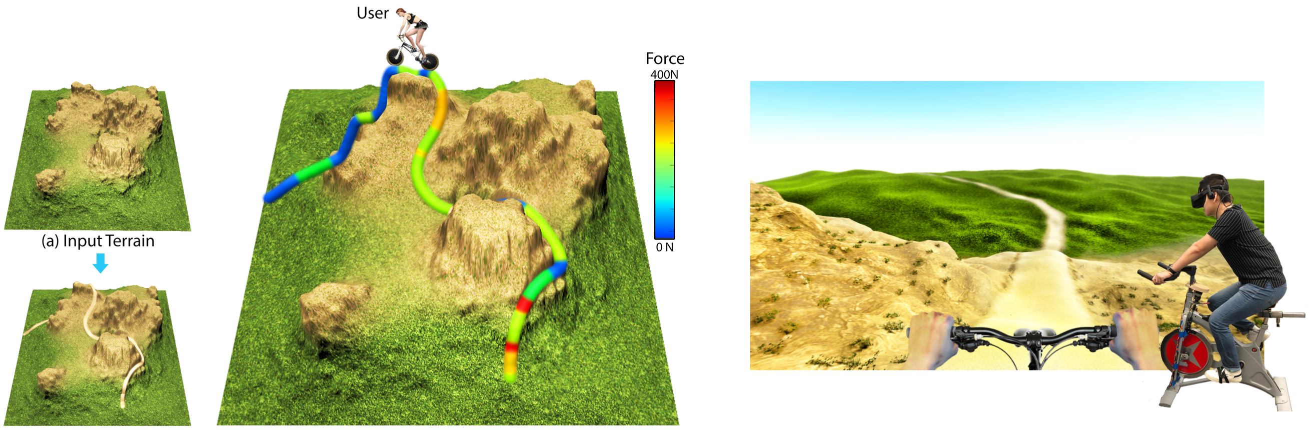
Exertion-Aware Path Generation

WANWAN LI[*], George Mason University
BIAO XIE[*], University of Massachusetts Boston
YONGQI ZHANG, George Mason University
WALTER MEISS, University of Massachusetts Boston
HAIKUN HUANG, University of Massachusetts Boston
LAP-FAI YU, George Mason University



(b) Exertion-Aware Path (c) Force Feedback (d) Exergaming on a VR Bike

Fig. 1. Given (a) a terrain as input, our approach automatically generates (b) an exertion-aware path that satisfies the user-specified exertion goals such as the the total work and perceived level of path difficulty. (c) Forces that a user experiences along the path. (d) The user bikes along the generated path in virtual reality via a VR headset and our custom-built exercise bike whose feedback force changes with the generated path, giving a highly immersive experience.

We propose a novel approach for generating paths with desired exertion properties, which can be used for delivering highly realistic and immersive virtual reality applications that help users achieve exertion goals. Given a terrain as input, our optimization-based approach automatically generates feasible paths on the terrain which users can bike to perform body training in virtual reality. The approach considers exertion properties such as the total work and the perceived level of path difficulty in generating the paths. To verify our approach, we applied it to generate paths on a variety of terrains with different exertion targets and constraints. To conduct our user studies, we built an exercise bike whose force feedback was controlled by the elevation angle of the generated path over the terrain. Our user study results showed that users found exercising with our generated paths in virtual reality more enjoyable compared to traditional exercising approaches. Their energy expenditure in biking the generated paths also matched with the specified targets, validating the efficacy of our approach.

CCS Concepts: • **Computing methodologies → Graphics systems and interfaces**; **Virtual reality**;

Additional Key Words and Phrases: procedural modeling, level design, path generation, haptics

ACM Reference Format:
Wanwan Li, Biao Xie, Yongqi Zhang, Walter Meiss, Haikun Huang, and Lap-Fai Yu. 2020. Exertion-Aware Path Generation. *ACM Trans. Graph.* 39, 4, Article 115 (July 2020), 14 pages. https://doi.org/10.1145/3386569.3392393

1 INTRODUCTION

"The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal."

— Ivan Sutherland, 1965

Ivan Sutherland, inventor of the first virtual reality (VR) head-mounted display, envisioned that the ultimate display would deliver virtual contents that are highly realistic in terms of visuals and haptics. The tremendous efforts of computer graphics researchers have led to the invention of generative models and procedural modeling algorithms capable of synthesizing visually stunning virtual contents such as streets [Chen et al. 2008], roads [Beneš et al. 2014; Galin et al. 2010; Nishida et al. 2016], terrains [Cordonnier et al. 2018; Guérin et al. 2017], cities [Parish and Müller 2001; Vanegas

[*] Equal Contributors

Authors' addresses: Wanwan Li and Lap-Fai Yu, Computer Science Department, George Mason University, 4400 University Dr, Fairfax, VA 22030, US.

0730-0301/2020/7-ART115 $15.00
https://doi.org/10.1145/3386569.3392393

et al. 2012], and worlds [Emilien et al. 2015]. However, as these virtual contents are mainly employed for visualization purposes, haptic properties such as the force feedback afforded by these virtual contents are typically not considered in their synthesis.

In this paper, we propose to generate virtual content with the consideration of haptic properties afforded by the content. More specifically, we propose an optimization-based approach to generate paths on a terrain that are both visually realistic and associated with desired exertion properties. Users can experience the generated paths in virtual reality through riding an exercise bike whose resistance is adjusted according to the elevation angle of the paths. The users' haptic feelings of the paths match with their visual perception, lead to enhanced realism and immersiveness. For example, they would feel that it is physically more demanding to ride uphill than downhill as the bike's resistance increases with the elevation angle of the generated paths. After riding the generated paths in virtual reality, users will also accomplish their exertion goals (in terms of total work done). Leveraging our exertion-aware paths and setup, biking in a virtual world resembles biking in the real world.

Our approach is motivated by the widespread popularity of exergames: a genre of games that prompt players to exercise. Successful exergame titles include Just Dance, Your Shape Fitness Evolved, and EA Sports Active, etc. With the growing popularity of consumer-grade virtual reality devices, a recent trend is to devise VR-based exergames as epitomized by recent VR exergame titles such as BoxVR, Ninja Legends, and Beat Saber. Following this trend, NordicTrack, a major gym equipment manufacturer, has introduced a VR bike that allows users to work out while biking in a virtual environment.

One major challenge in developing exergames is game level design, which is typically done manually by level designers under current practice. The design process is tedious as it involves tuning a level to satisfy multiple non-trivial goals and constraints such as helping the player achieve a certain amount of exercise, ensuring the visual realism and aesthetic quality of the game level, and constraining the game duration. Creating a plausible level typically requires many rounds of trials-and-errors. By solving the exergame level design problem as an optimization problem, our approach enables game level designers to generate exergame levels, paths with desirable exertion effects in our case, in a fast, scalable, and automatic manner. Our computational design framework also provides ample and explicit controls to designers to generate paths satisfying certain design preferences by specifying constraints. The major contributions of our work include:

- Proposing a novel problem statement of generating paths with desired exertion properties over a terrain, which can be employed for immersive virtual reality applications;
- Devising an optimization-based approach to automatically generate exertion-aware paths that help users achieve exertion goals. A generated path can be applied to control the force feedback of an exercise bike such that a user's haptic experience matches with his visual perception as he bikes along the path;
- Conducting user studies to validate the enhanced user experience and exertion effects brought by the generated paths.

2 RELATED WORK

We briefly review literature relevant to our problem statement of generating paths with the consideration of force feedback properties.

2.1 Procedural Modeling

Procedural modeling is widely applied to generate virtual world content such as terrains, rivers, roads, buildings, and cities in a scalable manner, reducing the cost of production by manual design. We refer our readers to surveys of procedural virtual world modeling [Smelik et al. 2014] and terrain modeling [Galin et al. 2019].

For road generation, Galin et al. [2010] encoded the slope of a specified terrain as well as natural obstacles as part of their objective cost functions that were procedurally minimized. Along a similar direction, Galin et al. [2011] procedurally generated hierarchical road networks that connected cites, villages, and towns given a large terrain; while Beneš et al. [2014] applied traffic simulations in the neighborhood to generate major roads in an urban setting. Moreover, Nishida et al. [2016] proposed an example-based approach to incorporate user input with its underlying terrain to generate and adapt roads in existing road networks. These research efforts mainly focus on the interaction among the urban and natural components on the given terrain instead of the human experience in navigating the generated roads. Our work focuses on the player's physical experience of the generated paths as they ride along the paths in virtual reality with a device (e.g., an exercise bike) capable of providing force feedback.

Procedure generation techniques have been applied for game design and development, from 2D platform games [Compton and Mateas 2006] to first-person games [Cardamone et al. 2011], and recently also to a ski-slalom game played on Wii [Dimovska et al. 2010] for physical rehabilitation.

While some research efforts focus on generating virtual worlds for a game, others consider gameplay experience and the player's skills for adjusting game levels automatically through algorithmic means. For example, Shi et al. [2018] and Jenning-Teats et al. [2010] proposed rule-based methods to dynamically adjust game level difficulty based on the real-time performance of the player. Similarly, Hooshyar et al. [2018] applied a data-driven approach to consider an individual's skills in generating an educational game. For a comprehensive review on game content generation, please refer to the survey by Hendrikx et al. [2013]. In contrast, our work focuses on considering the physical exertion and immersive experience of the player in procedural game generation.

2.2 Exergaming

Exergames refer to a genre of games that encourage people to exercise while playing a game. Exergames use motion sensing devices such as a depth sensor (e.g. Microsoft Kinect, PlayStation Camera) or a motion controller (e.g. Wii Remote, PlayStation Move, Nintendo Joy-Con) to capture body movement. The genre has gradually evolved from console exergames such as Microsoft Fitness and Nintendo Wii Sports to VR platform games and games played on sport equipment (e.g., VirZOOM bike). The use of motion sensing devices for game design enables the tracking and incorporation of the player's motion and physical experience in the game mechanics,

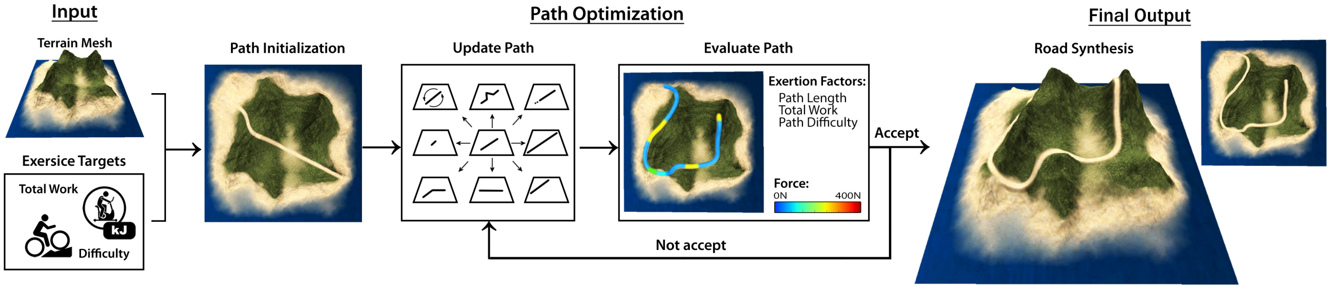


Fig. 2. Overview of our approach.

making exergames engaging as an alternative form of exercises. For example, ICAROS [2017] enables players to train different muscle groups to improve their reflex, balance, and coordination using fitness equipment while wearing a VR headset. VirZoom and CSE Entertainment developed exergaming systems that allow players to bike in virtual scenes by wearing a VR headset when using an exercise bike.

