Doctoral Dissertation

Generating Motion Effects for Moving Objects in 4D Content

Sangyoon Han (한 상 윤)

Department of Computer Science and Engineering

Pohang University of Science and Technology

2023

4D 컨텐츠에서 움직이는 다양한 물체에 대한 모션 효과의 자동 생성

Generating Motion Effects for Moving Objects in 4D Content

Generating Motion Effects for Moving Objects in 4D Content

by

Sangyoon Han
Department of Computer Science and Engineering
Pohang University of Science and Technology

A dissertation submitted to the faculty of the Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of philosophy in the Computer Science and Engineering

Pohang, Korea
11. 11. 2023
Approved by
Seungmoon Choi
Academic advisor

Generating Motion Effects for Moving Objects in 4D Content

Sangyoon Han

The undersigned have examined this dissertation and hereby certify that it is worthy of acceptance for a doctoral degree from POSTECH

11. 11. 2023

Committee Chair	Seungmoon Choi	(Seal)
Member	Sung H. Han	(Seal)
Member	Inseok Hwang	(Seal)
Member	Jaesik Park	(Seal)
Member	Sunghyun Cho	(Seal)

DCSE 20162686

한 상 윤. Sangyoon Han
Generating Motion Effects for Moving Objects in 4D Content,
4D 컨텐츠에서 움직이는 다양한 물체에 대한 모션 효과의 자동 생성
Department of Computer Science and Engineering , 2023, 125p, Advisor : Seungmoon Choi. Text in English.

ABSTRACT

4D content enhances the user's multimedia experience using various sensory effects such as motion, vibration, heat, and wind with audiovisual content. One of the most frequently used 4D effects is the *object-based motion effect.* The object-based motion effect refers to the movement of a chair emphasizing the object movements of interest in a scene. In the scene, several objects often simultaneously appear, and each object moves with high degrees of freedom (DoF) by its parts. In contrast, commercial motion platforms used in 4D theaters support only limited DoF and workspace. Therefore, it is difficult to author object-based motion effects.

To facilitate the production of such effects, many techniques for automatically generating motion effects by analyzing audiovisual streams have been proposed. However, these methods generated motion effects by focusing only on the translation of a single rigid body object. This dissertation presents an algorithm for automatically generating motion effects that delicately express the complex movements of various objects. To this end, we introduce a new concept of motion proxy, which abstracts the multiple movements of high DoFs into a single 3-DoF motion, and convert it with a motion effect.

First, we developed an algorithm generating motion effects for a rigid body. We compressed a 6-DoF motion of a rigid body into a 3-DoF motion proxy by combining translation and rotation according to the object size. Next, we developed an algorithm that expresses the high DoF movements of multiple articulated bodies. To inclusively represent the movements of the articulated bodies, we calculated the motion proxy by combining the movements of parts according to their speed and size. Finally, we automatically generated motion effects from all moving components in a scene, such as rigid bodies, articulated bodies, and particle movements. In this process, we used computer vision techniques such as scene flow estimation to automatically extract the motion information of a scene and calculated a motion proxy by combining the extracted motions according to their importance.

In each step, we implemented the optimal algorithm by performing several user studies to determine the algorithm details and optimal parameters. The experimental results confirmed that our algorithm creates convincing motion effects that improve user multimedia experiences better than the previous methods.

Contents

List of Tables

List of Figures

I. Introduction

Viewing multimedia content satisfies our eyes and ears. However, we want more: as a new form of media, mulsemedia (multiple sensorial media) leverages more than the two senses of vision and audition to expand our sensory experiences [2, 3]. Mulsemedia has already permeated our daily lives. In arcades, a racing game is equipped with a motion platform simulating the movement of a vehicle [4]. Also gaining popularity are 4D theaters, where the audience's viewing experience is enhanced with various sensory effects, such as motion, vibration, scent, and wind, in synchrony with audiovisual content [5]. Vive [6] and Meta [7], the major companies for virtual reality (VR) headsets, are conducting intensive research to provide improved haptic feedback through handheld controllers during interaction in VR.

Mulsemedia content is delivered to us through three stages: production, distribution, and rendering [8, 3]. At present, a critical bottleneck is in the production of sensory effects because it still relies on manual authoring. This costly process prevents faster and wider dissemination of mulsemedia content. For facilitation, a few research groups have proposed algorithms to automate sensory effect production based on audiovisual content [9, 10, 11, 12, 13]. Our works described in this paper are in line with such endeavors, with an emphasis on *motion effects*.

According to the taxonomy of 4D effects established by Lee et al. [9], motion effects used in 4D films can be classified into four classes: camera-based, object-based, vibration and impact, and context, based on the audiovisual grounds used for the design. Here, an *object-based motion effect* refers to the movement of a motion chair that applies physical movement to the viewer's body to emphasize

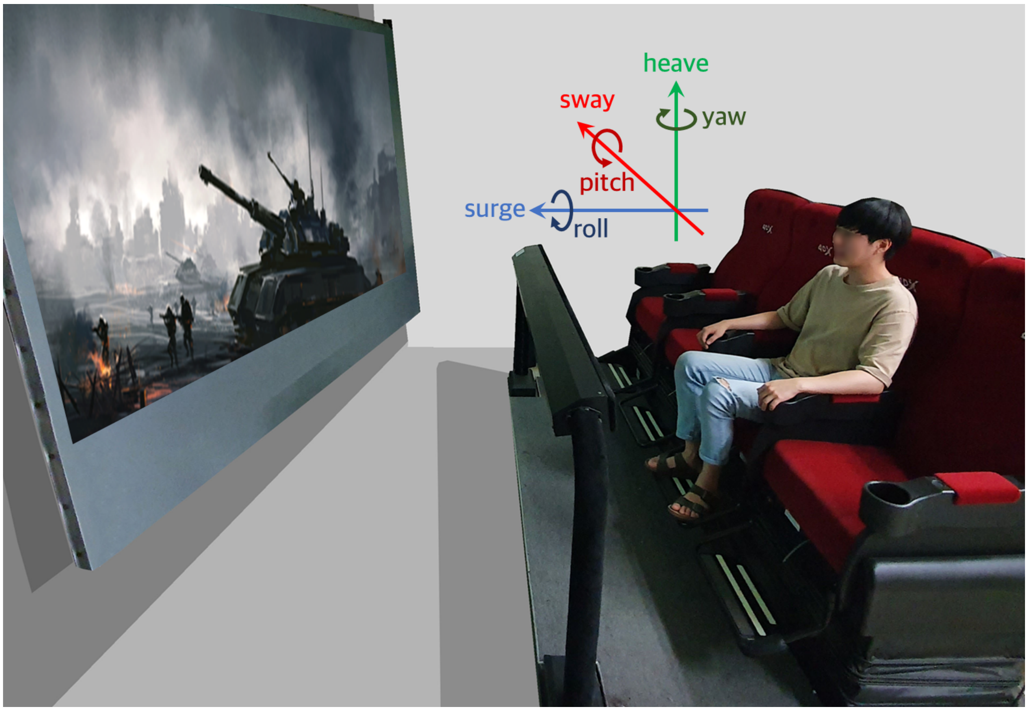


Figure 1.1: A person watching a scene with accompanying motion effects. A scene is full of many moving components, but a motion platform used in 4D theaters only supports roll, pitch, and heave motion. The motion platform is used in all the experiments reported in this paper.

the movement of an object of interest, e.g., the fighting Captain America (Figure 1.1). Among the four classes of motion effects, object-based motion effects are the most frequently used in 4D films. As shown in Figure 1.1, a scene has numerous moving components, including rigid bodies, articulated bodies, and particles. In contrast, the standard motion platforms used in 4D theaters only support limited degrees of freedom (DoF) and workspace. Since the motion platforms can't reproduce all movements, the object-based motion effects should be carefully designed, focusing on the important movements.

Motivated by this, this dissertation proposes automatic algorithms for generating motion effects for various objects. Our approach combines many movements into a representative movement according to their importance and converts the combined movement into a motion effect. To this end, we introduce the concept of *motion proxy*.

1.1 Organization

The rest of this thesis is structured as follows. Chapter II introduces the background and related works on the production of multisensory effects, focusing on motion effects. In Chapter III, we present a method of creating motion effects for a rigid body. To express the 6-DoF motion of a rigid body with a 3-DoF motion effect, we first introduce the concept of a motion proxy. Here, the motion proxy represents the movement of a rigid body by combining its translation and rotation with a movement of a single point. Chapter IV describes a motion generation method for multiple articulated bodies. To represent complex movements of multiple articulated bodies as a motion effect, we extend the concept of a motion proxy. We calculated an object motion proxy, representing the movements of an articulated body, by combining the movements of object parts, and a scene motion proxy, representing the movements of all articulated bodies, by combining the object motion proxies. In Chapter V, we generate motion effects for all the components consisting of a scene. We compute a scene motion proxy based on the scene flow for every pixel on the image.

1.2 Contributions

The major contributions of our research are:

1. The first proposal of the concept of motion proxy, representing multiple movements by a single movement by combining them.

2. The optimal combinations of movements according to various objects in

calculating a motion proxy.

3. The motion cueing algorithms designed considering visual perception for moving objects and vestibular perception for motion effects.
4. The shared design guidelines earned through the research.

These outcomes are expected to expedite and benefit the production of 4D multisensory effects.

II. Related Work

2.1 Authoring Multisensory Effects

Multisensory system presents immersive experiences by providing various sensory effects together with audiovisual content, such as motion [**?**, 14, 15, 16], vibration [17, 18], heat [19], scent [20, 21], and wind [22, 23]. The first multisensory system was invented in the 1960s. Sensorama platform [24], developed by Heilig, created the experience of riding a motorcycle in New York using a stereoscopic color display, a stereo sound system, fans, odor emitters, and a motion chair. Since then, mulsemedia systems have been steadily developed and applied to various domains, such as broadcasting [25, 26], theaters [27, 28], museums [29, 30], virtual reality [31, 32, 33, 34, 35, 36], and interactive environments [37].

Researchers have designed novel stimulation methods to deliver multisensory experiences, with focused efforts on haptics. For example, Danieau et al. [38] demonstrated that stimulating the user's head and arms using low-cost force-feedback devices can elicit convincing walking sensations, similarly to riding a motion platform. Danieau and colleagues [39] also introduced the concept of haptic cinematography and the associated taxonomy while showcasing a few haptic emphasis techniques for such films. Sra et al. [40] presented a device providing proprioceptive feedback using galvanic vestibular stimulation to replace an expensive motion platform.

In general, mulsemedia content is streamed to users through three stages: (1) production, (2) distribution, and (3) rendering [8, 3]. First, sensory effects are created in synchrony with audiovisual content (production). Second, the sensory effects are formalized into the data to be synchronized, stored, and transmitted with the audiovisual media (distribution). Finally, the sensory effects are rendered through various multisensory devices to users (rendering).

Among the three, the production stage is the biggest bottleneck preventing the proliferation of mulsemedia. In industry, designers use in-house authoring programs to create multisensory effects [41, 42, 43, 26]. These tools have similar interfaces using multiple timelines, one for each sensory effect. While useful, the designers must annotate all video segments that will accompany multisensory effects. To reduce such efforts, there have been attempts to detect video segments for which sensory effects will be provided and determine the corresponding effect types using the neural networks trained on audiovisual information [44, 45, 11, 46, 47]. However, the designers still need to make specific sensory effects manually, which leaves the design process extremely time-consuming. Another useful asset is an effect library, which stores frequently-used sensory effects and enables the designers to load and reuse them whenever necessary [48, 49, 50]. Such effect libraries are useful and can speed up the authoring and production in some cases, e.g., vibration effects conveying the sensations of walking, gunfire, and vibration.

2.2 Automatic Generation of 4D Effects

In the past decade, we have seen increasing research interests in the algorithmic generation of 4D effects from existing audiovisual data, which is essential to accelerate the spread of mulsemedia and related VR applications. This section describes the previous research outcomes mainly pertinent to motion effects. Motion effects have a central role among the various 4D sensory effects [9].

Shin et al. [51] were the first to propose a synthesis framework for motion effects that emphasize camera motion for point-of-view (POV) shots, which are named camera-based motion effects in [9]. The camera trajectory is estimated from sequential images and then used to compute the command to a motion chair using a washout filter. Lee et al. [9] developed much more efficient video-based generation algorithms of camera-based motion effects with quasi-real time performance and compelling viewer experiences. These algorithms were adapted by Seo et al. [52] to substitute camera-based motion effects with vibrotactile effects as a much less expensive alternative. Lim et al. [53] proposed an algorithm for styling camera-based motion effects using a texture image to provide more realistic sensations of vehicle riding, e.g., on a bumpy road.

As for object-based motion effects, the first approach was presented by Lee et al. [10], where 2-DoF motion effects (roll and pitch in Figure 1.1) are converted from the 2D trajectory of the object motion projected to the screen. An object of interest is tracked by a computer vision algorithm, and its 2D screen velocity is fed to a washout filter. This strategy of viewer-centered rendering matches the chair's motion to the shift of the audience's visual attention, which results in plausible motion effects for many scenes. Goh et al. [54] used saliency estimation with the optical flow to generate motion effects for visually salient objects. In Zhao et al. [55], a force vector is extracted from the velocity of an object of interest tracked in the image. It is rendered using a haptic glove that provides force feedback. These algorithms all operate on the screen space, where the rotation of an object may not be disclosed clearly. All the aforementioned generation algorithms for object-based motion effects are applicable only to a single rigid body.

Additionally, Lee and Choi [56] proposed a sound-to-vibration translation algorithm for selective audio-tactile feedback (e.g., not responding to background music) by relating perceptual loudness and roughness between sound and touch. Later, this algorithm was adapted to present motion effects that accentuate impact and collision events [9]. Recently, Li et al. [13] designed a method to provide localized vibrotactile effects by combining the multisensory information extracted from audio signals and image sequences. For gameplay spectators, Yun et al. [12]

proposed a motion synthesis algorithm that detects gunfire sounds using a deep learning model and generates a motion effect that resembles gun recoil.

Motion effects can also elicit the sensations of walking by leveraging the cyclic nature of human gaits. Amemiya and colleagues conducted a series of studies on walking sensation rendering using a motion platform [57, 58, 59]. For example, they demonstrated that motion effects in the heave direction can deliver more compelling walking sensations than motion effects in the yaw direction for users wearing head-mounted displays (HMDs) [59]. Recently, Lee et al. [60] proposed a data-driven framework for automatically generating the motion effects that provide users with walking sensations. Locomotion data in different gaits, e.g., walking, running, and stumping, was measured using motion sensors, and the captured data was converted to motion effects through a few steps of signal processing algorithms.

One last notable approach is synthesizing the motion effects with specific perceptual properties, presented by Han et al. [61]. By conducting dissimilarity estimation and adjective rating for various motion profiles, they identified *smooth-rough* and *irregular-regular* as two primary perceptual dimensions of motion effects. They presented several methods to synthesize a new motion effect with specific properties by interpolating multiple motion effects in the authoring space consisting of the *smooth-rough* and *irregular-regular* axes. Their user study showed that the synthesized motion effects express the target characteristics well.

2.3 Motion Cueing Algorithms

A motion platform, which is used for motion effects, can not reproduce the object movements in a scene due to its hardware limits, such as the maximum displacement and velocity. Motion cueing algorithms (MCA) is a technique to reproduce the desired motion within the hardware limits. Motion platforms and

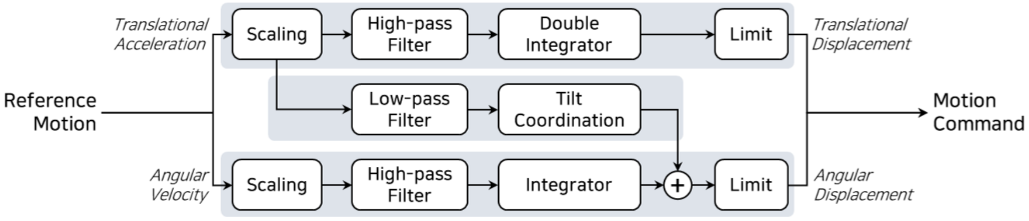


Figure 2.1: Classical washout filter.

MCAs were originally developed for flight simulations in the early 1970s [62], and they have recently been applied to the mulsemedia and 4D systems. The MCAs can be categorized into two groups: filter-based and optimization-based.

Filter-based MCAs rely on scaling down and filtering the reference motion so that the produced motion commands lie within the motion platform's workspace. The most widely used algorithm is a classical washout filter. An overall flow of the classical washout filter is depicted in Figure 2.1. The washout filters remove low-frequency motion to prevent the motion platform from exceeding its workspace [63]. The classical washout filter has two channels for translation and rotation. These channels scale the translational acceleration and angular velocity and feed them to high-pass filters. Then, the filtered acceleration and angular velocity are integrated twice and once, respectively, to be position (or angle) commands. Since this process does not guarantee that the motion command is within the maximum range of the motion platform, the motion commands need to be limited. This algorithm also includes a channel for *tilt-coordination*. For the simulation of a sustained acceleration to the body, this channel feeds the acceleration to a low-pass filter and adds the filtered acceleration to the rotational motion command. Here, the rotation rate can be limited to prevent a user from perceiving the rotational motion due to tilt-coordination.

While the washout filters can compute the motion commands in real-time, the parameters, such as gain and cutoff frequency, should be manually tuned

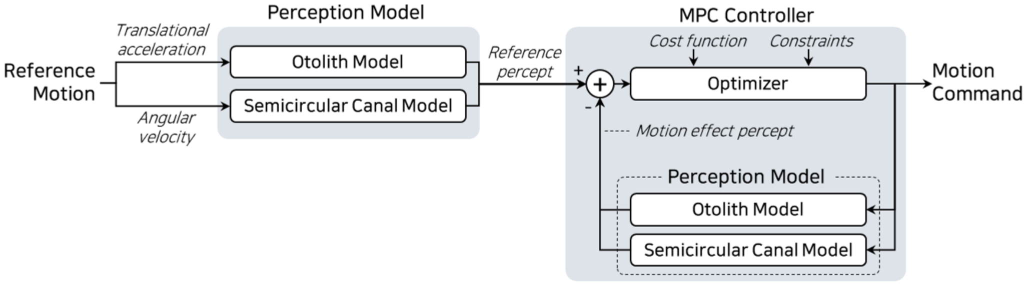


Figure 2.2: Model predictive control.

through a trial-and-error process, and the low-frequency motion information is lost. The variances of classical washout algorithms, such as adaptive washout [64] and linear optimal control [65], have been proposed to compensate for these shortcomings, but they do not completely overcome them.

Recently, several optimization-based MCAs have been developed. Unlike filter-based MCAs, optimization-based MCAs optimize motion commands to follow the reference trajectory, taking into account the constraints of the motion platform. The representative approach is model predictive control (MPC). MPC is a methodology that optimizes current control inputs based on a process model and future trajectories while considering constraints [?]. The optimization is governed by a cost function that quantifies the difference between the reference output and the output generated by the process model. This algorithm has the advantage of taking the platform limit into account explicitly. Particularly, the computational models of vestibular perception can be used as process models to reduce perceptual error [66, 67, 68].

Figure 2.2 depicts how the MPC algorithm embedding the perception models generates commands from the reference motion. First, the perceived acceleration and angular velocity are calculated by the perception models for a finite future trajectory and fed to the MPC controller. The MPC controller calculates the motion command that satisfies the constraint and minimizes the cost function.

Here, the constraint may be a hardware constraint, such as the maximum displacement, velocity, or acceleration of the motion platform. The cost function can be simply defined as the difference between the input and output motion percepts. Optionally, the cost function can include the weight of the platform displacement and velocity to ensure the stability of the platform.

In the cost function, the weight for displacement allows the motion commands to converge gradually to zero. Also, the perception models play the role of the band-pass filter, like the high-pass filters of a classical washout filter. Additionally, *tilt-coordination* can be explicitly embedded in the MPC's process model to consider the acceleration due to gravity when the platform is tilted.

However, because of their high computational complexity, optimization-based algorithms are difficult to apply in real-time applications of high sampling rates, such as above 50 Hz. See [69, 67, 70] for more detailed reviews of MCAs.

III. Generating Motion Effects for a Rigid Body

In this chapter, we propose an automatic algorithm for synthesizing convincing object-based motion effects for the movement of *a rigid body* from a given object motion trajectory. While previous approaches focused on only the 2D position of an object on the screen, our method represents both the 3D position and orientation of an object in the camera space by introducing a new concept of a *motion proxy*. The motion proxy is determined based on the results of a perceptual experiment that presents an optimal additive rule of the translation and rotation information scaled by the object's visual size. We generated a motion effect for a rigid body by feeding the motion proxy to a motion cueing algorithm (MCA). We investigated which MCA is most suitable for object-based motion effects among several candidates in terms of user experiences through a user experiment.

3.1 Problem Formulation

As shown in Figure 3.1, a rigid body object has six DoFs to move in the 3D camera space. In the camera coordinate frame, we represent the object by its center position $\mathbf{p}^{cam} = (p_x, p_y, p_z)^T$ and frontal direction $\mathbf{d}^{cam} = (d_x, d_y, d_z)^T$, where $\mathbf{d}^{cam}$ is a unit vector. The size l denotes the subtended visual angle of the object. The object is projected onto a 2D image plane and displayed on the screen.

The object can freely translate and rotate in the camera frame, leading to a motion of 6 DoFs. In contrast, almost all commercial motion platforms used in 4D theaters have only 3 DoFs represented by $\mathbf{m} = (roll, pitch, heave)^T$. Then,

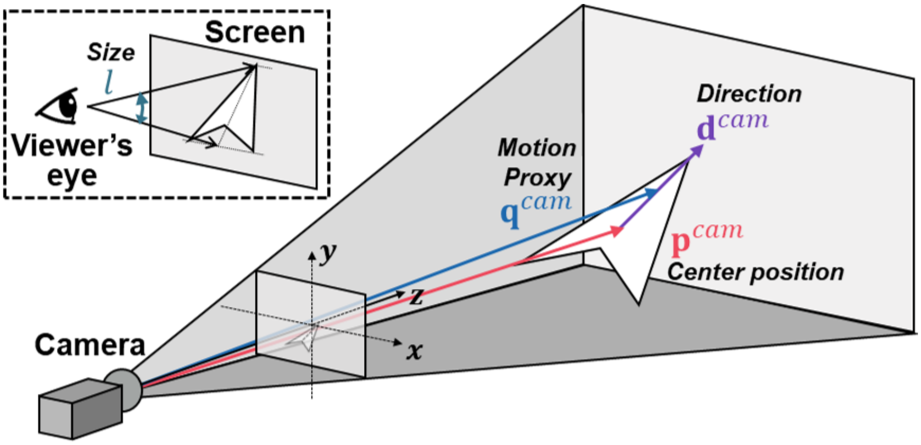


Figure 3.1: A moving object and its proxy in the 3D camera space.

our research problem is formulated to find the motion command $\mathbf{m}$ to the motion chair from the object motion $\mathbf{p}^{cam}$ and $\mathbf{d}^{cam}$ and the object size l. Note that the DoF must be reduced during the transformation.

Lee et al. [10] approach this problem by projecting $\mathbf{p}^{cam}$ to the image plane (so to the screen) and then converting the two horizontal and vertical positions to the roll and pitch motion commands. This simplicity may lead to insufficient transmission of the original motion information, e.g. if the translation or rotation of the object in the z-axis in Figure 3.1 is meaningful. While watching many 4D films, we indeed observed that both translation and rotation of objects are expressed as motion effects. We also noticed that the amplitudes of motion effects for translation and rotation depend on the object size. Therefore, we aimed to design *an algorithm creating the motion effects that simultaneously express the translation and rotation of an object in the motion platform's limited workspace and DoFs while seamlessly modulating the motion effect amplitude based on the object size.*

As the object translates, its center position $\mathbf{p}^{cam}$ moves, but it does not respond to a rotation. So motion effects for the center position represent only the object translation. As the object rotates, its frontal direction $\mathbf{d}^{cam}$ changes, but it does not respond to a translation. So motion effects for the frontal direction reflect only the rotation. Hence, both $\mathbf{p}^{cam}$ and $\mathbf{d}^{cam}$ must be included in determining the 3-DoF motion command $\mathbf{m}$ to express the 6-DoF object motion.

Based on these observations, we formulate a simple equation:

$$\mathbf{m} = w_T(l)(\mathbf{p}^{cam} - \mathbf{p}_0^{cam}) + w_R(l)(\mathbf{d}^{cam} - \mathbf{d}_0^{cam}), \tag{3.1}$$

where w_T and w_R are the scale factors transforming the translation and rotation of the object to the motion command, and $\mathbf{p}_0^{cam}$ and $\mathbf{d}_0^{cam}$ are the initial center position and frontal direction of the object. The initial values are for the motion command to begin at the neutral position of the motion chair. Here, the x, y, and z components of the object center position in the camera frame are mapped to the roll, heave, and pitch motion commands, respectively. Likewise, the x, y, and z components of the object frontal direction in the camera frame are mapped to the roll, heave, and pitch commands, respectively.

If we define

$$\mathbf{q}^{cam} = \mathbf{p}^{cam} + (w_R(l)/w_T(l))\mathbf{d}^{cam}, \tag{3.2}$$

$\mathbf{q}^{cam}$ stands for the point that is shifted from the object center by $w_R(l)/w_T(l)$ in the frontal direction (Figure 3.1). $\mathbf{q}^{cam}$ reflects both the object translation $\mathbf{p}^{cam}$ and rotation $\mathbf{d}^{cam}$. We refer to $\mathbf{q}^{cam}$ as the *motion proxy* for the object. Then (3.1) can be simplified to

$$\mathbf{m} = w_T(l)(\mathbf{q}^{cam} - \mathbf{q}_0^{cam}). \tag{3.3}$$

Our synthesis algorithm centers around the motion proxy $\mathbf{q}^{cam}$ while considering the object size l to decide the scale factors w_R and w_T.

In this chapter, we assume that the center position $\mathbf{p}^{cam}$, the frontal direction $\mathbf{d}^{cam}$, and the object size l are given, and then focus on motion effect synthesis using them. This information is readily available in computer-generated scenes, such as animation films and VR games. For regular 4D films, the information can be extracted from the image sequence using recent computer vision techniques, such as motion estimation [71], depth estimation [72], and object detection [73]. We leave an integration of such automatic feature extraction as future work and focus on motion generation.

3.2 Optimal Motion Scale Rules

To compute the motion proxy $\mathbf{q}^{\mathbf{cam}}$ by (3.2) and the motion command $\mathbf{m}$ by (3.3), we need to determine the values of $w_T(l)$ and $w_R(l)$, the two scaling factors from the object translation and rotation to the 3-DoF chair motion. We expected that their best values would depend on the object size l and performed a user experiment. Details are described in this section.

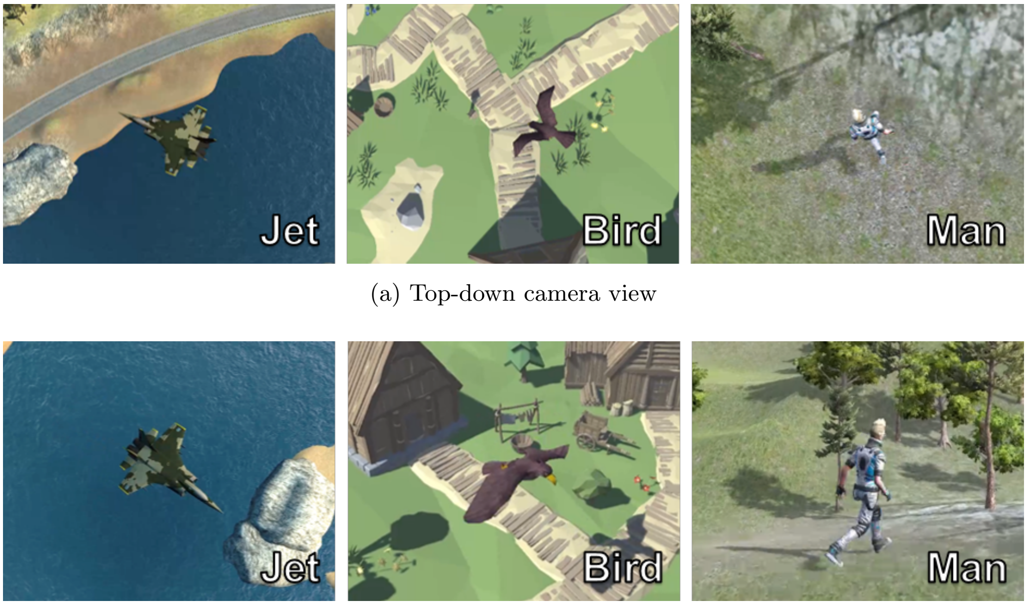
3.2.1 Methods

Participants

Twelve volunteers (seven males and five females; 19-31 years old with an average age of 24.7) with normal sensory abilities participated in the experiment. The experiment took about 120 min. The participants were paid approximately USD 20 after the experiment.

