Study 2: *City on Hill*.

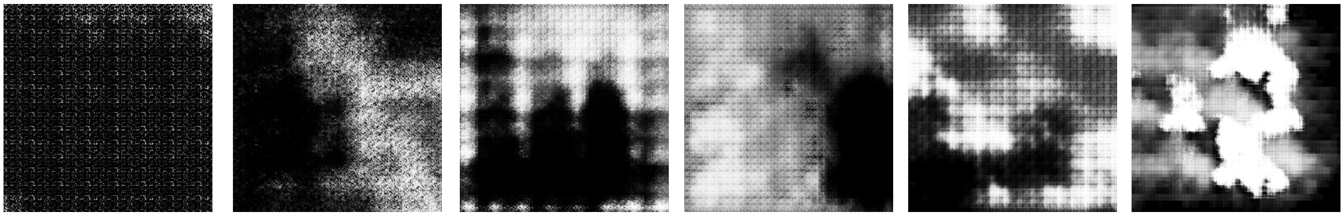
with the same workout profile: One is walking within an urban environment generated on a flat terrain called City on Ground as shown in Figure 1 (b). Another one is walking within an urban environment generated on our synthesized terrain called City on Hill as shown in Figure 1 (c). This user study is designed to investigate whether the compatibility between the elevation change in VR display and the elevation change on treadmill devices can significantly improve the user's workout experience.

3 User Study: Detailed Results

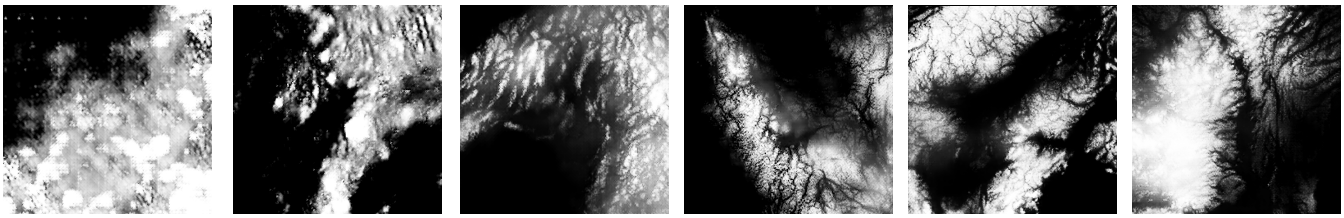
Detailed results of Study 1 are reported in Table 1 including the time and efforts users made during the terrain manual design process and the design errors made by the users. Collected data from User Study 1 include the number of error analyses (called efforts) and time complete the tasks (in minutes) and the design errors made by the users in the end, which include path elevation errors, smoothness errors, and water errors. Perception scores of Study 2 for *City on Ground* are reported in Table 2 and Perception scores of Study 2 for *City on Hill* are reported in Table 3. Details of the participant information are: For User Study 1, there are 9 males and 6 females among the 15 participants. For Study 2, among the 10 participants, there are 6 males, 4 females, and 3 have played VR games before. All of the participants are aged between 19 to 22 years old.

4 User Study 1: General Feedback

After User Study 1, we asked two questions about manual terrain design experiences including Q1: "How do you think of this designing experience in general?" and Q2: "Do you think this is an efficient way to design?". According to the user's answers to the questions, in general, from the user's experiences, it was extremely hard to consider all of these three terms together during the design,



(a) The initialization. (b) The $2K^{\text{th}}$ iteration. (c) The $4K^{\text{th}}$ iteration. (d) The $6K^{\text{th}}$ iteration. (e) The $8K^{\text{th}}$ iteration. (f) The $10K^{\text{th}}$ iteration.



(g) The $20K^{\text{th}}$ iteration. (h) The $40K^{\text{th}}$ iteration. (i) The $60K^{\text{th}}$ iteration. (j) The $80K^{\text{th}}$ iteration. (k) The $100K^{\text{th}}$ iteration. (l) The final result.

Figure 4: RaLSGAN training process. Grayscale images (512x512 pixels) show the high-resolution heightmaps generated with the RaLSGAN trained after a different amount of iterations. As shown in (a), the RaLSGAN trained before 2K iterations generate random noises. Through (b)-(l), randomly synthesized terrains look more and more realistic compared to the ground truth data from real-world terrain heightmaps.

even though some of them have a good result in minimizing the elevations error, it turns out to be a very large smoothness error in the end. However, most of them have a good result about the water error as the water region is textured as blue and is easy to check out. Some students that have the first time tried this type of design tool feels good about this designing experience in general. Someones think the controls were very intuitive and did not require a large amount of time to understand and learn. Someone even thought this designing experience was very similar to painting and creating art, so have lots of fun with it. But someone thinks it was challenging when trying to continually improve on the design and believe that the experience in a larger scale terrain design could get very difficult and messy. Someone thinks the error fluctuated greatly at some times and a lot of it was trial and error. And someone tried hard to make that terrain smooth. Some users do not think this is an efficient way to design especially on a larger scale or long term and think doing this seems very meticulous. Therefore, a conclusion that automation in terrain design process has its values and potential and our proposed approach can be applied to the terrain design for exertion games and release game designers' burden.