Furthermore, research has shown the positive effects of exergaming for rehabilitation and therapy such as weight control, balance enhancement, and cognitive-motor training. Bohm-Morawitz et al. [2016] and Staiano et al. [2013] studied the use of exergames to help adolescents and adults to achieve their weight loss. Padala et al. [2017] and Wüest et al. [2014] discovered that exergaming can effectively improve an elder's balance and movement performance. Ogawa et al. [2016] and Schoene et al. [2014] found that exergaming can improve cognitive and dual-task functions. Mueller et al. [2011; 2014] and Isbister et al. [2015] provided guidelines of designing exergames. One of the important remarks for the success of previous research is integrating exercises with virtual content to motivate players to achieve exercise goals.

However, most of the existing exergame content and settings are manually constructed and tested. Such a trial-and-error approach is time-consuming and expensive. Inspired by previous works [Xie et al. 2018; Zhang et al. 2019b] on assembling exergaming levels by considering human poses, we propose a computational design framework that enables path-finding on a terrain to achieve desired exertion goals. In addition, we create an engaging exertion experience that allows players to bike the generated paths in virtual reality while training their lower body strength.

2.3 Immersive Virtual Environments

Immersiveness is an important factor to consider in designing virtual reality content. Previous research [Weibel et al. 2008] found that presence, a form of immersion, is highly connected to a user's enjoyment level in a virtual reality setting. Therefore, to create a realistic virtual environment, researchers have investigated the visual [Huang et al. 2015], sound [Pfeiffer et al. 2014; Zhang et al. 2019a], and locomotion [Nilsson et al. 2014; Sra et al. 2016] aspects of a virtual environment to enhance the sense of presence for virtual experience. In particular, research on creating immersive biking experience has focused on the cycling aspect of speed [Löchtefeld et al. 2016], force feedback control [Herpers et al. 2008], and road surface texture [Rakhmatov et al. 2018]. However, the virtual environments of these works were either manually created or did not consider the physical aspects of a player in the virtual biking experience. In contrast, our approach considered the exertion and perception aspects of the virtual training experience such as the total work done and the perceived difficulty of the path as optimization terms in the path generation. Considering the physical interactions such as the force feedback that the virtual content affords in VR will make the virtual training experience more realistic, immersive, and hence more enjoyable and engaging.

2.4 Pathfinding

Tremendous amount of research on pathfinding has been conducted to overcome challenges in applications such as robotic simulation, traffic simulation and games. Algorithms based on graph search (e.g., A*), sampling (e.g., RRT), agents, and optimization, etc. have been devised for pathfinding and motion planning.

For example, Wilkie et al. [2011] used the first-in-first-out property of traffic to enable A* search algorithm to plan for large-scale vehicle routes. Karaman et al. [2011] used the anytime property of RRT* to improve efficiency during online motion planning. Best et al. [2017] proposed AutonoVi, a optimization-based maneuver planning for autonomous vehicle navigation that supports dynamic maneuvers. Huang et al. [2017] proposed an agent-based approach to consider human perception and wayfinding for generating paths. Nathan et al. [2005] proposed and applied a Partial-Refinement A* (PRA*) approach for pathfinding in real-time strategy games. Refer to the comprehensive reviews by Algfoor et al. [2015] and Gonzalez et al. [2015] on pathfinding and motion planning.

Pathfinding in an open terrain is often assumed to be isotropic; the cost of the path is independent of the traveling direction. There are some research works that investigate anisotropic pathfinding. Inspired by Galin et al. [2010], we uniformly discretize the terrain and assume our path generation problem is anisotropic such that the costs associated with the generated path depend on the traveling direction. Such costs associate physical exertion effects with geometric information such as the elevation of the path in the optimization.

3 OVERVIEW

Figure 2 shows an overview of our approach. Given a terrain, a total work target, and a target level of perceived path difficulty as input, the goal is to generate an exertion-aware path over the terrain.

The user can ride an exercise bike along the generated path whose elevation angle will be used to adjust the force feedback of the bike based on a physical model, such that the user's haptic feeling matches with his visual perception of the path. Upon completing the path, the user will have also achieved the target amount of exertion (i.e., achieved the total work target).

The core of our approach is an optimization framework. The total work target and the target level of perceived path difficulty are encoded as optimization goals. The optimization proceeds iteratively. At each iteration, the generated path will be evaluated in terms of the exertion it will induce to the user biking along the path. It will also be evaluated in terms of the perceived difficulty of the path predicted by a classifier trained with a dataset of real bike paths annotated with difficulty ratings. The optimization proceeds until a path that satisfies all the desired properties is generated. Our approach then synthesizes a road surface along the generated path and slightly refines the terrain to make the cross-section of the road surface horizontal for a natural riding experience.

The user can then bike along the generated path to achieve his exertion goal, using a VR bike whose force feedback is controlled by the elevation angle of the generated path.

4 TECHNICAL APPROACH

4.1 Problem Formulation

4.1.1 Representation. Figure 3 depicts the representation of our problem. The goal of our approach is to generate a path $\mathcal{P}$ on the input 3D terrain by optimization to satisfy the desired properties.

Akin to Galin et al. [2010], on the xz-plane we construct a 2D Hermite curve $\mathcal{R}$ which is mapped to the 3D terrain to generate the path $\mathcal{P}$. The xz-plane is overlaid with an $N \times N$ grid called path grid $\mathbf{M}$. The 2D Hermite curve $\mathcal{R}$ is constructed by a sequence of control points. Each control point is located at a grid intersection as shown in Figure 3. For notation convenience, we also use $\mathbf{M}$ to denote an $N \times N$ occupancy matrix of the path grid. An entry of $\mathbf{M}$ stores the index of a control point that is located at the entry's corresponding grid intersection on the xz-plane; otherwise it stores a value of -1 to indicate that no control point is located there. We use $N = 20$ by default in our implementation.

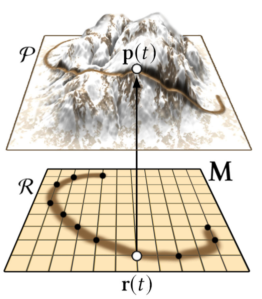


Fig. 3. Problem representation. Black dots refer to curve $\mathcal{R}$'s control points.

The 2D Hermite curve $\mathcal{R}$ is C^1 continuous. We parameterize curve $\mathcal{R}$ such that a point on curve $\mathcal{R}$ is denoted by $\mathbf{r}(t)$, where $t \in [0, 1]$. $\mathbf{r}(0)$ and $\mathbf{r}(1)$ refer to the start point and end point of curve $\mathcal{R}$.

The 2D curve $\mathcal{R}$ on the xz-plane is mapped to the path $\mathcal{P}$ on the surface of the 3D terrain using a height map function H of the terrain. Every 2D point $(\mathbf{r}_x(t), \mathbf{r}_z(t))$ on curve $\mathcal{R}$ on the xz-plane has its corresponding elevation $H(\mathbf{r}_x(t), \mathbf{r}_z(t))$ on the 3D terrain. Upon mapping and parameterization, the path $\mathcal{P}$ on the 3D terrain is given by $\mathbf{p}(t) = (\mathbf{r}_x(t), H(\mathbf{r}_x(t), \mathbf{r}_z(t)), \mathbf{r}_z(t))$, where $t \in [0, 1]$. $\mathbf{p}(0)$ and $\mathbf{p}(1)$ refer to the start point and end point of path P.

4.1.2 Optimization Objective. The goal of the optimization is to find a path $\mathcal{P}$ that minimizes the total cost function $C_{\text{total}}(\mathcal{P})$ which evaluates the overall error of the path $\mathcal{P}$ generated through path grid $\mathbf{M}$. The total cost function $C_{\text{total}}(\mathcal{P})$ is defined as:

$$C_{\text{total}}(\mathcal{P}) = w_{\mathrm{L}} C_{\mathrm{L}}(\mathcal{P}) + w_{\mathrm{W}} C_{\mathrm{W}}(\mathcal{P}) + w_{\mathrm{D}} C_{\mathrm{D}}(\mathcal{P}), \tag{1}$$

where the path length cost $C_{\mathrm{L}}(\mathcal{P})$ encodes the prior path length; the total work cost $C_{\mathrm{W}}(\mathcal{P})$ encodes the target amount of total work induced on the user for completing path $\mathcal{P}$ on a VR bike; and the path difficulty cost $C_{\mathrm{D}}(\mathcal{P})$ encodes the target level of perceived path difficulty with respect to the variation of the path elevation throughout the biking experience. The w_{L}, w_{W}, and w_{D} represent the respective blending weights of these three cost terms. We provide details of the costs in the following section.

4.2 Cost Terms

Our optimization framework considers a number of cost terms to guide the generation of a path to carry certain exertion-related properties. While our framework is extensible to incorporate additional considerations and constraints, we focus on three cost terms related to the exercise duration, how much total work the generated path induces on the user, and the perceived level of difficulty during the biking experience. Interestingly, people usually consider factors like these when choosing a bike path in the real world to exercise.

4.2.1 Path Length Cost. Different players may prefer different exercise duration for achieving different training goals. Given a certain biking speed, the duration of the exercise depends on the total length of the path. To adjust the length of the path, we introduce the path length cost which measures the difference between the length $L(\mathcal{P})$ of the generated path and a path length target ρ_{L}. Path length cost $C_{\mathrm{L}}(\mathcal{P})$ is defined as:

$$C_{\mathrm{L}}(\mathcal{P}) = 1 - \exp(-\left(\frac{L(\mathcal{P}) - \rho_{\mathrm{L}}}{\sigma_{\mathrm{L}}}\right)^2), \tag{2}$$

where we empirically set $\sigma_{\mathrm{L}} = 2\rho_{\mathrm{L}}$. $L(\mathcal{P})$ is the length of path $\mathcal{P}$ calculated through line integral:

$$L(\mathcal{P}) = \int_{\mathcal{P}} d\mathbf{p} = \int_0^1 |\mathbf{p}'(t)| dt \tag{3}$$

4.2.2 Total Work Cost. Another important consideration is the total work done for completing path $\mathcal{P}$ by riding on a VR bike. Different types of exercises induce different total works. For example, cardiovascular exercises typically expend more energy than weight-training exercises do per training session [Tinsley 2017]. It is challenging for a level designer to manually tune a level (a path in our case) to associate it with an arbitrary total work target as it involves non-trivial physics considerations of the force and movement of the user during the exergame.

