Exercise Intensity-driven Level Design

Biao Xie*, Yongqi Zhang*, Haikun Huang, Elisa Ogawa, Tongjian You and Lap-Fai Yu, *Member, IEEE*
*Equal Contributors

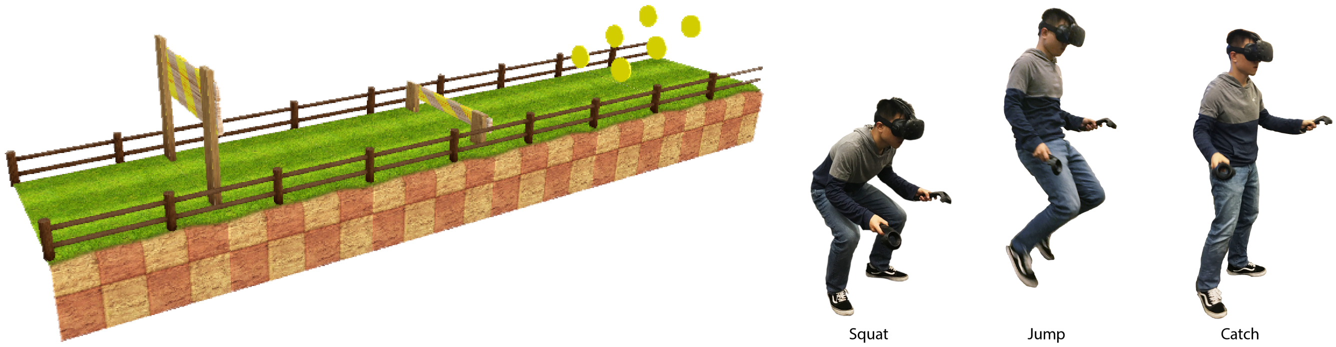


Fig. 1: Our approach is capable of synthesizing level designs with respect to target exercise parameters such as calories burned, exercise intensity level and duration for virtual reality-based exergaming. Left: part of a synthesized level composed of different chunks that require different player's motions to pass. Right: some example motions performed by the player to complete the level.

Abstract—Games and experiences designed for virtual or augmented reality usually require the player to move physically to play. This poses substantial challenge for level designers because the player's physical experience in a level will need to be considered, otherwise the level may turn out to be too exhausting or not challenging enough. This paper presents a novel approach to optimize level designs by considering the physical challenge imposed upon the player in completing a level of motion-based games. A game level is represented as an assembly of chunks characterized by the exercise intensity levels they impose on players. We formulate game level synthesis as an optimization problem, where the chunks are assembled in a way to achieve an optimized level of intensity. To allow the synthesis of game levels of varying lengths, we solve the trans-dimensional optimization problem with a Reversible-jump Markov chain Monte Carlo technique. We demonstrate that our approach can be applied to generate game levels for different types of motion-based virtual reality games. A user evaluation validates the effectiveness of our approach in generating levels with the desired amount of physical challenge.

Index Terms—Virtual Reality, Level Design, Procedural Modeling, Exergaming

1 Introduction

In view of the rapid popularity of virtual reality devices, many game companies transform their gaming platform from personal computers to virtual reality headsets. This transition poses a new challenge to level designers to create games that take players' physical comfort into account. Traditionally, video games are designed to be played with a game controller which involves only hand and finger movement, and the player's fatigue level is not an important concern nor consideration. In contrast, many virtual reality games require full body motions such as squatting, jumping and catching (see Figure 1 for some examples), with each motion causing a certain amount of fatigue. The player could quickly feel exhausted if the level is not designed well, which may prompt him to stop or quit the game. Balancing the exercise intensity level and the fun factor of the game is non-trivial and challenging.

We propose a novel optimization-based approach to address this challenge. As shown in Figure 1, our approach can be applied to synthesize a level with a desired exercise intensity level. By formulating the design problem as an optimization, the level designer can easily balance factors concerning the exercise intensity level of the game as well as other design factors and constraints. We implemented our optimization framework as a practical game engine plugin and demonstrated how it could be used to generate different types of games. The major contributions of this paper include:

- We propose a novel problem statement of optimizing level designs with respect to exercise intensity. Such concept could be applied to improve the user experience of different motion-based games, which constitute a majority of games played on virtual or augmented devices.
- We propose a novel optimization-based approach to automatically synthesize game levels while considering the player's physical experience. We demonstrate by experiments that our approach works for different types of motion-based games.
- We validate the effectiveness of our approach in generating optimized games levels through user evaluations.

- *Biao Xie is with the University of Massachusetts Boston. E-mail: Biao.Xie001@umb.edu.*
- *Yongqi Zhang is with the University of Massachusetts Boston. E-mail: Yongqi.Zhang001@umb.edu.*
- *Haikun Huang is with the University of Massachusetts Boston. E-mail: Haikun.Huang001@umb.edu.*
- *Elisa Ogawa is with the University of Massachusetts Boston. E-mail: Elisa.Ogawa001@umb.edu.*
- *Tongjian You is with the University of Massachusetts Boston. E-mail: Tongjian.You@umb.edu.*
- *Lap-Fai Yu is with the University of Massachusetts Boston. E-mail: craigyu@cs.umb.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

2 Related Work

2.1 Exercise Intensity in Game Design

The extent of physical challenge is considered as one of the most critical factors in designing motion-based games [25]. There have been studies about the relationship between the level of physical challenge and how much fun a game offers. Sorenson and Pasquier [32] concludes that a

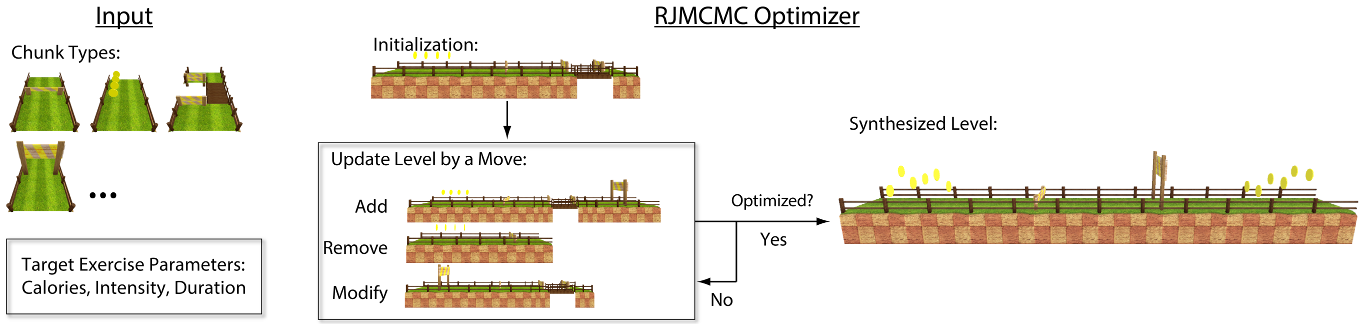


Fig. 2: Overview of our approach.

player derives the most fun from a game when the level is neither too simple nor difficult to complete. Pasch et al. [25] and Sinclair et al. [31] use the dual flow model to explain the importance of matching the physical challenge of a motion-based game with the player's physical fitness and skill level, which is critical to the game's attractiveness and effectiveness. While it is found important to appropriately set the physical challenge level, in current practice the setting is usually done manually and tested by a trial-and-error approach [11, 17], which is non-trivial, tedious and largely dependent on the experience of the level designer. It would be beneficial to quantify the exercise intensity involved in a game level so that scientific analysis and optimization can be performed. Our approach demonstrates how exercise intensity considerations can be incorporated into the automatic synthesis of a game level.