Devices

The motion chair (4DX, CJ 4DPLEX; Figure 1.1) used in the experiment had three DoFs for roll (± 4°), pitch (± 7°), and heave (± 4 cm). The chair is for four people, and the participant sat in the second seat from left during the experiment (and in all the other experiments). 2D images were projected onto a 94-inch screen using an polarized projector (EB-W16SK, Epson Corp.).



(a) Top-down camera view

(b) Other camera views

Figure 3.2: Three video clips used in user experiments.

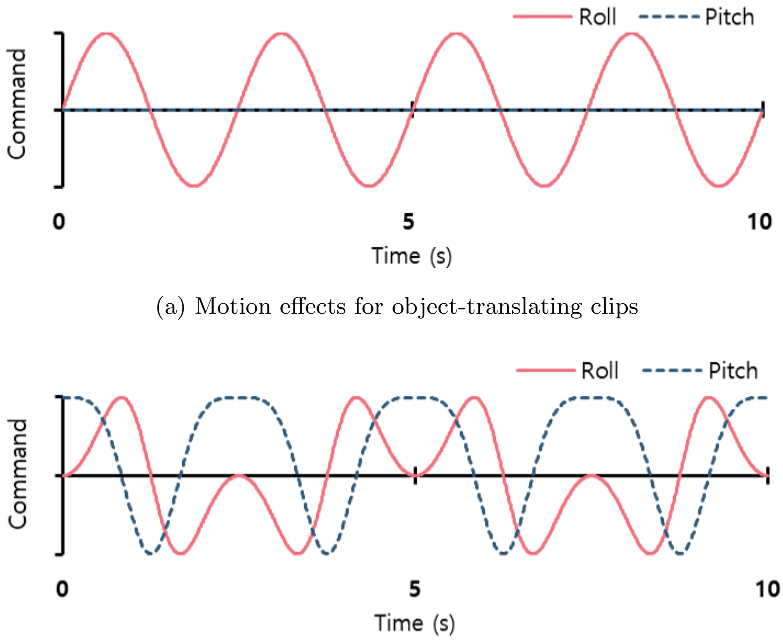
Videos and Motion Effects

We made 30 video clips in which objects of varying sizes (the subtended visual angles 1.91°, 3.62°, 5.52°, 7.41°, and 9.28°) moved (translated or rotated) in each of the three scenes (*Jet*, *Bird*, and *Man*; Figure 3.2a), using Unity3D. All the objects were initially at the center of the video, and all the cameras were stationary in a top-down view. To avoid the object size changing during playback, we used an orthographic projection camera model that did not distort the object size depending on the position and also constrained the object motion in the depth direction. Each video clip was 10-s long.

In object-translating clips, the objects reciprocated from right to left and then to right. The object center position $\mathbf{p}^{cam}$ was given as:

$$\mathbf{p}^{cam}(t) = (\sin(0.8\pi t), 0, 0)^T, \tag{3.4}$$

which has a fundamental period of 2.5 s. The orthographic projection was set to make the objects' displacement on the screen 32.9° in visual angle, which covered



(a) Motion effects for object-translating clips

(b) Motion effects for object-rotating clips

Figure 3.3: Motion effects used in the motion scaling experiment.

most of the screen horizontally.

In object-rotating clips, the objects initially faced the top of the screen and rotated only around the z-axis, following the angle:

$$\theta(t) = \pi \left(\cos(0.4\pi t) - 0.5\right), \tag{3.5}$$

which has a fundamental period of 5 s. The object alternately rotated clockwise and counterclockwise in one period. The frontal direction vector $\mathbf{d}^{cam}$ of the objects is given by:

$$\mathbf{d}^{cam}(t) = (\cos(\theta(t)), \sin(\theta(t)), 0)^T. \tag{3.6}$$

Procedure

The experiment comprised two sessions. In Session 1, we estimated the relationship between the physical amplitudes of motion effects and the perceived magnitudes. The motion effects for the object-translating clips, $\mathbf{m}_T(t)$, were generated by scaling the object center position, and those for the object-rotating clips, $\mathbf{m}_R(t)$, were by scaling the frontal direction (Figure 3.3):

$$\begin{aligned} \mathbf{m}_T(t) &= A\,\mathbf{p}^{cam}(t) = \left(A\sin(0.8\pi t), 0, 0\right)^T, \\ \mathbf{m}_R(t) &= A\,\mathbf{d}^{cam}(t) = \left(A\cos(\theta(t)), A\sin(\theta(t)), 0\right)^T, \end{aligned} \tag{3.7}$$

where A is one of 0.76°, 1.52°, 2.28°, 3.04°, and 3.80°.

Only the motion stimuli were provided to participants without visual stimuli in Session 1. The procedure was designed using the magnitude estimation method with free modulus [74]. Session 1 had two blocks of 20 trials (5 amplitudes and 4 repetitions). Each block was for one of the two motion effect sets for object translation and rotation. In each trial, participants first experienced the reference motion effect of the median amplitude ($A = 2.28°$) and then the comparison motion effect. Afterward, they gave a positive number that best represented the perceived magnitude of the comparison effect with respect to the reference effect. They could freely choose a number (modulus) representing the perceived magnitude of the reference effect and were asked to answer the perceived magnitude of the comparison effect by scaling it to the modulus. The order of the motion amplitudes was randomized per participant. After Session 1, participants were expected to become familiar with the motion effects of various amplitudes.

Session 2 consisted of three blocks, one for each visual scene, for each of object translation and rotation. Each block had 20 trials (5 object sizes and 4 repetitions). In each trial, participants watched an object-translating or -rotating clip of size l played together with the corresponding reference motion effect of the median amplitude. The reference motion effect did not depend on the object's visual size. Participants responded with the perceived magnitude of a motion effect that would best match the visually-perceived object motion while using the perceived magnitude of the reference motion effect as the modulus of rating. This *cross-modal* matching equates the perceived magnitude in one sensory modality to that in another modality [75], and the relationships between two such variables follow the power law [76]. Participants were informed that the reference motion effects were the same as those of Session 1. The order of the blocks was balanced across participants. The order of the object sizes was randomized per participant.

In both sessions, half of the participants first rated for object translation, while the other participants first rated for rotation. We gave participants a 5-min break after half of Session 1, a 10-min break after Session 1, and a 5-min break after half of Session 2. A 1-min break was also given between the blocks of Session 2. Participants were presented with white noise sound through noise-canceling headphones to block auditory cues from the motion chair. The rating data were independently standardized using the mean deviation standardization [77] to reduce the individual deviations.

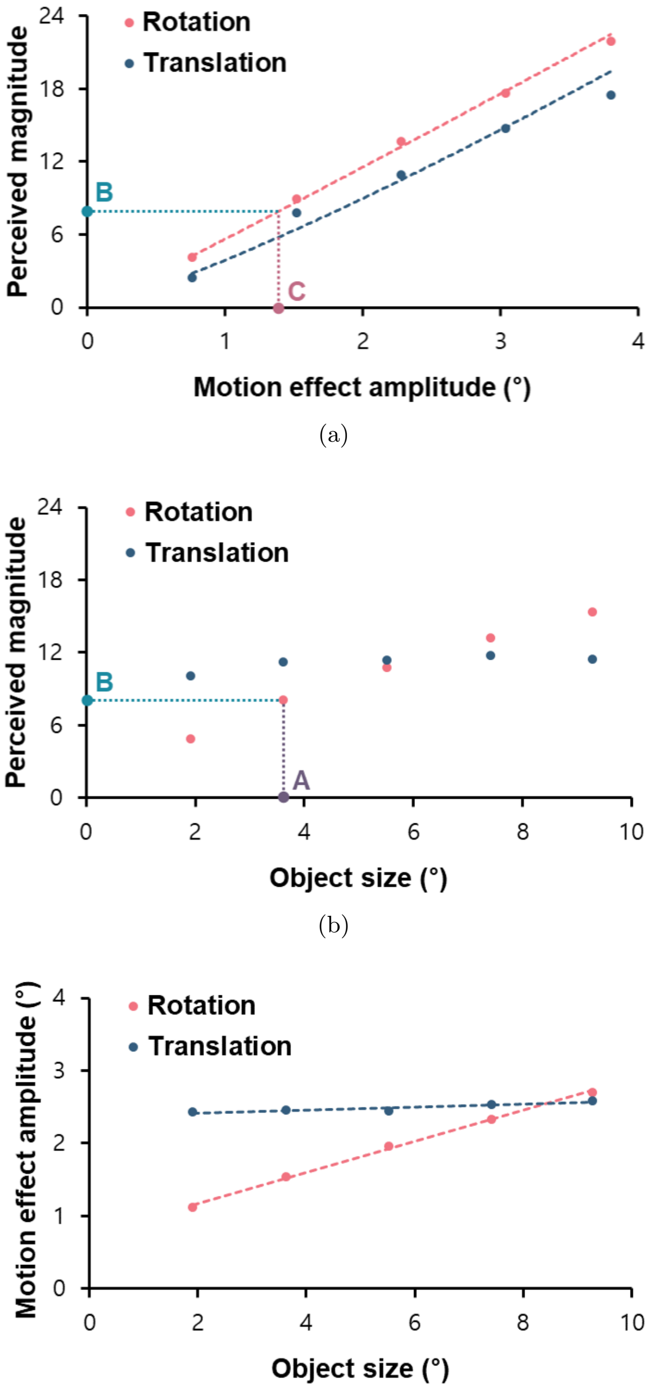
3.2.2 Results and Discussion

From the experimental data collected in Session 1, we obtained psychophysical magnitude functions that relate the physical amplitude to the perceived magnitude of motion effects. The data were fit to Stevens' power law [78], one of the most established empirical laws in cognitive psychology, given by $\psi = k\phi^{\alpha}$. Here ψ is the perceived magnitude, ϕ is the stimulus intensity, α is the power exponent that depends on the sensory modality and the stimulus condition, and k is an arbitrary constant.

Figure 3.4a shows the two psychophysical magnitude functions for motion effects for object translation and rotation. The equations are:

$$\begin{aligned} \psi_T &= 3.8985\, A^{1.2039} \;\; (R^2 = 0.9737), \\ \psi_R &= 5.6332\, A^{1.0372} \;\; (R^2 = 0.9978), \end{aligned} \tag{3.8}$$

where A is the motion amplitude in (3.7). All the coefficients of determination (R^2) were higher than 0.97, indicating a very good fit.



(a)

(b)

(c)

Figure 3.4: User study results for motion scaling. (a) Psychophysical magnitude functions of motion effects. (b) Perceived magnitudes of a motion effect expected for a moving object with different visual sizes. (c) Physical amplitudes of motion effects matching a moving object of different sizes.

Figure 3.4b summarizes the data of Session 2, which shows the average desired perceived magnitudes of motion effect for each visual object size in the translating and rotating clips. Then we combine the information in Figure 3.4a and 3.4b as follows: (1) select an object size (e.g., point A in Figure 3.4b), (2) find the corresponding perceived magnitude of a motion effect that would best match the object size using Figure 3.4b (point B), and (3) compute the physical amplitude of a motion effect resulting in the desired perceived magnitude using the inverse of the corresponding magnitude function in Figure 3.4a (point C).

Applying this procedure to all the data in the plots of Figure 3.4b shows the relationship between the physical amplitude of motion effect and the object size. The converted data indicated good fit by both linear regression and power law. For simplicity, we applied linear regression (see dotted lines in Figure 3.4c), such that

$$\begin{aligned} w_T(l) &= 0.0211\, s + 2.3730 \ \ (R^2 = 0.8578), \\ w_R(l) &= 0.2133\, s + 0.7474 \ \ (R^2 = 0.9973), \end{aligned} \tag{3.9}$$

where w_T and w_R are the scale factors in (3.1)—note that they have the same role with A in (3.7). For translation, the slope in $w_T(l)$ is very small, and it makes $w_T(l)$ nearly constant regardless of the object size l (but the slope was significantly different from 0; $p = 0.02$). For rotation, the slope in $w_R(l)$ is approximately 10 times greater than the slope in $w_T(l)$, with a very high $R^2 (> 0.99)$.

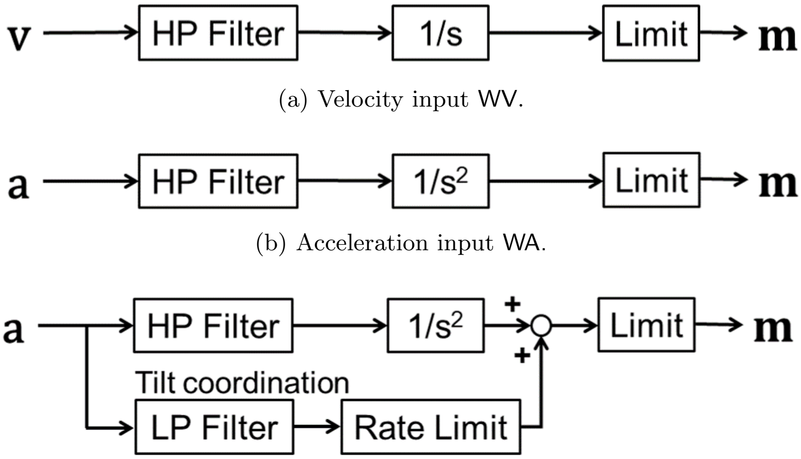
Using (3.1) with the motion scaling rules in (3.9) allows us to determine the position-based motion command that results in the motion effect that would have the perceived magnitude best agreeing to the visually-perceived object size. The specific weight values in (3.9) may depend on the selection of motion and visual stimuli, but their general behaviors are likely to extend to a wider class of motion and visual stimuli for the purpose of object motion effect synthesis. This can be better understood by examining the behavior of the motion proxy $\mathbf{q}^{cam}$

in (3.2) in light of the motion scaling rules found. Since $w_T(l)$ for translation is nearly constant, $\mathbf{q}^{cam}$ is a linear function of $w_R(l)$ for rotation. Since $w_R(l)$ increases as l increases, the motion proxy $\mathbf{q}^{cam}$ is shifted from the center position $\mathbf{p}^{cam}$ in the frontal direction $\mathbf{d}^{cam}$ as the object becomes larger to the viewers' eyes. Therefore, motion effect generation using the motion proxy conforms to the implication in (3.9) that the perceptual role of rotation becomes more important as the object size increases.

3.3 Motion Cueing Algorithms

Actual motion commands are constrained by the limited performance of the motion chair. The simple rule in (3.1) is insufficient, and we need refined MCAs that satisfy the chair's constraints while preserving the key idea in motion proxy. To this end, we adopt the two most-widely used approaches of washout filter and MPC. Because we express the 6-DoF object motion by the translation of motion proxy, we take only the processing pipeline for translation from the two approaches. We designed three algorithms using each approach while also considering visual and vestibular perception. The input to the MCAs, the velocity $\mathbf{v}$ and the acceleration $\mathbf{a}$, is obtained by scaling the motion proxy $\mathbf{q}^{cam}$ using (3.3) and differentiating it once and twice, respectively. This $\mathbf{v}$ and $\mathbf{a}$, expressed in the chair coordinate frame, are transformed to motion commands $\mathbf{m}$ by the MCAs. We then conducted a user study to compare the performance of the six MCAs.

Before applying MCAs, we scale the motion proxy $\mathbf{q}^{cam}$ using (3.3) to transform it to the motion commands $\mathbf{m}$. We differentiate this scale-adjusted $\mathbf{q}^{cam}$ and then smooth the result using local regression [79] to obtain the velocity $\mathbf{v}$ and the acceleration $\mathbf{a}$. This $\mathbf{v}$ and $\mathbf{a}$, expressed in the chair coordinate frame, are used as input to MCAs.



(a) Velocity input WV.

(b) Acceleration input WA.

(c) Acceleration input with tilt coordination WAT.

Figure 3.5: Three washout filters.

3.3.1 Washout Filter

The washout filter is essentially a set of high-pass filters that remove the low-frequency energy pushing the motion chair to its workspace limit. We designed three washout filters given below.

Washout Filter with Velocity Input (WV)

WV (Figure 3.5a) uses the motion velocity **v** as input. This is the same one used in Lee et al.[10], motivated by the fact that the object velocity is detected from visual information (i.e., optical flow). **v** is fed to a first-order Butterworth high-pass filter with a cutoff frequency of 1.0 Hz. The motion command is obtained by integrating the filtered velocities. It is then limited not to exceed the motion range.

Washout Filter with Acceleration Input (WA)

Here the input is the motion acceleration **a**, respecting that the vestibular organs (otoliths) detect only acceleration. The input goes to a second-order Butterworth high-pass filter with a cutoff frequency of 2.5 Hz and then to a double integrator (Figure 3.5b). WA and WV are similar, but different in that WA regards the initial velocity as 0.

Washout Filter with Acceleration Input and Tilt Coordination (WAT)

WAT is the combination of WA and tilt coordination (Figure 3.5c). Tilt coordination is a technique for simulating sustained acceleration, such as gravity and centrifugal force, by tilting a motion chair for a relatively long time. For example, the tilting angle of θ induces the acceleration of $g\theta$, where g is the gravitational constant. This technique is generally implemented as a low-pass filter, and the filtered output is rate-limited to prevent the sensation of rotation. In WAT, the acceleration divided by g is fed to a first-order Butterworth low-pass filter with a cutoff frequency of 0.1 Hz, and the maximum rate was set to 1°/s. This filtered acceleration is added to the output of WA.

3.3.2 Model Predictive Control

MPC is a control methodology that optimizes the current control input based on a process model and a future trajectory while considering constraints. The optimization is governed by an objective function quantifying the difference between the reference output and the output produced by the process model. Constraining the optimization by the maximum displacements of the motion chair, we designed three MPC algorithms as follows.

MPC with Velocity Input (MV)

The input is the motion velocity, similar to WV. The process model is the unity transfer function to express the perceived velocity as it is. Then MV solves the following optimization problem: Find $\mathbf{m}$ such that

$$\begin{aligned} \mathbf{m} = \arg\min_{\mathbf{m}} \quad & \|\mathbf{v} - \mathbf{v_m}\|^2 \\ \text{subject to} \qquad & \mathbf{m}(t) = \int_0^t \mathbf{v_m}(t')dt',\ |\mathbf{m}(t)| \leq \mathbf{m}_{max}, \end{aligned} \tag{3.10}$$

where $\mathbf{v_m}$ is the velocity of the motion command, and $\mathbf{m}_{max}$ is the maximum displacement of the motion chair.

MPC with Perceived Acceleration Input (MA)

MA uses the motion acceleration as input. A vestibular perception model is included to optimize the motion commands based on the perceived acceleration. The otolith transfer function from the physical acceleration a to the perceived acceleration $\hat{a}$ is given by [63]:

$$\frac{\hat{a}}{a} = \frac{K(\tau_n s + 1)}{(\tau_S\, s + 1)(\tau_L\, s + 1)}, \tag{3.11}$$

where K, τ_n, τ_S, and τ_L are the model coefficients of the otoliths (0.4, 13.2, 0.66, and 5.33, respectively), and s is a Laplacian variable. This transfer function has good responsiveness in 0.01–0.5 Hz and effectively works as a band-pass filter. Then MA solves the following optimization problem:

$$\begin{aligned} \mathbf{m} = \arg\min_{\mathbf{m}} \qquad & \|\hat{\mathbf{a}} - \hat{\mathbf{a}}_\mathbf{m}\|^2 \qquad (3.12) \\ \text{subject to} \qquad & \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}a(t),\ \hat{a}(t) = \mathbf{C}\mathbf{x}(t), \\ & \dot{\mathbf{x}}_\mathbf{m}(t) = \mathbf{A}\mathbf{x}_\mathbf{m}(t) + \mathbf{B}a_\mathbf{m}(t),\ \hat{a}_\mathbf{m}(t) = \mathbf{C}\mathbf{x}_\mathbf{m}(t), \\ & \mathbf{m}(t) = \int_0^t \int_0^{t'} \mathbf{a_m}(t'')dt''dt',\ |\mathbf{m}(t)| \leq \mathbf{m}_{max}, \end{aligned}$$

where $\mathbf{a_m}$ is the acceleration of the motion command, a and $a_\mathbf{m}$ represent one of the three components of $\mathbf{a}$ and $\mathbf{a_m}$, respectively, and $\mathbf{x}$ is the state vector. $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ are the state-space model realizing the transfer function (3.11) for each component, given by

$$\mathbf{A} = \begin{pmatrix} -\frac{1}{\tau_L} - \frac{1}{\tau_S} & 1 \\ -\frac{1}{\tau_L \tau_S} & 0 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} \frac{K\tau_n}{\tau_L \tau_S} \\ \frac{K}{\tau_L \tau_S} \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 1 & 0 \end{pmatrix}. \tag{3.13}$$

Note that the lines 2–3 in (3.12) and the state space model in (3.13) are applied to each of the 1D component in $\mathbf{a}$ and $\mathbf{a_m}$.

MPC with Perceived Acceleration Input and Tilt Coordination (MAT)

MAT also considers the acceleration caused by tilting the simulator. Its formulation is very similar to the optimization problem of MA in (3.12). The only difference is that the term $\mathbf{B}a_m(t)$ in the fourth line is replaced by $\mathbf{B}(a_m(t) + m(t)g/R)$, where g is the gravitational constant, and R is the distance between the chair's center of rotation and the viewer's vestibular organ.

3.4 Evaluation on Motion Cueing Algorithms

Washout filters and MPCs have been widely used for flight or vehicle simulators, where the vestibular stimulation accurate to the real cases is essential. For 4D applications, what matters more is enhancing the audiences' experiences, and it was unclear which of the six MCAs would be best suited for object motion effects. Hence, we performed a user study to compare the benefits of the six MCAs.

3.4.1 Methods

Participants

Eighteen participants (11 males and 7 females; 19–29 years old with an average age of 25.2) with normal sensory abilities were recruited for this experiment. The experiment took approximately 50 min, and the participants were paid USD 10 upon completion.

Videos and Motion Effects

We made three 20-s video clips of moving objects, one for each of the three scenes (*Jet*, *Bird*, and *Man* in Figure 3.2a), using Unity3D. The camera orientations were fixed to the top-down view. The 2D horizontal positions of the objects and the cameras were made using Perlin noise with different amplitudes and frequencies. Only for the *Man* clip, we manually controlled the man's position using keyboard input to make straight motions with abrupt direction changes. All the objects were constrained within the camera frame for 20 s.

Each video clip was played with the motion effects generated by each of the six MCAs. The velocities and accelerations of the object in the world coordinate frame are used as input to MCAs, as the washout filters and MPC were originally devised to track the movement in the world frame. An evaluation of our algorithm with motion proxy input will be described in Section 3.7.

Procedure

The experiment comprised three sessions, one for each video clip. In each trial, we presented one of the five motion effect sets with a video clip to participants twice. Participants could experience them more times when they wanted to. After each trial, they answered a questionnaire that included the following four questions:

Q1. Harmony: Did the motion effects match the object motion?

Q2. Synchronicity: Were the motion effects and the object motion synchronized in time?

Q3. Fatigue: Did the motion effects make you tired?

Q4. Preference: Did you like the motion effects

All the questions were answered with a number on a continuous scale of 1–7. In the data analysis, we inverted the scale of Q3.

The order of the video clips was balanced across participants. In each session, the order of the motion effect sets was randomized per participant. Participants took a 5-min rest after each session to prevent motion sickness and fatigue. During the experiment, we presented participants with white noise sound through noise-canceling headphones to block any auditory cues.

3.4.2 Results

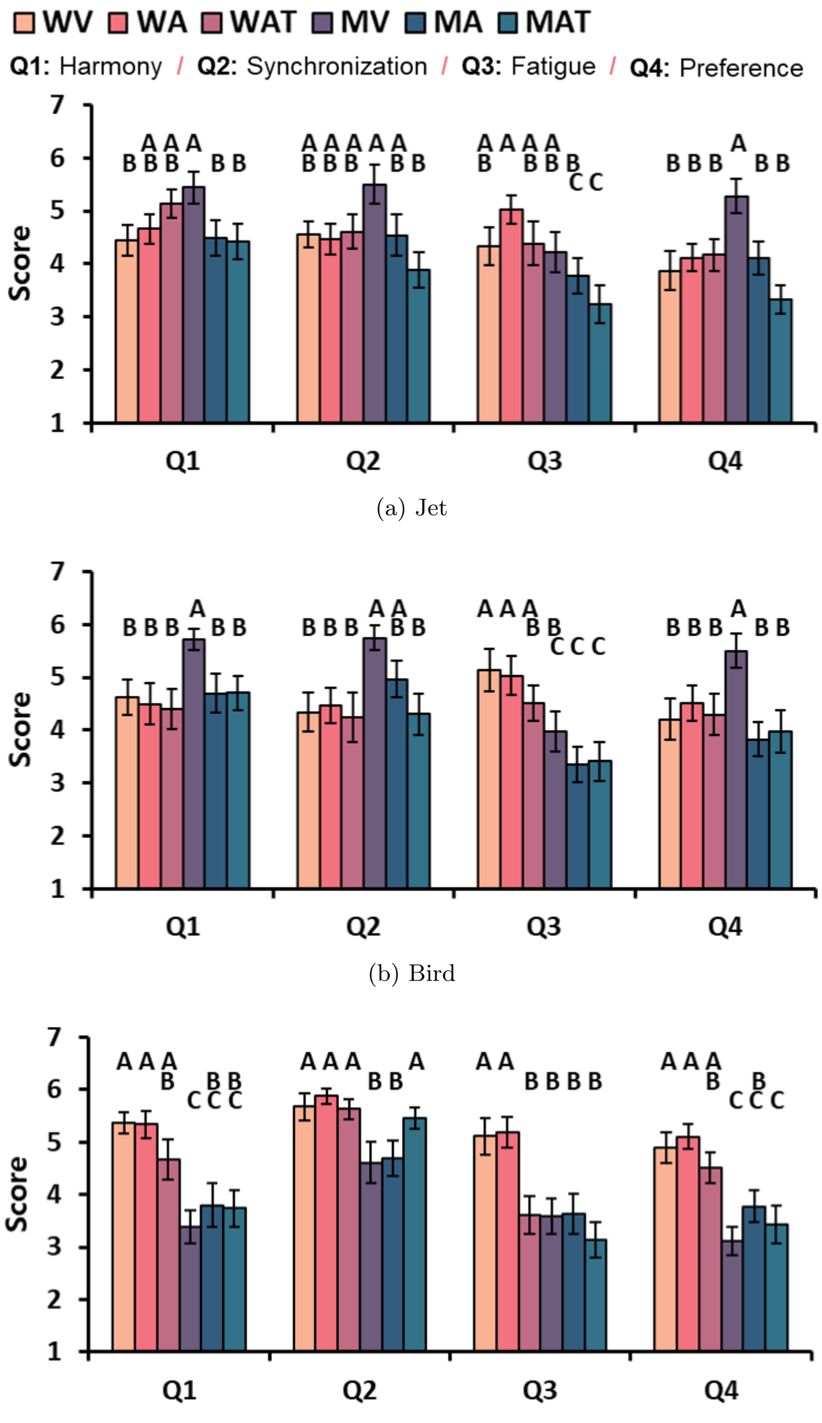
Experimental results are shown in Figure 3.6. The motion effect set was statistically significant for all the subjective metrics and the video clips (one-way repeated-measure ANOVA; $p < 0.001$). The results of the post-hoc SNK tests are also indicated in Figure 3.6.