To incorporate such a consideration, we formulate the expected total work $W(\mathcal{P})$ for a generated path $\mathcal{P}$ according to a physics model and compare it with the desired total work target ρ_{W}. The total work cost $C_{\mathrm{W}}(\mathcal{P})$ is defined as:

$$C_{\mathrm{W}}(\mathcal{P}) = 1 - \exp(-\left(\frac{W(\mathcal{P}) - \rho_{\mathrm{W}}}{\sigma_{\mathrm{W}}}\right)^2), \tag{4}$$

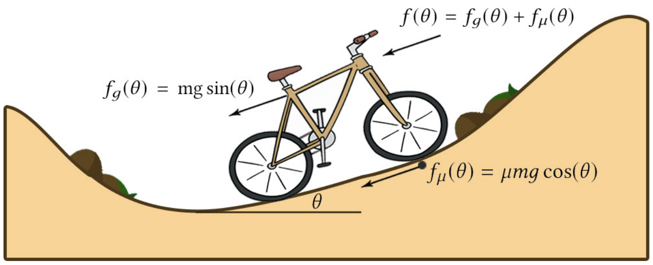

$f(\theta) = f_g(\theta) + f_\mu(\theta)$
$f_g(\theta)$ = mg sin(θ)
$f_\mu(\theta) = \mu mg\cos(\theta)$
θ

Fig. 4. Resistance force. A bike experiences a resistance force f which varies with the path elevation angle θ. The resistance force is composed of the gravitational force f_g and frictional force f_μ.

where we empirically set $\sigma_W = 2\rho_W$. $W(\mathcal{P})$ is the total work of completing path $\mathcal{P}$ calculated through line integral:

$$W(\mathcal{P}) = \lambda \int_{\mathcal{P}} f(t)d\mathbf{p} = \lambda \int_0^1 f(t)|\mathbf{p}'(t)|dt, \tag{5}$$

where $f(t)$ is the resistance force experienced by the user along the path $\mathbf{p}(t)$ as the path elevation angle $\theta(t)$ changes. From the directional derivative of the height map function H, we have $\theta(t) = \tan^{-1}\left(\frac{\mathrm{d}\mathbf{r}(t)}{\mathrm{d}t} \cdot \nabla H(\mathbf{r}(\mathbf{t}))\right)$.

λ is a parameter for scaling the distance of the path in the virtual world down to the real moving distance of a pedal of the exercise bike that the user rides. Suppose a path in the virtual world is 1,000m long, with $\lambda = 0.1$ set by default, the user will just need to ride 100m on the exercise bike to complete the path.

According to the physics model shown in Figure 4, a bike traveling on a slope would experience a resistance force f composed of the gravitational force f_g and the frictional force f_μ. The resistance force varies with the path elevation angle θ. By decomposition of forces, it is calculated as:

$$f(\theta) = mg\sin(\theta) + \mu mg\cos(\theta), \tag{6}$$

where m is the mass; g is the gravitational constant; and μ is the friction coefficient of the terrain surface.

In practice,the resistance force $f(t)$ in (5) is simulated by the feedback force of an exercise bike that the user rides. The exercise bike can give a discrete set of feedback forces $F = \{f_i\}$ through its pedals. Figure 5 (bottom) shows the feedback forces $\{f_1, ...f_6\}$ of the exercise bike used in our experiments, which are designed by the bike's manufacturer.

For simplicity, we assume mass $m = 70$kg which is slightly above the average global body mass [Walpole et al. 2012]. We want to associate biking on a horizontal path (i.e., $\theta = 0$) with a small feedback force (f_2) of the exercise bike. In this case, the user only experiences the frictional force as the resistance force according to Figure 4. Using (6), we calculate the friction coefficient $\mu = \frac{f_2}{mg}$.

As the user bikes along the path $\mathbf{p}(t)$, we compute the resistance force $f(\theta)$ from (6) that he should experience in reality according to the physics model. Then a feedback force $f_i \in F$ that is closest to $f(\theta)$ is given by the exercise bike to approximate the resistance force $f(\theta)$. Essentially, the user experiences a feedback force f_i that increases with the path elevation angle realistically.

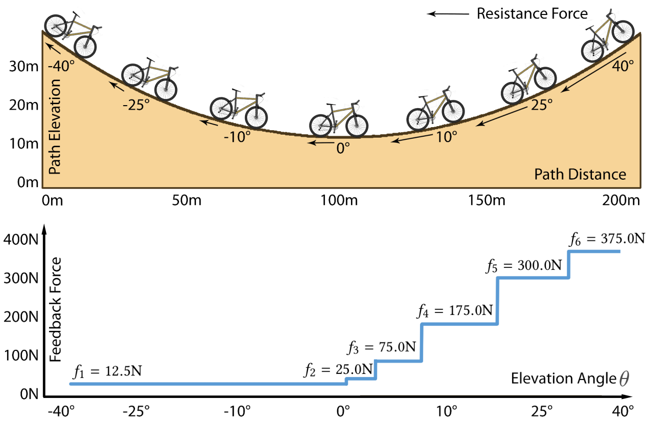


Fig. 5. Top: A path whose elevation angle changes smoothly from −40° to 40°. Bottom: the force feedback of the exercise bike used to approximate the resistance force experienced at the corresponding elevation angle.

As the resistance force $f(t)$ is simulated by the feedback force f_i of the exercise bike, for computing the total work cost $C_W(\mathcal{P})$, we set $f(t) = f_i$ to calculate the total work $W(\mathcal{P})$ in (5).

Figure 5 depicts the feedback force assigned according to the path elevation angle. For going uphill, we consider elevation angle up to 40° that corresponds to a 84% gradient and is very challenging [Neef 2013]. For going downhill ($\theta < 0°$), the user experiences a net force pushing him downwards according to (6). We assume that the user only rides using a very small force in this case that corresponds to the minimum feedback force f_1 of the exercise bike.

Note that we could also associate biking on a horizontal path with a feedback force other than f_2 of the exercise bike. For example, if we want to simulate biking on a rough surface (e.g., a muddy road) with a high friction, we could associate it with a higher feedback force level (e.g., f_3). In this case, we will obtain a larger μ and the user will experience a larger simulated frictional force.

4.2.3 *Path Difficulty Cost.*

The bumpiness of a path is related to the perceived level of biking difficulty. For example, a trail on a bumpy mountain is usually perceived as more difficult to bike than a trail on a plain. Websites such as the MTB Project collect people's ratings of the biking difficulty of different trails.

As the perception of biking difficulty is subjective, we formulate the path difficulty cost by using a classifier trained with difficulty ratings of real-world trails given by bikers. This cost measures the difference between the perceived difficulty $D(\mathcal{P})$ of the generated path and the desired path difficulty target ρ_D. It is defined as:

$$C_D(\mathcal{P}) = 1 - \exp(-\left(\frac{D(\mathcal{P}) - \rho_D}{\sigma_D}\right)^2), \tag{7}$$

where we empirically set $\sigma_D = 0.25$ in our implementation. $D(\mathcal{P}) \in [0, 1]$ is the perceived difficulty of path $\mathcal{P}$ evaluated using the classifier, with a larger value referring to a more difficult path.

To train a classifier to predict the difficulty of a path, we collected more than 30,000 paths from the MTB Project contributed by passionate bikers excited to share their knowledge of local biking trails with others. The difficulty of a path is voted by MTB users.

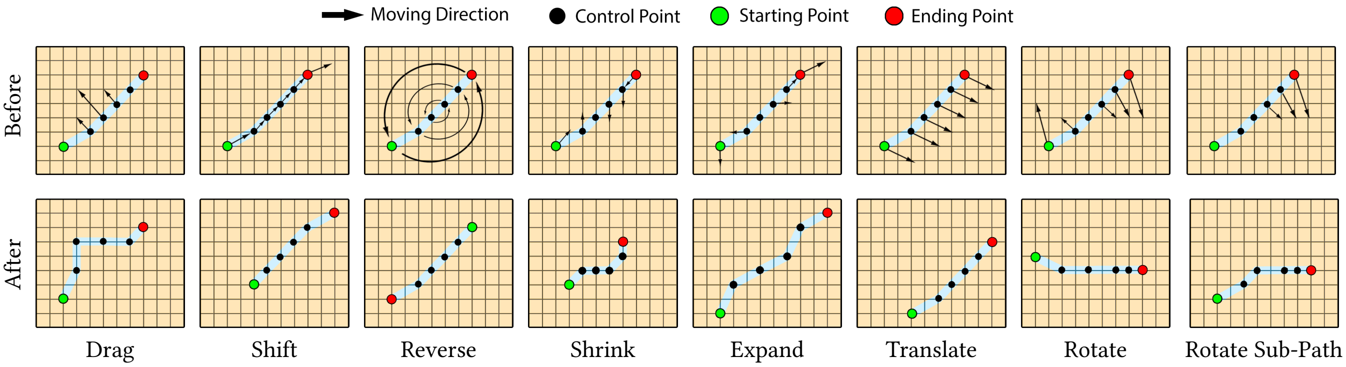


Fig. 6. Move strategies. Each move is applied to the control points of the 2D Hermite curve $\mathcal{R}$ on the path grid. The arrows indicate the moving directions of the control points. Each move modifies curve $\mathcal{R}$, which in turn modifies path $\mathcal{P}$ on the 3D terrain.

The difficulty ratings range from 1 to 6, with 1 meaning easy and 6 meaning hard. The number of votes for a path varies from 1 to 100. We dropped the paths with fewer than 10 votes.

We computed the average difficulty of each path using its votes. We then grouped the paths into two difficulty groups: easy and hard. The easy group contains paths with an average difficulty rating lower than 3; the hard group contains paths with an average difficulty rating higher than 4. We ignored paths with an average difficulty rating between 3 and 4 as their difficulty levels are ambiguous. Finally, we obtained 1, 000 easy paths and 1, 000 hard paths.

Each collected path is encoded as a sequence of sample points denoting the elevation along the path. The number of sample points of a path varies from 100 to 1, 000. A sample point is taken every 0.1 mile approximately. Figure 7 visualizes the two groups of paths. Visually, the hard paths tend to be bumpier than the easy paths.