2.2 Exergaming

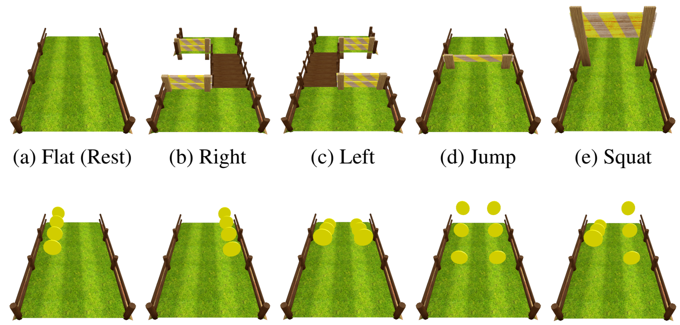
The widespread popularity of human-computer interaction devices for gaming such as depth sensors (e.g., Microsoft Kinect) and motion controllers (e.g., Wii Remote) have given rise to a new genre of games known as fitness games or exergames, which refer to video games that are also a form of exercises [2, 31]. Recently, with the growing popularity of household virtual reality devices such as the Oculus Rift and the HTC Vive, companies have started to explore the possibility of developing exergames that are played via virtual reality devices, which have aroused much commercial and research interests [9, 10, 28]. For example, VirZOOM [36] and CSE Entertainment [7] developed exergaming systems that allow players to exercise on a cycling machine while navigating through a virtual scene viewed via a virtual reality headset. ICAROS [16] integrated virtual reality headsets with fitness equipment to create highly immersive VR experiences which entertain the player while he is working out. Using virtual reality devices for exergaming is becoming increasingly popular.

Recent research has validated the positive effects of exergaming for rehabilitation and therapy purposes such as weight control, balance enhancement and cognitive-motor training. Staiano et al. [33] suggested that exergames could be used as a strategic tool to motivate young adults to lose weight. Kim et al. [22] introduced an unsupervised virtual reality-based exergaming program that could help enhance muscle strength and restore physical function on elders. Agmon et al. [1], Schoene et al. [26] and Ogawa et al. [24] found that exergaming could reduce the risk of elders from falling. In addition, Wüest et al. [37] validated that exergaming is effective for improving movement- and balance-related physical performance of elders.

In sum, exergaming brings health benefits to players, motivating them to exercise and improve their health conditions. One key element for the success of such kind of systems is the provision of virtual reality content with an appropriate level of exercise intensity for the player [28, 34]. However, most of the existing exergames either for research or general purposes require a level designer to manually design and develop a suitable exercise intensity level. In contrast, our approach is capable of generating optimized game levels automatically for exergaming in virtual reality.

2.3 Procedural Level Design

Procedural techniques have been successfully applied for designing game levels [6]. For example, rule-based [14] and learning-based



(a) Flat (Rest) (b) Right (c) Left (d) Jump (e) Squat

(f) C-Left (g) C-Right (h) C-Up (i) C-Down (j) C-Swipe

Fig. 3: Different types of chunks for assembling a game level for the running example game, *Reflex*. The chunk types at the top requires the player to avoid obstacles, while the chunk types at the bottom requires the player to catch coins.

approaches [29] (e.g., via neural networks [18]) have been applied for generating levels for Super Mario Bros. Refer to the book by Short and Adams [30] and the survey by Hendrikx [13] for a comprehensive review and discussion on procedural level design.

Our approach is inspired by previous procedural game content generation works which are driven by the player's gaming experience [27, 38] or "emotion" [35] during the gameplay. However, to the best of our knowledge, there is no previous work which explicitly uses exercise intensity metrics as the criteria to optimize a game level, which is our novelty. Along a similar direction, Dimovska et al. [8] propose to adapt a game level to the measured performance of the player in a skiing game played with Wii, and demonstrates through a preliminary evaluation study that such adaption could achieve better rehabilitation effects on the player. Their results inspire us to investigate how exercise intensity metrics can be comprehensively employed for driving level design.

By using procedural techniques, game content can be created in a fast and scalable manner, and can even be generated on-the-fly during the gameplay. Another advantage is that random variation among the generated game content can keep the game fresh and interesting to the player. The major challenge of automatic content generation or virtual world synthesis approaches [15, 39] lies in controlling the procedural techniques to generate game levels or virtual worlds that satisfy the designer's constraints and goals [30]. To tackle this problem, we devise an optimization framework upon which different design goals and constraints can be incorporated and hence automatically considered by the optimizer in synthesizing a level design. In particular, our framework incorporates exercise intensity level, a critical design factor to consider in motion-based games, in the game level synthesis process. Hence our framework is especially useful for designing motion-based games for virtual or augmented reality experiences.

3 Overview

Figure 2 shows an overview of our approach. Given different types of chunks for assembling a game level plus their exercise intensity properties, our approach runs an optimization to assemble different game levels using a varying number of chunks. In each iteration of the optimization, the assembled game level is evaluated for the physical difficulty it imposes on the player to complete the level. The game level is iteratively updated by the optimizer until a desirable extent of physical difficulty is achieved.

Reflex (Running Example). To facilitate the illustration of our approach, we focus on a motion-based game called *Reflex* created for illustration and experiment purposes. The game design of *Reflex* mimics that of a game called *Reflex Ridge* played using the Microsoft Kinect, which has been employed for previous physical therapy study [19]. While our game logic is similar, we changed the interaction and visualization design such that the player plays the game through an HTC Vive virtual reality device rather than a Kinect. The HTC Vive tracks the player's motion such as stepping left or right, jumping and crouching down, while the player sees the virtual world through a VR headset. This gives the player an even more immersive experience than playing through a Kinect and a TV screen.

Game Logic: A game level of the *Reflex* game consists of a runway assembled by a number of chunks. The chunks belong to one of the chunk types as depicted in Figure 3. As the game starts, the runway starts rolling towards the player. Depending on the chunk type, the player passes each chunk by moving his body to bypass any obstacle or by catching all coins with his hands. The game ends when the player has reached the end of the runway. Depending on the chunks used to assemble a level, completing a level requires different extent of physical movements and hence imposes a different amount of exercise intensity on the player. The supplementary video shows a demo of the gameplay.

4 Technical Approach

The goal of our approach is to synthesize levels optimized with respect to a desired level of exercise intensity and other design factors, which are encoded by cost terms.

4.1 Formulation

Let $l = (c_1, c_2, ..., c_n)$ denote a level, which consists of a number of chunks c_i assembled in a sequential order. Each chunk c_i belongs to a certain chunk type T_i, where $T_i \in T$ and T is the set of all chunk types. For example, the running example game has 10 chunk types as depicted in Figure 3.

The quality of a level l is evaluated by a total cost function $C_{\text{Total}}(l)$:

$$C_{\text{Total}}(l) = \mathbf{C}_{\text{I}}\mathbf{w}_{\text{I}}^{\text{T}} + \mathbf{C}_{\text{P}}\mathbf{w}_{\text{P}}^{\text{T}}, \tag{1}$$

where $\mathbf{C}_{\text{I}} = [C_{\text{I}}^{\text{c}}, C_{\text{I}}^{\text{m}}, C_{\text{I}}^{\text{v}}]$ is a vector of intensity costs and $\mathbf{w}_{\text{I}} = [w_{\text{I}}^{\text{c}}, w_{\text{I}}^{\text{m}}, w_{\text{I}}^{\text{v}}]$ is a vector of weights. C_{I}^{c}, C_{I}^{m} and C_{I}^{v} encode the exercise intensity considerations: the total amount of calories burned to complete the level, the mean intensity throughout the experience and the variance in intensity among the chunks. $\mathbf{C}_{\text{P}}$ is a vector of game-specific prior costs encoding design priors such as duration and variation between adjacent chunks, and $\mathbf{w}_{\text{P}}$ stores the weights of these costs. We will provide details of these priors costs in Section 4.4.

4.2 Chunk Type Intensity and Duration

Each chunk type T_i is associated with an intensity level I_i and a duration D_i for completing the chunk, which is estimated by the designer or found empirically as follows.

In exercise science, heart rate is commonly used as a metric to evaluate the intensity level of a certain type of exercise [23]. Exercises with different intensity levels are needed to achieve different training purposes. We quantify the intensity level associated with each chunk type similarly. The intensity level I_i of each chunk type T_i is found empirically by the following steps:

1. First, a level consisting of an infinite number of chunks with chunk type T_i is prepared.

2. Ten healthy participants (5 males and 5 females) were recruited to play a level for about 5 minutes. All participants were college students, whose ages range from 18 to 24. None of them were athletes. The heart rate of each participant during this game-play was recorded by a wearable heart rate recorder (Polar T31).

Chunk Type	Intensity	Chunk Type	Intensity
Flat	12.6%	C-Left	28.1%
Right	14.2%	C-Right	26.0%
Left	14.2%	C-Up	27.6%
Jump	55.1%	C-Down	31.5%
Squat	32.5%	C-Swipe	25.4%

Table 1: Intensity level of each type of chunk for the running example game, *Reflex*.