The results for two video clips, *Jet* and *Bird*, whose motions were generated by Perlin noise, showed similar tendencies. Here MV showed the best scores in all the metrics, except Q3 (Fatigue) for *Bird*. This is presumably because object motion perception depends only on vision, unlike self-motion perception that relies on both visual and vestibular sensations. This is supported by the poor scores of MA and MAT, which use motion acceleration as input. Also, MV may benefit from the fact that it preserves the most spectrum in the object motion including the low-frequency components. For this reason, MV could have had the worst scores in almost all the metrics for *Man*. The object in *Man* moved to one side for about 6 s, but MV cannot filter out such motion unlike other MCAs. Thus, the motion command easily reached the maximum range and did not respond to the object motion further. In contrast, the three washout filters explicitly filter out the sustained motion, and the perception model used in MA and MAT had the role of a band-pass filter.

The three washout filters had scores comparable to or higher than MA and MAT in all the metrics except Q3 for all the video clips. They did not show significant differences between them.

The inclusion of tilt-coordination did not lead to significant differences between the two washout filter methods (WA vs. WAT) or the two MPC methods (MA vs. MAT). It seems that the tilt coordination technique was not effective because of the small workspace of the motion chair used for 4D films.

In summary, MV should be the best candidate for automatic synthesis of object motion effects for general video clips. Based on these results, we adopt MV to synthesize motion effects. However, it sometimes does not respond well to sustained motions. In this case, other methods, such as WV and WA, should be considered.



(a) Jet

(b) Bird

(c) Man

Figure 3.6: User study results for six motion cueing algorithms. Error bars represent standard errors. Motion effects sets marked with the same letter indicate that they did not show statistically significant differences by the SNK tests.

3.5 Estimating Visually Perceived Velocity

When watching 4D films, the audience perceives an object motion through vision. As the visually-perceived motion does not perfectly match the actual object motion, we note a possibility that creating motion effects based on the actual motion leads to discrepancies between the visually-perceived motion and the motion effects. A remedy can be computing the motion vector that viewers visually perceive for motion proxy based on visual perception. Particularly, we formulate the motion vector by referring to Duncker's two modes of object motion perception [80] and motion field equations [81].

Object motion can be perceived in two ways: subject-relative and object-relative perception [80]. In subject-relative perception, the object motion is perceived in the viewer's local coordinate frame, whereas the relative displacement between two objects is perceived in object-relative perception. Humans adopt the perception mode depending on numerous factors, including visual characteristics and their attention. Since the quantitative knowledge about the factor effects significantly lacks [82], we designed our algorithm to support both modes so that designers can select a mode that generates more plausible effects.

3.5.1 Subject-Relative Perception Mode

People can perceive the horizontal and vertical velocity of an object as optic flow and the depth velocity by integrating monocular and binocular cues [83]. So we compute the vertical and horizontal components using motion field equations for the optical flow and the depth component by scaling the actual depth velocity.

Assume that the camera with a focal length f moves with the translational velocity $\mathbf{t}^{cam} = (t_x, t_y, t_z)^T$ and the rotational velocity $\omega^{cam} = (\omega_x, \omega_y, \omega_z)^T$ (Figure 3.7a). The motion proxy $\mathbf{q}^{cam} = (q_x^{cam}, q_y^{cam}, q_z^{cam})^T$ projects onto an image point $\mathbf{q}^{img} = (q_x^{img}, q_y^{img}, f)^T$, and its 3D relative velocity to the camera

$\mathbf{v}^{cam} = (v_x^{cam}, v_y^{cam}, v_z^{cam})^T$ is observed in the image as the horizontal and vertical velocity $\mathbf{v}^{img} = (v_x^{img}, v_y^{img})^T$. Then the following relationships (3.14)-(3.16) hold [81]:

$$\mathbf{v}^{cam} = -\mathbf{t}^{cam} - \omega^{cam} \times \mathbf{q}^{cam},\ \mathbf{q}^{img} = \frac{f}{q_z^{cam}}\mathbf{q}^{cam},\ \text{ and } \mathbf{v}^{img} = \frac{d}{dt}\mathbf{q}^{img}, \tag{3.14}$$

which gives:

$$\mathbf{v}^{img} = \frac{1}{f}\mathbf{G}\omega^{cam} + \frac{1}{q_z^{cam}}\mathbf{H}\mathbf{t}^{cam}, \tag{3.15}$$

where

$$\begin{aligned} \mathbf{G} &= \begin{pmatrix} q_x^{img} q_y^{img} & -(q_x^{img})^2 - f^2 & f q_y^{img} \\ (q_y^{img})^2 + f^2 & -q_x^{img} q_y^{img} & -f q_x^{img} \end{pmatrix} \text{ and} \\ \mathbf{H} &= \begin{pmatrix} -f & 0 & q_x^{img} \\ 0 & -f & q_y^{img} \end{pmatrix}. \end{aligned} \tag{3.16}$$

Denoting the subject-relative velocity vector of the object by $\mathbf{v}^{sub} = (v_x^{sub}$, v_y^{sub}, $v_z^{sub})^T$, we can obtain the depth component v_z by scaling down v_z^{cam} based on the distances from the camera to the image plane and to the object (Figure 3.7b) by

$$v_z^{sub} = f\frac{v_z^{cam}}{q_z^{cam}} \tag{3.17}$$

Since $v_x^{sub} = v_x^{img}$ and $v_y^{sub} = v_y^{img}$,

$$\mathbf{v}^{sub} = \frac{1}{f}\mathbf{G}'\omega^{cam} + \frac{1}{q_z^{cam}}\mathbf{H}'\mathbf{t}^{cam}, \tag{3.18}$$

where

$$\mathbf{G}' = \begin{pmatrix} & \mathbf{G} & \\ -q_y^{img} & q_x^{img} & 0 \end{pmatrix} \text{ and } \mathbf{H}' = \begin{pmatrix} & \mathbf{H} & \\ 0 & 0 & -f \end{pmatrix}. \tag{3.19}$$

3.5.2 Object-Relative Perception Mode

In this mode, we calculate the motion vector $\mathbf{v}^{obj}$ representing the relative velocity between the object and its surroundings. Here, we refer to a statement

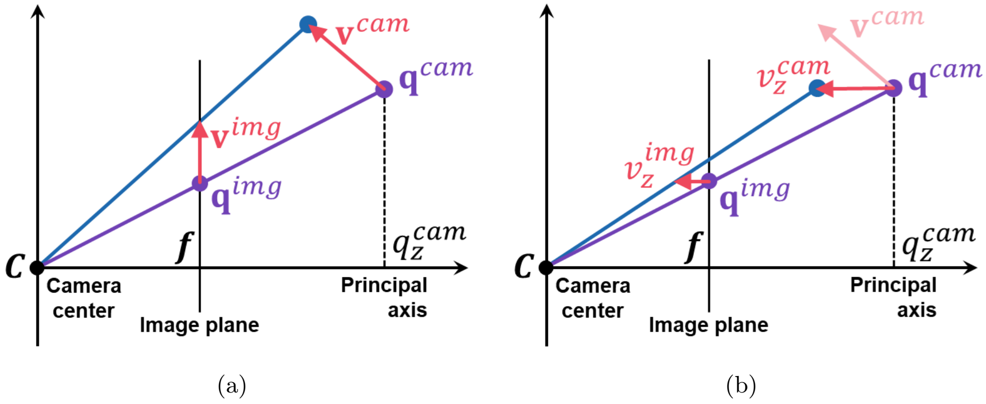


Figure 3.7: Motion vectors to determine the subject-relative velocity. (a) Horizontal and vertical components. (b) Depth component.

by Duncker [80]: when the relative displacement between an object and its background is given, the motion will be assigned to the object and immobility to the surroundings. Hence, we assign the object-relative velocity to the difference between the subject-relative velocity of the background, $\mathbf{v}_{bg}^{sub}$, and the subject-relative velocity of the object, $\mathbf{v}^{sub}$, such that

$$\mathbf{v}^{obj} = \mathbf{v}^{sub} - \mathbf{v}_{bg}^{sub} = \mathbf{H}' \left(\frac{1}{q_z^{cam}} \mathbf{t}^{cam} - \frac{1}{Z_{bg}} \mathbf{t}_{bg}^{cam} \right), \tag{3.20}$$

where Z_{bg} is the depth of the background point projecting to $\mathbf{q}^{img}$, and $\mathbf{t}_{bg}^{cam}$ is the camera velocity relative to the background point.

This equation (3.20) has some implications. First, the motion vector of this mode is not affected by camera rotation. Second, due to the camera motion, the object motion can be canceled out or even captured as moving in the opposite direction.

3.6 Synthesis of Object Motion Effects

We integrate the key ideas and derivations, as well as the major results of the formative user studies, described so far into a camera space synthesis algorithm of

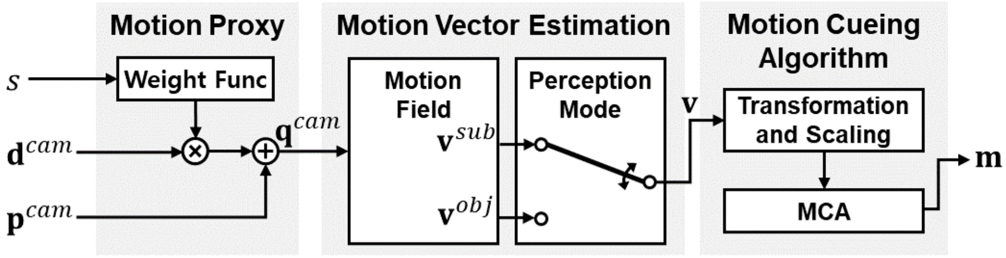


Figure 3.8: Flow of our synthesis algorithm for object motion effects.

object motion effects, as depicted in Figure 3.8. Given a moving object of interest, the input of our algorithm consists of its center position, frontal direction, and size. The output is a motion command to a 3-DoF motion chair. In the first step, our algorithm computes the motion proxy $\mathbf{q}^{cam}$ from the center position $\mathbf{p}^{cam}$, frontal direction $\mathbf{d}^{cam}$, and size l by (3.2). The two scale factors for translation and rotation, $w_T(l)$ and $w_R(l)$, are determined by (3.9) based on the experimental results of Section 3.2. In the second step, our algorithm estimates the visual velocity $\mathbf{v}$ of the motion proxy $\mathbf{q}^{cam}$ in the subject-relative mode using (3.18) or in the object-relative mode using (3.20), rather than the actual velocity. 4D designers can select a mode that leads to more plausible motion effects. Finally, our algorithm transforms the visual velocity $\mathbf{v}$ to the motion command $\mathbf{m}$ by scaling and feeding $\mathbf{v}$ (or its acceleration $\mathbf{a}$) to an MCA.

3.7 Performance Evaluation

Finally, we conducted a summative user study to compare the user experiences elicited by the object motion effects synthesized by our algorithms including the two visual perception modes and the conventional algorithm [9, 10].

3.7.1 Methods

Participants

Twenty-four people (13 males and 11 females; 20–30 years old with an average age of 22.7) with normal sensory abilities participated in this experiment. The experiment took about 75 min, and the participants were paid USD 13 upon completion.

Videos and Motion Effects

Using Unity3D, we made three 20-s video clips each showing the three moving objects of *Jet*, *Bird*, and *Man* in Figure 3.2b. The 6-DoF camera motions and the object positions were generated by Perlin noises with different amplitudes and frequencies. In *Man*, we added natural curves in the terrain to make horizontal object motions, and the man always stood upright moving on the terrain. All the objects were translated in 3D and constrained within the camera frame for 20 s. The parameters of Perlin noise used in this experiment were different from those used in Section 3.4. To diversify the scenes, we varied the initial camera pose for each scene (*Jet*: top-down view, *Bird*: 45° high angle view, and *Man*: side view). We also added a random stopping condition to the object and camera motions for participants to clearly evaluate the synchronicity between visual motion and motion effect.

Five sets of motion effects were generated for this experiment.

- Object-relative mode (OR): Motion effects synthesized by our algorithm (perception mode: object-relative, MCA: MV).
- Subject-relative mode (SR): Motion effects synthesized by our algorithm (perception mode: subject-relative, MCA: MV).
- Object and camera classes (OC): Addition of the two motion effects for

object and camera classes generated by [9, 10].

- Object class (O): Object class motion effects synthesized by [10].
- Random (R): Randomly generated motion effects by Perlin noise.

Procedure

For familiarization, we first conducted a training session in which participants experienced all the motion effect sets once with a video clip of each main session. The main experiment comprised three sessions, one for each video clip. In each trial, one of the five motion effect sets was presented twice with a video clip. Participants could experience them more times when they wanted to. To eliminate the influence of retinal eccentricity (motion sensitivity is better in the peripheral retina than in the fovea [84]), we instructed participants to keep their eyes on the object while watching the video clips.

At the end of each trial, participants answered a questionnaire that included the following eight questions:

Q1. Harmony, hit: Did the motion effects follow the expected motion well?

Q2. Harmony, false alarm: Were the motion effects played even though you did not expect any motion?

Q3. Synchronicity: Were the motion effects and the object motion well synchronized in time?

Q4. Comfort: Did the motion effects feel comfortable?

Q5. Distraction: Were the motion effects distracting from focusing on the object motion?

Q6. Enjoyment: Did you enjoy watching the video with the motion effects?

Q7. Preference: Did you like the motion effects?

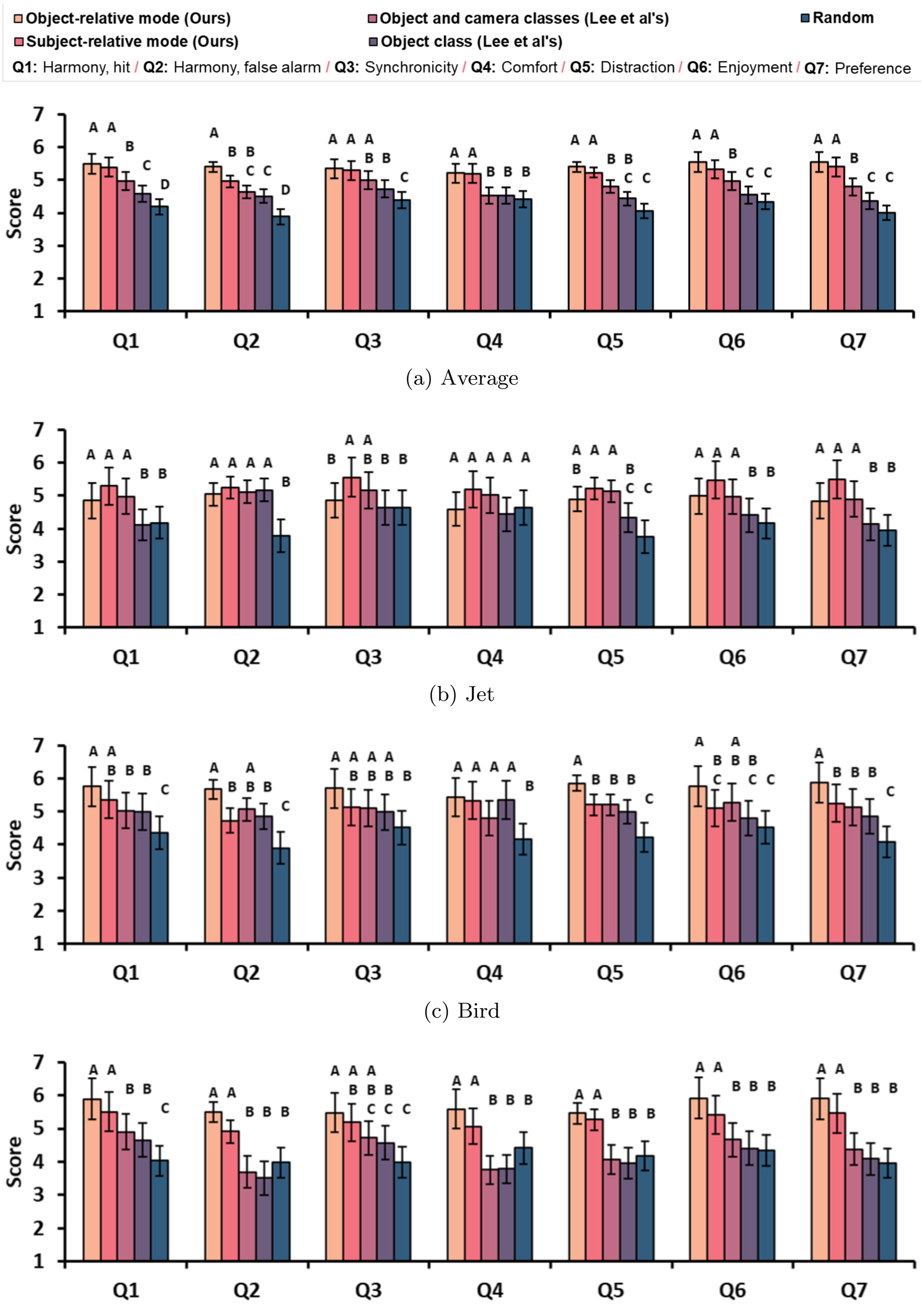
Q8. Free Comment: Please leave a comment regarding the motion effects.

Participants rated Q1–Q7 on a continuous scale of 1–7 by selecting a position on a horizontal line. The two ends of the horizontal lines were labeled with symmetric positive and negative answers, for example, "Strongly disagree" at the left end and "Strongly agree" at the right end. Participants freely responded to Q8 using a keyboard. In the data analysis, we inverted the scales of Q2 and Q5.

The order of the video clips was balanced across participants, and the order of the motion effect sets was randomized per participant. Participants took a 30-s rest after each trial and a 3-min rest after each session to prevent motion sickness and fatigue. During the experiment, we presented participants with white noise sound through noise-canceling headphones to block any auditory cues.

3.7.2 Results

The experimental results are shown in Figure 3.9. We performed a two-way ANOVA on each question using the motion effects set and the video clip as the independent variables. The motion effects set was statistically significant ($p < 0.0001$) for all the subjective metrics and all the video clips. The video clip was also significant for all the metrics except for Q3 (Synchronicity) and Q6 (Enjoyment). We observed significant interactions between the motion effects set and video clip in all the metrics. We then used the SNK test for post-hoc multiple comparisons (also shown in Figure 3.9).



(a) Average

(b) Jet

(c) Bird

(d) Man

Figure 3.9: User experiment results for five synthesis algorithms. Error bars represent standard errors. Motion effects sets marked with the same letter indicate that they did not show statistically significant differences by the SNK tests.

On average (Figure 3.9a), the OR (object-relative mode) motion effects showed the best scores in all the metrics, followed by the SR (subject-relative mode) motion effects. These two new object motion effect synthesis methods led to significantly higher scores in all the metrics except for Q2 (Harmony, false alarm) and Q3 (Synchronicity) than the two previous methods of O (object class) and OC (object and camera class) and the random effects R.

The effect of visual perception mode depended on the video clip, suggesting that the participants could have relied on different visual perception modes for the three video clips. OR was rated clearly higher than SR for most of the metrics in *Man* (Figure 3.9d). Also, OR was more favored than SR in *Bird* (Figure 3.9c). We speculate that this is because object motion is predominantly perceived through object-relative perception [85]. In contrast, SR was slightly more favored in *Jet* (Figure 3.9b). *Jet*'s background, a vast and textureless lake, may have provided weak cues for estimating the object motion, forcing the participants to depend more on subject-relative perception.

Also by analyzing the comments from the participants, we summarize the main reasons for the improved performance of OR and SR. First, OR and SR consider both the translation and rotation of object motion, while O and OC from [9, 10] consider only the translation. Seven participants commented on this, e.g., "It was good because dynamic and fine motions (flipping, direction changes, etc.) were well expressed." (*Jet*, OR) and "It was unsatisfactory because it didn't express the fine motions such as flipping." (*Jet*, OC). Second, OR and SR map the object depth motion to the heave motion of a motion platform. In O and OC, only horizontal and vertical motions are considered. Although the OC effects include some depth motions, they represent the camera motion rather than the object motion. Four participants reported on this, e.g., "The depth motion was felt clearly." (*Jet*, SR).

In conclusion, automatic synthesis of object motion effects in *object-relative mode* and *subject-relative mode* showed better perceptual performance than the others. 4D effect designers are likely to create more convincing effects by leveraging the two methods.

3.8 Conclusion

In this chapter, we have presented an automatic synthesis algorithm that generates object motion effects by abstracting the 6 DoF motion of a rigid body and utilizing human vestibular and visual perception models. This algorithm is designed to take three input variables: the center position, frontal direction, and size of an object. At each step, we performed a user experiment to find the optimal design parameters for implementation, and the results of those steps are integrated into a single algorithm. The final summative user experiment indicated that our algorithm could produce perceptually better motion effects than the current state of the art.

The contributions of this work can be summarized as follows: (1) We propose a novel concept, motion proxy, which enables a simultaneous expression of the translation and rotation of a 6-DoF moving object with a 3-DoF motion chair. Also, the computation rule of motion proxy is perceptually optimized by a perceptual experiment; (2) The algorithm produces compelling 4D experiences by considering human perception while fully utilizing a motion platform's workspace and DoF; and (3) We formulate the visually perceived velocity for an object (or the motion proxy) to create motion effects synchronized with what humans visually perceive.

This algorithm has the potential for extension to 2-DoF and 6-DoF motion chairs. For a 2-DoF chair, we can simply use the 2D components of the motion proxy. For a 6-DoF chair, we can directly map the 6-DoF object motion to the

6-DoF motion command. Here the scale of the rotation commands may depend on the object size.

Current limitations include: (1) Our algorithm does not automatically extract the input parameters from image sequences. Recent computer vision technologies [71, 72, 73] can be used for that purpose; and (2) Our algorithm cannot produce motion effects representing the motion of multiple objects or the local motion of part of an object. Future work will concentrate on addressing these limitations.

IV. Generating Motion Effects for Multiple Articulated Bodies

The algorithm in Chapter III could effectively generate the motion effects of a single rigid body moving in the 3D camera space. However, the algorithm did not capture the local movement of object parts as well as the movements of multiple objects. This chapter presents an automatic algorithm for synthesizing the object-based motion effects that express the movements of multiple *articulated bodies*[1] inclusively. Articulated bodies require a very high degree of freedom (DoF) for motion description. To express complex movements of multiple articulated bodies with a motion effect, the concept of motion proxy is extended. The motion proxy is determined by linearly combining the velocities based on the size and speed of object parts. Among the several methods of calculating a motion proxy, the best combination was investigated by user studies.

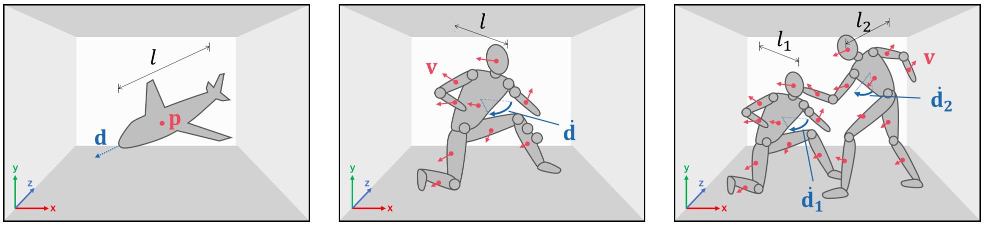
4.1 Problem Formulation

In this section, we formulate a problem for generating motion effects for moving articulated bodies. We first briefly summarize the algorithm pipeline described in Chapter III. Then, based on it, we extend the motion proxy concept to a single articulated body and multiple articulated bodies.

In Chapter III, to compress the 6-DoF object motion to a single movement, we defined the motion proxy for the object as

$$\mathbf{q} = \mathbf{p} + cl\mathbf{d}, \tag{4.1}$$

[1] Here, "articulated" means made of sections connected by joints. The human body is a good example of an articulated body; see Figure 4.1.



(a) Rigid body (b) Single articulated body (c) Multiple articulated bodies

Figure 4.1: Examples of a moving object (or objects) in the 3D camera space. A red point or arrow indicates the position $\mathbf{p}$ or velocity $\mathbf{v}$ of an object (or a part). A blue arrow represents the direction vector $\mathbf{d}$ or its derivative $\dot{\mathbf{d}}$. l is the frontal length of an object measured on the screen.

where $\mathbf{q} = (q_x, q_y, q_z)$ is the motion proxy, $\mathbf{p} = (p_x, p_y, p_z)$ and $\mathbf{d} = (d_x, d_y, d_z)$ are the position and direction vectors of a rigid body, respectively, in the 3D camera space, l is the object's frontal length projected onto the screen, and c is a constant (Figure 4.1a). The latter term, $cl\mathbf{d}$, is to reflect more rotation as the object size increases. (4.1) can also be expressed by its derivatives in the viewer's perspective as

$$\dot{\mathbf{q}} = \mathbf{v} + cl\dot{\mathbf{d}}, \tag{4.2}$$

where $\dot{\mathbf{q}}$ is the velocity of the motion proxy, and $\mathbf{v}$ and $\dot{\mathbf{d}}$ are the linear velocity and the derivative of the direction vector of the object in the viewer's perspective. To reproduce the movement of the motion proxy within the limited workspace of a motion platform, such as the maximum displacement and velocity, the velocity (or its acceleration) of the motion proxy is fed to a motion cueing algorithm (MCA) to generate a motion command as

$$\mathbf{m} = \mathrm{MCA}(\dot{\mathbf{q}}), \tag{4.3}$$

where $\mathbf{m} = (roll, heave, pitch)$ is the 3-DoF motion command to a motion chair.

Unlike a rigid body, an articulated body consists of many joints and links, leading to a motion of a very high DoF. While watching many 4D films, we discovered two key patterns in manually-crafted motion effects for articulated

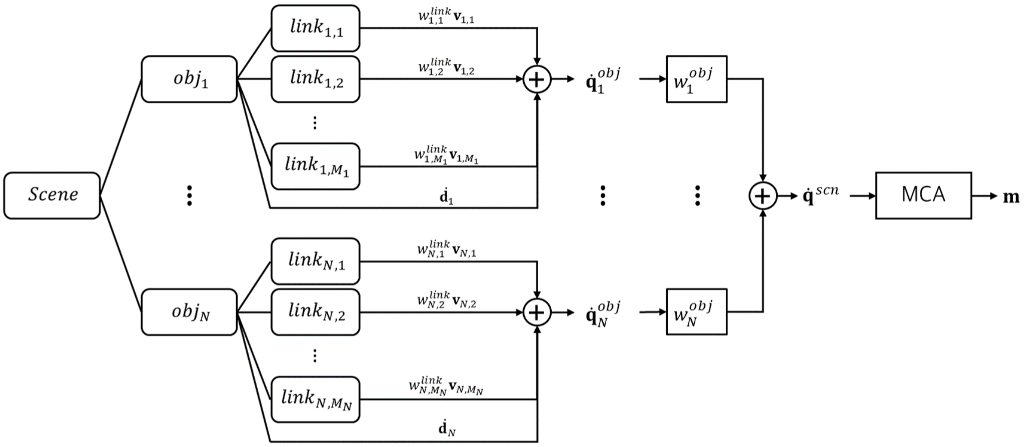


Figure 4.2: Overall flow of our motion effect generation algorithm for multiple articulated bodies. The velocity of an object motion proxy, $\dot{\mathbf{q}}^{obj}$, representing the object's movement, is calculated from the velocities and weights of the object's links. Then, the velocity of a scene motion proxy, $\dot{\mathbf{q}}^{scn}$, representing the movements of all the objects in the scene, is calculated from the weights and $\dot{\mathbf{q}}_{obj}$ of the objects. The velocity of the scene motion proxy, $\dot{\mathbf{q}}^{scn}$, is converted to a motion command by a motion cueing algorithm (MCA).

bodies. First, the local movements of an object or its links, both translation and rotation, are expressed in a motion effect. Second, when multiple objects appear simultaneously in the scene, their movements are combined somehow into a motion effect. Therefore, we designed *an algorithm for generating the motion effects that simultaneously express the translations and rotations of multiple articulated bodies while respecting the motion platform's limited DoFs and performance.*

To this end, we extend the original formulation of the motion proxy as follows. For an articulated object i in the scene (Figure 4.1b), we determine its motion proxy by

$$\dot{\mathbf{q}}_i^{obj} = \sum_{j=1}^{M_i} w_{i,j}^{link} \mathbf{v}_{i,j} + cl_i \dot{\mathbf{d}}_i, \tag{4.4}$$

where $\dot{\mathbf{q}}^{obj}$ is the velocity of the motion proxy, M_i is the number of the links, $w_{i,j}^{link}$ is the normalized weight of link j, $\mathbf{v}_{i,j}$ is the visual velocity of link j, c

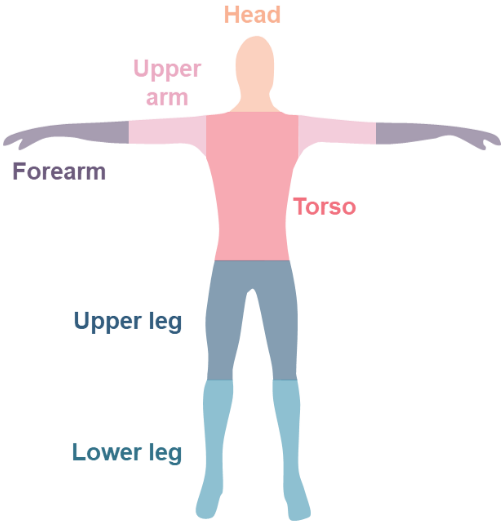


Figure 4.3: Configuration of the human model used in this study. The human model consists of a head, a torso, two upper arms, two forearms, two upper legs, and two lower legs.

is a constant, l_i is the line length screen-projected from the 3D line connecting the object's center and outermost point in the direction $\mathbf{d}_i$, and $\dot{\mathbf{d}}_i$ is the derivative of the direction vector. Here, the term for the linear link velocities is a weighted generalization of the linear velocity term in (4.2). Similarly, one can use a weighted sum of the direction derivatives of all links for the term related to rotational velocity. Instead, we include only $\dot{\mathbf{d}}_i$, the direction vector of the root link. In our tests, this simpler form leads to more plausible motion effects for object rotation.