As the paths have different lengths, to align the paths for training, we extracted 11 features from each path:

- Path length;
- Highest and lowest elevations of the path;
- Sum of ascent (elevation increase) along the path;
- Sum of descent (elevation decrease) along the path;
- Minimum and maximum curvature;
- Accumulative curvature;
- Number of peaks and valleys;
- Variance in the elevations of the sample points of the path.

We trained a feedforward artificial neural network for predicting the difficulty of a path based on its features. Each feature was normalized to $[0, 1]$. The neural network has 1 hidden and fully-connected layer with 15 hidden units. The neural network has 11 trainable parameters. It uses RELU as the activation function and SGD as the loss function. We randomly selected 500 paths from each difficulty group to construct the training dataset, and used the remaining paths as the testing data. The prediction accuracy of our classifier is about 82%. Each prediction also gives a probability that indicates how likely the input path is classified into the hard group. For example, if the probability of a given path is 0.2, it means that the path has a 20% chance to be classified as a hard path and a 80% chance to be classified as an easy path. We use this probability as a metric $D(\mathcal{P})$ to evaluate the difficulty of a generated path $\mathcal{P}$.

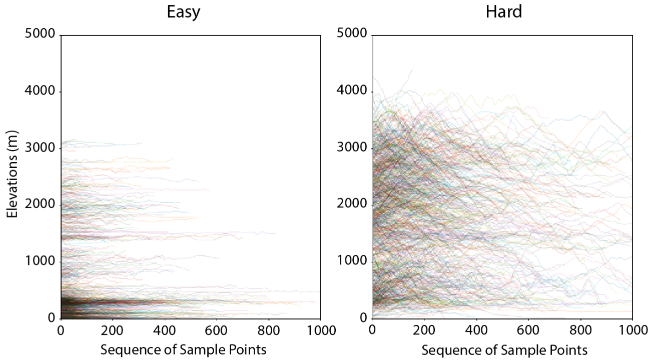


Fig. 7. Elevations of the paths from the easy and hard groups. The paths correspond to real-world trails of the MTB Project. The elevation along each path is shown. The hard paths are generally bumpier than the easy paths.

Our classifier is able to correctly classify 86% of the easy paths and 78% of the hard paths. We used a neural network as our classifier it achieved a higher prediction accuracy and robustness compared to alternative classifiers such as random forests and SVM in our preliminary tests.

4.3 Path Optimization

4.3.1 Initialization. We initialize the path grid $\mathbf{M}$ to generate a random line, which contains a sequence of control points of a 2D Hermite curve, near the center of the grid. To ensure that the random line stays within the terrain boundary, we consider both the target path length ρ_{L} and the terrain width ω. The coordinates of the center (x_c, y_c) of the line are randomly initialized in the range $[\frac{\rho_{\mathrm{L}}}{2}, \omega - \frac{\rho_{\mathrm{L}}}{2}]$. By selecting a random angle $\phi \in [0, \pi]$, we generate the start point of the line as $(x_c - \frac{\rho_{\mathrm{L}}}{2}\cos(\phi), x_c - \frac{\rho_{\mathrm{L}}}{2}\sin(\phi))$ and the end point as $(x_c + \frac{\rho_{\mathrm{L}}}{2}\cos(\phi), x_c + \frac{\rho_{\mathrm{L}}}{2}\sin(\phi))$. The control points on this line are then used to construct a 2D Hermite curve $\mathcal{R}$ that is mapped to the 3D terrain to generate an initial path $\mathcal{P}$ as described in Section 4.1.

4.3.2 Process. The optimization proceeds iteratively. At each iteration, our approach applies a move to modify the control points of the 2D Hermite curve $\mathcal{R}$ so as to modify curve $\mathcal{R}$ and the path $\mathcal{P}$

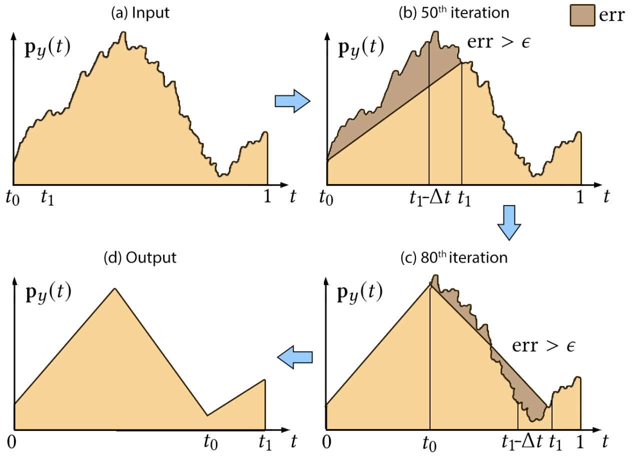


Fig. 8. Attaining high coplanarity of local road surfaces through our post-processing method. (a) Input terrain that is bumpy. (b) At the 50th iteration, as the accumulated approximation error exceeds the threshold ϵ, the terrain between t_0 and $t_1 - \Delta t$ is approximated by a linear segment. t_0 is then reset to $t_1 - \Delta t$ to start searching for the next linear segment. (c) At the 80th iteration, as the accumulated error exceeds the threshold ϵ again, the next linear segment is formed. (d) The output is an approximation of the input terrain by a series of linear segments.

on the 3D terrain. The move is randomly selected from one of the eight different types of moves shown in Figure 6:

- **Drag:** Some control points in the middle are dragged aside by a random amount so that the curve is dragged like a spring.
- **Shift:** The first and last control points move forward or backward by a random amount so that the curve shifts like a snake.
- **Reverse:** The first and last points of the curve swap with each other, reversing the travel direction of the path.
- **Shrink:** The first and last control points move towards each other to shrink the curve.
- **Expand:** The first and last control points move away from each other to elongate the curve.
- **Translate:** All control points translate by the same amount along a random direction so that the curve translates.
- **Rotate:** The whole curve is rotated with a random angle.
- **Rotate Sub-Path:** One side of the curve is rotated with a random angle while the other side stays.

To constrain the solution to be valid, we drop a move if it results in a 2D Hermite curve that is out of the boundary of the path grid or is self-colliding. To regularize the resulting curve to be smooth, a move is also dropped if the resulting curve makes a sharp corner, i.e., there exists control points (x_{k-1}, y_{k-1}), (x_k, y_k), and (x_{k+1}, y_{k+1}) such that $[x_k - x_{k-1}, y_k - y_{k-1}] \cdot [x_{k+1} - x_k, y_{k+1} - y_k] > 0$.

We use the Metropolis Hastings algorithm [Chib and Greenberg 1995] with a simulated annealing [Kirkpatrick et al. 1983] state-searching step to optimize the path. At each iteration, a new path $\mathcal{P}'$ is proposed by applying a randomly-selected move on the current path $\mathcal{P}$ as described above. The new path $\mathcal{P}'$ is accepted with a probability defined based on the Metropolis criterion:

Algorithm 1 Turning Points Extraction

```
function LinearApproximationError($\mathbf{p}_y(t), t_0, t_1$)
    totalError = $\int_{t_0}^{t_1} \left| \mathbf{p}_y(t) - \left( \mathbf{p}_y(t_0) + (t - t_0) \frac{\mathbf{p}_y(t_1) - \mathbf{p}_y(t_0)}{t_1 - t_0} \right) \right| \mathrm{d}t$
return totalError/$(t_1 - t_0)$
procedure Linear Approximation($\mathbf{p}_y(t), \epsilon$)    ▷ $\epsilon$: max error
    $t_0 \leftarrow 0$    ▷ 0: time stamp of start point
    $t_1 \leftarrow \Delta t$    ▷ $\Delta t$: a small time step
    $T \leftarrow \{0\}$    ▷ $T$: a set of time stamps of turning points
    while $t_1 < 1$ do
        err ← LinearApproximationError($\mathbf{p}_y(t), t_0, t_1$)
        while $t_1 < 1 \wedge$ err $< \epsilon$ do
            $t_1 \leftarrow t_1 + \Delta t$    ▷ $t_0$ stays, $t_1$ moves forward
            err ← LinearApproximationError($\mathbf{p}_y(t), t_0, t_1$)
        $t_0 \leftarrow t_1 - \Delta t$, $T \leftarrow T \cup \{t_0\}$    ▷ add new turning point
    return $T \cup \{1\}$    ▷ add time stamp of end point
```

$$Pr(\mathcal{P}'|\mathcal{P}) = \min\left(1, \frac{\gamma(\mathcal{P}')}{\gamma(\mathcal{P})}\right), \tag{8}$$

where $\gamma(\mathcal{P})$ is a Boltzmann-like function comprising the total cost:

$$\gamma(\mathcal{P}) = \exp(-\frac{1}{t} C_{\text{total}}(\mathcal{P})), \tag{9}$$

and t is the temperature parameter of simulated annealing, which drops from an initial high value (we set it to 1 empirically) to a low value near zero over the iterations. Essentially, the initial high temperature allows the optimizer to explore the solution space more aggressively with a high acceptance probability. As the temperature decreases over iterations, the optimizer becomes more greedy and it is less likely to accept worse paths than before. The optimization terminates if the absolute change in $C_{\text{total}}(\mathcal{P})$ is smaller than 3% over the past 50 iterations.

4.3.3 Parameter Settings. By default, we set the weights as $w_{\text{L}} = 0.5$, $w_{\text{W}} = 0.3$, and $w_{\text{D}} = 0.2$. The targets ρ_{L}, ρ_{W}, and ρ_{D} are set according to the specific goals of the experiments.

Note that we attempted solving the optimization using CMA-ES similar to some existing works [Al Borno et al. 2012; Naderi et al. 2017]. However, we found that due to the complex optimization landscape and the curse of dimensionality (there were about 30 2D control points), the optimization was sensitive to the initialization of the first population in CMA-ES and was also memory-intensive. The optimizer tended to get trapped at a poor local optimum if the initialization was poor. Therefore, we discretize the solution space to reduce the problem complexity and apply simulated annealing with flexible moves to extensively explore the solution space. A path is optimized in a few minutes (<5,000 iterations) typically.

4.4 Road Synthesis

With a generated path $\mathcal{P}$, our approach generates a road along this path on the terrain. The user bikes on this road in virtual reality. To make this road visually appealing and to deliver a natural and comfortable biking experience, we incorporate design concepts, namely, coplanarity and horizontality, from road engineering [Douglas 2016;

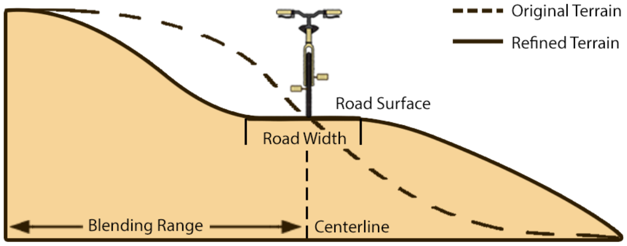


Fig. 9. A terrain is refined to create a horizontal road surface.