3. The intensity I_i is calculated by the Karvonen formula [20]:

$$H = (H_{\max} - H_{\text{rest}})I_i + H_{\text{rest}}, \tag{2}$$

$$I_i = \frac{H - H_{\text{rest}}}{H_{\max} - H_{\text{rest}}}, \tag{3}$$

where $I_i \in [0,1]$; $H_{\max} = 220 - a$ with a being the participant's age; H is the average measured heart rate of the participant during the game level; H_{rest} is the resting heart rate of the participant.

4. The duration D_i (in seconds) associated with chunk type T_i is calculated as the average time the participants take to finish each chunk of that type.

Table 1 shows the intensities of different types of chunks used for synthesizing a level for the running example game, *Reflex*, found following the above procedure.

4.3 Intensity Costs

Three costs are defined to encode the exercise intensity considerations of a level l.

Calorie Cost. This cost compares the amount of energy needed to complete the level with a target amount of energy expenditure expressed in terms of calories:

$$C_{\text{I}}^{\text{c}}(l) = \frac{1}{N}\left|\sum_{c_i} G(c_i)D(c_i) - \rho_{\text{cal}}\right|, \tag{4}$$

where $D(c_i)$ returns the average duration for completing chunk c_i based on its chunk type as determined in Section 4.2. ρ_{cal} is the target total amount of calories burned. N is a normalization constant set as $N = \kappa_1\kappa_2\kappa_3$, where κ_1 is the upper bound on the number of chunks used to assemble a level (set as 200 for *Reflex*), κ_2 is the calories burned in finishing a chunk of the type associated with the maximum intensity level, and κ_3 is the duration needed to finish a chunk of the type associated with the longest duration. $G(c_i)$ is set based on the formulas and measurements from Keytel et al. [21]:

$$G(c_i) = (\mu_1 + \mu_2 H(c_i) + \mu_3\omega + \mu_4 a)\mu_5, \tag{5}$$

where $H(c_i)$ returns the average heart rate (computed by Equation 2) in playing chunk c_i according to the intensity level of its chunk type; ω is the weight and a is the age of the player. We use 70 beats per minute as the default resting heart rate of the player.

The coefficients are gender-specific [21]. For a male player, $\mu_1 = -55.0969$, $\mu_2 = 0.6309$, $\mu_3 = 0.1988$ and $\mu_4 = 0.2017$. For a female player, $\mu_1 = -20.4022$, $\mu_2 = 0.4472$, $\mu_3 = -0.1263$ and $\mu_4 = 0.0740$. For both genders, $\mu_5 = \frac{1}{4.184\times 60}$.

Mean Intensity Cost. This cost evaluates the mean intensity of a level with respect to a target mean intensity:

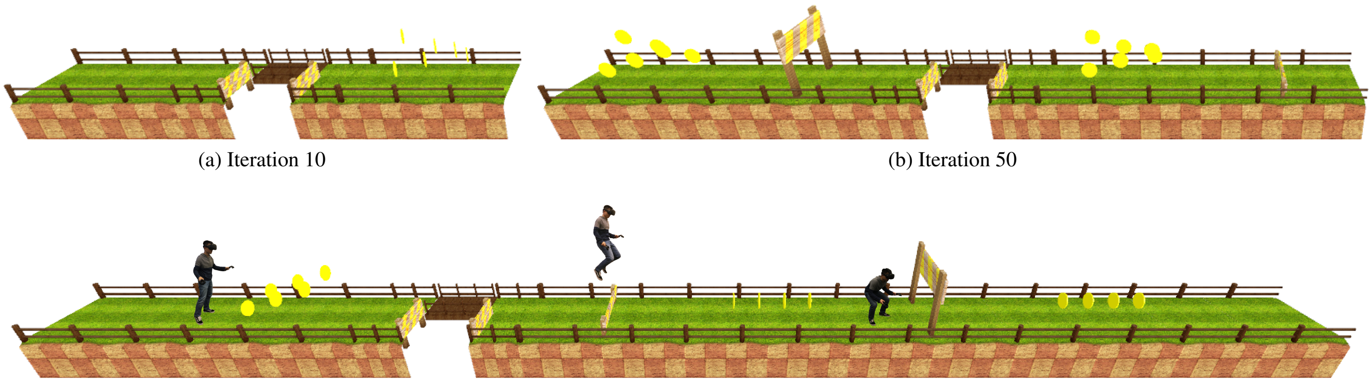
$$C_{\text{I}}^{\text{m}}(l) = \frac{1}{|l|}\sum_{c_i} I(c_i) - \rho_{\text{m}}, \tag{6}$$

where ρ_{m} is the target mean intensity (e.g., 50%). If a level is too intense, this cost will be high and more "resting" chunks (i.e., flat chunks) will be added by the optimizer to lower the average intensity.

Intensity Variation Cost. A cost is also defined to evaluate how much variation in intensity exists among the selected chunks. A high cost means large variation:

$$C_{\text{I}}^{\text{v}}(l) = \frac{1}{|l|}\sum_{c_i} (I(c_i) - \bar{I})^2 - \rho_{\text{v}}, \tag{7}$$

where ρ_{v} is a target variance in intensity and $\bar{I}$ is the mean of the intensity of the chunks.



(a) Iteration 10

(b) Iteration 50

(c) Iteration 200 (result)

Fig. 4: Synthesized levels throughout an optimization. For visualization convenience, we set the optimizer to assemble a rather short level in this example by using a small target duration parameter ρ_{d}.

4.4 Prior Costs

Different prior costs can be defined to encode game-specific level design considerations. For example, game levels of a certain total duration may be desired. It may also be preferable to assemble a level using chunks of different chunk types, so that the level will appear more diverse and fun. For our experiments, we define two prior costs for illustration.

Duration Cost. A cost is defined to softly constrain the level to be of a certain duration:

$$C_{\mathrm{P}}^{\mathrm{d}}(l) = 1 - \exp\left(-\frac{(\sum_{c_i} D(c_i) - \rho_{\mathrm{d}})^2}{2\sigma^2}\right), \tag{8}$$

where ρ_{d} is the target duration of the game level and σ is set as $0.5\rho_{\mathrm{d}}$. These parameters are used as the mean and standard deviation of a Gaussian distribution for evaluating how close the duration of the level is with respect to the target duration.

Adjacent Chunk Variation Cost. We also define a cost to penalize levels that are "monotonic", i.e., consisting of adjacent chunks with the same chunk type:

$$C_{\mathrm{P}}^{\mathrm{v}}(l) = \frac{1}{|l|-1} \sum_{c_i, c_{i+1}} \Gamma(c_i, c_{i+1}), \tag{9}$$

where c_i and c_{i+1} are adjacent chunks. $\Gamma(c_i, c_{i+1})$ returns 1 if c_i and c_{i+1} are of the same chunk type, otherwise it returns 0.

Other types of prior costs can be similarly defined and incorporated into the optimization framework depending on the specific requirements of a game under design.

4.5 Optimization

We synthesize a level by optimizing it with respect to the total cost function $C_{\mathrm{Total}}(l)$. To solve the optimization, we apply a Markov chain Monte Carlo technique, namely, simulated annealing with a Metropolis-Hastings state searching step. As a level could be assembled by an arbitrary number of chunks, our optimization is performed in a trans-dimensional solution space. To effectively sample solutions from spaces of levels assembled by different numbers of chunks, we employ the reversible-jump Markov chain Monte Carlo technique [12]. First, we define a Boltzmann-like objective function:

$$f(l) = \exp\left(-\frac{1}{t} C_{\mathrm{Total}}(l)\right), \tag{10}$$

where t is the temperature parameter of simulated annealing. At each iteration of the optimization, our approach applies a move to alter the current level l to create a proposed level l'. The move belongs to one of the following types:

- *Add a Chunk*: a chunk with a randomly selected chunk type is added to a random location of the current level l to create a proposed level l';
- *Remove a Chunk*: a chunk is randomly selected and removed from the current level l to create a proposed level l';
- *Modify a Chunk*: a chunk is randomly selected from the current level l and its chunk type is modified to a randomly selected chunk type to create a proposed level l'.

The selection probabilities of the add, remove and modify moves are set as p_{a}, p_{r} and p_{m}. Unless otherwise specified, we use $p_{\mathrm{a}} = 0.4$, $p_{\mathrm{r}} = 0.2$ and $p_{\mathrm{m}} = 0.4$ in our experiments such that the add and modify moves are selected with a higher probability.