Figure 4.2 illustrates the process of computing the object motion proxy. As the link weight w^{link} increases, the object motion proxy $\mathbf{q}^{obj}$ reflects the motion of the corresponding link more. In this study, we use a human model as an articulated body and divide it into ten parts, as shown in Figure 4.3. When the weights excluding the torso are all zero, the object motion proxy $\mathbf{q}^{obj}$ represents only the motion of the torso.

Next, we consider the case where multiple articulated objects simultaneously appear in the scene (Figure 4.1c). We define a scene motion proxy $\dot{\mathbf{q}}^{scn}$ as the weighted sum of the object motion proxies $\dot{\mathbf{q}}^{obj}$ of all objects:

$$\dot{\mathbf{q}}^{scn} = \sum_{i=1}^{N} w_i^{obj} \dot{\mathbf{q}}_i^{obj}, \tag{4.5}$$

where $\dot{\mathbf{q}}^{scn}$ is the velocity of the scene motion proxy representing the movements of all the objects in the scene, N is the number of the objects, and w^{obj} represents the normalized weight for each object; also see Figure 4.2. For example, if there are two objects with object weights of 1 and 0, only one object's motion is reflected in the scene motion proxy. When the object weights are both 0.5, the motions of the two objects are combined equally.

To determine the link and object weights, w^{link} and w^{obj}, we use *size* and *speed* as the main features. This allows us to reflect in the motion proxy the motion of a link or an object more if it is larger or faster. These two features belong to the key variables for estimating salient regions in image sequences [86, 87]. To compute the size of a link/object, two options can be considered: the visual area projected onto the screen and the volume in the 3D camera space. Our tests did not find significant differences in the resulting motion effects. We use the visual area as it has an additional merit of emphasizing the movement of a closer link/object from the camera. Similarly, we choose the magnitude of the visual velocity on the screen as the speed feature of a link/object. Based on these two features, we designed and tested various methods to set the link and object weights. A few best candidates are presented in Section 4.2 and 4.4, respectively.

In this chapter, we directly extract the values of all variables in (4.4) and (4.5) every frame using Unity3D. The visual area of each link is calculated as the area of the region on the screen to which its mesh is projected. The visual velocity of each link, $\mathbf{v}_{i,j} = (v_{i,j,x}^{vis}, v_{i,j,y}^{vis}, v_{i,j,z}^{vis})$, is computed as in Chapter III. The vertical and horizontal components of the visual velocity are determined from the derivative of its position projected onto the screen as

$$v_{i,j,x}^{vis} = \frac{\partial}{\partial t}\left(f\frac{p_{i,j,x}}{p_{i,j,z}}\right) \quad \text{and} \quad v_{i,j,y}^{vis} = \frac{\partial}{\partial t}\left(f\frac{p_{i,j,y}}{p_{i,j,z}}\right) \tag{4.6}$$

where t is time, $\mathbf{p}_{i,j}(t) = (p_{i,j,x}(t), p_{i,j,y}(t), p_{i,j,z}(t))$ is the position vector of each link in the camera space, and f is the focal length of the camera ($f = 781.2$ in this study). The depth component is computed by scaling the depth velocity as

$$v_{i,j,z}^{vis} = f\frac{v_{i,j,z}}{p_{i,j,z}} \tag{4.7}$$

where $v_{i,j,z} = \partial p_{i,j,z}/\partial t$. We used the magnitude of this velocity, $\|\mathbf{v}_{i,j}\|$, as the feature for speed.

4.2 Single Articulated Body

In this section, we design several methods that set the link weights, w^{link}, for a single articulated body to obtain the object motion proxy's velocity, $\dot{\mathbf{q}}^{obj}$, by (4.4). As described earlier, each link weight is determined based on the two features of area and speed. We first define requirements for a valid link weighting method and then present a few weighting policies satisfying the requirements.

4.2.1 Requirements on Link Weights

Valid link weighting methods need to meet the following two conditions: First, faster or larger links are weighted higher. Second, visually identical links are equally weighted. The first condition can be easily satisfied if link weights monotonically increase with the size or speed.

The second condition requires more care. Suppose a situation where an object includes two links that have the same visual area s and are moving at the same velocity $\mathbf{v}$. Then, we would expect that the two links contribute equally to motion effects, and their weights must be the same. This observation can be formalized as follows. Let $link(s, \mathbf{v})$ be a link with the visual area s and the velocity

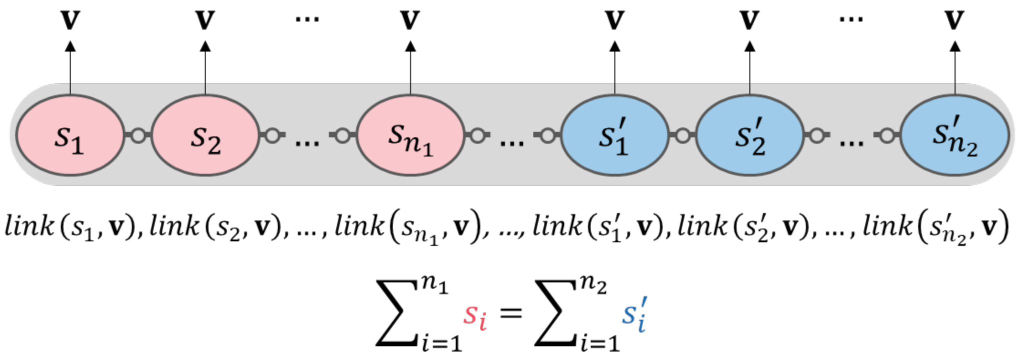

v v ··· v v v ··· v
s_1 s_2 ... s_{n_1} ... s'_1 s'_2 ... s'_{n_2}
$link(s_1,\mathbf{v}), link(s_2,\mathbf{v}), \ldots, link(s_{n_1},\mathbf{v}), \ldots, link(s'_1,\mathbf{v}), link(s'_2,\mathbf{v}), \ldots, link(s'_{n_2},\mathbf{v})$
$\sum_{i=1}^{n_1} s_i = \sum_{i=1}^{n_2} s'_i$

Figure 4.4: A case in which two groups of consecutive links should be equally weighted.

$\mathbf{v}$. Assume that an object has two groups of consecutive links, $\{link(s_1, \mathbf{v}_1), \cdots, link(s_{n_1}, \mathbf{v}_{n_1})\}$ and $\{link(s'_1, \mathbf{v}'_1), \cdots, link(s'_{n_2}, \mathbf{v}'_{n_2})\}$, as depicted in Figure 4.4. When the areas of the two groups are equivalent, i.e., $\sum_{i=1}^{n_1} s_i = \sum_{i=1}^{n_2} s'_i$ and all the links have the same velocity $\mathbf{v}$, the two link groups have identical visual effects. This gives a constraint that the two groups must have the same sums of the link weights, such that

$$\sum_{i=1}^{n_1} w^{link}\left(link(s_i, \mathbf{v})\right) = \sum_{i=1}^{n_2} w^{link}\left(link(s'_i, \mathbf{v})\right). \tag{4.8}$$

Many seemingly reasonable weighting methods violate this condition. For example, one can set a link weight to be directly proportional to the link velocity, i.e., $w^{link}(link(s_i, \mathbf{v})) = \mathbf{v}$, but it violates the second condition in (4.8) when $n_1 \neq n_2$; the sum of weights is $n_1\mathbf{v}$ for one link group and $n_2\mathbf{v}$ for the other group.

4.2.2 Weighting Policies

We present three methods to compute the link weights while satisfying the two requirements described in Section 4.2.1.

Size

This policy gives each link a weight proportional to its size (visual area) as

$$w_{i,j}^{link} = \frac{s_{i,j}}{\sum_{k=1}^{M_i} s_{i,k}}, \tag{4.9}$$

where $s_{i,j}$ is the visual area of the link j of object i. This method allows the motion proxy to reflect the movement of a larger link more. For example, more intense motion effects result from kicks than punches because legs are larger than arms.

Momentum

In this policy, the link weight is proportional to both its area and speed, i.e., its momentum, as

$$w_{i,j}^{link} = \frac{s_{i,j}\|\mathbf{v}_{i,j}\|}{\sum_{k=1}^{M_i} s_{i,k}\|\mathbf{v}_{i,k}\|}. \tag{4.10}$$

It emphasizes the movement of a faster or larger link, so producing a more dynamic motion effect for fast link movements.

Single

Occasionally, expressing the movement of only one link may have a merit in delivering a more focused motion effect. This is achieved by setting the weight of the selected link to 1, while all the other links have zero weights.

Comparisons

Figure 4.5 shows an example illustrating the effects of the three weighting policies. The animation shows that a character lowers its arms, and its left arm (right to readers) moves in a longer distance than its right arm. The torso remains to be almost stationary. For this animation, *Size* creates a weak motion effect because the arm sizes (so their visual areas) are relatively small. *Momentum* makes the greatest motion effect by responding to the quick movements of the

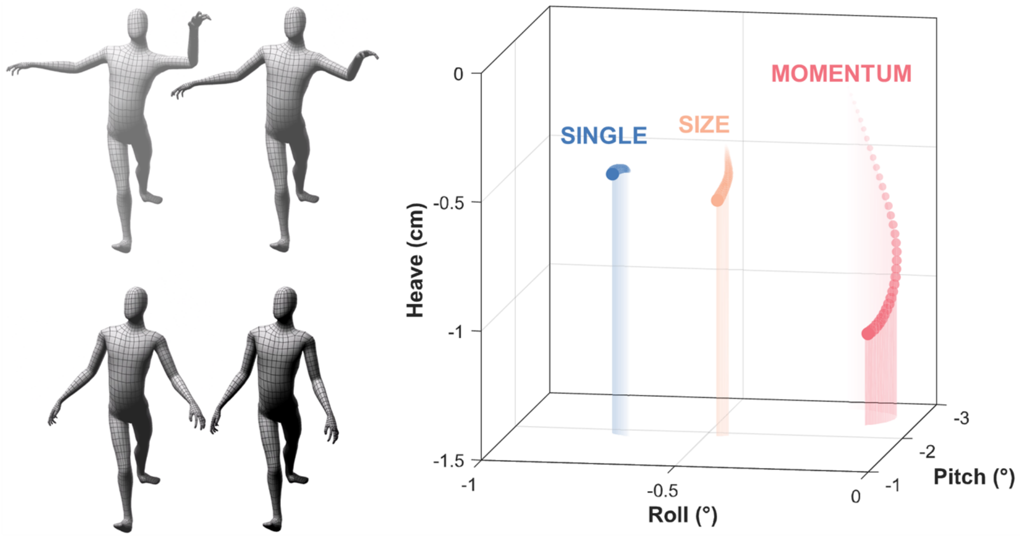


Figure 4.5: An animation segment (taken from Video #1 in Table 4.1) and motion commands generated by the three link weighting policies for object motion proxy. In this animation, the character lowers its arms, and the left arm moves in a larger distance than the right arm. In all images, thicker colors indicate later events in time.

arms (albeit small). *Single*, set to capture only the torso's movement, makes the least motion effect.

4.2.3 Motion Cueing Algorithms

Integrating the velocity of the object motion proxy, calculated by (4.4), and mapping it to a motion command may exceed the motion chair's workspace. To prevent this, we scale the object motion proxy $\mathbf{q}^{obj}$ by the scale factor α to transform the motion proxy from the viewer's perspective space into the motion chair space. Then, we feed the scaled motion proxy, $\alpha\mathbf{q}^{obj}$, to an MPC algorithm to convert it to a motion command $\mathbf{m}$.

In this study, we adopt an MPC with velocity input, which showed the best performance in Section 3.3. The MPC algorithm solves the following optimization problem: Find $\mathbf{m}$ such that

$$\begin{aligned} \mathbf{m} = \arg\min_{\mathbf{m}} \quad & \|\alpha\dot{\mathbf{q}} - \dot{\mathbf{m}}\|^2 + \epsilon\|\mathbf{m}\|^2 \\ \text{subject to} \quad & |\mathbf{m}| \leq \mathbf{m}_{max}, \end{aligned} \tag{4.11}$$

where $\mathbf{m}$ and $\dot{\mathbf{m}}$ are the motion command and its derivative, respectively, and $\mathbf{m}_{max}$ is the maximum displacement of the motion chair. The scale factor α depends on the specific settings of the camera space and the workspace of the motion chair. We use $\alpha = 1.16 \times 10^{-4}$; see Section 4.3.1 for the motion chair used in our work. The other parameter ϵ controls how quickly the motion command $\mathbf{m}$ converges to zero (the neutral position), and we use $\epsilon = 0.3$.

4.3 Evaluation on a Single Articulated Body

We performed a user experiment on many scenes showing a single articulated body in order to compare the benefits of the three link weighting methods.

4.3.1 Methods

Participants

Twenty-seven volunteers (10 females and 17 males; 18–38 years old with an average age of 23.1) with normal sensory ability participated in this experiment. The experiment took no more than 1 hour, and the participants were paid approximately USD 13 after the experiment.

Devices

The motion chair (4DX, CJ 4DPLEX; Figure 1.1) used in the experiment had three DoFs for roll (±4°), pitch (±7°), and heave (±4 cm). The chair is for four people, and the participant sat in the second seat from the left during the experiment. 2D images were projected onto a 94-inch screen using a polarized projector (EB-W16SK, Epson Corp.). We blocked auditory cues by presenting

Table 4.1: Animations used in the user experiment on a single articulated body. All the animations are from CMU mocap database [1].

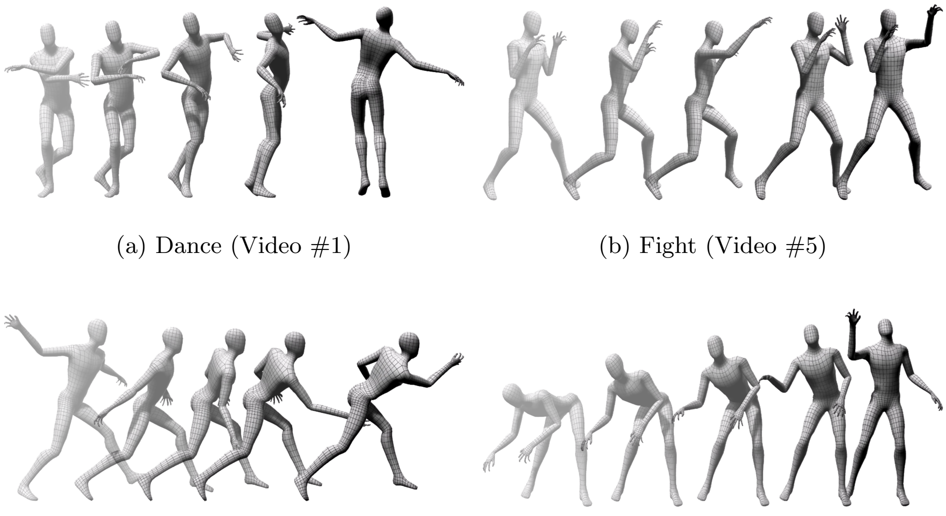
#	Category	Animation ID	Description	Duration (s)
1	Dance	05_02	Expressive arms, pirouette	9.3
2		05_04	Sideways arabesque, folding arms, bending back	10.0
3		120_07	Mickey Dance	9.3
4	Fight	135_01	Bassai	51.0
5		14_01	Boxing	46.6
6		15_13	Boxing	77.7
7	Sport	124_01	Baseball pitch	5.3
8		124_13	Underhand fast pitch	7.2
9		79_02	Swimming	6.5
10	Daily actions	127_02	Range of motion	14.6
11		14_12	Washing windows	22.9
12		76_05	Swatting at pesky bug	6.5

a white noise sound using a noise-canceling headphone during the entire experiment.

Experimental Conditions

From the CMU mocap database [1], we selected 12 animations showing an articulated motion of a single person. Then, we prepared 12 videos using Unity3D, three each for the four categories: dance (Video #1–#3), fight (Video #4–#6), sport (Video #7–#9), and daily actions (Video #10–#12), as described in Table 4.1. The camera orientation was fixed to the side view facing the center of a person in all videos. Example animation sequences from selected videos are shown in Figure 4.6. The full videos are available in the supplemental materials.

For each of the 12 videos, we generated three sets of motion effects using the three weighting methods, SIZE, MOMENTUM, and SINGLE, as described in Section 4.2.2. SINGLE was made by capturing the movement of only the charac-



(a) Dance (Video #1)

(b) Fight (Video #5)

(c) Sport (Video #8)

(d) Daily actions (Video #11)

Figure 4.6: Examples of animated motions used in the user study for a single articulated body.

ter's torso. In this case, SINGLE is equivalent to the motion effect produced by the algorithm of Chapter III, serving as the baseline. We also tested alternatives, e.g., capturing the movement of only the link with the highest momentum adaptively. However, such selections produced excessively intensive motion effects, so we used the torso for SINGLE.

Procedure

For familiarization, participants had a practice session in which they experienced all three motion effects for the three shortest videos (#7, #9, and #12). The main session comprised 12 blocks of trials, one for each of the 12 videos. Each block consisted of three trials in which three motion effects were presented with the video. In each trial, participants could experience the pair of a video and a motion effect as many times as they wanted. The order of the video was randomized across participants, and the order of the motion effects was counterbalanced across the videos. Participants took a 3-min break after the practice session and a 5-min break after the sixth block in the main session.

After experiencing each motion effect, participants answered a questionnaire that included the following seven questions ($\neg$ indicates a negative question):

Q1. Harmony: The motion effect matched the movements of the object.

Q2. $\neg$Causality: I experienced motion effects that I did not understand why they had been provided.

Q3. Fatigue: I felt tired after experiencing the motion effect.

Q4. Distraction: I was distracted from watching the video by the motion effect.

Q5. Detail: The motion effect described detailed movements of the object.

Q6. Preference: I liked the motion effect.

Q7. Free Comment: Please leave a comment regarding the motion effect.

For each question from Q1 to Q6[2], participants expressed the extent of agreement between 0 and 100 by selecting a position on a horizontal line. On the line, descriptors were marked at every 25 points: 0–strongly disagree, 25–disagree, 50–neutral, 75–agree, and 100–strongly agree. For Q7, participants entered a response using a keyboard. In the data analysis, we inverted the scores of three questions (Q2, Q3, and Q4) so that a high score indicates a better experience in every measure.

4.3.2 Results and Discussion

The experimental results are shown in Figure 4.7. We performed a two-way repeated-measures ANOVA on every question from Q1 to Q6 using *Motion Effect* and *Video* as the independent variables. We then applied the SNK test for post-hoc multiple comparisons. *Motion Effect* was statistically significant for

[2] Three out of the six questions were designed to imply negative meanings for balancing.

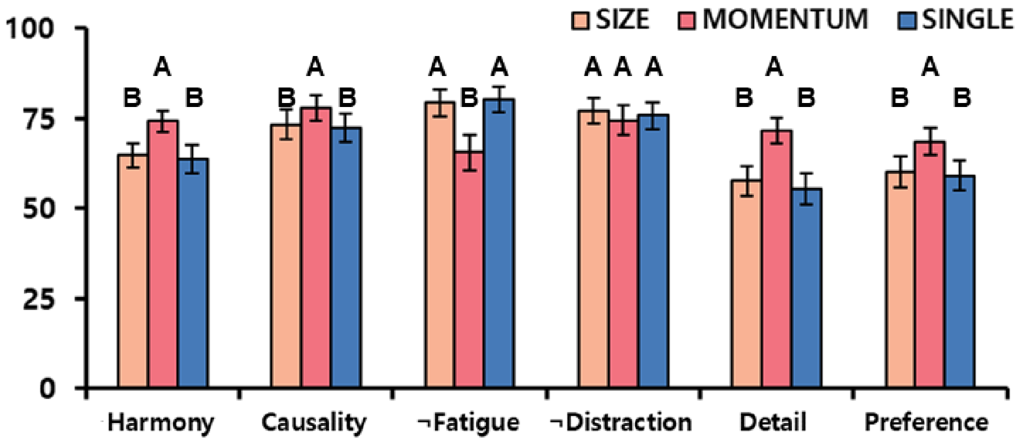
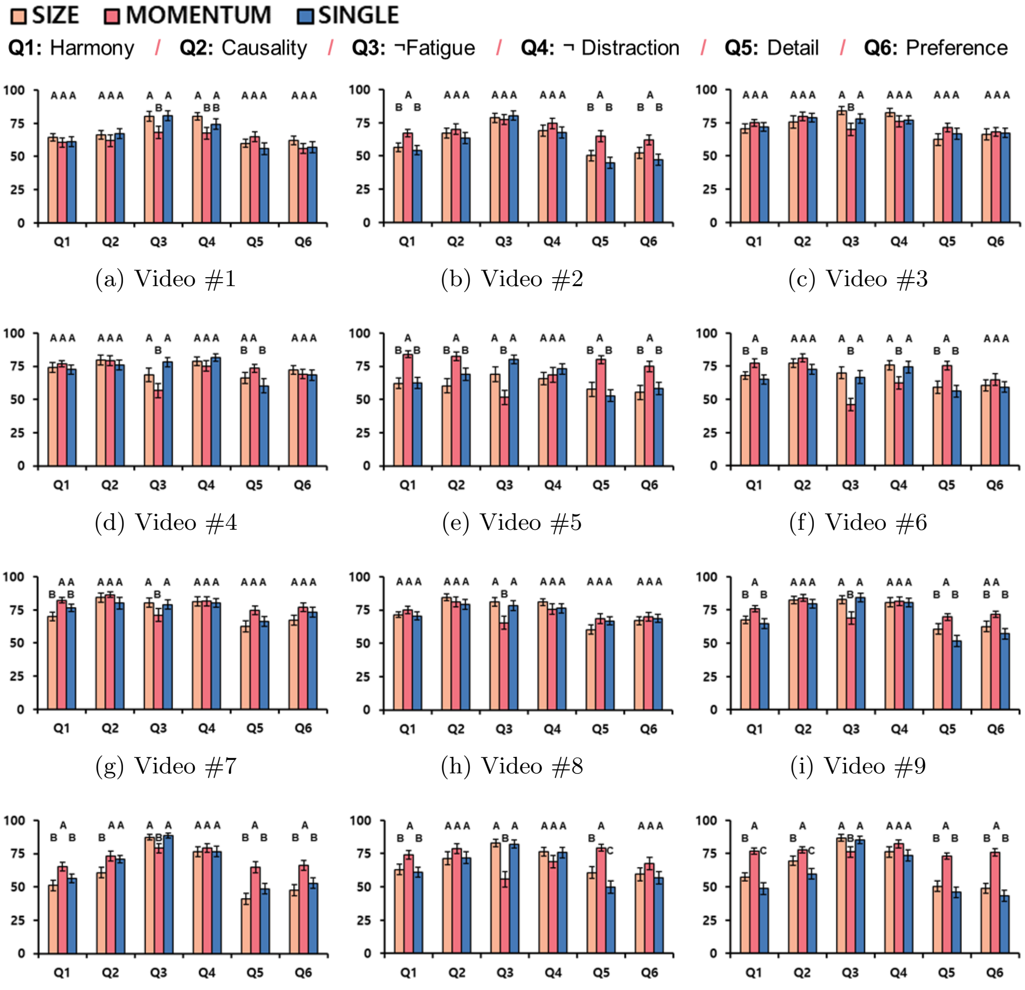


Figure 4.7: Overall results of the user experiment for a single articulated body. Error bars represent standard errors. Motion effect sets marked with the same letters indicate that they did not show statistically significant differences by the SNK tests. A higher score indicates a better performance.

Q1 (Harmony; $F(2, 52) = 29.52$, $p < .0001$), Q2 (¬Causality; $F(2, 52) = 10.25$, $p = .0002$), Q3 (Fatigue; $F(2, 52) = 54.79$, $p < .0001$), Q5 (Detail; $F(2, 52) = 33.62$, $p < .0001$), Q6 (Preference; $F(2, 52) = 14.32$, $p < .0001$), but not for Q4 (Distraction; $F(2, 52) = 1.06$, $p = .3538$). *Video* was significant for every measure ($p < .0001$). We also observed significant interactions between *Motion Effect* and *Video* in every measure. More detailed results of the statistical tests are available in the supplemental materials. Also see Figure 4.8 for the results of the individual videos.



(a) Video #1 (b) Video #2 (c) Video #3

(d) Video #4 (e) Video #5 (f) Video #6

(g) Video #7 (h) Video #8 (i) Video #9

(j) Video #10 (k) Video #11 (l) Video #12

Figure 4.8: User experiment results of single articulated body rendering per video. Error bars represent standard errors. Motion effect sets grouped with the same letters indicate that they did not show statistically significant differences by the SNK tests. A higher score indicates a better result for every measure.

On average, MOMENTUM showed the best scores in Q1 (Harmony), Q2 (Causality), Q4 (Detail), and Q6 (Preference) with statistical significance (Figure 4.7). This result indicates that for the source videos, the motion effects generated by MOMENTUM were better in matching the visual events, conveying the sense of result from the visual cause, and capturing detailed and fast object movements. Therefore, the participants favored MOMENTUM more than SIZE and SINGLE. However, MOMENTUM was inferior in Q3 (Fatigue), which implies that its motion effects can be more drastic and tiring. In the results of the individual videos, MOMENTUM also outperformed SIZE and SINGLE in most measures except Q3 (Fatigue) (Figure 4.8).

These results were supported by the participant's comments about MOMENTUM. For example, one participant said, "It was good to move slowly in smooth parts and to move vividly in abrupt parts." (Video #7). However, for long videos, some participants mentioned, "It was the most lively. But because the video was long, I was tired, and my satisfaction decreased." (Video #10) and "It would be perfect if the intensity were weaker." (Video #8).

In comparison, since SIZE does not emphasize fast movements, the weights of SIZE were more evenly distributed than those of MOMENTUM. Thus, motion effects made using SIZE could feel unfocused, resulting in relatively low scores. Four participants reported that some object movements were missed or not expressed well with SIZE. SINGLE also received low scores similar to SIZE. It was mainly because SINGLE represented only the torso's movement without considering the arms' or legs' movements. It was pointed out by three participants. Nevertheless, SIZE and SINGLE scored comparably to MOMENTUM in Video #3, #4, #6, #7, #8, and #11, in which the movements of all the body parts were highly correlated to each other. This implies that using only the torso's movement may be sufficient to create motion effects in some cases.