Hayes 2015], in constructing this road. The terrain is also slightly refined in forming this road.

4.4.1 Coplanarity of Local Road Surfaces. Figure 8 shows the effects of maintaining the coplanarity of local road surfaces. While path $\mathcal{P}$ is generated on the terrain whose surface could be bumpy, the road built on top of the path should be locally linear to prevent the user from biking up and down too frequently, which may result in dizziness. Such local linearity also suits the VR exercise bike hardware as it prevents the bike's force feedback from changing too frequently, which may wear out the motor otherwise.

The key idea for achieving high coplanarity of local road surfaces is to approximate the path with a series of linear segments, each of which is associated with a coplanar road surface. To this end, we devise a post-processing method to refine the elevation of the original path. A new path $\mathcal{P}'$ is formed which approximates the original path $\mathcal{P}$ by a number of linear segments. More specifically, the new path's elevation $\mathbf{p}'_y(t)$ is an approximation of the original path's elevation $\mathbf{p}_y(t)$. The new path resembles the original path by keeping the key features of the original path such as its major turning points. The road is built on top of the new path $\mathcal{P}'$.

The post-processing method consists of two major steps: (1) extracting the major turning points from the original path following Algorithm 1; (2) and linearly interpolating the extracted turning points to form a new path $\mathcal{P}'$ to approximate the original path $\mathcal{P}$. Figure 8 illustrates this process.

Note that while Section 4.2 uses the original path $\mathcal{P}$ for deriving the costs, we use the new path $\mathcal{P}'$ instead of the original path $\mathcal{P}$ to evaluate the path length cost in (2) and the total work cost in (4). That is, we evaluate $C_\mathrm{L}(\mathcal{P}')$ and $C_\mathrm{W}(\mathcal{P}')$ instead of $C_\mathrm{L}(\mathcal{P})$ and $C_\mathrm{W}(\mathcal{P})$ in optimizing the paths for our experiments. This is because we had our user bike along the new path $\mathcal{P}'$ rather than the original path $\mathcal{P}$ in our experiments to avoid the high bumpiness of the input terrain. Using $\mathcal{P}'$ for evaluating these costs is then more accurate. If the designer would rather generate a road closely following the bumpy terrain and did not apply the post-processing method for linear approximation, for example, for simulating a vigorous mountain biking experience, the original path $\mathcal{P}$ could have been used for evaluating the costs.

4.4.2 Horizontality of Road Cross Slope. As shown in Figure 10, a tilted road would cause a bike to slide or even flip on its side due to gravity. The cross slope of a road refers to the road's slope taken across the road's centerline. It is important to keep the cross slope of a road horizontal. Inspired by the geometric design of roads, we incorporate such a consideration in creating a realistic road along the path for delivering a natural biking experience.

To create a road with a horizontal cross slope, we slightly refine the terrain along the road by a blending operation (Figure 9). More specifically, we design an α-map blending function for refining the input terrain's height map $H(x,z)$ so that the terrain is flat along the cross slope of the road; and the sides (within a certain blending range) of the road blend smoothly into the natural landscape.

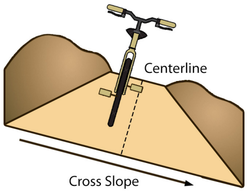


Fig. 10. A tilted road.

We make the road's cross slope horizontal by setting the terrain near the center of the road to have the same height as the elevation of the path $\mathcal{P}'$. As the elevation of path $\mathcal{P}'$ is $\mathbf{p}'_y(t)$, the new height map $H'(x,z)$ of the terrain after blending is:

$$H'(x,z) = \alpha(x,z)\mathbf{p}'_y(t^*(x,z)) + (1-\alpha(x,z))H(x,z), \qquad (10)$$

where $t^*(x,z) = \operatorname{argmin}_t |\mathbf{r}(t) - (x,z)|$ corresponds to the point $\mathbf{r}(t)$ on the 2D Hermite curve $\mathcal{R}$ that is closest to the point (x,z) on the xz-plane. $d(x,z) = \min(|\mathbf{r}(t) - (x,z)|)$ denotes the minimal distance from pixel (x,z) to the 2D Hermite curve $\mathcal{R}$. The α-map blending function $\alpha(x,z)$ is defined as:

$$\alpha(x,z) = \begin{cases} 1 & d(x,z) < w \\ 0 & d(x,z) > \kappa \\ \frac{1}{2} + \frac{1}{2}\cos(\frac{d(x,z)-w}{\kappa-w}\pi) & w \le d(x,z) \le \kappa, \end{cases} \qquad (11)$$

where w is the road's half width and κ denotes a blending range. The blending decays from 1 to 0 as $d(x,z)$ increases from w to κ.

5 EXPERIMENTS

5.1 Implementation

We implemented our approach using an Alienware machine equipped with an Intel Core i7-9700 CPU, an NVIDIA GeForce RTX 2070 graphics card, and 32GM of RAM. The implementation was done in Python. A path is generally optimized in about 1,000 to 5,000 iterations, depending on input terrain and the specified targets which affect the optimization difficulty. Optimizing a path generally takes about 1 to 5 minutes on our machine.

5.1.1 Custom-built Bike. Figure 12 shows the custom-built bike used in our user study. It was built based on a Schwinn Evolution SR Indoor Cycle Bike. We built an Arduino MEGA2560 R3 with a motor to control the resistance and sensors to monitor the cadence. The resistance level of the original bike could be adjusted by turning the resistance control knob. We replaced the manufacturer's knob with our own stepper motor that was controlled by the Arduino. Then the Arduino could adjust the resistance according to the elevation angle of the generated path to give the desired feedback force. As the force feedback is

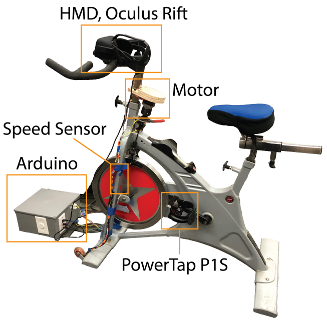
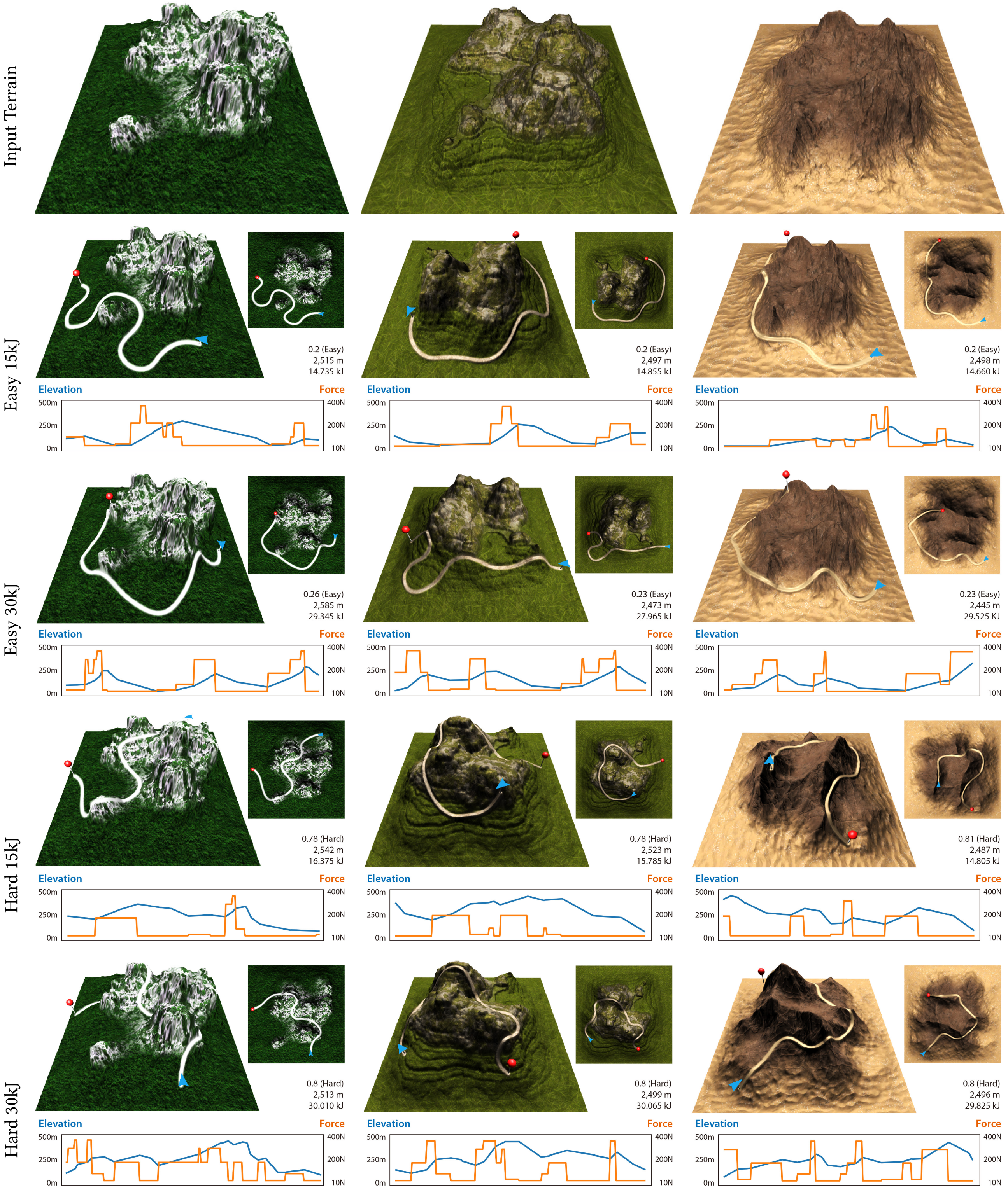
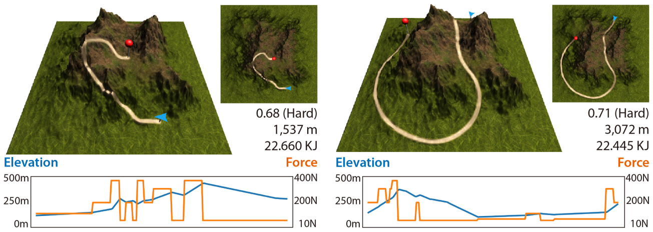


Fig. 12. Our custom-built bike.