To decide whether to accept the proposed level l', our approach compares the total cost value $C_{\mathrm{Total}}(l')$ of the proposed level l' with the total cost value $C_{\mathrm{Total}}(l)$ of the original level l. To maintain the detailed balance condition in the trans-dimensional optimization, our approach accepts the proposed level l' with the following acceptance probability $Pr(l'|l)$ specified based on the Metropolis criterion:

For an *Add a Chunk* move,

$$Pr(l'|l) = \min\left(1, \frac{p_{\mathrm{r}}}{p_{\mathrm{a}}} \frac{Z-|l|}{|l'|} \frac{f(l')}{f(l)}\right), \tag{11}$$

For a *Remove a Chunk* move,

$$Pr(l'|l) = \min\left(1, \frac{p_{\mathrm{a}}}{p_{\mathrm{r}}} \frac{|l|}{Z-|l'|} \frac{f(l')}{f(l)}\right), \tag{12}$$

For a *Modify a Chunk* move,

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

Note that for formulation simplicity, we assume each chunk type T_i can only be selected Z_i times rather than an infinite number of times, so that the dimensionality of the solution space has an upper limit. In other words, a level can be assembled by up to $Z = \sum_i Z_i$ chunks. We set $Z_i = 20$ for each chunk type in our experiments

We apply simulated annealing to efficiently explore the solution space, which is controlled by the temperature parameter t. At the beginning of the optimization, t is set to be high such that the optimizer aggressively explores the solution space to locate a good start. Throughout the optimization, the temperature t is lowered gradually until it reaches a low level which is near zero. By default, we set $t = 1.0$ at the beginning of the optimization and decreases it by 0.1 every 3,000 iterations, until it reaches zero. Essentially, the optimizer becomes more greedy to refine the solution. The optimization is set to terminate

Fig. 5: Another exergame, *Longbowman*, used for our experiments. Left: player's view of the game. Right: three types of enemies (*Zombunny*, *Zombear* and *Helephant*) with different movement speeds and health points, which require different amount of player's efforts to tackle.

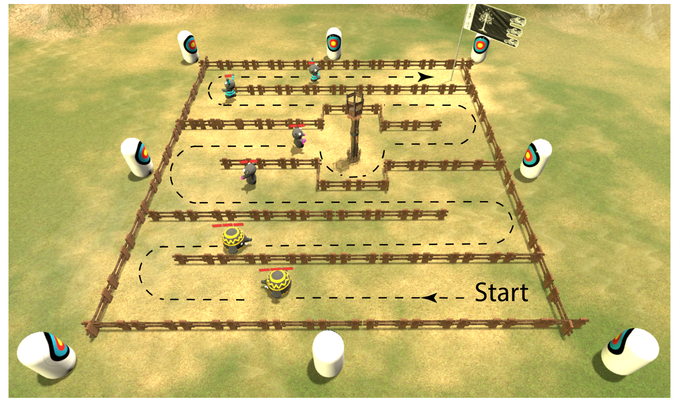


Fig. 6: *Longbowman*'s game world. The black dashed line denotes the predefined walking path of the enemies. Our approach synthesizes a level in which enemies appear in an order of sequence optimized with respect to the exercise intensity required for clearing the level.

if the change in the total cost is less than 3% over the previous 50 iterations.

Unless otherwise specified, we set the weights as $w_{\mathrm{I}}^{\mathrm{c}} = 1.0$, $w_{\mathrm{I}}^{\mathrm{m}} = 0.5$ and $w_{\mathrm{I}}^{\mathrm{v}} = 0.1$ in our optimization. The weights of all the prior cost terms are set as 0.1. Note that the designer can control the synthesis to emphasize certain design goals by changing the weights.

Figure 4 shows the levels synthesized throughout an optimization. For visualization convenience, in this illustrative example we synthesize a rather short level by specifying a target duration ρ_{d} corresponding to about 8 chunks. Figure 13 shows parts of two different levels synthesized with the middle two chunks fixed by the designer. Our approach is capable of synthesizing the rest of the levels with respect to the optimization goals. This functionality could be helpful to the level designer in practice, in case he wants to constrain certain parts of a level to certain types, while letting the optimizer complete the design.

5 Experiments & Results

5.1 Implementation

We implemented our approach on an Alienware PC equipped with an Intel Core i7-5820K CPU and 32GB of memory. The optimization framework is implemented in C# as a plugin for the Unity game engine. The games used for experiments are implemented in Unity using the SDK of the HTC Vive. Depending on the game, synthesizing a game level consisting of 90 chunks (for a *Reflex* level of about 180 seconds) typically takes about 30,000 optimization steps, which can be finished in about 10 seconds based on our current implementation.

We demonstrate how our approach can be applied to synthesize levels for two games, *Reflex* and *Longbowman*, in the following. The levels are used for user evaluation tests that are described in Section 6.

5.2 Reflex

We first synthesize three different levels for the running example game, *Reflex* (refer to Sections 3 and 4 for details of the game logic). The three levels are synthesized with physical difficulty levels of *easy*, *medium* and *hard*, specifically, with target mean intensity level ρ_{m} set as 15%, 30% and 40% respectively; and target calories burned ρ_{cal} set as 10 kcal, 15 kcal and 20 kcal. The target duration time ρ_{d} is set as 180 seconds for all three levels, and the target intensity variation ρ_{v} is set as 0.8 to make the level entertaining by introducing variation among trucks.

Figure 8 shows the synthesized levels and some screenshots during the gameplay. It can be observed that all three synthesized levels have a duration of about 180 seconds. The *easy* level consists mostly of chunks of low intensity (e.g., "Flat", "Right", "Left"); the *medium* level is composed of chunks of mixed intensities; while the *hard* level consists mostly of chunks of high intensity (e.g., "Jump", "Squat", "C-Up"). We validate in our user evaluation tests in Section 6 that the levels indeed impose extents of exercise intensity and calories burned on players as specified.

5.3 Longbowman

To demonstrate the generality of our approach and validate its effectiveness in generating level designs with desired physical difficulty properties, we implement another motion-based game called *Longbowman*. This game mimics a popular shooting game called *Longbow* available on Steam VR, which is played via an HTC Vive or an Oculus Rift. Figure 5 shows the player's view of the game and different types of enemies.

Game Logic. The game logic of *Longbowman* resembles that of *Longbow*. The player is tasked with the goal of shooting incoming enemies under a tower with his bow and arrow. The player wears an HTC Vive virtual reality device to play the game, moving two controllers to mimic actions of pulling his bow, aiming and shooting an arrow. Figure 7 shows a player doing a shot. As the player is constantly holding two controllers in a bow-pulling pose, soreness and tiredness easily develop, which add to the exercise intensity of the virtual reality experience.

A level of *Longbowman* is composed of a sequence of chunks, realized as waves of enemies that appear in the game. The player stands on a tower located at the center of the game world, which gives him the widest field of view of the game world, and flexibility to turn around and shoot. As the game starts, a wave of enemies is spawned at the lower-right corner of the game world every 10 second according to the current chunk type of the level (see Figure 6). The enemies follow a predefined path trying to walk to the upper-right corner to reach a flag, at which point they will disappear and become unavailable for shooting. The player's goal is to shoot as many enemies as possible to get high scores. The game ends either when the player has eliminated the last wave of enemies or when all enemies have escaped by reaching the flag. The supplementary video shows a demo of the gameplay.

Fig. 7: Shooting a target in the *Longbowman* game using the HTC Vive controllers.

Approach. As *Longbowman* is a 3D shooting type of game, its concept of chunks is quite different from that of *Reflex*. In *Longbowman*, a chunk refers to a wave of enemies that will appear in the game and that the player needs to shoot.

We define three types of enemies: *Zombunny*, with one health point (i.e., killed by 1 hit) and high moving speed; *Zombear*, with two health points and medium moving speed ; and *Helephant*, with three health points and low moving speed. Figure 5 shows their appearances.