Interestingly, we observed that a motion effect can affect the perception of a character's properties in the video. One participant commented, "I felt the more intensive the motion effect is, the stronger the person moved." (Video #9). Meanwhile, some participants reported that the video context influenced their expectations of motion effects, e.g., "I tended to rate strong motion effects for martial arts more positively." (Video #7). In summary, *Momentum* showed the best performance. However, its abrupt movements sometimes disrupt users' satisfaction. Keeping this shortcoming in mind, we adopt *Momentum* to compute the object motion proxy for each of the multiple articulated bodies.

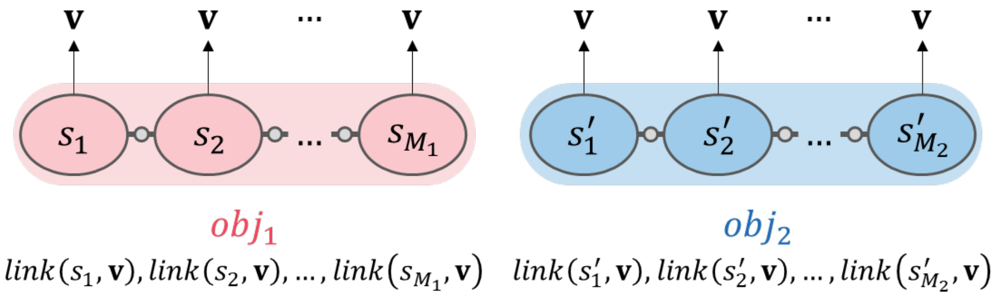
4.4 Multiple Articulated Bodies

This section presents a few methods to set the object weights, w^{obj}, when multiple articulated bodies appear in the scene. As in Section 4.2.1, we define requirements for the object weights and design weighting policies satisfying the requirements. The velocity of the scene motion proxy, $\dot{\mathbf{q}}^{obj}$, is calculated by substituting the weights into (4.4) and (4.5) and then converted to a motion command $\mathbf{m}$ through an MCA.

4.4.1 Requirement on Object Weights

Valid object weighting methods need to meet the following two conditions: First, faster or larger objects are weighted more. Second, visually identical objects are equally weighted. The first condition can be easily satisfied if the object weights monotonically increase with the size or speed.

For the second condition, suppose that there are two objects, obj_1 and obj_2, consisting of $\{link(s_1, \mathbf{v}_1), \cdots, link(s_{M_1}, \mathbf{v}_{M_1})\}$ and $\{link(s'_1, \mathbf{v}'_1), \cdots, link(s'_{M_2}, \mathbf{v}'_{M_2})\}$, respectively, as depicted in Figure 4.9. If their sizes are the same, i.e, $\sum_{i=1}^{M_1} s_i = \sum_{i=1}^{M_2} s'_i$, and they all have the same velocity $\mathbf{v}$, then the two objects



v
v
…
v
v
v
…
v
s_1
s_2
…
s_{M_1}
s'_1
s'_2
…
s'_{M_2}
obj_1
obj_2
$link(s_1, \mathbf{v}), link(s_2, \mathbf{v}), \ldots, link(s_{M_1}, \mathbf{v})$
$link(s'_1, \mathbf{v}), link(s'_2, \mathbf{v}), \ldots, link(s'_{M_2}, \mathbf{v})$

Figure 4.9: A case where two objects are equally weighted.

have identical visual effects. This gives a constraint that the two objects must have the same object weights, such that:

$$w^{obj}(obj_1) = w^{obj}(obj_2). \tag{4.12}$$

This second condition can be violated by many weighting methods. For example, one can set an object weight to be directly proportional to the sum of the velocities of all the links constituting the object, i.e., $w^{obj} = \sum_{j=1}^{M} \mathbf{v}_j$, but it violates the second condition when $M_1 \neq M_2$; the object weight is $M_1\mathbf{v}$ for one object and $M_2\mathbf{v}$ for the other.

4.4.2 Weighting Policies

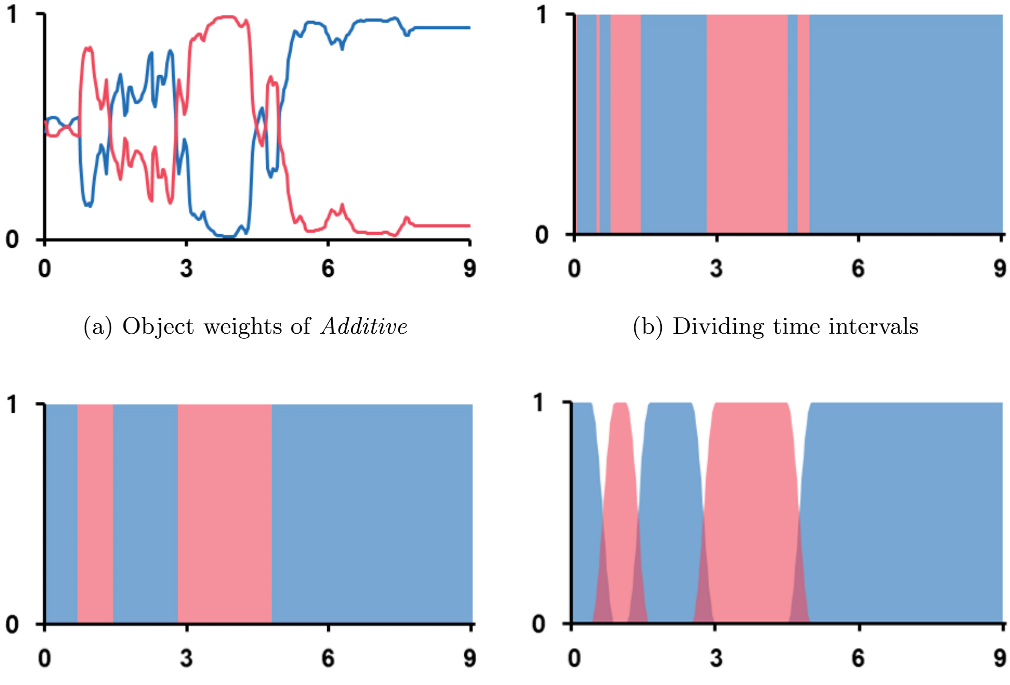
We design four methods that compute the object weights while satisfying the two requirements described in Section 4.4.1.

Additive

This method sets the object weight w^{obj} as the sum of the momentums, i.e., size × speed, of all links comprising the object:

$$w_i^{obj} = \frac{\sum_{j=1}^{M_i} s_{i,j} \|\mathbf{v}_{i,j}\|}{\sum_{k=1}^{N} \sum_{j=1}^{M_i} s_{k,j} \|\mathbf{v}_{k,j}\|}, \tag{4.13}$$

where M_i is the number of links of object i and N is the number of all the objects in the scene. In this method, the higher the link weights constituting the object



(a) Object weights of *Additive*

(b) Dividing time intervals

(c) Removing short intervals

(d) Smoothing transitions

Figure 4.10: Overall process of *Salient*. These plots are for Video #4 in Table 4.2. Red and blue lines (or areas) indicate the object weights for the attacker and the victim, respectively.

are, the more weighted the object is. Therefore, this method can be advantageous for capturing the movements of large and fast objects.

Uniform

This method gives the same weights to all objects, i.e., $w_i^{obj} = 1/N$. For example, if there are two objects, the scene motion proxy $\mathbf{q}^{scn}$ reflects the two object motion proxies, $\mathbf{q}_1^{obj}$ and $\mathbf{q}_2^{obj}$, equally.

Salient

When multiple objects appear, one object may attain the viewer's attention almost exclusively, whereas the movements of other objects are neglected. For example, imagine the Iron Man fighting against many (insignificant) enemies.

This method, *Salient*, is designed for such cases and attempts to express the movement of the most important object at that instant. We use the object weights calculated by the *Additive* policy to judge the importance of each object. *Salient* determines the object weights in three steps, as illustrated in Figure 4.10. Here, we denote the object weight of *Additive* by w^{add} and the object weight of *Salient* by w^{sal}. First, we make time intervals based on which object has the greatest w^{add} in each interval (Figure 4.10a). In each interval, we set w^{sal} of the object with the highest w^{add} to 1, and w^{sal} of the other objects to 0 (Figure 4.10b). Very short intervals (less than 1 s) are merged to their preceding intervals to achieve stable motion (Figure 4.10c). Finally, for a continuous blending of the object motion proxies, w^{sal} is smoothed using the half period of the cosine profile with a period of 1 s (Figure 4.10d).

Single

This method captures only the movement of one object among the many in the scene. The target object gets the full weight of 1, while the weights of all the other objects are zeros.

Comparisons

Figure 4.11 shows an example illustrating the effects of the four weighting policies. The paired animation shows that the victim (left character) punches first, but the attacker (right character) avoids it and raises its right leg to kick. For this animation, when capturing the victim's movement, *Single* makes a motion effect expressing the victim's pulling the right arm. In contrast, when capturing the attacker's movement, it makes a motion effect for the attacker's raising the right leg. *Additive* responds to both movements. *Uniform* makes a motion effect expressing half the two movements. Lastly, *Salient* regards only the victim's movement as important and makes a motion effect for the victim.

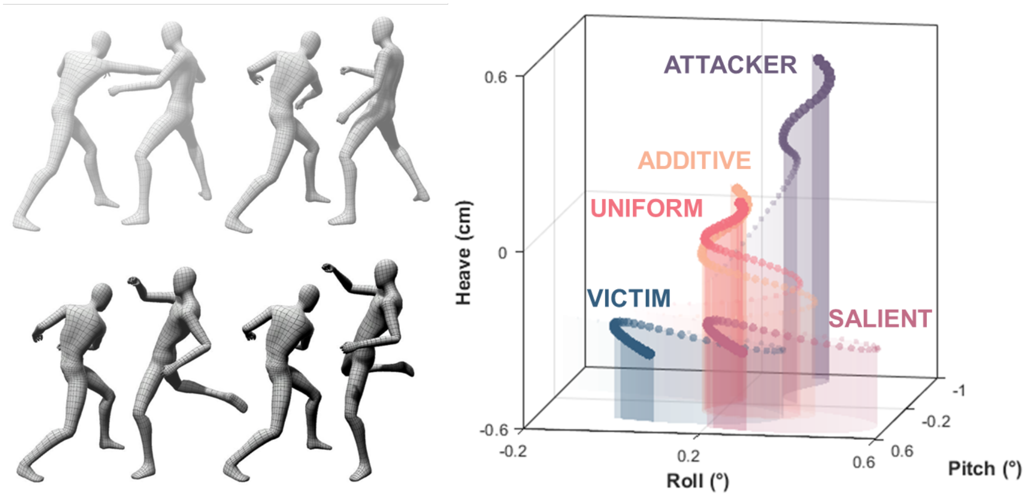


Figure 4.11: An animation segment (taken from Video #6 in Table 4.2) and motion commands generated by the four object weighting policies for scene motion proxy. In this animation, the victim (left character) punches first, but the attacker (right character) avoids it and raises its right leg to kick. In all images, thicker colors indicate later events in time.

4.4.3 Post-Processing and Motion Cueing Algorithms

The velocity of the scene motion proxy, $\dot{\mathbf{q}}^{scn}$, obtained by (4.4) and (4.5), is converted into a motion command $\mathbf{m}$. We use *Momentum*, which showed the best performance in Section 4.3, to determine the link weights w^{link}. Additionally, to prevent the abrupt chair movements pointed out in the previous user study (Section 4.3.2), we apply the following attenuation rule, which reduces the variations of a motion effect over time:

$$\widehat{\dot{\mathbf{q}}}^{scn} = \beta\|\dot{\mathbf{q}}^{scn}\|^{\gamma}\frac{\dot{\mathbf{q}}^{scn}}{\|\dot{\mathbf{q}}^{scn}\|}, \tag{4.14}$$

where $\widehat{\dot{\mathbf{q}}}^{scn}$ is the velocity of the attenuated motion proxy, β is a scale factor that makes $\widehat{\dot{\mathbf{q}}}^{scn}$ and $\dot{\mathbf{q}}^{scn}$ have a similar level of signal energy, and γ is a gain for attenuation ($0 < \gamma \leq 1$).

Decreasing γ lowers the fluctuations of the scene motion proxy by following the power function that has γ as the exponent. As a result, when the object movements are strong, using the attenuated scene motion proxy, instead of the original proxy, decreases the motion effect intensity. When the object movements are weak, it rather increases the motion effect intensity. This can be easily understood by comparing what happens to (4.14) if $\gamma = 1$ or 0.5. According to our experience, the range of γ for perceptually-best motion effects is 0.5–0.7. We set $\beta = 6.0$ and $\gamma = 0.7$ in the user study reported in Section 4.5.

Finally, the attenuated motion proxy, $\widehat{\dot{\mathbf{q}}}^{scn}$, is converted into a motion command through scaling and MCA, as described in Section 4.2.3.

4.5 Evaluation on Multiple Articulated Bodies

We performed a user experiment to compare the benefits of the four weighting policies that determine the scene motion proxy. The experiment used many scenes where two articulated bodies interacted with each other.

4.5.1 Methods

Participants

Twenty volunteers (9 females and 11 males; 19–29 years old with an average age of 23.4) with normal sensory ability participated in this experiment. The experiment took approximately 90 min. The participants were paid approximately USD 22 after the experiment.

Experimental Conditions

From the Unity Asset Store, we selected eight animation pairs, as described in Table 4.2. Each animation pair showed the articulated motions of two people, the attacker and the victim, fighting each other. Then, we prepared eight videos for them using Unity3D. The camera orientation was fixed to the side view facing the center of two people in all videos. Some animation sequences from selected videos are shown in Figure 4.12.

For each of the eight videos, we used four weighting methods described in Section 4.4.2 to generate five motion effects: ADDITIVE, UNIFORM, SALIENT, ATTACKER, and VICTIM. ATTACKER and VICTIM were made using *Single* by capturing the movement of only the attacker and only the victim, respectively.

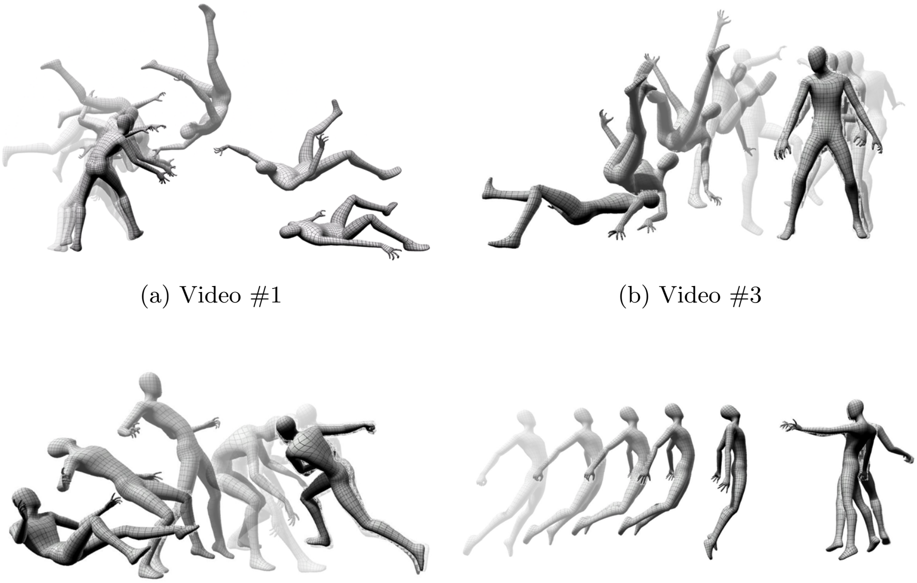
Procedure

For familiarization, participants had a practice session in which they experienced all five motion effects for the three shortest videos (#2, #3, and #6). The main session comprised eight blocks of trials, one for each video. Each block consisted of five trials in which the five motion effects were presented with the video. In each trial, participants could experience the pair of a video and a motion effect as many times as they wanted. The order of the video was randomized, and the order of the motion effects per video was balanced across participants using the balanced Latin square. Participants had a 3-min break after the practice session and a 5-min break after the fourth block in the main session.

After experiencing each motion effect, participants answered the same questionnaire described in Section 4.3. In this questionnaire, we slightly modified Q2 (Causality) to “It was easy to understand the relationship between the video and the motion effect.” This was to test whether the participant could find out which of the two characters the motion effect was focused on. Participants’ response methods to the questions were also the same with the previous experiment. In the data analysis, we inverted the scores of two questions (Q3 and Q4) so that a high score indicates a better experience in every measure.

Table 4.2: Animations used in the user experiment on multiple articulated bodies. All the animations are from wemakethegame[4].

#	Animation	Description	Duration (s)
1	Judo, Setmotion 1	The attacker lifts the victim and throws it back.	10.8
2	Judo, Setmotion 2	The attacker throws the victim while rolling backward.	7.7
3	Judo, Setmotion 6	The attacker trips the victim over with its foot.	8.1
4	Judo, Setmotion 7	The attacker holds the victim's head and trips the victim over with its foot.	9.1
5	Rogue Finisher, Finisher 1	The attacker hits the victim four times and stabs the victim. Then, the attacker pushes the victim back with its body.	10.4
6	Rogue Finisher, Finisher 4	The attacker avoids the victim's punch and kicks the victim. Then the attacker hits the victim twice, and the victim falls down.	8.1
7	Rogue Finisher, Reversal Finisher 1	The attacker blocks the victim's punch and stabs the victim several times. The victim then falls down.	9.0
8	Superhero, Neck Finisher	The attacker pulls the victim from a distance with a superpower, and the victim falls down.	9.8



(a) Video #1

(b) Video #3

(c) Video #6

(d) Video #8

Figure 4.12: Examples of animated motions used in the user study for multiple articulated bodies.

4.5.2 Results and Discussion

The experimental results averaged over the eight videos are shown in Figure 4.13. We performed a two-way repeated-measures ANOVA on every question from Q1 to Q6 using *Motion Effect* and *Video* as the independent variables. We then used the SNK test for post-hoc multiple comparisons. *Motion Effect* was statistically significant for every question: Q1 (Harmony; $F(4, 76) = 34.1$, $p < .0001$), Q2 (Causality; $F(4, 76) = 17.56$, $p = .0002$), Q3 (Fatigue; $F(4, 76) = 6.92$, $p < .0001$), Q4 (Distraction; $F(4, 76) = 3.34$, $p = .0141$), Q5 (Detail; $F(4, 76) = 40.87$, $p < .0001$), and Q6 (Preference; $F(4, 76) = 30.55$, $p < .0001$). *Video* was significant for only Q3 (Fatigue; $F(7, 133) = 2.86$, $p = .0081$). We observed significant interactions between *Motion Effect* and *Video* in five measures; Q1 (Harmony; $F(28, 532) = 6.85$, $p < .0001$), Q2 (Causality; $F(28, 532) = 4.74$, $p < .0001$), Q3 (Fatigue; $F(28, 532) = 2.89$, $p < .0001$), Q5

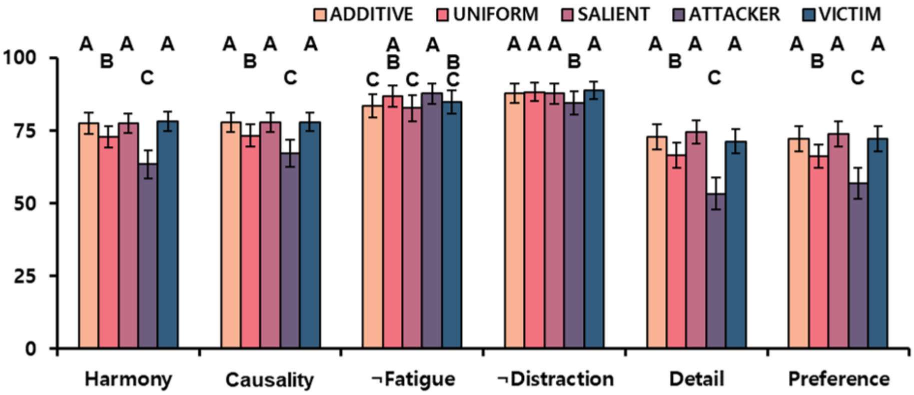


Figure 4.13: Average results of the user experiment for multiple articulated bodies. Error bars represent standard errors. Motion effect sets marked with the same letter indicate that they did not show statistically significant differences by the SNK tests. A higher score indicates a better performance.

(Detail; $F(28, 532) = 7.73$, $p < .0001$), and Q6 (Preference; $F(28, 532) = 6.43$, $p < .0001$), but not in Q4 (Distraction; $F(28, 532) = 0.90$, $p = .6166$). More detailed results of the statistical tests are provided in the supplemental materials. In summary, ADDITIVE, SALIENT, and VICTIM showed the best scores in all measures except Q3 (Fatigue), followed by the UNIFORM (Figure 4.13). ATTACKER received the lowest scores in all metrics except Q3 (Fatigue).

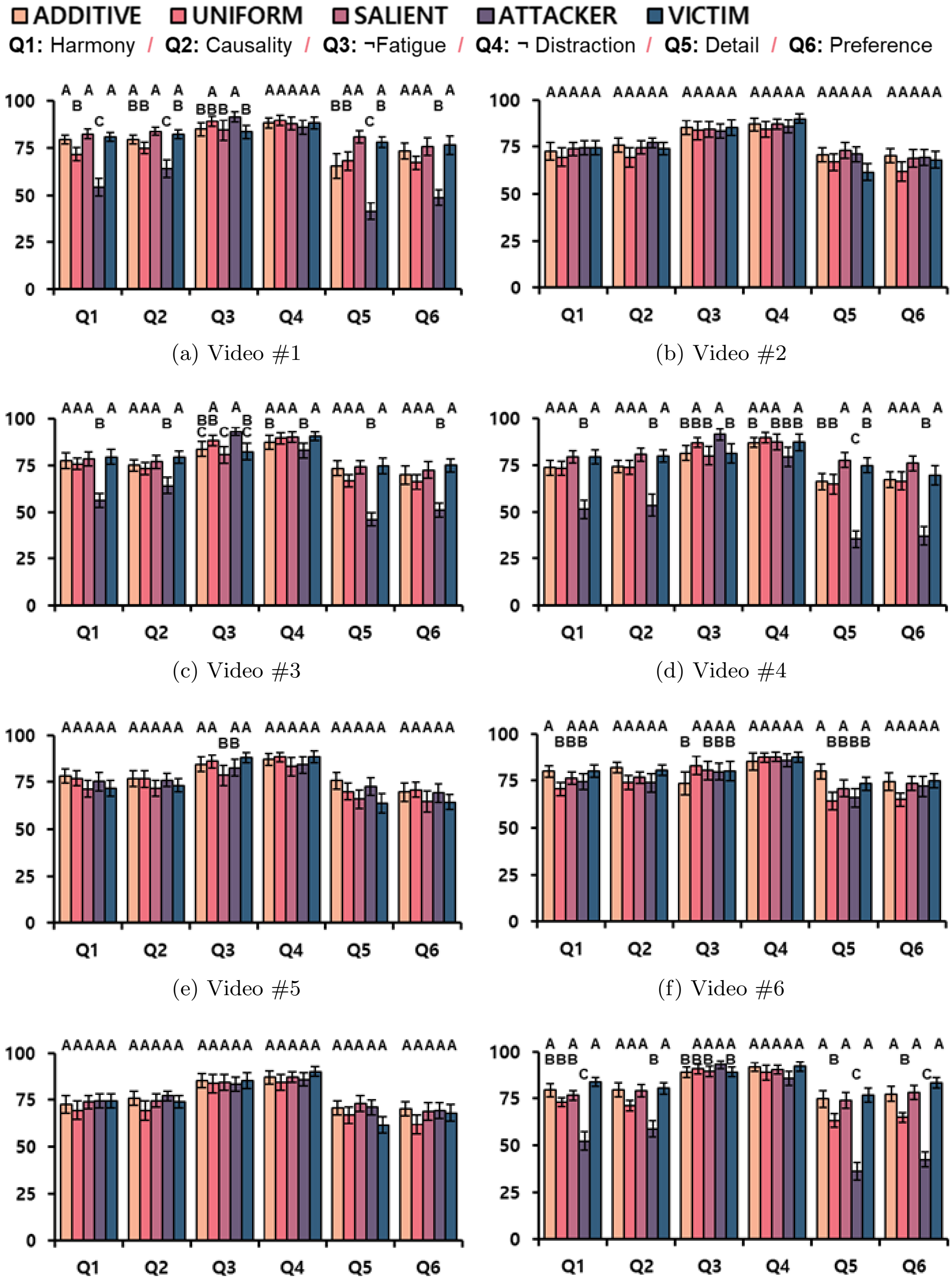
VICTIM elicited significantly better viewer experiences than ATTACKER even though both were made using the same weighting policy *Single*. It is presumably because the movement of the victim was more dynamic than the attacker in most videos. The attacker usually showed short and fast movements, such as punching, kicking, and stabbing. On the contrary, the victim exhibited long-lasting and large movements, such as staggering and falling. Eight participants pointed out that ATTACKER missed important movements, while only four said the same about VICTIM. For the same reason, ATTACKER made the participants less tired, leading to high scores in Q3 (Fatigue). However, VICTIM was not always better than ATTACKER. ATTACKER and VICTIM had similar scores in

Video #2, #5, #6, and #7, where the attacker's movements were as large and dynamic as the victim's. In Video #5, ATTACKER scored slightly higher than VICTIM in Q1 (Harmony), Q2 (Causality), Q5 (Detail), and Q6 (Preference), although the differences were insignificant.

Both ADDITIVE and SALIENT received the highest scores (similar to VICTIM). It seems that the object weights changing dynamically and quickly by both methods well captured the important movements of the attacker and the victim. However, participants frequently complained that ADDITIVE and SALIENT presented excessively strong motion effects. Also, SALIENT sometimes missed important movements; one participant said, "No motion was given for pulling out the hand after stabbing." (Video #5). As illustrated in Figure 4.15, SALIENT did not capture the attacker's quick movement as it was removed by the merging process. The same participant also commented on this: "I could not understand what the motion effect focused on." (Video #7). This problem can be improved by dynamically adjusting the window size during motion merging or using a better attention model.

UNIFORM resulted in lower scores than ADDITIVE, SALIENT, and VICTIM, due to the weak strength of its motion effects as reported by five participants. This is because the movements of the two objects can be canceled out when they are evenly combined, without emphasizing the movement of a more important object. The weak intensity of UNIFORM also led to the high scores of Q3 (Fatigue).

As in Section 4.3, we observed that motion effects could alter the perceived properties of characters and actions. Two participants commented on this: "The movement was excessive, and it felt like a *heavy* person was standing up." (Video #1) and "It felt like *the attacker was hitting very weakly* due to the weak motion effect." (Video #5).



(a) Video #1

(b) Video #2

(c) Video #3

(d) Video #4

(e) Video #5

(f) Video #6

(g) Video #7

(h) Video #8

Figure 4.14: User experiment results for multiple articulated bodies per video. Error bars represent standard errors. Motion effect sets marked with the same letters indicate that they did not show statistically significant differences by the SNK tests. A higher score indicates a better performance.

In conclusion, ADDITIVE, SALIENT, and VICTIM showed better perceptual performance than the others. In our videos, the attacker and the victim were distinguishable. However, the attacker and the victim may not be clearly distinguished in some cases, e.g., when two people beat each other up. In this case, ADDITIVE and SALIENT may have a merit because both vary the object weights dynamically. Although we did not test animations for more than two objects, we expect that ADDITIVE and SALIENT can produce more plausible motion effects in such cases, where the viewer's attention can be dispersed to multiple objects.

4.6 Limitations

Our algorithm synthesizes convincing motion effects based on the link movements of articulated bodies. However, it requires information on the sizes and velocities of many links. Such information is directly accessible in computer-generated content, such as animation movies and VR games, as in our implementation. For regular 4D movies, the information can be extracted from the image sequence using recent computer vision techniques, such as pose estimation [88], depth estimation [72], and object detection [73]. We leave an integration of such automatic feature extraction as future work.