(a) *Rocky Hill* (b) *Terrace Hill* (c) *Desert Hill*

Fig. 11. Paths generated on different terrains with different exertion goals. In each row, the path difficulty target (e.g., easy) and total work target (e.g., 15kJ) are shown on the left. The path length target is 2, 500m for all. The path difficulty, path length, total work, as well as the elevation and feedback force plots of each generated path are shown. The blue arrow and the red pin refer to the start and end.



(a) Short Path

(b) Long Path

Fig. 13. A long path and a short path generated using long and short path length targets on the same input terrain. Both paths used the same total work target $\rho_W = 22.5$kJ. Our approach generates (a) a short uphill path which is more intensive; and (b) a long downhill path which is less intensive.

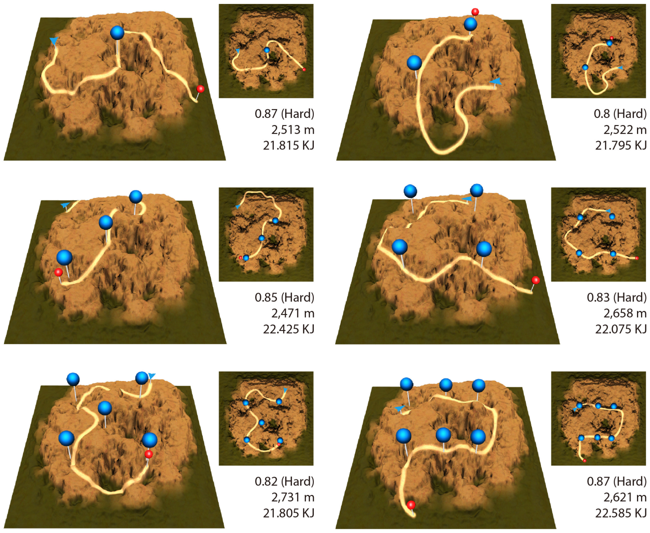


Fig. 14. Paths generated with specified landmarks (in blue). All paths were generated with the same path length target $\rho_L = 2,500$m and total work target $\rho_L = 22.5$kJ. The generated paths pass the landmarks closely.

tuned by a resistance knob controlled by a stepper motor, the motor speed determines the speed of change of force feedback. We tuned our motor to avoid sudden changes. Using more discrete levels used can also smoothen the transition of force feedback between different levels.

A reed switch was mounted on the bike's frame near the wheel to measure the speed. Two magnets were attached to the wheel, triggering the reed switch when they passed. This allowed speed measurement by the Arduino with a high degree of accuracy. The Arduino sent the speed readings to the computer through serial ports so that our program could move the user's bike in virtual reality according to the measured speed.

Since we wanted to validate the effectiveness of our approach, we replaced the original pedals with PowerTap P1S power meter pedals. Per the manufacturer's instructions, the left pedal was used for measuring the power output of a user as he was biking. The original bike allowed a total of 10 levels of feedback forces. However, we dropped the first three levels as the forces were very small and almost identical. We also dropped the last level as it was very heavy to ride. Figure 5 shows the magnitudes of the levels of feedback forces used.

5.1.2 Terrain Generation. To conduct our experiments, we generated different terrains including a *Rocky Hill*, a *Terrace Hill*, and a *Desert Hill*. We used a terrain generator called World Creator, which can generate, design, blend, mix, paint sculpt, erode and simulate realistic terrains, to generate the height maps of the terrains. As one feature of the World Creator, different types of terrains can be generated through different filters that carry specific geological properties. For example, the *Rocky Hill* was generated using the canyon filter, rocky-hills filter, smooth slope filter, and the zero-edge filter.

For realistic rendering, we simulated the appearance of the terrain using three different levels of textures, namely, the road texture, the plain texture, and the mountain texture. We used an alpha blending function similar to that described in Section 4.4 to blend the road texture smoothly into the texture of the natural landscape (i.e., mountain or plain) near the sides of the road.

The path $\mathcal{P}'$ generated in Section 4.4 that approximates the original path $\mathcal{P}$ by a number of linear segments may not be smooth, resulting in corners between linear segments. While we use the elevation angles from path $\mathcal{P}'$ to control the discrete feedback force of the bike following Section 4.2, for visual appeal only, we smooth the corners by fitting small quadratic Bézier curves at the corners.

5.2 Different Terrains and Exertion Goals

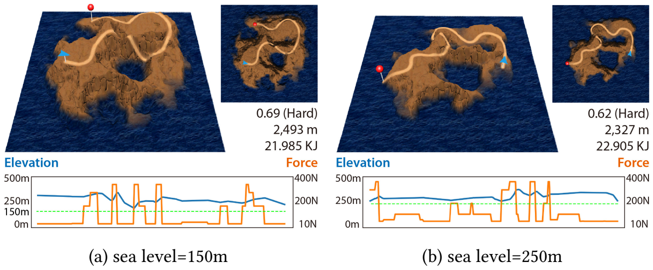
We experimented our approach with optimizing paths on three different terrains. Each generation used the same path length target $\rho_L = 2,500$m. The paths were generated using four combinations of exertion goals:

(1) Easy and Low Work ($\rho_D = 0.2$, $\rho_W = 15$kJ);
(2) Easy and High Work ($\rho_D = 0.2$, $\rho_W = 30$kJ);
(3) Hard and Low Work ($\rho_D = 0.8$, $\rho_W = 15$kJ);
(4) Hard and High Work ($\rho_D = 0.8$, $\rho_W =30$kJ);

Figure 11 shows the terrains and the generated paths. All generated paths have a length which is close to the path length target of 2,500m. As the plots below each sub-figure show, the feedback force pattern changes with the path elevation correspondingly. The paths generated with larger total work targets tend to have larger average forces than the ones generated with smaller targets. On the other hand, given the same total work target, paths generated with different path difficulties tend to have different distributions on the same terrain: Easy paths tend to lie on the plain area where the elevations vary in smaller scales; hard paths tend to lie on the mountainsides where the elevations vary in larger scales. We conducted user studies based on some of these generated paths. We discuss the results in Section 6.

5.3 Path Length

Figure 13 shows a short path and a long path generated using a short and long path length target respectively. The input terrain is called *Corner Hill*. The short path and long path were generated with a target of $\rho_L = 1,500$m and $\rho_L = 3,000$m respectively. The same



(a) sea level=150m

(b) sea level=250m

Fig. 15. Paths generated with the avoidance of forbidden zones which refer to sea water in this example. (a) A path generated as the sea level is 150. (b) A path generated as the sea level rises to 250m.

total work target of $\rho_W = 22.5$kJ was used for both paths. As the results show, to achieve the same amount of total work, the short path is more intensive in terms of force requirement as it goes uphill while the long path is less intensive as it goes downhill mostly.

5.4 Landmarks

Figure 14 shows paths generated by specifying landmarks on the terrain that the paths should pass. The input terrain is called *Sandy Hill*. The landmarks specified by the designer follow the dot patterns on the six faces of a dice. The paths were generated with the same path length target $\rho_L = 2,500$m and the same total work target $\rho_L = 22.5$kJ.

We extend the formulation by adding a landmark cost that encourages a generated path to pass the landmarks. Denote the positions of the landmarks on the 3D terrain as $\{(x_i, y_i, z_i)\}$. The landmark cost $C_K(\mathbf{p})$ is defined as:

$$C_K(\mathbf{p}) = 1 - \exp(-\left(\frac{1}{\sigma_K \omega K} \sum_{i=1} d(x_i, z_i)\right)^2), \tag{12}$$

where $d(x_i, z_i)$ denotes the minimal distance, defined in (11), between landmark i and the 2D Hermite curve $\mathcal{R}$ on the xz-plane. K is the number of landmarks used. ω is the width of the terrain. $\sigma_K = 0.25$ is set empirically. As Figure 13 shows, the generated paths pass the landmarks either exactly or very closely.

5.5 Forbidden Zones

Figure 15 shows paths generated with the avoidance of forbidden zones referring to sea water. The input terrain is called *Island Hill*. The sea water covers the land according to the sea level. The paths were generated with the same path length target $\rho_L = 2,500$m and the same total work target $\rho_L = 22.5$kJ. We enabled such a consideration by adding a forbidden zone cost $C_F(\mathbf{p})$ defined as:

$$C_F(\mathbf{p}) = 1 - \exp(-\left(\frac{1}{\sigma_F} \frac{\int_{\mathcal{P}} \Gamma(\mathbf{p}) d\mathbf{p}}{\int_{\mathcal{P}} d\mathbf{p}}\right)^2), \tag{13}$$

where $\Gamma(\mathbf{p}) = 1$ if path $\mathbf{p}$ is below the sea level and $\Gamma(\mathbf{p}) = 0$ otherwise. $\sigma_F = 0.25$ is set empirically. The cost penalizes if path $\mathbf{p}$ enters the forbidden zones. As Figure 15 shows, the generated paths avoid getting into the sea water. Alternatively, the designer could manually specify obstacles on a terrain to avoid similarly. Figure 16 shows an example using the same path length and total work targets.

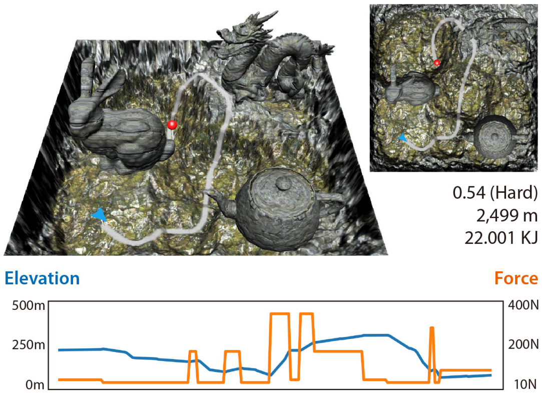


Fig. 16. Paths generated with the avoidance of manually specified obstacles which are the bunny, dragon, and teapot in this case.

6 USER STUDY

We conducted user study experiments to test the enjoyment and exertion induced on users by our generated paths. The experiments were done with a head-mounted display (HMD) VR headset and a custom-built exercise bike, which were connected to an Alienware laptop showing the generated paths in a game-like setting. The laptop was equipped with an Intel Core i7-8750H CPU, an NVIDIA GeForce GTX 1070 graphics card, and 16GB of memory. The VR headset was the Oculus Rift, which used an OLED screen with $2,160 \times 1,200$ resolution at 90Hz. Our custom-built exercise bike allowed our program to change its feedback force and monitor its cadence and power output while a user was biking.