There are nine types of chunks, consisting of one, two or three occurrences of *Zombunny*, *Zombear* or *Helephant* in a wave. In general, a chunk made of one occurrence of *Zombunny* is the easiest to clear, while a chunk made of three occurrences of *Helephants* is the hardest. We measure the exercise intensity associated with each chunk type following the same procedure described in Section 4.2 based on heart

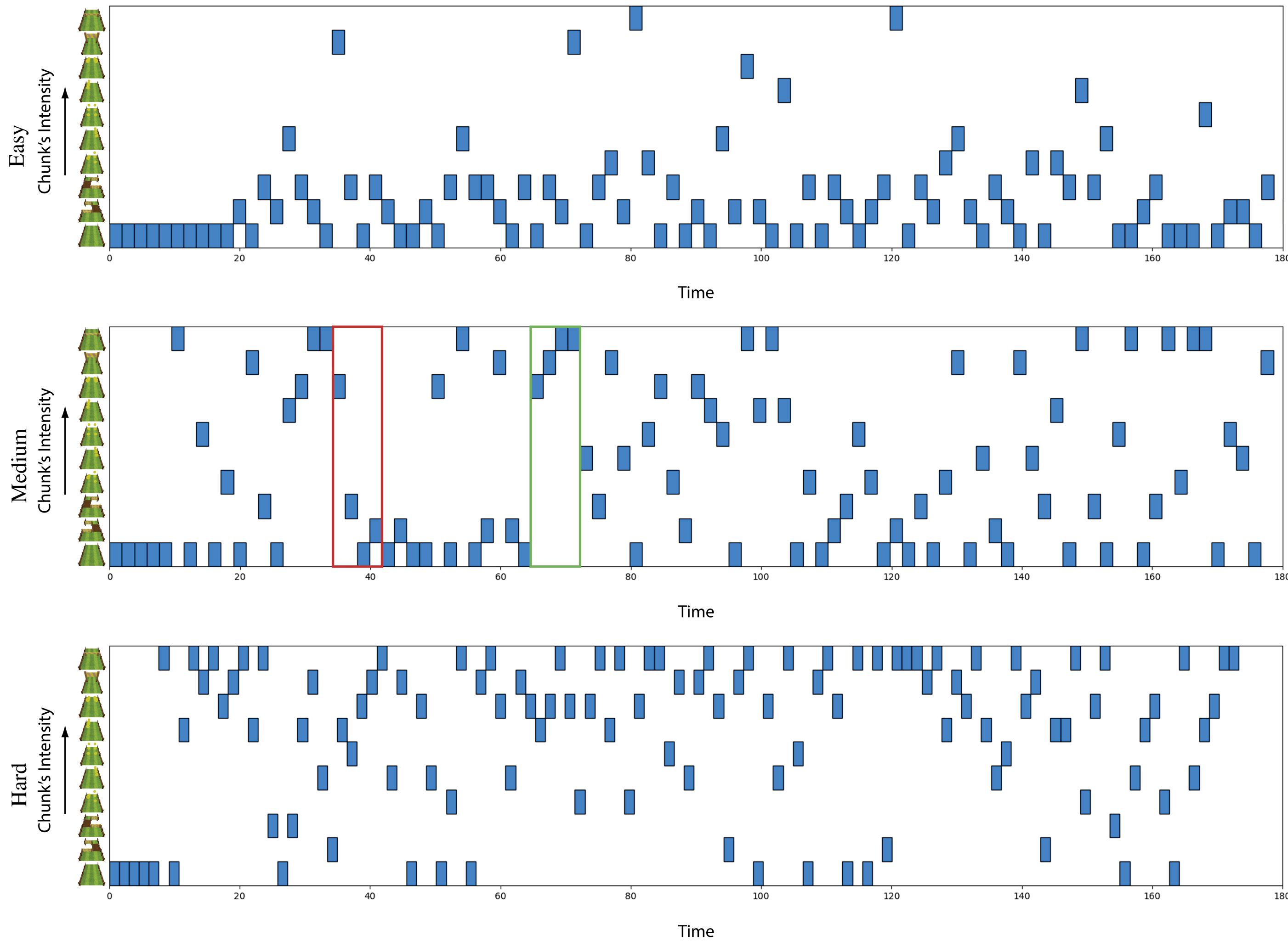


Fig. 8: Levels of the game *Reflex* synthesized with different mean intensity levels. Each blue bar refers to a chunk that appears at a certain time of the level.

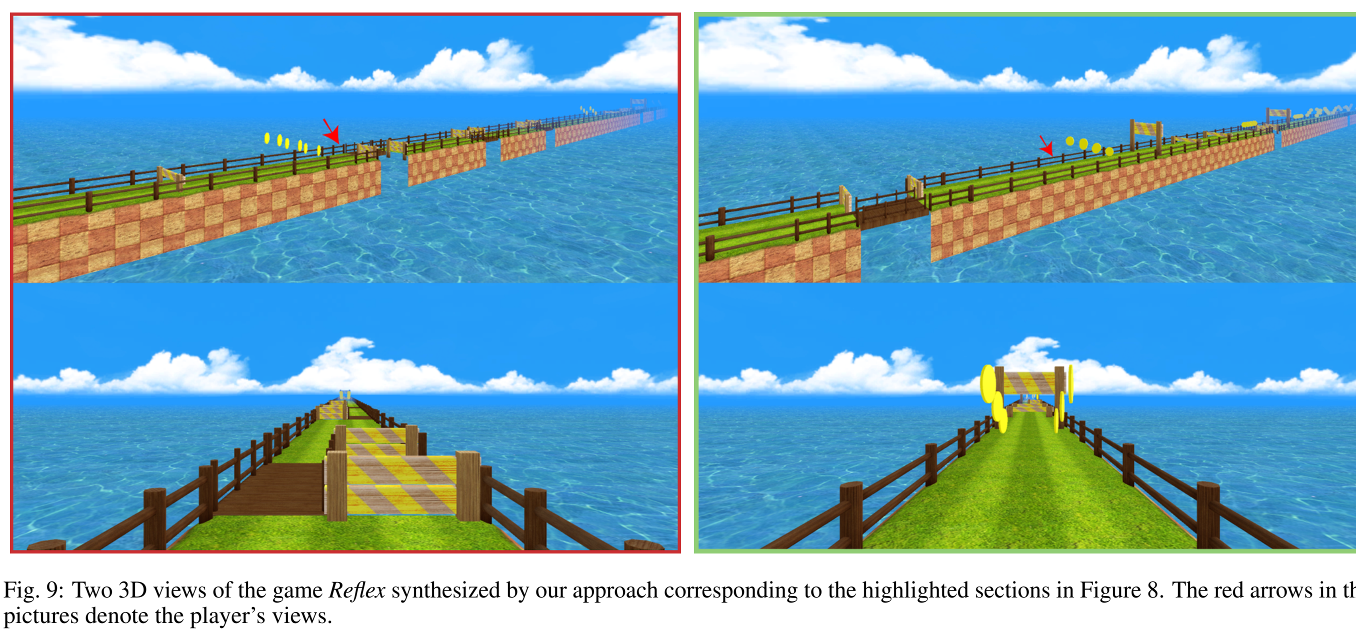


Fig. 9: Two 3D views of the game *Reflex* synthesized by our approach corresponding to the highlighted sections in Figure 8. The red arrows in the pictures denote the player's views.