5 User Study 2: General Feedback

After User Study 2, we ask a few questions about users' VR walking experience in *City on Hill* and *City on Ground*. According to the user's feedback about the VR experience, someone liked the *City on Hill* that view downhill was very exciting. Someone liked the terrain a lot and thought it was cool to look back and see the path walked and the hills surrounding. Someone liked climbing the mountains and liked that the VR visual matched up to what was felt when the treadmill's elevation changed and liked how the game and treadmill interacted with the incline. Someone thought it was cool to see something moving instead of just staying in the same place while walking on a treadmill as a change of scenery helps enjoy the workout more. Some users felt very much enjoyed that can exercise while having fun. Some users think the game aspect of it makes them forget that doing something boring like exercise. Someone hoped love to see more VR games in the future based around treadmills and similar exercises (like running in place). Someone believed this game was very well in simulating a realistic walk through a city on the hill while encouraging physical activity. Some users think that walking on a treadmill is boring, however, having realistic scenery through VR helps them go through the process of using the treadmill for physical activity. In general, *City on Hill* was more impressive than *City on Ground* as more users emphasize how the compatibility between the elevation change in VR display and the elevation change on treadmill device indeed improves their workout experience. This coincides with the conclusion of the statistical test that shows VR walking experience in City on Hill is much more enjoyable, immersive, fun, realistic, and relaxing than the VR walking experience in City on Ground.

6 User Study 1: Manual Terrain Design Results

We recruited 15 student users to design a terrain using the Unity3D terrain editor extended with a plugin we developed to visualize their design errors including the path elevation errors, smoothness errors, and water errors. As shown in Figure 3, users tend to propose a trivial solution that only elevates the surrounding area of the terrain where the path is passing through to make it match with the target path. But obviously, this results in an unrealistic virtual environment. Also, lack of the terrain details makes the result not realistic too. Therefore, compared to users' manual designs, our proposed approach has an faster speed that needs ignorable manual efforts but results in fewer design errors while keeping the terrains realistic which are synthesized with well-trained RaLSGAN.

7 High-Resolution Heightmap

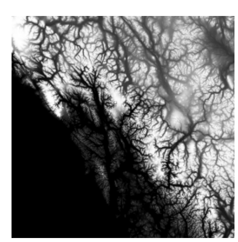


Figure 5: Sample.

In our proposed solution, 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). The training process on the SGAN is applied on two separate convolutional or fully-connected neural networks which are generator G and discriminator D. The generator captures the potential distribution of real data samples and generates new data samples, namely, fake data. The discriminator is a classifier to distinguish whether the input data is real or fake. Then the training process is achieved with an optimization process through a min-max loss function, ending with the fact that the "worse" performance of the discriminator results in the "best" performance of the generator. The loss function for discriminator D is defined as: $L_D(D,G) = \log D(x) + \log(1 - D(G(z)))$, where data x is from the real data distribution while z is from the random distribution (noise) called a latent vector, and $G(z)$ is the output of the GAN's generator. On the other side, the goal is to maximize the loss function for generator G which is $L_G(D,G) = \log(D(G(z)))$.

However, SGAN is hard to converge to a good solution for synthesizing realistic high-resolution images as the optimization process is a divergence maximization. Therefore, it becomes easier to introduce noises onto the synthesized images and even not be able to converge at all to synthesize realistic images. Therefore, by introducing a "relativistic discriminator" which estimates the probability that the given fake data is more realistic than randomly sampled real data, Relativistic GANs (RGANs) are significantly more stable and generate higher quality data samples than their non-relativistic counterparts. 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, in RaLSGAN, the Mean Square Error (MSE) loss functions for discriminator D and generator G are:

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

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

Figure 4 shows the training process of RaLSGAN. We download grayscale images (512x512 pixels) of high-resolution terrain heightmap images from the Kaggle website [1] which provides a 4GB dataset of earth terrain, height, and segmentation map images including more than 5,000 images. After a preprocessing of the raw data including adjusting the brightness and contrast, trimming to fixed sizes, redirecting to the correct folder, etc., we feed the dataset into the RaLSGAN as the real data given the parameter settings are latent Vector Z whose length is 128, 4 convolutional layers for discriminator D (learning rate=0.0001, Adam optimizer) and Generator G (learning rate=0.0025, Adam optimizer), and the batch size is 64.

As shown in Figure 4 (a), the RaLSGAN trained after 2K iterations are still generating random noises that do not look like terrain. However, after 100K iterations, the final result shown in Figure 4 (j) looks realistic compared to the ground truth data downloaded from the real-world terrain height maps. Figure 5 shows sample data in the Kaggle terrain dataset. As we can see, both heightmaps as shown in Figure 4 (j) (generated) and in Figure 5 (downloaded) contain detailed geological structures such as the hydraulic erosions. Therefore, well-trained RaLSGAN can generate different terrains by adjusting latent vector z for terrain inverse procedural modeling.

References

[1] Earth Terrain, Height, and Segmentation Map Images. Kaggle inc. Terrain heightmap images access link: `https://www.kaggle.com/tpapp157/earth-terrain-height-and-segmentation-map-images`, 2022.