6.1 Enjoyment Test

To test the enjoyment of biking experiences using our VR setup, we recruited 23 participants to bike our generated paths (from Section 5.2) under three different exercise modes (VR mode, TV mode and Gym mode) as shown in Figure 17. We examined descriptive statistics on our participants' performance using a Friedman Test to detect if there was an overall difference across the three different modes. We also conducted Wilcoxon signed-rank tests to pinpoint where the difference was located in each pair of modes. (e.g., VR-TV pair). From the statistical analysis, we concluded that the participants generally preferred exercising under the VR mode. For more details of the analysis, please refer to our supplementary material.

6.2 Exertion Test

To verify the exertion effectiveness of the generated paths, we used the Hard 15kJ and Hard 30kJ paths generated for the *Rocky Hill* as shown in Figure 11 for an exertion test. We call these paths Hard15 and Hard30 for brevity. We wanted to validate if the participants achieved the total work targets (e.g., 15kJ and 30kJ) specified for optimizing these paths.

6.2.1 Participants. We recruited 10 participants, who were university students and staff, to bike the Hard15 and Hard30 paths. They were 26 years old on average and their average body mass index was 21kgm^{-2}. Refer to the supplementary material for more details.

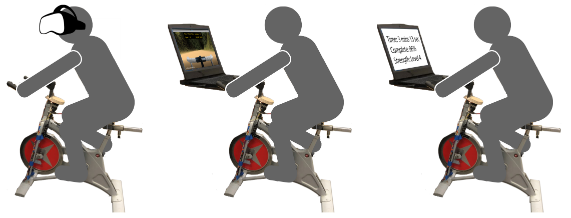

(a) VR (b) TV (c) Gym

Fig. 17. Three modes, delivered in random order, were used for the enjoyment test. (a) VR mode based on our VR bike system; (b) TV mode where the user biked while watching the game on a screen; (c) Gym mode where only biking information was displayed.

6.2.2 Procedure. Our evaluation procedure was IRB-approved. Each participant was briefed about the game control and given a warm-up session to get familiar with the game. Then we asked the participant to bike the two paths given in a randomized order. A path was completed when the participant reached the end point. There was a 5-minute break between biking the two paths.

6.2.3 Measurement. We used the PowerTap P1S power meter installed on our bike to record the power output and duration during the experiments for analysis. Before the evaluation, we calibrated the power meter. The captured data included duration in seconds, energy burned in kilojoules, average power in watts, and average cadence in round-per-minutes. The power meter was comfortable to use and did not require the participant to wear any other sensor.

6.2.4 Results. Figure 18 compared the average energy expenditure with the total work targets specified for optimizing the paths. We observed that overall the results were close to the targets. On average, the energy expenditure for the Hard15 path was 17.8kJ and that for the Hard30 path was 27.6kJ. Table 1 shows the descriptive statistics. We provided a possible explanation for the slight deviations from the targets in our supplementary material.

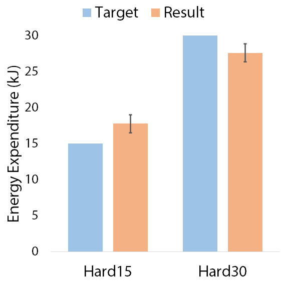
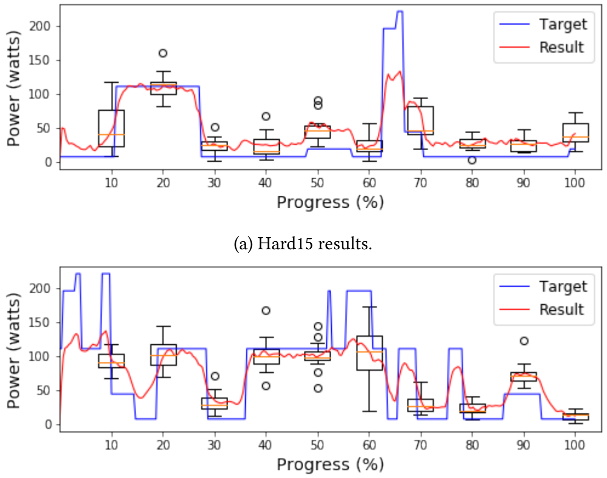


Fig. 18. Average energy expenditure in biking the Hard15 and Hard30 paths versus the targets. A significant difference was found with $p < 0.001$.

We examined descriptive statistics on our participants' energy expenditure using a paired t-test to detect whether there are differences in biking the Hard15 and Hard30 paths. A significant difference ($p < 0.01$) was found.

To observe the participants' energy expenditure during the experiment, we compared their power output results during the entire gameplay with the expected power output targets of the generated paths. Note that we convert the expected feedback forces of our generated paths into the expected power output targets for this comparison because the power meter measured power, not forces. We describe the conversion in our supplementary material. As Figure 19 shows, the participants' power output results follow the targets closely. Overall, we conclude that the participants' energy expenditure matched with the total work targets specified for optimizing the paths reasonably well, hence the generated paths help the participants achieve the exertion goals.



(a) Hard15 results.

(b) Hard30 results.

Fig. 19. Boxplots of the participants' power output in biking the Hard15 and Hard30 paths. The red line show the average result of the 10 participants. The blue line shows the expected power output target.

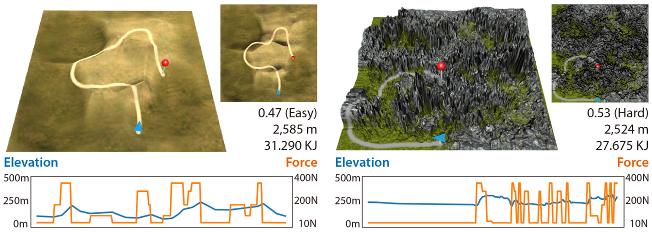
Table 1. Descriptive statistics for the Hard15 and Hard30 paths. The average completion duration, power output and energy expenditure of the participants are shown.

Path	Average Power(watt)	Time(second)	Energy(kJ)
Hard15	110.4 ± 16.5	162.7 ± 23.3	17.8 ± 2.4
Hard30	157.0 ± 23.6	178.1 ± 17.3	27.6 ± 1.6

6.2.5 User Feedback. We asked our participants to answer some open response questions at the end of our evaluation. Most participants felt that our VR bike was fun to play since the biking experience was realistic and they enjoyed biking up and down. Some of them preferred using our VR bike setup as a replacement for a typical exercise bike. One complained that the resolution of our game was not high enough. One wished that the hardware could be improved with a brake. Our supplementary material contains more user comments.

7 SUMMARY

We proposed a novel problem of generating paths over a terrain with desirable force feedback properties. We devised an optimization-based approach for generating such exertion-aware paths, which can be employed for virtual training experiences such as exergaming. By applying the generated paths to control the force feedback of an exercise bike, the immersiveness and realism of the virtual reality experience are enhanced as the user's haptic experience matches with his visual perception. We verified our approach for generating exertion-aware paths for a variety of scenarios and also through a user study consisting of an enjoyment test and an exertion test.



(a) *Plain*

(b) *Rocky Mountain*

Fig. 20. Failure cases. Our approach failed to find (a) a hard path on a plain and (b) an easy path on a rocky mountain.

7.1 Limitations and Future Work

We discuss some limitations and possible extensions of our approach. We measured the energy expenditure in joules instead of calories because the amount of calories burned could vary among people; different people could burn different amounts of calories when doing the same exercise (biking the same path), influenced by their gender, age, height, weight, and other factors. In future work, we could obtain basic body information of a user and employ that to generate a more personalized level that uses calories burned as an optimization objective to deliver more personalized training.

For our proof-of-concept experiments, we built our bike based on a common indoor cycling bike that supports 10 levels of feedback force in its original setting. The feedback force levels we used covered elevation angles up to 40° that corresponds to a 84% gradient, which was comprehensive in terms of angle coverage. However, a high-end exercise bike could offer more feedback force levels (e.g., 20 levels) which could provide more realistic and high-resolution haptic experience. Our approach could be easily extended to support more feedback force levels. We note that some high-end commercial exercise bikes are equipped with a live coach feature, allowing the user to receive personal coaching. These services usually required paid membership and could be expensive. Our approach provides an alternative, low-cost solution to generate and deliver exercising programs with desired exertion goals.

As shown in Section 4.2, our classifier achieved a 82% accuracy in classifying perceived difficulties of paths based on path features. Evaluating perceived difficulties in a user study was difficult because there are a number of factors which could affect the perceived difficulty of a biking experience but could be difficult to simulate. For example, we could not simulate some real-world factors (e.g., road textures, windiness, humidity) which are not related to paths but could have influenced the perceived difficulty ratings in our dataset. Therefore, in our approach, we consider the difficulty mainly based on the elevation features of a path. Moreover, we assume that the terrain's surface material is the same throughout the experience, and hence the friction coefficient is constant. In future work, we could include a variety of surface materials (e.g., mud, sand, concrete) carrying different physical properties such as friction properties for a terrain. We could also consider weather effects on the friction properties of the road. For example, a path on a rainy day should be more slippery than that on a sunny day, which could be simulated by decreasing the friction coefficient.

As far as the scope of our user study is concerned, the paths generated for our user evaluation are not compatible with a regular biking training session which is typically at least 30-minute long. For a real training, we could generate the desired path by using a longer path length target or a larger total work target in our optimization.

Moreover, we assume that the input terrain consists of a wide range of elevation which allows our optimizer to find a path that satisfies the exertion goals. If the terrain does not show desired variations in elevation, finding a satisfactory path may not be feasible. Figure 20 shows two failure cases, where our approach failed to find a hard path on the *Plain* terrain and an easy path on the *Rocky Mountain* terrain although the total work target of 30kJ and the path length target of $2,500$m were roughly achieved. To address this problem, future extensions might analyze a terrain to suggest possible exertion goals in a pre-processing step. Another extension is to explore optimizing both the terrain and the path concurrently to generate a solution that satisfies the exertion goals.

To demonstrate the novelty of our approach for generating paths with desired force feedback properties, we customized a bike which allows users to achieve exertion goals in virtual reality training. Our approach could be extended for generating virtual training content for other exercise machines such as a treadmill or an elliptical while leveraging the enjoyment brought about by virtual reality.

It would be interesting to extend our bike with a fan whose speed changes according to the speed of the bike. For example, the fan speeds up when the bike is going downhill so that the user can feel the wind. Such haptic feedback may create an even more engaging, realistic, and immersive VR exertion experience.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their constructive comments. We thank Daniel Lynch for his help in assembling the VR gym bike. We also thank Tongjian You and Elisa Ogawa for their advice and discussion on exergame design.

REFERENCES

Mazen Al Borno, Martin De Lasa, and Aaron Hertzmann. 2012. Trajectory optimization for full-body movements with complex contacts. *IEEE transactions on visualization and computer graphics* 19, 8 (2012), 1405–1414.