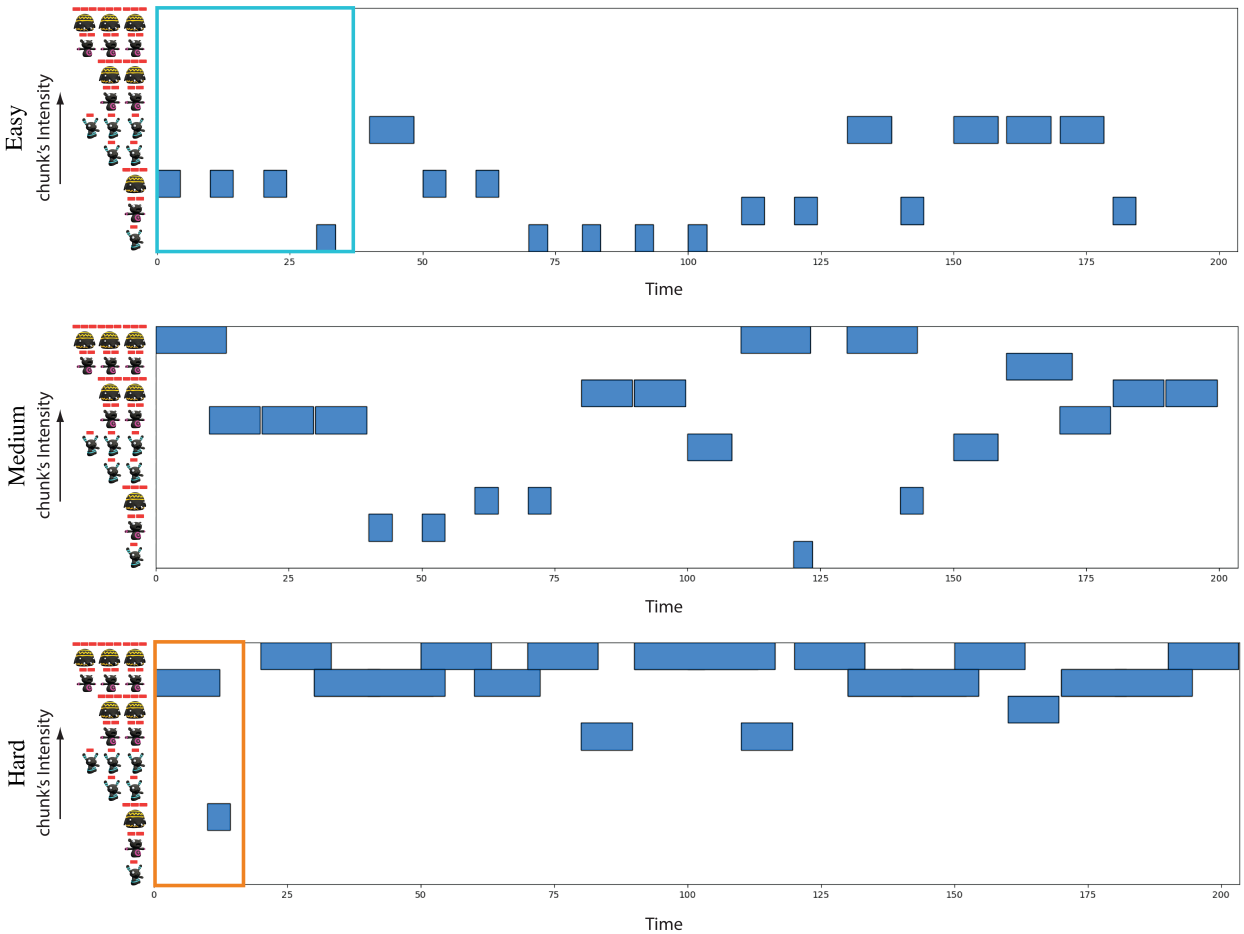
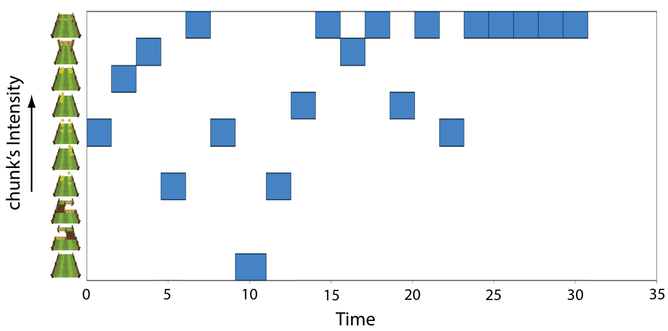
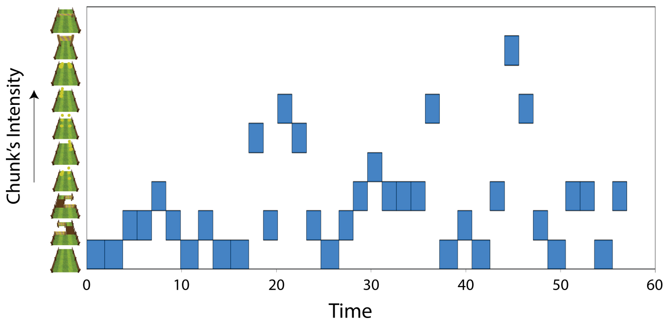


Fig. 10: Levels of the game *Longbowman* synthesized with different mean intensity levels. Each blue bar refers to a chunk that appears at a certain time of the level.

Fig. 11: Two 3D views of the game *Longbowman* synthesized by our approach corresponding to the highlighted sections in Figure 10. The yellow highlights in the pictures denote the player's views.



(a) 30-second level



(b) 60-second level

Fig. 12: Two levels synthesized with the same target calories burned (5 kcal) but different target duration (30 seconds and 60 seconds). The 30-second level has more high-intensity chunks while the 60-second level has more low-intensity chunks.

rates. A chunk made of three occurrences of *Helephants* imposes the highest exercise intensity (i.e., it is the tirdest to clear).

Results. We apply our optimization framework to synthesize different levels for *Longbowman*. Similar to *Reflex*, we synthesize three levels: *easy*, *medium* and *hard*, with target mean intensity level ρ_{m} set as 16%, 19% and 21% respectively; and target calories burned ρ_{cal} set as 15 kcal, 20 kcal and 40 kcal. We set the target duration time ρ_{d} as 180 seconds for all three levels.

Figure 10 depicts the synthesized levels and Figure 11 shows screenshots during the gameplay. Note that the chunk types shown on the left-hand-side of each plot are sorted according to their associated exercise intensity. It can be seen that the *easy* level is dominated with chunks of low intensity (e.g., "1 *Zombunny*", "1 *Zombear*"); the *medium* level is assembled by chunks of mixed intensities; and the *hard* level is dominated with chunks of high intensity (e.g., "3 *Helephants*", "3 *Zombears*"). We validate the exercise intensity levels that these levels impose on players in our user evaluation test.

5.4 Other Results

Different Duration. The designer can also synthesize levels with the same expected amount of calories burned but with different expected gameplay duration. Figure 12 shows an example for the *Reflex* game. To achieve the same amount of calories burned, the optimizer allocates more high-intensity chunks for the 30-second level while it allocates more low-intensity chunks for the 60-second level.

Fixing Chunks. The designer can synthesize a level with some of the chunks manually assigned and fixed. The optimizer with optimize the rest of the level automatically with respect to the fixed chunks. Figure 13 shows an example for the *Reflex* game.

6 Evaluation

We conducted user evaluation tests to validate the effectiveness of our approach in synthesizing game levels optimized with respect to desired physical difficulty levels.

6.1 Settings

We used the *easy*, *medium* and *hard* levels synthesized for the games *Reflex* and *Longbowman* for our user evaluation test. Figures 8 and 10 depict the levels. Through the data collected from the participants, we want to measure whether the mean exercise intensity levels and calories burned correspond to the target values specified in optimizing the levels. We also want to see whether The Borg Ratings of Perceived Exertion (RPE) given by the participants correspond to the general physical difficulty (i.e., easy, medium and hard) of the levels. RPE is frequently used in exercise sciences as a quantitative measure of perceived exertion during physical activity [3, 5].

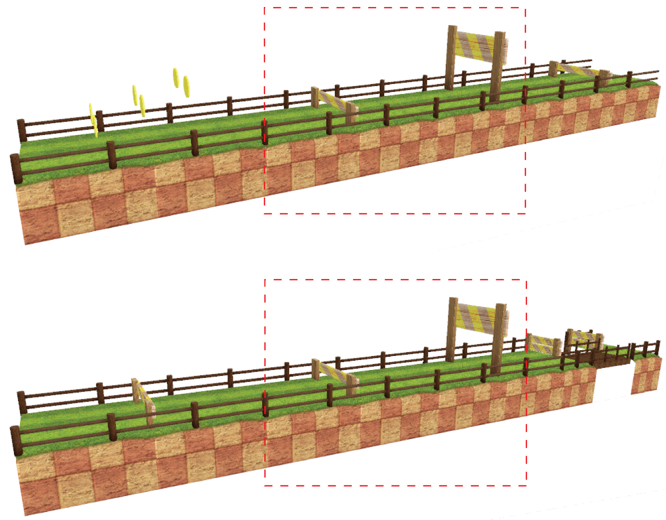


Fig. 13: Two different levels synthesized with the middle chunk (enclosed) fixed. Our approach is capable of synthesizing the rest of the levels with respect to the optimization objectives.

Participants. To conduct the evaluation, 21 participants were recruited to play each game. The participants were university students and staff, whose ages range from 16 to 37./blueNotably, none of recruited participants are preset of players who are invited to help defining chunk intensity. All participants were reported healthy and had experience using virtual reality devices.