Additionally, 4D effect designers need to tune parameters, such as α, β, γ, and ϵ in (4.11) and (4.14). These parameters should be adjusted considering various factors, e.g., genre, situation, context, and mood of the audiovisual content. Recently, Li et al. [13] showcased adjusting the intensity of vibrotactile effects based on the contextual information extracted from the psychoacoustic measures of sound. Such an approach can also be useful for our framework, where strong motion effects are preferred in intense scenes and gentle motion effects in sentimental scenes.

Finally, we tested our algorithm in the limited cases for two articulated

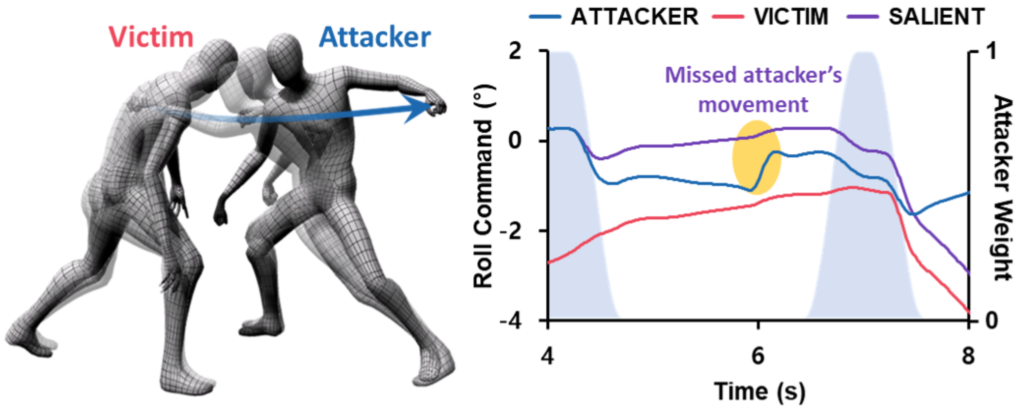


Figure 4.15: A case that SALIENT fails to capture the important movement in Video #5. In this video, the attacker (the right person) quickly withdraws his hand at 6 s. The right graph shows the roll commands of ATTACKER, VICTIM, and SALIENT. In the blue areas, the object weight for the attacker is used for SALIENT. SALIENT does not capture the attacker's movement at 6 s (yellow area), as it is removed by the merging process.

bodies in the scene. We have not tested for more than two objects, but we expect *Additive* and *Salient* to have advantages over the other weighting methods. In addition, while the camera moves freely in general movies, the camera was fixed in all the videos used in the user studies. In camera-moving scenes, the camera movement may be compensated by subtracting the background movement from the link velocities. Camera motion estimation techniques [89] can be applied for that purpose.

Our future work will focus on resolving the above limitations.

4.7 Conclusion

In this chapter, we have addressed a new research problem: enhancing the user experiences elicited by the multisensory 4D content that includes motion effects emphasizing the visual movements of multiple articulated bodies. Our approach is an algorithmic framework that generates object-based motion effects automatically by compressing the complex external and internal movements of articulated bodies into a single point: motion proxy. To this end, we have extended the computational definition of motion proxy from one rigid body (Chapter III) to an articulated body, and then to multiple articulated objects. For each case, a few weighting methods to determine the motion proxy are designed, and their relative merits and drawbacks are evaluated by user studies. Finally, major findings and our research experiences are summarized into design guidelines and current limitations. To the best of our knowledge, this work is the first attempt to automatically generate object-based motion effects for complex articulated bodies moving in the scene.

V. Generating Motion Effects for Scene Components

The algorithm discussed in Chapter IV successfully generates captivating motion effects for objects with complex structures. However, in a typical scene, there are multiple moving components, including objects, particles, and camera motion, that the algorithm does not account for. This limitation prompts us to propose an algorithm in this chapter that aims to generate motion effects that comprehensively capture the movements of all scene components. In particular, this chapter even covers a process for automatically extracting motion information from audiovisual content using computer vision technologies, whereas the previous chapters assumed that motion trajectories are given. By incorporating these techniques, we are able to address the broader scope of scene components and their corresponding motions. For a comprehensive representation of the movements of scene components, we extend the concept of motion proxy to the scene components. We designed several algorithms to create motion effects based on the motion proxy, and we further investigated the optimal algorithm through a user study.

5.1 Problem Formulation

In this section, we formulate a problem for generating motion effects for moving scene components. First, we briefly summarize the formulation of a motion proxy for multiple articulated bodies, described in Chapter IV. Building upon this foundation, we extend the concept of the motion proxy to scene components.

In Chapter IV, the motion proxy for articulated bodies was defined to compress the many movements of multiple parts and multiple objects into a single movement as

$$\dot{\mathbf{q}} = \sum_{i=1}^{N} w_i^{obj} \left(cl_i\dot{\mathbf{d}}_i + \sum_{j=1}^{M_i} w_{i,j}^{link}\mathbf{v}_{i,j} \right), \tag{5.1}$$

where $\dot{\mathbf{q}}$ is the 3D velocity of motion proxy, N is the number of objects, c is constant. For object i, w_i^{obj} is its weight, l_i is its length, $\dot{\mathbf{d}}_i$ is the derivative of the 3D direction vector, M_i is the number of links, $w_{i,j}^{link}$ is the weight of link j, and $\mathbf{v}_{i,j}$ is the visual velocity of link j in the 3D visual space. Here, the term for direction vectors was to express the object rotations, and the term for linear velocities was to express the link translations.

To reproduce the movement of the motion proxy within the limited workspace of a motion platform, the movement of the motion proxy is fed to a motion cueing algorithm (MCA) to generate a motion command as

$$\mathbf{m} = \mathrm{MCA}(\dot{\mathbf{q}}), \tag{5.2}$$

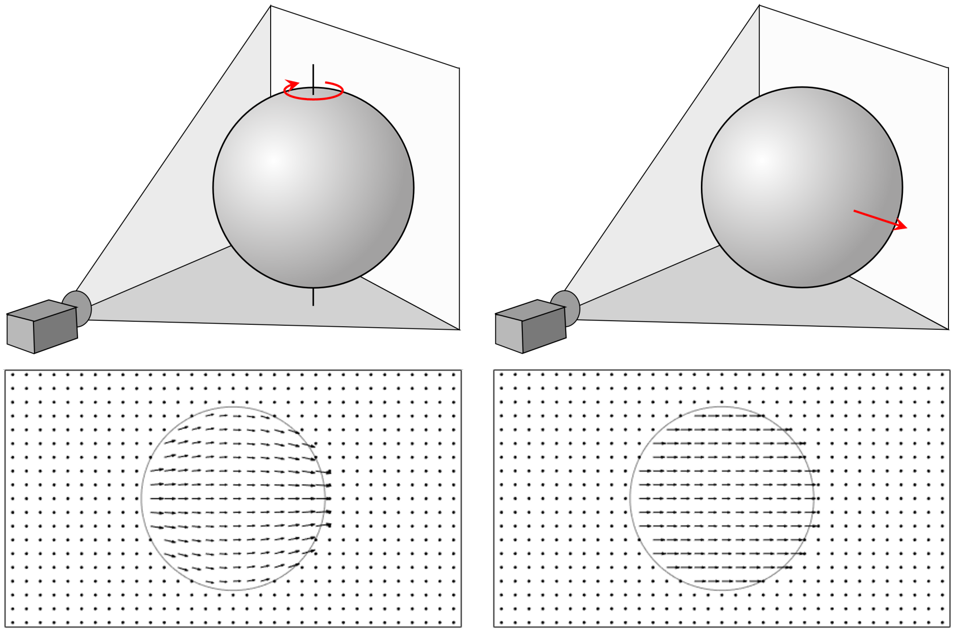
where $\mathbf{m} = (roll, heave, pitch)$ is the 3-DoF motion command to a motion chair.

In a video, the movements of all scene components are inherently accompanied by corresponding pixel movements. For example, an explosion accompanying the movement of particles is visualized as diverging movements of the corresponding pixels, and a camera motion is represented by the movement of entire pixels in the video in the opposite direction to the camera.

This implies that the movements of scene components can be represented by combining the individual pixel movements. Therefore, similar to motion proxies for multi-joint bodies, we define a motion proxy for scene components by aggregating the movements of all pixels within the image, as

$$\dot{\mathbf{q}} = \sum_{i=1}^{H}\sum_{j=1}^{W} w[i,j]\mathbf{v}[i,j], \tag{5.3}$$

where $\dot{\mathbf{q}} = (\dot{q}_x, \dot{q}_y, \dot{q}_z)$ is the velocity of the motion proxy, H and W are the image height and width of the audiovisual content, and $w[i,j]$ and $\mathbf{v}[i,j] =$



(a) Rotation

(b) Translation

Figure 5.1: Examples of the optical flows for the rotation and translation of a sphere.

$(v_x[i,j], v_y[i,j], v_z[i,j])$ are the weight and velocity of the pixel of the i-th row and j-th column. This equation does not incorporate the term for the direction vector in (5.1). This omission is due to the fact that the pixel movements involve both rotation and translation of an object, as illustrated in Figure 5.1.

To extract the values of the two parameters, $w[i,j]$ and $\mathbf{v}[i,j]$, from a video, our algorithm leverages two computer vision technologies: scene flow estimation and saliency detection. The core of our algorithm revolves around the computation of the motion proxy, which serves as a crucial foundation for generating convincing object-based motion effects. The details of our algorithm will be described in Section 5.2.

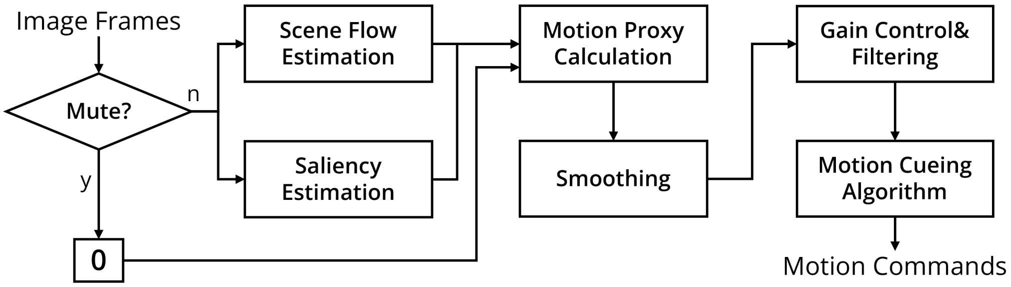


Figure 5.2: Overview of our algorithm to generate the object-based motion effects.

5.2 Motion Effects Synthesis Algorithm

This section describes a pipeline of our automatic algorithm to generate motion effects for scene components based on pixel movements (Figure 5.2). Our algorithm only requires timing information about (1) mute intervals and (2) shot transitions. Mute intervals encompass conversations or static scenes where motion effects are not desired. Shot transitions can be easily detected using computer vision techniques [90, 91]. However, we manually annotated such information for accuracy in this study. Our algorithm extracts pixel motions from each frame of the video source and combines these motions to compute a motion proxy (Section 5.2.1). Then, the motion proxy is modulated (Section 5.2.2) and converted to motion commands through an MCA (Section 5.2.3).

5.2.1 Calculation of Motion Proxy

Our algorithm determines the velocity of motion proxy, $\dot{\mathbf{q}}$ in (5.3) by combining pixel velocities, $\mathbf{v}[i,j]$ with corresponding weights, $w[i,j]$. To obtain such information, we employ two computer vision technologies: scene flow estimation [92] and saliency detection [93] (Figure 5.3). For each pixel i, j, the scene flow estimation estimates a displacement vector, $\mathbf{f}[i,j] = (f_x[i,j], f_y[i,j], f_z[i,j])$, between two consecutive frames in a 3D camera space. The saliency detection al-

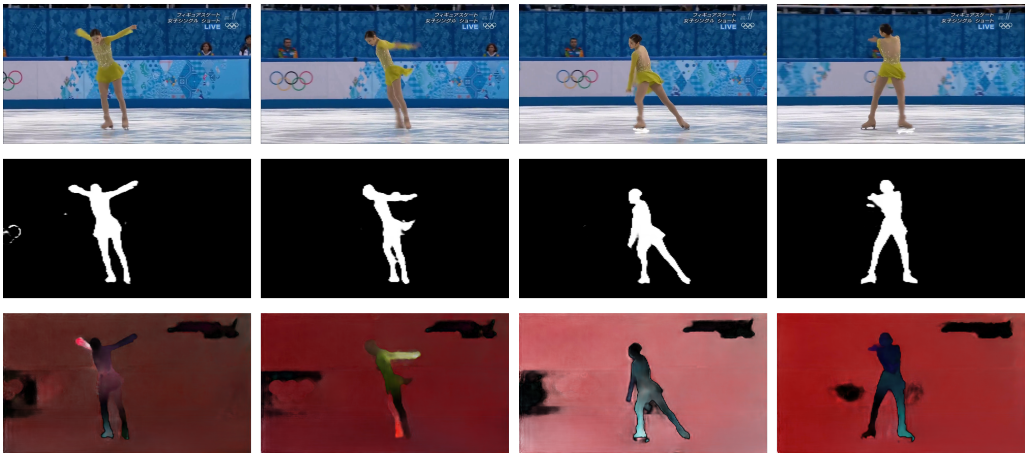


Figure 5.3: Examples of the results of two computer vision technologies on a figure skating video (*Skating* in Table 5.1). The images in the 1st, 2nd, and 3rd rows visualize the original images, the saliencies, and the scene flows, respectively. The scene flows in the x, y, and z axes are mapped to the hue, saturation, and value of color.

gorithm uses an image to estimate a saliency value, $s[i,j]$ $(0 \leq s[i,j] \leq 1)$, which was directly mapped to the pixel weight, i.e., $w[i,j] = s[i,j]$.

As in Section 3.5, we determined the pixel velocity by considering two modes of visual object motion perception by Duncker [80]: subject-relative and object-relative. In subject-relative perception, the object motion is perceived within the viewer's local coordinate frame, while in object-relative perception, the relative displacement between two objects is perceived. The choice of perception mode depends on various factors, including visual characteristics and attention. Since the quantitative knowledge about the factor effects significantly lacks [82], our algorithm supports both subject-relative and object-relative modes to accommodate both modes and provide designers with the flexibility to generate more plausible effects. For subject-relative perception, we defined the subject-relative velocity of a pixel as $\mathbf{v}^{sbj}[i,j] = \mathbf{f}[i,j]$. In object-relative mode, we calculated the velocity, $\mathbf{v}^{obj}[i,j]$, by compensating the scene flow with the background move-

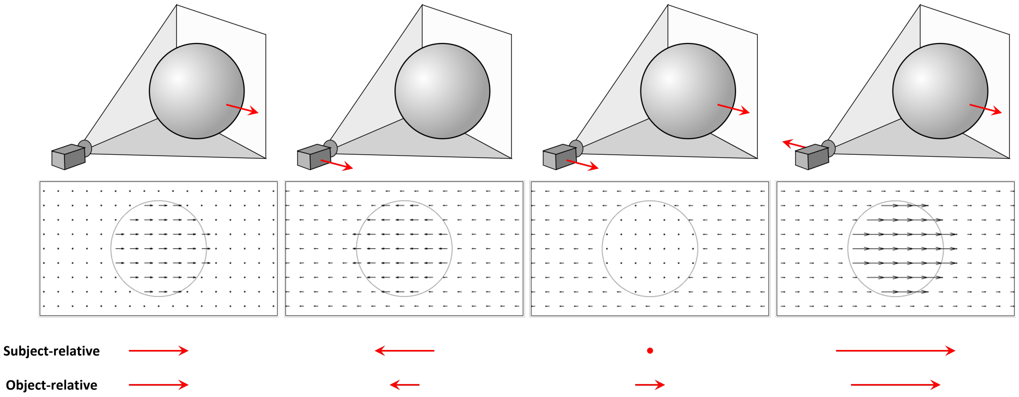


Figure 5.4: Velocities in subject- and object-relative modes for four scenarios. The images in the 1st, 2nd, and 3rd rows are (1) the scenes where the camera captures the object, (2) the optical flows for the scenes, and (3) the visual velocities obtained by the optical flows.

ment, as $\mathbf{v}^{obj}[i,j] = \mathbf{f}[i,j] - \mathbf{b}$. Here, $\mathbf{b}$ is the background velocity, which can be approximated by averaging scene flows of non-salient areas, as:

$$\mathbf{b} = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(1 - s[i,j])\mathbf{f}[i,j]}{\sum_{i=1}^{H}\sum_{j=1}^{W}(1 - s[i,j])}. \tag{5.4}$$

Figure 5.4 visualizes the two visual velocities for the four scenarios. In the first scenario, the camera is stationary, only the object is moving, resulting in no difference between the two visual velocities. When the object is stationary, and only the camera moves, both the background and the object move in the video, but the object moves more since the object is closer. For the scenario, where the camera moves along the object, the object is regarded as stationary in the subject-relative mode. In contrast, the object seems to move to the right by the background movement in the object-relative mode. In the last scenario, the object and the camera move in opposite directions. At this time, the subject-relative velocity is very large, but in the object-relative mode, the movement is slightly offset by the background movement.

The velocity of the motion proxy, $\dot{\mathbf{q}}$, is calculated based on the the pixel

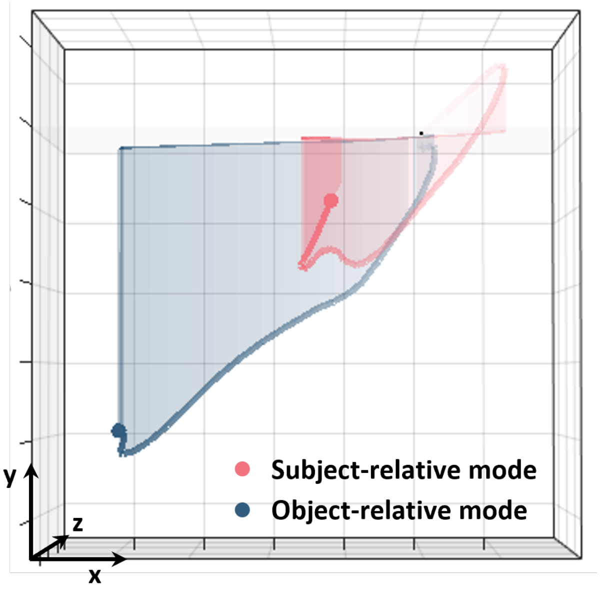


Figure 5.5: Visualization of the motion proxies in two visual modes for the clip in Figure 5.3.

velocities, $\mathbf{v}^{sbj}[i,j]$ or $\mathbf{v}^{obj}[i,j]$, and the saliency values, $s[i,j]$ by (5.3). During mute intervals, $\dot{\mathbf{q}}$ is forced to zero. In the case of shot transitions, it is set to the average of those of two adjacent frames. Finally, $\dot{\mathbf{q}}$ was smoothed using local regression [94] with a window size of 0.2 s.

Figure 5.5 illustrates the motion proxies computed from the two perception modes for the clip in Figure 5.3. In the clip, the skater and the background initially move to the right (indicated by the red color in the scene flow). Then the skater spins to the left (indicated by the blue color in the scene flow), in the opposite direction to the background movement. In subject-relative mode, the motion proxy moves following the movement in camera space. However, in object-relative mode, the motion proxy moves to the right and then to the left because the skater's movement is compensated by the background movement.

5.2.2 Gain Control and Filtering

To create a motion effect whose intensity matches well with the video, we applied four steps to the velocity of the motion proxy, $\dot{\mathbf{q}}$, through four steps: (1) modulation based on the object size, (2) filtering, (3) normalization, and (4) manipulation according to the genre.

The first step modulates the scale of $\dot{\mathbf{q}}$ based on the visual size of salient objects, which can be acquired by $\sum_{i=1}^{H} \sum_{j=1}^{W} s[i,j]$. This modulation is intended to create a stronger motion effect for larger objects. We tested motion effects created with three scaling conditions. The first condition was scaling with a constant value regardless of the object size. The second condition was to scale proportionally to the salient area. The third condition scaled the motion effect proportionally to the square root of the salient area. The motion effects obtained with constant scaling did not effectively emphasize the movements of larger objects. The two scaling options, based on the salient area, resulted in motion effects that were too weak for small objects, and excessively strong for large objects, especially in close-up scenes. In particular, the scaling proportional to the salient size produced a drastic motion effect when an object approached the camera in the depth direction. By combining the strengths of the three approaches, we designed a unified scaling method. The unified scaling is proportional to the square root of the salient area, but it limits the scale factor when the salient area is too small or large, as:

$$\dot{\mathbf{q}}' = \sqrt{S}\,\dot{\mathbf{q}},\ S = \min\left(\max\left(\sum_{i=1}^{H}\sum_{j=1}^{W} s[i,j], S_{min}\right), S_{max}\right) \tag{5.5}$$

where $\dot{\mathbf{q}}'$ is the velocity of the modulated motion proxy, S_{min} and S_{max} are the lower and upper bounds of the salient area, respectively. In this study, we set $S_{min} = 0.1$ and $S_{min} = 0.2$.

Next, we apply high-pass filters and low-pass filters to the modulated motion proxy, $\dot{\mathbf{q}}'$, resulting in the high-frequency velocity, $\dot{\mathbf{q}}'_h$, and the low-frequency velocity, $\dot{\mathbf{q}}'_l$. This filtering process is essential to generate reference inputs for our motion cueing algorithms, which operate on separate channels for high- and low-frequency motion components. We used a 5th-order Butterworth high-pass filter and a low-pass filter with a cutoff frequency of 1.0 Hz.

To ensure consistent and appropriate intensity levels of the generated motion effects across different videos, we normalize the two filtered motions, $\dot{\mathbf{q}}'_h$ and $\dot{\mathbf{q}}'_l$. This is because we have observed significant variations in the intensity of scene flows among different videos. Specifically, we divided $\dot{\mathbf{q}}'_h$ and $\dot{\mathbf{q}}'_l$ by the variance of $\dot{\mathbf{q}}'_h$ for each axis, because our motion cueing algorithms mainly focus on rendering the high-frequency motions of the motion proxy. Although normalization by the variance of the vector magnitude is a possible alternative, we opt for normalizing each axis individually to fully utilize the motion in all three degrees of freedom. This normalization allows the average velocity of each axis to be approximately 1.

Finally, we apply the signal manipulation to the normalized motions, $\dot{\mathbf{q}}'_h$ and $\dot{\mathbf{q}}'_l$, based on the genre of the video. To this end, we employ the following manipulation rule, as in Section 4.4.3:

$$\widehat{\dot{\mathbf{q}}}_h = \beta \|\dot{\mathbf{q}}'_h\|^{\alpha} \frac{\dot{\mathbf{q}}'_h}{\|\dot{\mathbf{q}}'_h\|}, \tag{5.6}$$

$$\widehat{\dot{\mathbf{q}}}_l = \beta \|\dot{\mathbf{q}}'_l\|^{\alpha} \frac{\dot{\mathbf{q}}'_l}{\|\dot{\mathbf{q}}'_l\|}, \tag{5.7}$$

where $\widehat{\dot{\mathbf{q}}}_h$ and $\widehat{\dot{\mathbf{q}}}_l$ is the attenuated or amplified high- and low-frequency velocity of the motion proxy, β is a scale factor to fine-tune the motion intensity, and α is a gain for attenuation and attenuation ($0 < \alpha$). When α is less than 1, weak motions are amplified more strongly than themselves while strong motions are attenuated weakly (see the green line in Figure 5.6). Conversely, when α is greater than 1, small movements become smaller and large movements become

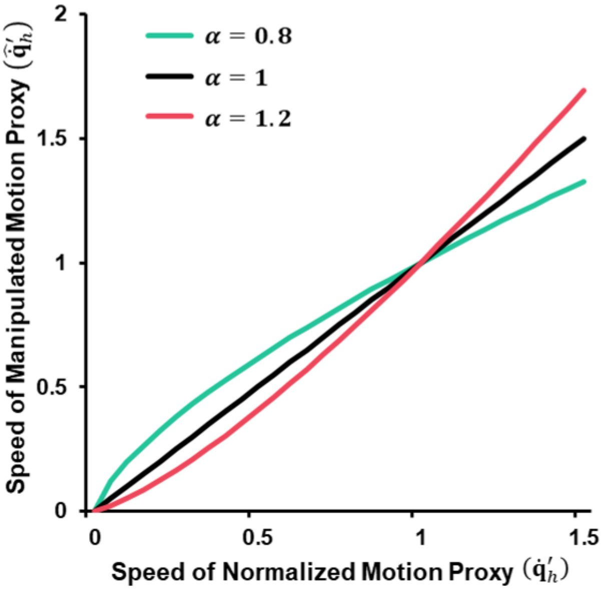
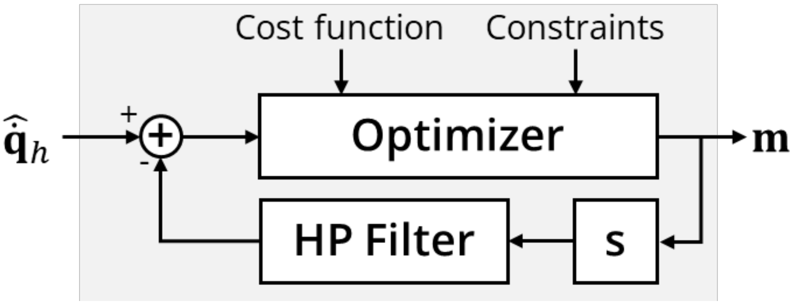


Figure 5.6: Plot of the manipulated speed of motion proxy depending on the exponent, α.

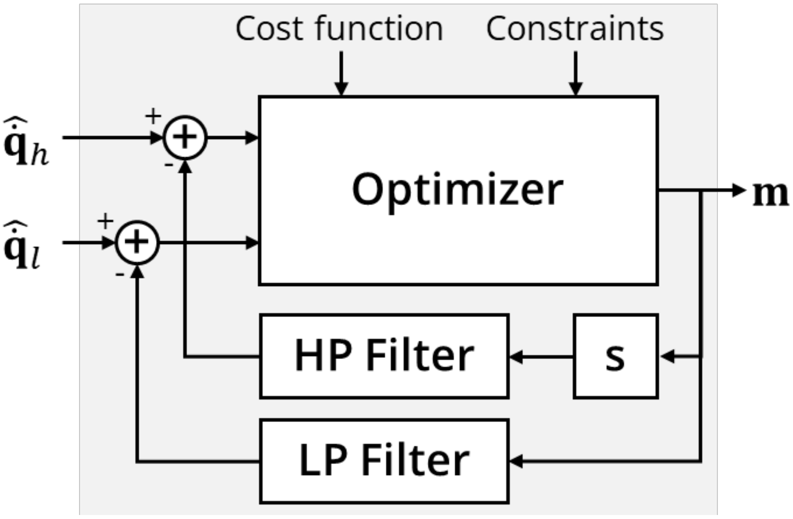
larger. In this study, we set $\alpha = 0.8$ for sport and dance videos to emphasize even the subtle movements. In contrast, for game videos, which typically involve fast and numerous movements, we set $\alpha = 1.2$ to attenuate small movements (see the red line in Figure 5.6). We set the value of β between 1.2 and 2.0, depending on the video.

5.2.3 Motion Cueing Algorithms

Integrating the velocity of the motion proxy and mapping it to a motion command may exceed the motion chair's workspace. To prevent this, we feed the low- and high-frequency motions of the motion proxy, $\dot{\mathbf{q}}'_h$ and $\dot{\mathbf{q}}'_l$ to an MCA to convert it to a motion command $\mathbf{m}$. Model predictive control (MPC) is an optimization-based control approach that considers process models and future trajectories while taking constraints into account. The optimization is governed by an objective function that minimizes the difference between the desired motion (reference) and the output produced by the process model. In this study,



(a) MPC with High-Pass Filters



(b) MPC with High-Pass Filters and Tilt-Coordination

Figure 5.7: Diagram for two MPCs.

we designed two MCAs by adopting the MPC. These MCAs generate a motion command, $\mathbf{m}$, that best aligns with the desired motion while considering the constraints imposed by the motion chair's workspace.