Zeyad Abd Algfoor, Mohd Shahrizal Sunar, and Hoshang Kolivand. 2015. A comprehensive study on pathfinding techniques for robotics and video games. *International Journal of Computer Games Technology* 2015 (2015), 7.

Elizabeth Behm-Morawitz, Jennifer Lewallen, and Grace Choi. 2016. A second chance at health: how a 3D virtual world can improve health self-efficacy for weight loss management among adults. *Cyberpsychology, Behavior, and Social Networking* 19, 2 (2016), 74–79.

Jan Beneš, Alexander Wilkie, and Jaroslav Křivánek. 2014. Procedural modelling of urban road networks. In *Computer Graphics Forum*, Vol. 33. 132–142.

Andrew Best, Sahil Narang, Daniel Barber, and Dinesh Manocha. 2017. Autonovi: Autonomous vehicle planning with dynamic maneuvers and traffic constraints. In *IROS*. IEEE, 2629–2636.

Luigi Cardamone, Georgios N Yannakakis, Julian Togelius, and Pier Luca Lanzi. 2011. Evolving interesting maps for a first person shooter. In *European Conference on the Applications of Evolutionary Computation*. Springer, 63–72.

Guoning Chen, Gregory Esch, Peter Wonka, Pascal Müller, and Eugene Zhang. 2008. Interactive procedural street modeling. In *ACM transactions on graphics (TOG)*, Vol. 27. ACM, 103.

Siddhartha Chib and Edward Greenberg. 1995. Understanding the metropolis-hastings algorithm. *The American Statistician* 49, 4 (1995), 327–335.

Kate Compton and Michael Mateas. 2006. Procedural Level Design for Platform Games.

Guillaume Cordonnier, Marie-Paule Cani, Bedrich Benes, Jean Braun, and Eric Galin. 2018. Sculpting mountains: Interactive terrain modeling based on subsurface geology. *IEEE TVCG* 24, 5 (2018), 1756–1769.

Dajana Dimovska, Patrick Jarnfelt, Sebbe Selvig, and Georgios N Yannakakis. 2010. Towards procedural level generation for rehabilitation. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. ACM, 7.

Robert A Douglas. 2016. *Low-volume road engineering: design, construction, and maintenance*. CRC Press.

Arnaud Emilien, Ulysse Vimont, Marie-Paule Cani, Pierre Poulin, and Bedrich Benes. 2015. Worldbrush: Interactive example-based synthesis of procedural virtual worlds. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 106.

Eric Galin, Eric Guérin, Adrien Peytavie, Guillaume Cordonnier, Marie-Paule Cani, Bedrich Benes, and James Gain. 2019. A Review of Digital Terrain Modeling. In *Computer Graphics Forum*, Vol. 38. 553–577.

Eric Galin, Adrien Peytavie, Eric Guérin, and Bedřich Beneš. 2011. Authoring hierarchical road networks. In *Computer Graphics Forum*, Vol. 30. 2021–2030.

Eric Galin, Adrien Peytavie, Nicolas Maréchal, and Eric Guérin. 2010. Procedural generation of roads. In *Computer Graphics Forum*, Vol. 29. 429–438.

David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. 2015. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems* 17, 4 (2015), 1135–1145.

Eric Guérin, Julie Digne, Eric Galin, Adrien Peytavie, Christian Wolf, Bedrich Benes, and Benoît Martinez. 2017. Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 228.

A. Hayes. 2015. *Building Bridges and Roads: Civil Engineers*. PowerKids Press.

Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 1 (2013), 1.

Rainer Herpers, Wolfgang Heiden, Michael Kutz, David Scherfgen, Ulrich Hartmann, Jens Bongartz, and Oliver Schulzyk. 2008. FIVIS bicycle simulator: an immersive game platform for physical activities. In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*. ACM, 244–247.

Danial Hooshyar, Moslem Yousefi, Minhong Wang, and Heuiseok Lim. 2018. A data-driven procedural-content-generation approach for educational games. *Journal of Computer Assisted Learning* 34, 6 (2018), 731–739.

Fu-Chung Huang, Kevin Chen, and Gordon Wetzstein. 2015. The Light Field Stereoscope: Immersive Computer Graphics via Factored Near-eye Light Field Displays with Focus Cues. *ACM Trans. Graph.* 34, 4, Article 60 (July 2015), 12 pages. https://doi.org/10.1145/2766922

Haikun Huang, Ni-Ching Lin, Lorenzo Barrett, Darian Springer, Hsueh-Cheng Wang, Marc Pomplun, and Lap-Fai Yu. 2017. Automatic Optimization of Wayfinding Design. *IEEE TVCG* (2017).

ICAROS. 2017. ICAROS - We make you fly. https://www.icaros.com/. Accessed 12-27-2019.

Katherine Isbister and Florian "Floyd" Mueller. 2015. Guidelines for the design of movement-based games and their relevance to HCI. *Human–Computer Interaction* 30, 3-4 (2015), 366–399.

Martin Jennings-Teats, Gillian Smith, and Noah Wardrip-Fruin. 2010. Polymorph: A model for dynamic level generation. In *Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Sertac Karaman, Matthew R Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. 2011. Anytime motion planning using the RRT. In *2011 IEEE International Conference on Robotics and Automation*. IEEE, 1478–1483.

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

Markus Löchtefeld, Antonio Krüger, and Hans Gellersen. 2016. DeceptiBike: Assessing the Perception of Speed Deception in a Virtual Reality Training Bike System. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*. ACM, 40.

Florian'Floyd' Mueller, Darren Edge, Frank Vetere, Martin R Gibbs, Stefan Agamanolis, Bert Bongers, and Jennifer G Sheridan. 2011. Designing sports: a framework for exertion games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2651–2660.

Florian Mueller and Katherine Isbister. 2014. Movement-based game guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2191–2200.

Kourosh Naderi, Joose Rajamäki, and Perttu Hämäläinen. 2017. Discovering and synthesizing humanoid climbing movements. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–11.

Matt de Neef. 2013. Gradients and cycling: an introduction. http://theclimbingcyclist.com/gradients-and-cycling-an-introduction/. Accessed 1-19-2020.

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 TVCG* 20, 4 (2014), 569–578.

Gen Nishida, Ignacio Garcia-Dorado, and Daniel G Aliaga. 2016. Example-Driven Procedural Urban Roads. In *Computer Graphics Forum*, Vol. 35. 5–17.

Elisa F Ogawa, Tongjian You, and Suzanne G Leveille. 2016. Potential benefits of exergaming for cognition and dual-task function in older adults: a systematic review. *Journal of Aging and Physical Activity* 24, 2 (2016), 332–336.

Kalpana P Padala, Prasad R Padala, Shelly Y Lensing, Richard A Dennis, Melinda M Bopp, Paula K Roberson, and Dennis H Sullivan. 2017. Home-based exercise program improves balance and fear of falling in community-dwelling older adults with mild Alzheimer's disease: A pilot study. *Journal of Alzheimer's Disease* 59, 2 (2017), 565–574.

Yoav IH Parish and Pascal Müller. 2001. Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 301–308.

Max Pfeiffer, Stefan Schneegass, Florian Alt, and Michael Rohs. 2014. Let me grab this: a comparison of EMS and vibration for haptic feedback in free-hand interaction. In *Proceedings of the 5th augmented human international conference*. ACM, 48.

Ruslan Rakhmatov, Arsen Abdulali, Waseem Hassan, Minji Kim, and Seokhee Jeon. 2018. Virtual Reality Bicycle with Data-Driven Vibrotactile Responses from Road Surface Textures. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*. IEEE, 1–9.

Daniel Schoene, Trinidad Valenzuela, Stephen R Lord, and Eling D de Bruin. 2014. The effect of interactive cognitive-motor training in reducing fall risk in older people: a systematic review. *BMC Geriatrics* 14, 1 (2014), 107.

Peizhi Shi and Ke Chen. 2018. Learning Constructive Primitives for Real-Time Dynamic Difficulty Adjustment inSuper Mario Bros. *IEEE Transactions on Games* 10, 2 (2018), 155–169.

Ruben M Smelik, Tim Tutenel, Rafael Bidarra, and Bedrich Benes. 2014. A survey on procedural modelling for virtual worlds. In *Computer Graphics Forum*, Vol. 33. 31–50.

Misha Sra, Sergio Garrido-Jurado, Chris Schmandt, and Pattie Maes. 2016. Procedurally generated virtual reality from 3D reconstructed physical space. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. ACM, 191–200.

Amanda E Staiano, Anisha A Abraham, and Sandra L Calvert. 2013. Adolescent exergame play for weight loss and psychosocial improvement: a controlled physical activity intervention. *Obesity* 21, 3 (2013), 598–601.

Nathan Sturtevant and Michael Buro. 2005. Partial pathfinding using map abstraction and refinement. In *AAAI*, Vol. 5. 1392–1397.

Grant Tinsley. 2017. Cardio vs Weight Lifting: Which Is Better for Weight Loss? https://www.healthline.com/nutrition/cardio-vs-weights-for-weight-loss. Accessed 10-24-2017.

Carlos A Vanegas, Ignacio Garcia-Dorado, Daniel G Aliaga, Bedrich Benes, and Paul Waddell. 2012. Inverse design of urban procedural models. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 168.

Sarah Catherine Walpole, David Prieto-Merino, Phil Edwards, John Cleland, Gretchen Stevens, and Ian Roberts. 2012. The weight of nations: an estimation of adult human biomass. *BMC public health* 12, 1 (2012), 439.

David Weibel, Bartholomäus Wissmath, Stephan Habegger, Yves Steiner, and Rudolf Groner. 2008. Playing online games against computer-vs. human-controlled opponents: Effects on presence, flow, and enjoyment. *Computers in Human Behavior* 24, 5 (2008), 2274–2291.

David James Wilkie, Jur Van den Berg, Ming Lin, and Dinesh Manocha. 2011. Self-aware traffic route planning. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

S Wüest, NA Borghese, M Pirovano, R Mainetti, R van de Langenberg, and ED de Bruin. 2014. Usability and effects of an exergame-based balance training program. *Games for Health Journal* 3, 2 (2014), 106–114.

Biao Xie, Yongqi Zhang, Haikun Huang, Elisa Ogawa, Tongjian You, and Lap-Fai Yu. 2018. Exercise intensity-driven level design. *IEEE TVCG* 24, 4 (2018), 1661–1670.

Yongqi Zhang, Biao Xie, Haikun Huang, Elisa Ogawa, Tongjian You, and Lap-Fai Yu. 2019b. Pose-Guided Level Design. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 554.

Zechen Zhang, Nikunj Raghuvanshi, John Snyder, and Steve Marschner. 2019a. Acoustic texture rendering for extended sources in complex scenes. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 222.