Measurements. To measure the fatigue level of the participant, we used a wearable activity tracker, the Polar heart rate sensor, to keep track of the heart rates of the participants (Figure 14). The participant wore this device throughout the whole experiment. A higher recorded heart rate corresponds to more calories burned and a higher fatigue level in general. For each level, we discard the first 30 seconds of heart rate data as it refers to the period that the participant was warming up. After finishing each level, the participant was also asked to give a rating of perceived exertion (RPE) in a 6 to 20 scale [4] to describe the physical efforts involved in completing the level as he perceived, with "6" indicating "No exertion", "13" indicating "Somewhat hard exertion" and "20" indicating "Maximal exertion". All measurements followed IRB regulations of the research institute.

Fig. 14: A Polar heart rate sensor worn over the chest.

Procedure. Before starting the test, a participant was first briefed about the game control and he was allowed to get familiar with the gameplay in a warm-up session. After that, he was asked to take a rest such that his heart rate got back to his resting heart rate. Then he played the *easy*, *medium* and *hard* levels of the game in a randomized order. After playing a level, he gave a RPE for the level. Then he was asked to take a rest so that his heart rate got back to the normal level before playing another level. This process was repeated until he finished playing each level of each game.

For *Reflex*, the participant was asked to complete each level by going through all its chunks. For *Longbowman*, the participant was asked to shoot as many enemies as possible. As a motivation, he was told before the test that the amount of monetary reward that he would receive as a participant would depend on the number of enemies he had shot, though at the end we would pay everyone the same amount of monetary reward regardless.

6.2 Results

Figure 15 shows the results of the user evaluation tests. For each level (*easy*, *medium* and *hard*) of each game, the average results (calories burned, exercise intensity, RPE) obtained from the participants are shown, which generally increase with the physical difficulty of the levels.

For *Reflex*, it can be observed that the average results follow the target values specified for synthesizing the levels quite closely. For both calories burned and exercise intensity, the results and target values differ only by about 10% or less. The RPEs for the *easy*, *medium* and *hard* levels are 8.96, 12.25 and 14.29 respectively, corresponding to "very light", "somewhat hard" and "hard" [4] according to Borg's scale.

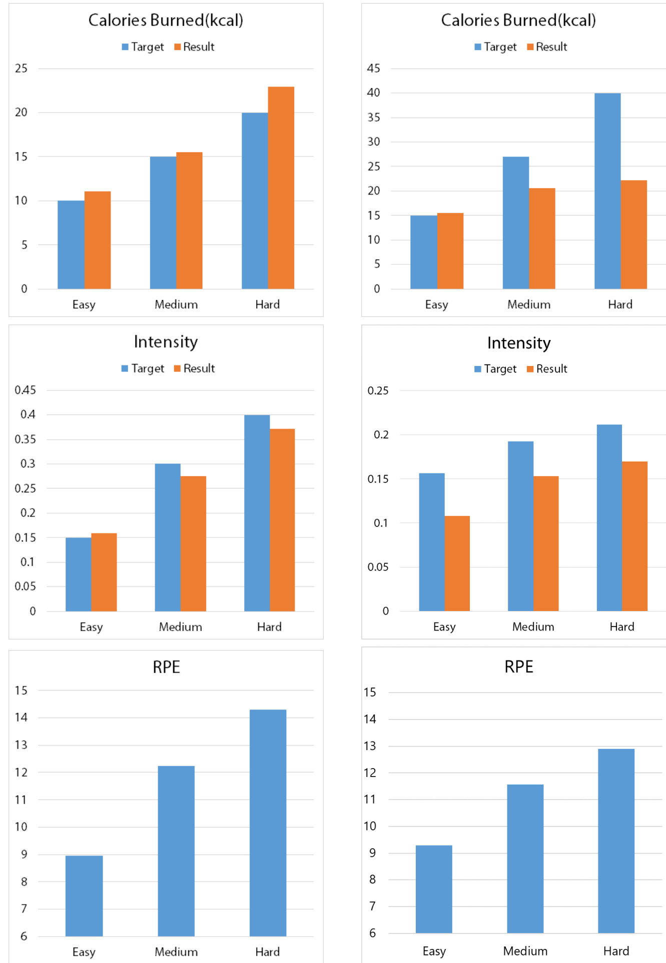
For *Longbowman*, more significant deviation is observed between the average results and the target values specified for synthesis, especially for the *medium* and *hard* level, though the results still increase with the physical difficulty of the levels as expected. The RPEs for the *easy*, *medium* and *hard* levels are 9.28, 11.57 and 12.90, corresponding to "very light", between "light" and "somewhat hard", and "somewhat hard" respectively.

One reason for the small deviation in the results of *Reflex* is that the participants were "forced" to complete each chunk of a level by moving their bodies accordingly in order to finish the whole level. In other words, the sequence of motions they performed were literally the same as expected, and hence the amount of calories burned and the exercise intensity levels.

In contrast, in *Longbowman*, some participants might have failed to complete some chunks (i.e., some participants might have failed or did not attempt to shoot some of the enemies before they escaped). In other words, some participants might not have gone through the whole sequence of body motions as expected to shoot the enemies, hence resulting in a lower amount of calories burned and exercise intensity level than expected. This deviation from expectation is especially pronounced for the *medium* and *hard* levels, probably because some participants felt that it was tiring to fully complete those levels. In fact, the percentages of enemies killed for the *medium* and *hard* levels are 73% and 53% respectively, in contrast to 96% for the *easy* level, which might justify the explanation.

The results hint that a level designer can apply our approach to automatically generate levels of a motion-based game that can help players to achieve calories burned and exercise intensity targets, which are specified as optimization parameters. How closely those targets are met depends on whether the player is compulsorily required to complete each chunk of a level. Not enforcing such compulsory requirement may result in more deviation from the target, yet the player may find the gaming experience more relaxing given the flexibility.

Our supplemental material contains the data of the user evaluation tests and further details of the user performances in different levels.



(a) *Reflex* (b) *Longbowman*

Fig. 15: Results of user evaluation tests of (a) *Reflex* and (b) *Longbowman*. For each game, the average calories burned and the average exercise intensity levels for the *easy*, *medium* and *hard* levels are shown, which are compared with the target values used for synthesizing the levels. The results show corresponding trends as the target values, with greater deviation observed for the *medium* and *hard* levels of *Longbowman*. Refer to the main text for more details. The ratings of perceived exertion (RPE) also increase with the physical difficulty of the levels for both games.

6.3 User Feedback

We spoke with the participants after the experiments. A majority of the participants found the two games entertaining. Some reported that exergaming changed the way they used to think of workout, which impressed them as repetitive, boring and not engaging. They said that they would be more motivated to exercise in virtual environments because of the appealing events and effects they could experience in virtual worlds.

A few participants reported that they have experienced a bit of dizziness during part of the running example game *Reflex*. They pointed out that when they were jumping, they felt insecure because they were worried about getting tripped. In future, a wireless virtual reality headset and setup with motion-tracking capability could potentially resolve this concern and enhance user experience in virtual reality-based exergaming.

7 Summary

We introduce a novel problem of procedurally generating game levels with consideration of exercise intensity. We also devise a novel, optimization-based framework capable of automatically generating such game levels, and demonstrate its effectiveness for designing the levels of different motion-based games. The flexibility of our framework allows it to be practically and generally applied to generate appropriate game levels for different motion-based games which can be played with virtual reality devices. We will release our tool as a game engine plugin to facilitate game design in the virtual reality community.

Limitations and Future Work. As far as the length of our synthesized game levels is concerned, each of the synthesized level used in our evaluation experiments only lasts for about three minutes. The exercise intensity that a participant experienced in each gameplay was not compatible to that of a 45-minute exercise session in a typical workout. We synthesized short levels for verifying our approach through experiments; in practice longer levels could be synthesized by using a larger target duration parameter ρ_d. Moreover, since the intensity of each chunk in a level is calculated with a preset of players, the system may not be generalize well to all players. Thus, a large-scale pretest that contains diverse demographics is needed for extending our approach to the general population. Also, we discovered that if a game (e.g., the *Longbowman* game) does not compulsorily require players to complete each chunk, the result may deviate from the target. One solution could be adding a real-time exercise intensity tracker to the system (e.g., using a heart rate sensor) and encouraging players to put more efforts to play if their exercise intensity falls below a threshold.

Besides, our games were played through the HTC Vive, which requires the participant to wear a wired headset during his gameplay. This constraint on our participants brought inconvenience to their movement.