MPC with High-Pass Filters

This method only renders the high-frequency velocity of the motion proxy, $\dot{\mathbf{q}}_h'$, using high-pass filters, which remove the low-frequency energy that could potentially push the motion chair beyond its workspace limits (Figure 5.7a). That is, the motion command, $\mathbf{m}$, is optimized so that its high-frequency velocity is matched to $\dot{\mathbf{q}}'_h$. It solves the following optimization problem: Find $\mathbf{m}$ such that

$$\mathbf{m} = \arg\min_{\mathbf{m}} \quad \epsilon_h \|\widehat{\dot{\mathbf{q}}}_h - \dot{\mathbf{m}}_h\|^2 + \epsilon_0 \|\mathbf{m}\|^2 \tag{5.8}$$

$$\begin{aligned} \text{subject to} \quad & \dot{\mathbf{x}}_h(t) = \mathbf{A}_h \mathbf{x}_h(t) + \mathbf{B}_h \dot{\mathbf{m}}(t), \\ & \dot{\mathbf{m}}_h(t) = \mathbf{C}_h \mathbf{x}_h(t) + \mathbf{D}_h \dot{\mathbf{m}}(t), \\ & \dot{\mathbf{m}}(t) = \frac{\partial}{\partial t} \mathbf{m}(t), \\ & |\mathbf{m}(t)| \le \mathbf{m}_{max}, \ |\dot{\mathbf{m}}(t)| \le \dot{\mathbf{m}}_{max}, \end{aligned} \tag{5.9}$$

where $\dot{\mathbf{m}}$ and $\dot{\mathbf{m}}_h$ is the velocity of the motion command and its high-frequency component, $\mathbf{m}_{max}$ and $\dot{\mathbf{m}}_{max}$ are the maximum displacement and velocity of the motion chair, and ϵ_h and ϵ_0 are the weights of the cost function, respectively. Increasing the value of ϵ_h enhances the matching of high-frequency velocities between the motion proxy and the motion command. Conversely, a higher value of ϵ_0 promotes faster convergence of the motion command towards zero. In our study, we set $\epsilon_h = 1.0$ and $\epsilon_0 = 0.1$. $\mathbf{x}_h$ is the 15-by-1 state vector. $\mathbf{A}_h$, $\mathbf{B}_h$, $\mathbf{C}_h$, and $\mathbf{D}_h$ are the state-space model realizing the three 5th-order Butterworth high-pass filters with a cutoff frequency of 1 Hz, connected in parallel, and the dimensions of these matrices are (15×15), (15×3), (3×15), and (3×3), respectively.

Also, we considered incorporating computational models for the vestibular system [63], as an alternative to the high-pass filters. However, after testing, we found that using these models resulted in the loss of most high-frequency components that represent object motions in the generated motion effects. Therefore, we decided not to employ them in our approach.

MPC with High-Pass Filters and Tilt-Coordination

We recognized that there are certain scenarios in which both the high-frequency and low-frequency motions of an object need to be represented in the motion effects. In particular, in scenes where an object exhibits continuous motion in one direction while the camera follows it, the *MPC with high-pass filters* produces somewhat flat and static motion effects. However, the motion command may exceed the workspace limit of the motion chair when continuously rendering the low-frequency velocity. To overcome this, we adopted and modified tilt-coordination. Tilt-coordination is a technique for simulating sustained acceleration, such as gravity and centrifugal force, by tilting a motion chair in roll and pitch for a relatively long time. In our case, we made adjustments to this technique to simulate sustained velocity by aligning the low-frequency component of the motion command with the low-frequency velocity of the motion proxy (Figure 5.7a). For example, this modification shifts a motion chair to the right when an object continuously moves to the right. We also applied tilt-coordination to the heave, in addition to roll and pitch, to convey the sensation of rising and falling. The method solves the following optimization problem: Find $\mathbf{m}$ such that

$$\mathbf{m} = \arg\min_{\mathbf{m}} \quad \epsilon_h \|\widehat{\mathbf{q}}_h - \dot{\mathbf{m}}_h\|^2 + \epsilon_l \|\widehat{\mathbf{q}}_l - \mathbf{m}_l\|^2 + \epsilon_0 \|\mathbf{m}\|^2 \tag{5.10}$$

$$\begin{aligned}
\text{subject to} \quad & \dot{\mathbf{x}}_h(t) = \mathbf{A}_h \mathbf{x}_h(t) + \mathbf{B}_h \dot{\mathbf{m}}(t), \\
& \dot{\mathbf{m}}_h(t) = \mathbf{C}_h \mathbf{x}(t) + \mathbf{D}_h \dot{\mathbf{m}}(t), \\
& \dot{\mathbf{m}}(t) = \frac{\partial}{\partial t} \mathbf{m}(t), \\
& \dot{\mathbf{x}}_l(t) = \mathbf{A}_l \mathbf{x}_l(t) + \mathbf{B}_l \mathbf{m}(t), \\
& \mathbf{m}_l(t) = \mathbf{C}_l \mathbf{x}_l(t) + \mathbf{D}_l \mathbf{m}(t), \\
& |\mathbf{m}(t)| \leq \mathbf{m}_{max}, \ |\dot{\mathbf{m}}(t)| \leq \dot{\mathbf{m}}_{max},
\end{aligned} \tag{5.11}$$

where ϵ_l is the weight, which makes the low-frequency component of the motion command and the low-frequency velocity of the motion proxy matched in tilt-coordination ($\epsilon_l = 0.25$ in this study), $\mathbf{m}_l$ is the low-frequency component of the motion command. $\mathbf{x}_l$ is the 15-by-1 state vector, and $\mathbf{A}_l$, $\mathbf{B}_l$, $\mathbf{C}_l$, and $\mathbf{D}_l$ are the state-space model realizing the three 5th-order Butterworth low-pass filters with a cutoff frequency of 1 Hz, connected in parallel, and their dimensions are (15×15), (15×3), (3×15), and (3×3), respectively.

5.3 Performance Evaluation

We conducted a user study to compare the user experiences elicited by the motion effects synthesized by our algorithm and the conventional algorithm [10]. In this experiment, we classified audiovisual content into three categories and compared the benefits of the four options of our algorithms according to the category.

5.3.1 Methods

Devices

The motion chair (4DX, CJ 4DPLEX; Figure 1.1) used in the experiment had three DoFs for roll (±4°), pitch (±7°), and heave (±4 cm). The chair is for four people, and the participant sat in the second seat from the left during the experiment. 2D images were projected onto a 94-inch screen using a polarized projector (EB-W16SK, Epson Corp.). Sound was played through NS-150 5-channel home theater speakers (Yamaha Corp.).

Experimental Conditions

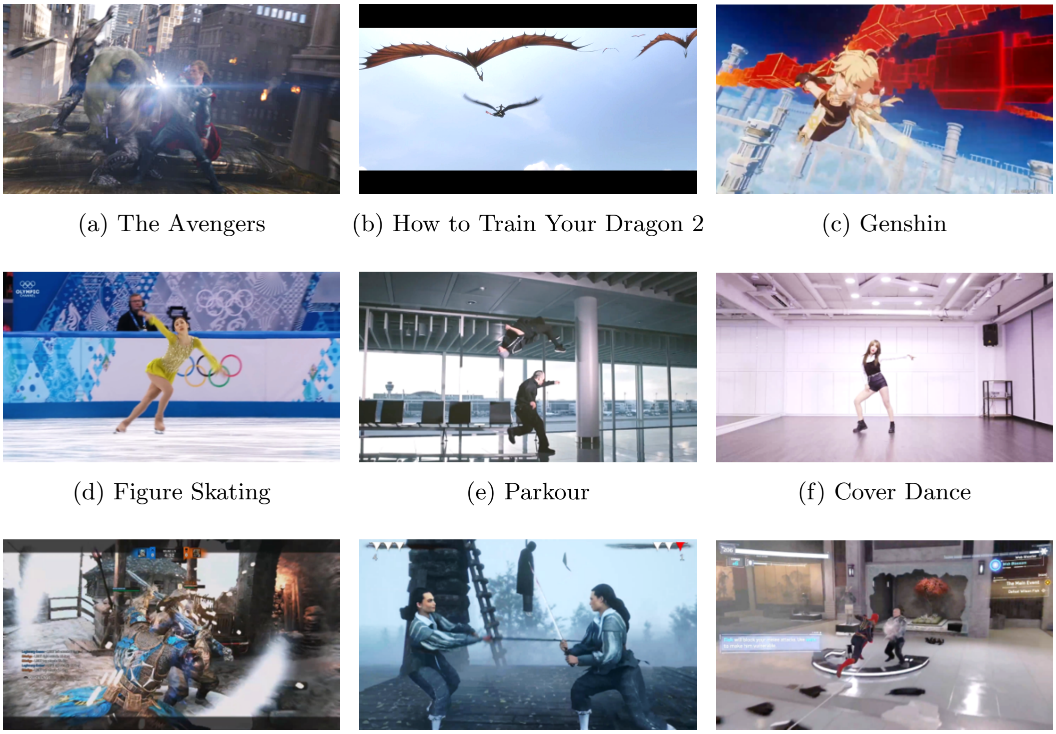
We classified audiovisual content, which has the potential to improve user experience with motion effects, into three categories: (1) Film and Game cinematic cut-scene, (2) Sport and Dance, and (3) Game. Three separate experiments, referred to as Exp. 1, Exp. 2, and Exp. 3, were conducted, each focusing on one of the three content categories.

For each of the three categories, we carefully selected and downloaded three videos from YouTube (Figure 5.8). To ensure dynamic and engaging content, we extracted approximately 90-s clips from each video, specifically choosing scenes that were not static. In Exp. 1, the two videos were from commercial films, The Avengers (*Avengers*) and How to Train Your Dragon (*Dragon*), and the other one was from a cinematic cut-scene of a game, Genshin Impact (*Genshin*). Exp. 2 used the videos of figure skating (*Skating*), parkour (*Parkour*), and cover dance performance (*Dance*). The videos for Exp. 3 contained the segments from fight-games, For Honor (*Honor*), Hellish Quart (*Quart*), and Spider-Man Remastered (*Spiderman*). The detailed description for the selected videos is summarized in Table 5.1.

For each of the nine videos, we used generated five sets of motion effects using our algorithm and conventional algorithm:

- SH: Motion effects synthesized by our algorithm (perception mode: subject-relative, MCA: *MPC with high-pass filters*).
- ST: Motion effects synthesized by our algorithm (perception mode: subject-relative, MCA: *MPC with high-pass filters and tilt-coordination*).
- OH: Motion effects synthesized by our algorithm (perception mode: object-relative, MCA: *MPC with high-pass filters*).
- OT: Motion effects synthesized by our algorithm (perception mode: object-relative, MCA: *MPC with high-pass filters and tilt-coordination*).
- Lee: Semi-automatically generated motion effects synthesized by [10].

In our four algorithms, we varied the gain factor, α in (5.7), according to the category of the video: $\alpha = 1.0$ for films and cinematic cut-scenes (Exp. 1), $\alpha = 0.8$ for sports and dances (Exp. 2), and $\alpha = 1.2$ for games (Exp. 3). The algorithms in Chapter III and IV can be considered as the baseline condition. However, they are not readily applicable to general content, because they require all motion trajectories of objects in a scene. Therefore, we selected Lee's algorithm [10] as the baseline condition and used CSRT (Channel and Spatial Reliability Tracking) tracker [95] to estimate 2D position of an object. For this condition, we manually annotated the initial bounding box of the main object, which served as the basis for generating the motion effect. Whenever the tracking failed, we re-annotated the bounding box. Then, the motion effects of Lee were generated by applying a washout filter to the 2D velocity of the object and mapping them to roll and pitch commands. Note that motion effects in all conditions are scaled to have intensities.



(a) The Avengers (b) How to Train Your Dragon 2 (c) Genshin

(d) Figure Skating (e) Parkour (f) Cover Dance

(g) For Honor (h) Hellish Quart (i) Spider-man Remastered

Figure 5.8: Video clips used in the user study. The videos of (a)-(c), (d)-(f), and (g)-(i) were used in Exp. 1, 2, and 3, respectively.

Table 5.1: Descriptions for the videos used in the user study.

Video tag	Description	Source	Duration
Avengers	Six superheroes fight and defeat numerous aliens.	1'42"–3'14" in [96]	92 s
Dragon	A character rides a dragon and flies freely through the sky.	0'16"–1'44" in [97]	88 s
Genshin	Two characters fight the witch in the air, but one of them disappears by the witch's spell.	0'21"–1'40" in [98]	79 s
Skating	A figure skater performs to the song 'Send in the Clown'.	0'42"–2'12" in [99]	89 s
Parkour	A man evades inspectors and goes through airport security performing parkour.	1'08"–2'31" in [100]	83 s
Dance	A woman dances to a song 'Solo' by Jennie.	0'39"–2'02" in [101]	83 s
Honor	Two soldiers, one with a sword and the other with two axes, fight in third-person view for two rounds.	2'25"–3'45" in [102]	79 s
Quart	Two soldiers with a long sword fight for four rounds in side view.	0'09"–1'19" in [103]	70 s
Spiderman	Spider-Man beat a boss while firing a web and flying in third-person view.	18'28"–19'57" in [104]	89 s

Procedure

For familiarization, participants had a practice session in which they experienced all five motion effects for one of the three videos. The main session comprised three blocks of trials, one for each of the three videos. Each block consisted of five trials in which five motion effects were presented with the video. In each trial, participants experienced the pair of a video and a motion effect twice. The order of the video was randomized across participants, and the order of the motion effects per video was balanced across participants using the balanced Latin square. Participants took a 3-min break after the practice session and each block in the main session.

After experiencing each motion effect, participants answered a questionnaire that included the following seven questions ($\neg$ indicates a negative question):

Q1. Harmony: The motion effect matched the movements in the video.

Q2. $\neg$ **Fatigue:** I felt tired after experiencing the motion effect.

Q3. Enjoyment: I enjoyed watching the video with the motion effect.

Q4. $\neg$**Causality:** I experienced motion effects that I did not understand why they had been provided.

Q5. $\neg$ **Distraction:** I was distracted from watching the video by the motion effect.

Q6. Preference: I liked the motion effect.

Q7. Free Comment: Please leave a comment regarding the motion effect.

For each question from Q1 to Q6[1], participants expressed the extent of the agreement by entering positive numbers between 0 and 100 using a keyboard. To assist participants in understanding the scale, descriptors were provided at the top of the questionnaire for every 25-point interval: 0–strongly disagree, 25–disagree, 50–neutral, 75–agree, and 100–strongly agree. For Q7, participants entered a response using a keyboard. In the data analysis, we inverted the scores of three questions (Q2, Q4, and Q5) so that a high score indicates a better experience in every measure.

[1] Three out of the six questions were designed to imply negative meanings for balancing.

Table 5.2: Participant information for three experiments.

Experiment	Category	Participants	Mean Age (SD)
Exp. 1	Film and Game cinematic cut-scene	20 (6F, 14M)	24.1 ± 2.9
Exp. 2	Sport and Dance	20 (6F, 14M)	21.8 ± 2.7
Exp. 3	Game	20 (7F, 13M)	19.9 ± 1.8

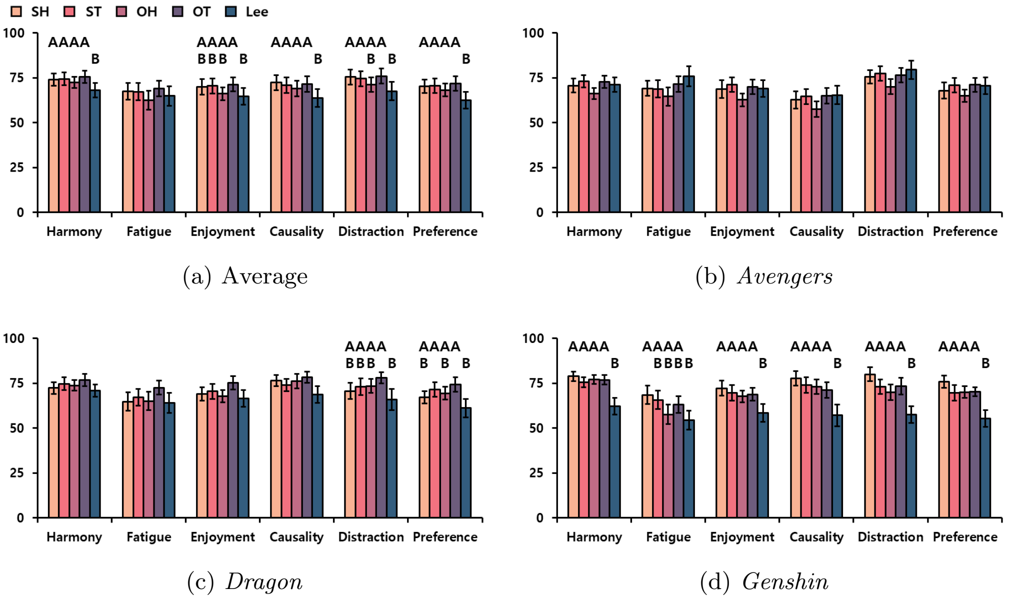
Participants

Each of the three experiments involved the participation of 20 people, as detailed in Table 5.2. None of them reported known sensorimotor disorders. The experiment took approximately 90 min per participant. The participants were paid approximately USD 23 after the experiment.

5.3.2 Results and Discussion

The experimental results for Exp. 1, 2, and 3 are shown in Figure 5.9, 5.10, and 5.11, respectively. For each experimental result, We performed a two-way repeated-measures ANOVA on every question from Q1 to Q6 using *Motion Effect* and *Video* as the independent variables. We then used the SNK test for post-hoc multiple comparisons.

In Exp. 1, *Motion Effect* was statistically significant in five measures; Q1 (Harmony; $F(4, 76) = 4.12$, $p = .0045$), Q3 (Enjoyment; $F(4, 76) = 3.28$, $p = .0155$), Q4 (Causality; $F(4, 76) = 4.85$, $p = .0016$), Q5 (Distraction; $F(4, 76) = 6.03$, $p = .0003$), and Q6 (Preference; $F(4, 76) = 4.10$, $p = .0046$); but not in Q2 (Fatigue; $F(4, 76) = 2.32$, $p = .0640$). *Video* was not significant for every measure. We observed significant interactions between *Motion Effect* and *Video* in five measures; Q1 (Harmony; $F(8, 152) = 3.98$, $p = .0003$), Q3 (Enjoyment; $F(8, 152) = 2.01$, $p = .0489$), Q4 (Causality; $F(8, 152) = 3.53$, $p = .0009$), Q5 (Distraction; $F(8, 152) = 3.96$, $p = .0003$), and Q6 (Preference; $F(8, 152) = 3.26$,



(a) Average

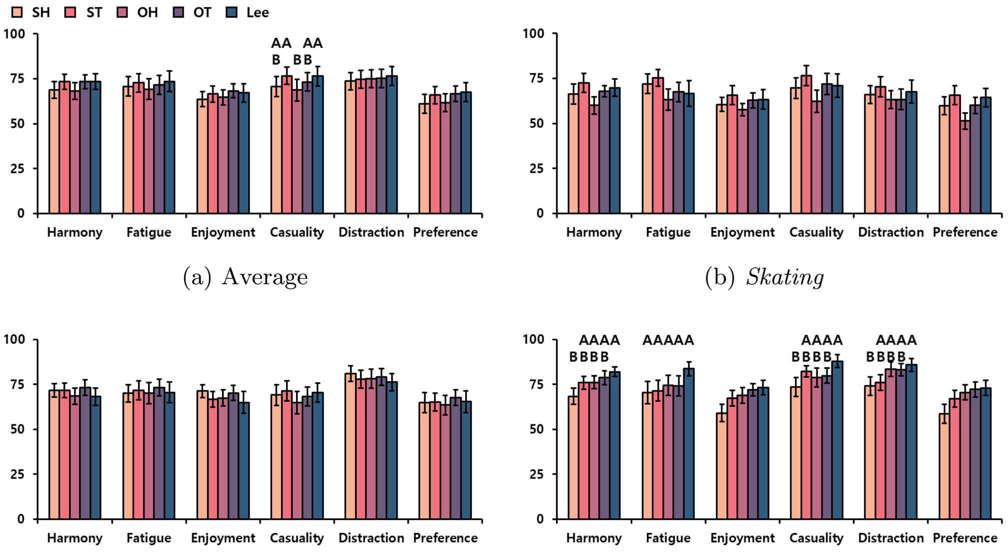
(b) *Avengers*

(c) *Dragon*

(d) *Genshin*

Figure 5.9: Experimental results for Exp. 1. Error bars represent standard errors. Motion effect sets marked with the same letters indicate that they did not show statistically significant differences by the SNK tests. The scales of negative questions are inverted so that a higher score indicates a better performance.

$p = .0019$). Our four methods, SH, ST, OH, and OT, received higher scores than Lee in Q1 (Harmony), Q2 (Causality), and Q6 (Preference). This may be because our algorithms inclusively respond to various components, including object motions, camera motions, and visual effects. This trend was particularly evident in *Genshin*, which featured numerous explosions and particle effects. Regarding this, six participants expressed their satisfaction with our algorithms, mentioning that our algorithms produced the motion effects that they expected. For example, one participant stated "The vibrating motion effect when the portal opened was appropriate." (OT). Conversely, six participants criticized Lee for the absence of motion effects for special effects, e.g., "There was no motion effect for the special effects, so it was less immersive." (Lee). However, in the case of *Avengers*, there were no significant differences because the movement of one superhero was noticeable in most shots. In *Dragon*, the algorithms with tilt-coordination, ST



(a) Average

(b) *Skating*

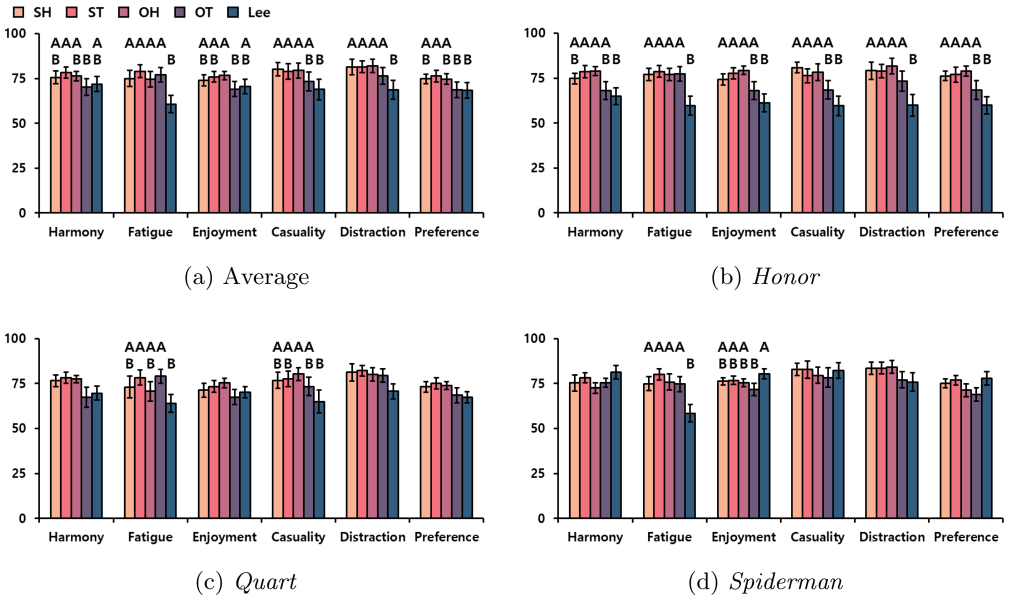
(c) *Parkour*

(d) *Dance*

Figure 5.10: Experimental results for Exp. 2. Error bars represent standard errors. Motion effect sets marked with the same letters indicate that they did not show statistically significant differences by the SNK tests. The scales of negative questions are inverted so that a higher score indicates a better performance.

and OT, were preferred over Lee, likely due to the longer duration of shots and the continuous movement of the main object, the dragon, in the sky. In all three videos, there was no significant difference in Q2 (Fatigue), because all the motion effects were scaled to have similar intensities.

In Exp. 2, *Video* was significant for most measures; Q1 (Harmony; $F(2, 38) = 5.03$, $p = .0115$), Q3 (Enjoyment; $F(2, 38) = 3.35$, $p = .0456$), Q4 (Causality; $F(2, 38) = 8.52$, $p = .0009$), Q5 (Distraction; $F(2, 38) = 12.24$, $p < .0001$), and Q6 (Preference; $F(2, 38) = 3.38$, $p = .0446$). However, *Motion Effect* was only statistically significant in Q4 (Causality; $F(4, 76) = 3.49$, $p = .0114$) and there were no significant interactions in any measures. That is, all the algorithms received similar scores. This result can be attributed to the fact that all three videos focused on the movement of a single person. In such cases, our algorithms and Lee's algorithm inherently produce almost identical motion effects. In par-



(a) Average

(b) *Honor*

(c) *Quart*

(d) *Spiderman*

Figure 5.11: Experimental results for Exp. 3. Error bars represent standard errors. Motion effect sets marked with the same letters indicate that they did not show statistically significant differences by the SNK tests. The scales of negative questions are inverted so that a higher score indicates a better performance.

ticular, in *Dance*, there were few differences between the motion profiles by the two visual perception modes (SH vs. OH and ST vs. OT), since the camera was static. Moreover, it seems that Lee scored higher than our algorithms because of the manual process to ensure accurate object tracking. For *Skating*, 17 people pointed out that the motion effects of our four algorithms were too intense when spinning, e.g., “Even if the spinning wasn’t that strong, the motion effect was too strong.” (ST), while no one did not point out for Lee. The spinning was mainly expressed as roll motion in Lee and as pitch motion in our algorithms, respectively, but our algorithm produced more intense motion effects, as depicted in Figure 5.12. We speculate that this interval contributed the most to the lower scores of our algorithms. We believe that this issue could be improved by attenuating such high-frequency motions, as supported by one participant’s comment: “I felt that almost all motion effects matched well with the video except for the

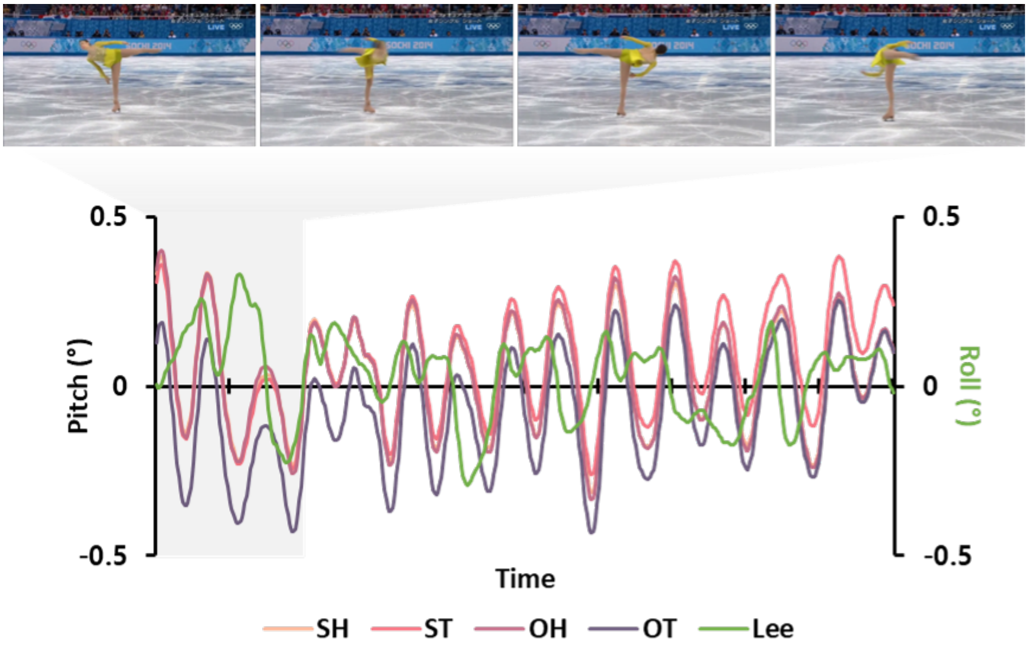


Figure 5.12: A scene of a skater spinning in *Skating* and its motion effects.

spinning.” (ST).