Additionally, the HTC Vive headset is approximately 12 cm-tall, 19 cm-wide and 12 cm-deep (measured from the front to the back of the headset), and its weight is one pound. The weight and size could make some participants feel uncomfortable during the gameplay.

When it comes to "intensity", our approach only considers the exercise intensity that a synthesized game level imposes upon the player. Other aspects, such as the intensity in terms of mental demand (e.g., staying alert), and the visual experience that the player will go through in a synthesized level could also be important to consider. For example, a level that continuously requires the player to respond quickly and precisely in order to win could be mentally challenging and tiring as the player needs to stay highly focused for a prolonged period of time. The visual content of the levels should also be optimized to reduce the potential problems of eye fatigue and dizziness.

Our approach uses heart rates to characterize the exercise intensity levels associated with different chunk types. Though heart rate is a commonly used metric in exercise science, alternative metrics such as oxygen consumption could be used. Our formulation (Equation 5) considers basic body metrics of the player such as the player's age, gender and weight in synthesizing a level. However, to synthesize more personalized level designs for training purposes, a more comprehensive fitness test of the player could be performed to obtain more detailed body metrics such as VO_2 Max, as well as flexibility and muscular strengths of different parts of his body. Applying such metrics for synthesizing appropriate levels of a motion-based game could be an interesting future research direction.

Finally, it would be interesting to extend our approach to consider more sophisticated and long-term health effects that the synthesized game levels may bring. Such investigations will facilitate the development of virtual reality-based exergames for fitness and other health-related purposes, such as weight loss and rehabilitation.

REFERENCES

[1] M. Agmon, C. K. Perry, E. Phelan, G. Demiris, and H. Q. Nguyen. A pilot study of wii fit exergames to improve balance in older adults. *Journal of geriatric physical therapy*, 34(4):161–167, 2011.

[2] I. Bogost. The rhetoric of exergaming. *Power*, 2005.

[3] G. Borg. Perceived exertion as an indicator of somatic stress. *Scandinavian journal of rehabilitation medicine*, 2(2):92, 1970.

[4] G. Borg. *Borg's perceived exertion and pain scales.* Human kinetics, 1998.

[5] G. A. Borg. Psychophysical bases of perceived exertion. *Med sci sports exerc*, 14(5):377–381, 1982.

[6] K. Compton and M. Mateas. Procedural level design for platform games. In *Proceedings of the Second AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE'06, pages 109–111. AAAI Press, 2006.

[7] CSE Entertainment. CSE Entertainment Ltd. `https://cse.fitness/`, 2017. Accessed 07-14-2017.

[8] D. Dimovska, P. Jarnfelt, S. Selvig, and G. N. Yannakakis. Towards procedural level generation for rehabilitation. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, page 7. ACM, 2010.

[9] L. Donath, R. Rössler, and O. Faude. Effects of virtual reality training (exergaming) compared to alternative exercise training and passive control on standing balance and functional mobility in healthy community-dwelling seniors: a meta-analytical review. *Sports medicine*, 46(9):1293–1309, 2016.

[10] S. Finkelstein, A. Nickel, Z. Lipps, T. Barnes, Z. Wartell, and E. A. Suma. Astrojumper: Motivating exercise with an immersive virtual reality exergame. *Presence: Teleoperators and Virtual Environments*, 20(1):78–92, 2011.

[11] K. Gerling, R. Mandryk, et al. Custom-designed motion-based games for older adults: A review of literature in human-computer interaction. *Gerontechnology*, 12(2):68–80, 2014.

[12] P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[13] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup. Procedural content generation for games: A survey. *ACM Trans. Multimedia Comput. Commun. Appl.*, 9(1):1:1–1:22, Feb. 2013.

[14] B. Horn, S. Dahlskog, N. Shaker, G. Smith, and J. Togelius. A comparative evaluation of procedural level generators in the mario ai framework. In *International Conference on Foundations of Digital Games*, 2014.

[15] H. Huang, N.-C. Lin, L. Barrett, D. Springer, H.-C. Wang, M. Pomplun, and L.-F. Yu. Automatic optimization of wayfinding design. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[16] ICAROS. ICAROS - We make you fly. `https://www.icaros.com/`, 2017. Accessed 07-14-2017.

[17] K. Isbister. *Motion-Based Game Design: How to Design Quality Movement Games for Wii, Kinect, and Move*. Morgan Kaufmann Publishers Inc., 2013.

[18] M. Jennings-Teats, G. Smith, and N. Wardrip-Fruin. Polymorph: A model for dynamic level generation. In *Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2010.

[19] E.-K. Ji and S.-H. Lee. Effects of virtual reality training with modified constraint-induced movement therapy on upper extremity function in acute stage stroke: a preliminary study. *Journal of physical therapy science*, 28(11):3168–3172, 2016.

[20] J. Karvonen and T. Vuorimaa. Heart rate and exercise intensity during sports activities. *Sports Medicine*, 5(5):303–311, 1988.

[21] L. Keytel, J. Goedecke, T. Noakes, H. Hiiloskorpi, R. Laukkanen, L. van der Merwe, and E. Lambert. Prediction of energy expenditure from heart rate monitoring during submaximal exercise. *Journal of sports sciences*, 23(3):289–297, 2005.

[22] J. Kim, J. Son, N. Ko, and B. Yoon. Unsupervised virtual reality-based exercise program improves hip muscle strength and balance control in older adults: a pilot study. *Archives of physical medicine and rehabilitation*, 94(5):937–943, 2013.

[23] B. D. Levine. : what do we know, and what do we still need to know? *The Journal of physiology*, 586(1):25–34, 2008.

[24] E. F. Ogawa, T. You, and S. G. Leveille. Potential benefits of exergaming for cognition and dual-task function in older adults: a systematic review. *Journal of aging and physical activity*, 24(2):332–336, 2016.

[25] M. Pasch, N. Bianchi-Berthouze, B. van Dijk, and A. Nijholt. Movement-based sports video games: Investigating motivation and gaming experience. *Entertainment Computing*, 1(2):49–61, 2009.

[26] D. Schoene, T. Valenzuela, S. R. Lord, and E. D. de Bruin. The effect of interactive cognitive-motor training in reducing fall risk in older people: a systematic review. *BMC geriatrics*, 14(1):107, 2014.

[27] N. Shaker, G. N. Yannakakis, and J. Togelius. Towards player-driven procedural content generation. In *Proceedings of the 9th conference on Computing Frontiers*, pages 237–240. ACM, 2012.

[28] L. A. Shaw, B. C. Wünsche, C. Lutteroth, S. Marks, and R. Callies. Challenges in virtual reality exergame design. In S. Marks and R. Blagojevic, editors, *Proceedings of the 16th Australasian User Interface Conference (AUIC), 2015*, pages 61–68. Australian Computer Society, 2015.

[29] P. Shi and K. Chen. Online level generation in super mario bros via learning constructive primitives. In *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*, pages 1–8. IEEE, 2016.

[30] T. X. Short and T. Adams. *Procedural Generation in Game Design*. A K Peters/CRC Press, 2017.

[31] J. Sinclair, P. Hingston, and M. Masek. Considerations for the design of exergames. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 289–295. ACM, 2007.

[32] N. Sorenson and P. Pasquier. The evolution of fun: Automatic level design through challenge modeling. In *ICCC*, pages 258–267, 2010.

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

[34] T.-L. Sun and C.-H. Lee. An impact study of the design of exergaming parameters on body intensity from objective and gameplay-based player experience perspectives, based on balance training exergame. *PloS one*, 8(7):e69471, 2013.

[35] J. Togelius and G. N. Yannakakis. Emotion-driven level generation. In *Emotion in Games*, pages 155–166. Springer, 2016.

[36] VirZOOM. Virzoom. `https://www.virzoom.com/`, 2017. Accessed 07-14-2017.

[37] S. Wüest, N. A. Borghese, M. Pirovano, R. Mainetti, R. van de Langenberg, and E. D. de Bruin. Usability and effects of an exergame-based balance training program. *GAMES FOR HEALTH: Research, Development, and Clinical Applications*, 3(2):106–114, 2014.

[38] G. N. Yannakakis and J. Togelius. Experience-driven procedural content generation. *IEEE Trans. Affect. Comput.*, 2(3):147–161, July 2011.

[39] L.-F. Yu, S. K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. Osher. Make it home: automatic optimization of furniture arrangement. *ACM Trans. Graph.*, 30(4):86, 2011.