In Exp. 3, *Motion Effect* was significant for every measure; Q1 (Harmony; $F(4, 76) = 3.35$, $p = .0139$), Q2 (Fatigue; $F(4, 76) = 23.98$, $p < .0001$), Q3 (Enjoyment; $F(4, 76) = 3.19$, $p = .0177$), Q4 (Causality; $F(4, 76) = 5.48$, $p = .0006$), Q5 (Distraction; $F(4, 76) = 5.34$, $p = .0008$), and Q6 (Preference; $F(4, 76) = 3.60$, $p = .0096$). *Video* was significant only for Q4 (Causality; $F(2, 38) = 3.83$, $p = .0306$). We found significant interactions between *Motion Effect* and *Video* in four measures; Q1 (Harmony; $F(8, 152) = 2.70$, $p = .0084$), Q3 (Enjoyment; $F(8, 152) = 3.99$, $p = .0003$), Q4 (Causality; $F(8, 152) = 2.23$, $p = .0284$), and Q6 (Preference; $F(8, 152) = 3.58$, $p = .0008$). Our algorithms, SH, ST, and OH, showed better performance in Q2 (Fatigue), Q4 (Causality), Q5 (Distraction), and Q6 (Preference). This suggests that our algorithms performed well in situations where multiple characters were moving rapidly. In contrast, Lee’s algorithm struggled in such scenarios, as parts or all of the objects were frequently out of the frame. Notably, Lee recorded a low score of Q2 (Fatigue) in *Spiderman*, likely because the jitters of the tracked bounding box were represented as a rough motion effect. Meanwhile, OT performed worse than Lee in terms of Q1 (Harmony) and Q3 (Enjoyment). This may be because viewers tend to focus on short and rapid movements of characters in games, so the motion effects representing camera motion and tilting were not matched to participants' expectations. Furthermore, 16 participants pointed out the need to emphasize the motion effects during hitting in all three videos, e.g., "I liked the motion effects for characters interacting with the background, but I was disappointed with the motion effects for hitting between characters." (*Spiderman*).

It is interesting to note that some participants expressed a preference for the amplification of motion effects in our algorithms, e.g., "The dynamics of this motion effect were the best. (OT, *Spiderman*)", while there were no comments on the attenuation. In addition, one participant reported, "I could distinguish the difference in motion effects depending on the visual size of the objects."

In conclusion, our algorithms generally performed better than the conventional algorithm, particularly in videos with multiple objects or diverse components. Our algorithms also offer the advantage of overcoming the limitations associated with object-tracking-based methods, such as objects leaving the frame. Although our four algorithms were hardly distinguishable, there were situations where each algorithm had advantages. Therefore, 4D designers can leverage our algorithmic options depending on the specific requirements of a given scenario, thereby creating more convincing and tailored motion effects.

5.4 Comparison of Our Algorithms

In the previous section, we confirmed that our algorithm demonstrates superior performance compared to the conventional algorithm. However, we rarely observed significant differences among the four options of our algorithm. Therefore, we conducted an additional experiment comparing these options using clips displaying distinct profiles of the four motion effects.

5.4.1 Methods

Participants

Sixteen people (6 females and 10 males, 23–30 years old with an average age of 26.69) participated in this experiment. None of them reported known sensorimotor disorders. The experiment took approximately 60 min per participant.

Devices

We used the same devices as in Section 5.3.1.

Experimental Conditions

We selected four clips from the nine videos used in Section 5.3.1: *Avengers*, *Dragon*, *Skating*, and *Spiderman*. We cropped the video to about 10 to 20 s where the motion profiles of the four options are distinguished. Their durations are as follows. *Avengers*: 2'27"–3'37" (10 s), *Dragon*: 0'16"–0'31" (15 s), *Skating*: 1'13"–1'32" (19 s), and *Spiderman*: 18'40"–19'00" (20 s).

For each of the four clips, we generated four sets of motion effects using the four options of our algorithm:

- SH: Motion effects synthesized by our algorithm (perception mode: subject-relative, MCA: *MPC with high-pass filters*).
- ST: Motion effects synthesized by our algorithm (perception mode: subject-relative, MCA: *MPC with high-pass filters and tilt-coordination*).
- OH: Motion effects synthesized by our algorithm (perception mode: object-relative, MCA: *MPC with high-pass filters*).

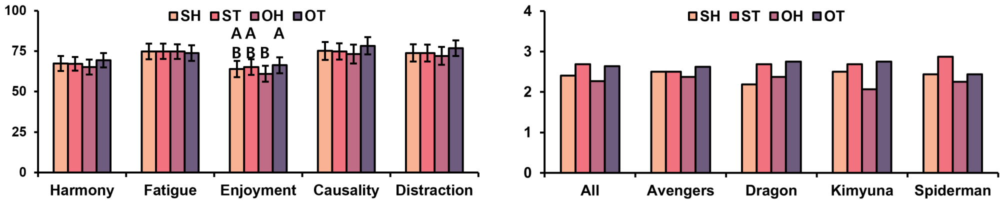
- OT: Motion effects synthesized by our algorithm (perception mode: object-relative, MCA: *MPC with high-pass filters and tilt-coordination*).

In this experiment, we increased the weight for tilt-coordination, ϵ_l in (5.10), from 0.25 to 0.75 to enhance its effect. All other parameters were set to the same values as in the previous experiment.

Procedure

The experiment was conducted following a similar procedure to the previous experiment. For familiarization, participants had a practice session in which they experienced all four motion effects for one of the four videos. The main session comprised four blocks of trials, one for each of the four videos. Each block consisted of four trials in which four motion effects were presented with the video. In each trial, participants experienced the pair of a video and a motion effect three times. The order of the video was randomized across participants, and the order of the motion effects per video was balanced across participants using the balanced Latin square. Participants took a 3-min break after the practice session and each block in the main session.

After experiencing each motion effect, participants answered the same questionnaire described in Section 5.4.1. In this questionnaire, we modified Q6 (Preference) to "Rank the four motion effects in the order of preference." (Preference in rank). This was done to differentiate clearly the ratings among the four options. For Q1–Q5, participants' response methods to the questions were also the same with the previous experiment. For Q6, participants ranked the four motion effects in descending order of preference without ties. In the data analysis, we inverted the scores of three questions (Q2, Q4, and Q5) so that a high score indicates a better experience in every measure. Also, the rank data obtained from Q6 (Preference) was inverted to a preference score by (6-*rank*).



(a) Average scores

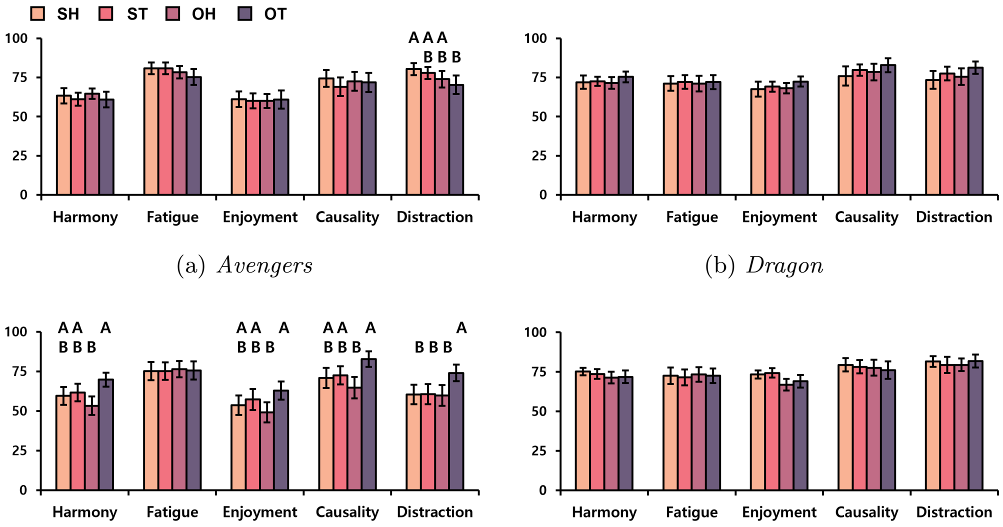
(b) Average rank

Figure 5.13: Experimental results for comparing the four options of our algorithm Exp. 1. Error bars represent standard errors. Motion effect sets marked with the same letters indicate that they did not show statistically significant differences by the SNK tests. The scales of negative questions are inverted so that a higher score indicates a better performance.

5.4.2 Results and Discussion

The experimental results are shown in Figure 5.13 and 5.14. We performed a two-way repeated-measures ANOVA on every question from Q1 to Q5 using *Motion Effect* and *Video* as the independent variables. We then used the SNK test for post-hoc multiple comparisons. *Motion Effect* was statistically significant for only Q3 (Fatigue; $F(3, 45) = 3.33, p = .0278$). *Video* was significant for Q1 (Harmony; $F(3, 45) = 10.09, p < .0001$), Q3 (Fatigue; $F(3, 45) = 6.96, p = .0006$), and Q5 (Distraction; $F(3, 45) = 8.50, p = .0071$). We observed significant interactions between *Motion Effect* and *Video* in two measures; Q4 (Causality; $F(9, 135) = 2.20, p = .0259$) and Q5 (Distraction; $F(9, 135) = 2.66, p = .0071$). The scores of Q6 (Preference) are ordinal data. We only tested the main factor of *Motion Effect* and *Video* over all videos and for each video using a Friedman test. However, none of the significant effects of *Motion Effect* and *Video* were observed for all videos, and each video.

Overall, SH received slightly higher scores than the others, but statistically significant differences among the four options were hardly identified. However, we could observe some significant differences among the four methods for in result on



(a) *Avengers*

(b) *Dragon*

(c) *Skating*

(d) *Spiderman*

Figure 5.14: Experimental results for each video. Error bars represent standard errors. Motion effect sets marked with the same letters indicate that they did not show statistically significant differences by the SNK tests. The scales of negative questions are inverted so that a higher score indicates a better performance.

each video, particularly in clips of *Avengers* and *Skating*, where the camera follows a main character with sustained movements. In *Skating*, OT generally showed better performance than the other methods in terms of Harmony ($F(3, 45) = 3.14, p = .0345$), Enjoyment ($F(3, 45) = 2.95, p = .0426$), Causality ($F(3, 45) = 4.05, p = .0125$), and Distraction ($F(3, 45) = 5.12, p = .0039$). We presume that OT received high scores because it effectively captured the camera's low-frequency movement as well as the skater's movements. Six participants commented on this, e.g., "The motion effect matched well both the movements of the camera and the skater." However, in *Avengers*, despite the camera continuously following a moving subject, OT was rated more distracting than SH ($F(3, 45) = 3.07, p = .0374$). This is because high-frequency motion is relatively less expressed by the term for tilt-coordination in (5.10). Four people pointed out the weak intensity of OT, e.g., "The motion effects were too weak when the character was moving."

In conclusion, we recommend adopting SH in general scenes. But, we expect that OT may have a merit in scenes where the camera continuously follows a moving object if the algorithm is tuned to enhance tilt-coordination and maintain the expression of high-frequency motions. Designers will be able to choose the appropriate method according to the scene.

5.5 General Discussion

In this study, we focused on generating object-based motion effects by considering two visual modes. This allowed us to create motion effects that capture both object and camera motions. Consequently, there is no need to generate object- and camera-based motion effects independently. Further research can explore the comparison between our algorithm and an approach that combines these two types of motion effects.

Our approach based on combining the movements of pixels overcomes the limitations of the previous approach based on object tracking. Object tracking requires manual annotation, and it has limitations when dealing with occlusion or when objects partially or completely leave the frame. In contrast, our approach is free from these limitations and produces more reliable results. Also, our method is robust to errors in scene flow, since the motion proxy aggregates the scene flow, thereby it is not significantly affected by the errors. Moreover, as computer vision techniques advance, our algorithm can further benefit from these advancements.

Our approach can create motion effects matching the overall visual flow of a scene. However, depending on the situation, a motion effect focusing on a single object could be preferred, which can be achieved by utilizing object segmentation technology. For example, in scenarios where audiences are expected to relate to a specific character, especially in games, motion effects can be selectively provided to emphasize the character's movement for viewers to immerse in the character.

5.6 Conclusion

In this chapter, we have presented a fully automatic algorithm for extracting motion information from scene segments and generating motion effects for various scene components, including objects, particles, and camera motion. Unlike the algorithms discussed in previous chapters (Chapter III and IV), which relied on given object motion trajectories, this algorithm utilizes scene flow estimation and saliency detection techniques to extract motion information from the scene. This approach overcomes the limitations of the conventional algorithms based on object-tracking, such as occlusion and objects leaving the frame. Additionally, it allows for detecting key components and extracting their movements without requiring human assistance, thereby facilitating the efficient production of object-based motion effects. For the comprehensive representation of scene component movements, we compressed the extracted movements into a single point, extending the computational definition of motion proxy from multiple articulated bodies. To the best of our knowledge, this work is the first attempt to generate convincing motion effects for immersive 4D experiences through a fully automatic process.

VI. Guidelines for Motion Effects

In this chapter, we present guidelines for 4D effect designers in authoring object-based motion effects on the basis of our observations and qualitative feedback from participants in user studies, throughout our research.

Express various components. Scenes consist of numerous components, including objects, particles, visual effects, and a camera, and their movements can all be portrayed through motion effects. Motion effects should encompass and appropriately respond to the movements of these components. One effective approach is to combine the overall movements instead of solely focusing on the movement of an individual component, as in our approach.

Focus on users' attention. Various factors can serve as cues for motion effects, but people expect motion effects to be presented for the cues that catch their attention. This is supported by the finding that methods emphasizing salient movements were preferred by participants. However, when unexpected motion cues were provided, even if they were consistent with the content, participants perceived them as distracting. In contrast, when expected motion cues were not provided, participants evaluated it as mismatched with the content, resulting in decreased satisfaction. Therefore, it is crucial to identify what users are likely to attend in the scene and align it with the motion effects.

Make it dynamic but not tiring. As mentioned in Section 4.3, participants preferred dynamic motion effects, but they became tired when the effects lasted for a long time, which decreased user satisfaction. Thus, intense motion effects emphasizing fast movements should be carefully used, especially for movies and games with a long duration. To prevent abrupt movements, the motion intensity can be adjusted by scaling and applying the attenuation rule in Eq. (4.14). We set the exponent γ to 0.7 in the experiments, but using a lower value can create smoother motion effects with less deviations in motion speed. Of course, a higher value can be advantageous for more dynamic situations.

Modulate it according to context. As mentioned earlier, the motion intensity should be carefully tuned according to the context of audiovisual content. Dynamic motion effects can be preferred in fighting scenes, while gentle ones are in dance scenes. Moods and circumstances are also important. For example, to emphasize a tense situation, the motion effect can amplify even the fine movements in the scene. Besides, motion effects can be leveraged to augment a character's properties, such as strengths and weights. A motion effect weaker than a visual movement can emphasize such properties as the weakness and lightness of an object. In contrast, a stronger motion effect can accentuate the strength or the heaviness.

Utilize both visual and auditory information. The inclusion of sound information could enhance the generation of motion effects that emphasize specific events. As mentioned earlier in Section 5.3.2, motion effects need to be amplified for the events related to hits and collisions. While it is challenging to detect hits and collisions using visual information alone, incorporating sound information could facilitate easier detection.

Tilt it based on shot length. Motion effects should be tailored according to the length of shots. For example, the option of tilt-coordination was favored for longer shots featuring sustained movements in Section 5.4. However, this approach may not be ideal, especially for shorter shots (less than 3 s). The tilt cause discomfort and be perceived as irrelevant to the content. Hence, using different motion cueing options based on the shot length would be beneficial.

Utilize various animation data. The algorithm in Chapter IV can generate object-based motion effects by taking various animation data as the input. The animation data can be collected using a motion capture system or extracted from animation databases, e.g., the CMU mocap database [1]. The algorithm can use such animation data to yield motion effects for specific movements, such as walking and gestures. Putting them together can lead to a motion effect library that provides a huge amount of motion effect segments. These motion effect segments can be transformed and stitched into a motion effect for an articulated body to fit the content. Also, a motion effect for the movement of multiple objects can be created by blending and merging the motion effects of the individual objects. This approach can be particularly effective in generating motion effects for the movements that are not shown visually in the content but can be inferred, e.g., the locomotion in a first-person view. Lee et al. [60] approached this problem by collecting gait motion data using a motion sensor and generating motion effects for the character's gait that is not visually revealed in first-person shooting games. As an alternative, our algorithm can be applied using the animations of human locomotion. In our test, the synthesized motion effects for running animations showed similar motion profiles (Fig. 6.1).

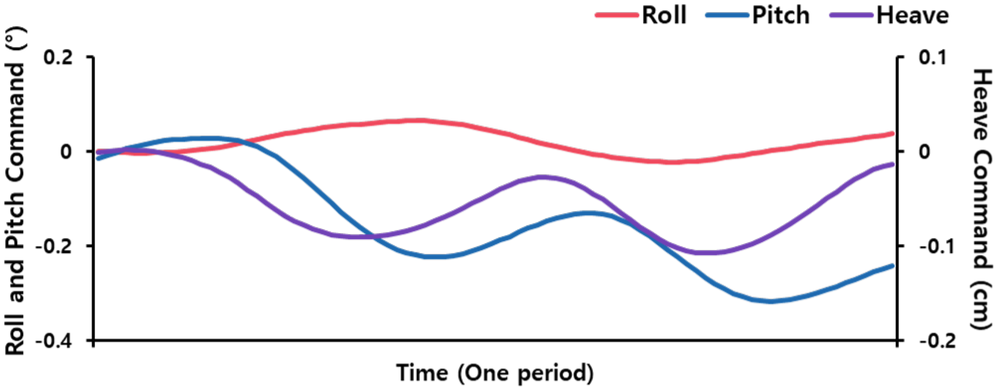


Figure 6.1: Motion effect generated by our algorithm for a running animation. The input animation is 09_11 from the CMU mocap database [1].

VII. Conclusion

In this thesis, we have presented innovative methods for automatically generating object-based motion effects to enhance 4D experiences. Finally, in this chapter, we summarize our contributions and propose future works that should be pursued.

7.1 Contributions

The contribution of my thesis can be summarized as follows:

- We proposed automatic algorithms for generating motion effects for various objects, aiming to enhance the immersive media viewing experience. Unlike previous approaches that focused on rendering the 2D velocity of a single object, our method is the first to offer object-based motion effects for diverse forms of objects.

- We introduce the concept of motion proxy to represent the complex movements of objects with high DoFs using a 3-DoF motion chair. We first defined the concept of a motion proxy as a combination of the direction and position of a rigid body object. Then we extended this concept to encompass multiple articulated bodies and scene components. The motion proxy captures the essential movements of objects and scene elements, providing a condensed representation of their motions within a scene, enabling the generation of motion effects for a wide range of objects.

- Our approach extends beyond motion effects and can be applied to various sensory stimuli. By utilizing vibrator arrays or force feedback devices, we can provide users with a sense of movement corresponding to the motion proxy's movements. We expect such applicability to be a cost-effective alternative to expensive motion platforms.

- We design robust algorithms for automatic motion effect generation, considering both users' visual perception and vestibular sensory cues, as well as hardware limitations. By incorporating insights from human perception mechanisms and accounting for hardware constraints, our algorithms create plausible object-based motion effects.
- Our algorithms can deliver a superior user experience compared to the existing method. By going beyond the limitations of the existing algorithm based on object tracking, our approach enables a comprehensive representation of complex 3D movements within scenes, resulting in a more immersive and captivating experience.

Our research contributes to enhancing the overall media viewing experience and paves the way for future developments in this area.

7.2 Future Works

While this thesis has made significant contributions to generating object-based motion effects, there are several avenues for future research and development. While we utilized saliency detection to identify major objects in the scene, we did not differentiate between individual objects during processing. By incorporating techniques such as object segmentation, we could selectively provide motion effects more focused on specific objects. For example, for TPS games, we could tailor the motion effects to react only to the movements of the player-controlled character. Furthermore, our algorithm primarily relied on visual information to generate motion effects. However, incorporating auditory information could create even more convincing object-based motion effects. Sound can provide valuable cues for objects' movements. For instance, hitting and collisions could be emphasized by amplifying the motion effects in synchronization with prominent sound events. By combining visual and auditory information, we can enhance the overall realism and impact of the motion effects.

요 약 문

4D 컨텐츠는 시청각 컨텐츠와 함께 모션, 진동, 열, 바람과 같은 다양한 감각 효과를 활용하여 사용자의 멀티미디어 경험을 향상시킨다. 가장 빈번히 사용되는 4D 효과 중의 하나는 '물체 기반의 모션 효과'이다. 물체 기반의 모션 효과란 장면에서 주요한 물체의 움직임을 강조하기 위하여 사용자가 앉아있는 의자(모션 플랫폼)를 움직이는 것을 말한다. 4D 컨텐츠에서는 한 장면에 여러 물체가 동시에 등장하기도 하며, 각 물체는 여러 부위에 의해 높은 자유도로 움직이기도 한다. 반면, 극장에서 사용되는 모션 의자는 대부분 제한된 동작 범위와 3자유도만을 가지고 있기 때문에 그러한 물체의 움직임을 표현하는데 한계가 있다. 따라서, 물체 기반의 모션 효과를 만들어내는 것은 매우 어렵다.

이러한 모션 효과의 생산을 가속하기 위해 영상을 분석하여 모션 효과를 자동으로 생성하는 기술들이 제안된 바 있지만, 이 방법들은 하나의 강체의 병렬이동(translation)에 대해서만 집중하여 모션 효과를 생성하고자 하였다. 본 논문에서는 다자유도의 복잡한 움직임들을 정교하게 표현하는 모션 효과를 자동으로 생성하는 알고리즘을 제시한다. 이를 위해, 모션 프록시(Motion proxy)라는 새로운 개념을 제시하여 다자유도의 움직임을 하나의 3자유도 움직임으로 축약하고 그것을 모션 효과로 변환한다.

먼저, 우리는 강체에 대한 모션 효과 생성 알고리즘을 개발하였다. 우리는 강체의 크기에 따라 병렬이동과 회전을 조합하여, 강체의 6자유도 움직임을 3자유도의 모션 프록시로 표현하고자 하였다. 다음으로, 우리는 다중 다관절체(Articulated body)의 여러 부위의 복합적인 움직임을 모두 표현하는 모션 효과 생성 알고리즘을 개발하였다. 우리는 다관절체의 움직임을 포괄적으로 표현할 수 있도록 수많은 부위의 움직임은 각 부위의 속도와 크기에 따라 조합하여 모션 프록시를 계산하였

다. 마지막으로, 우리는 움직이는 강체, 다관절체, 입자의 움직임 등의 장면의 모든 움직이는 요소로부터 모션 효과를 완전 자동으로 생성하였다. 특히, 이 과정에서 장면의 움직임을 자동으로 추출하기 위하여 장면 흐름 추정(Scene flow estimation)과 주목도 탐지(Saliency detection)과 같은 컴퓨터 비전 기술을 활용하였고, 추출된 움직임을 중요도에 따라 조합하여 모션 프록시를 계산하였다.

각 단계에서 우리는 세부적인 알고리즘과 최적의 파라미터를 결정하기 위해 여러 번의 사용자 평가 실험을 수행하여 최적의 알고리즘을 구현하였다. 사용자 평가 결과, 우리가 제안하는 알고리즘이 기존의 기술보다 뛰어난 수준의 물체 기반 모션 효과를 자동으로 생성해내고 있음을 확인할 수 있었다.

References

[1] CMU. Carnegie-mellon mocap database, 2003.

[2] Douglas Paulo De Mattos, Débora C. Muchaluat-Saade, and Gheorghita Ghinea. Beyond multimedia authoring: On the need for mulsemedia authoring tools. *ACM Computing Surveys*, 54(7), 2021.

[3] Alexandra Covaci, Longhao Zou, Irina Tal, Gabriel-Miro Muntean, and Gheorghita Ghinea. Is multimedia multisensorial?-a review of mulsemedia systems. *ACM Computing Surveys*, 51(5), 2018.

[4] Wikipedia contributors. Sim racing — Wikipedia, the free encyclopedia, 2022. [Online; accessed 1-September-2022].

[5] Wikipedia contributors. 4d film — Wikipedia, the free encyclopedia, 2022. [Online; accessed 1-September-2022].

[6] Meta. Meta: Social Metaverse Company, 2022.

[7] VIVE. Vr headsets, games, and metaverse life: United states, 2022.

[8] Fabien Danieau, Anatole Lécuyer, Philippe Guillotel, Julien Fleureau, Nicolas Mollet, and Marc Christie. Enhancing audiovisual experience with haptic feedback: A survey on hav. *IEEE Transactions on Haptics*, 6(2):193–205, 2012.

[9] Jaebong Lee, Bohyung Han, and Seungmoon Choi. Motion effects synthesis for 4d films. *IEEE Transactions on Visualization and Computer Graphics*, 22(10):2300–2314, 2015.

[10] Jaebong Lee, Bohyung Han, and Seungmoon Choi. Interactive motion effects design for a moving object in 4d films. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pages 219–228. ACM, 2016.

[11] Chien-Hsing Chou, Yu-Sheng Su, Che-Ju Hsu, Kong-Chang Lee, and Ping-Hsuan Han. Design of desktop audiovisual entertainment system with deep learning and haptic sensations. *Symmetry*, 12(10), 2020.

[12] Gyeore Yun, Hyoseung Lee, Sangyoon Han, and Seungmoon Choi. Improving viewing experiences of first-person shooter gameplays with automatically-generated motion effects. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, 2021.

[13] Yaxuan Li, Yongjae Yoo, Antoine Weill-Duflos, and Jeremy Cooperstock. Towards context-aware automatic haptic effect heneration for home theatre environments. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*. ACM, 2021.

[14] Jan Gugenheimer, Dennis Wolf, Gabriel Haas, Sebastian Krebs, and Enrico Rukzio. Swivrchair: A motorized swivel chair to nudge users' orientation for 360 degree storytelling in virtual reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, page 1996–2000. ACM, 2016.

[15] Lung-Pan Cheng, Patrick Lühne, Pedro Lopes, Christoph Sterz, and Patrick Baudisch. Haptic turk: A motion platform based on people. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 3463–3472. ACM, 2014.

[16] Dhruv Jain, Misha Sra, Jingru Guo, Rodrigo Marques, Raymond Wu, Justin Chiu, and Chris Schmandt. Immersive terrestrial scuba diving using virtual reality. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, page 1563–1569. ACM, 2016.

[17] Ali Israr and Ivan Poupyrev. Tactile brush: Drawing on skin with a tactile grid display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2019–2028. ACM, 2011.

[18] Seungmoon Choi and Katherine J. Kuchenbecker. Vibrotactile display: Perception, technology, and applications. *Proceedings of the IEEE*, 101(9):2093–2104, 2012.

[19] Roshan Lalintha Peiris, Wei Peng, Zikun Chen, Liwei Chan, and Kouta Minamizawa. Thermovr: Exploring integrated thermal haptic feedback with head mounted displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, page 5452–5456. ACM, 2017.

[20] Takamichi Nakamoto and Hai Pham Dinh Minh. Improvement of olfactory display using solenoid valves. In *2007 IEEE Virtual Reality Conference*, pages 179–186. IEEE, 2007.

[21] Jas Brooks, Pedro Lopes, Judith Amores, Emanuela Maggioni, Haruka Matsukura, Marianna Obrist, Roshan Lalintha Peiris, and Nimesha Ranasinghe. Smell, taste, and temperature interfaces. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, 2021.

[22] Taeyong Moon and Gerard J. Kim. Design and evaluation of a wind display for virtual reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 122–128. ACM, 2004.

[23] Eduardo C. Rodrigues, Estevao B. Saleme, and Celso A. S. Santos. A haptic system for switching wind temperatures based on ultrasonic vibrations, peltier elements, and electrical resistances for multisensory applications. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, pages 73–80. ACM, 2021.

[24] Wikipedia contributors. Sensorama — Wikipedia, the free encyclopedia, 2022. [Online; accessed 1-September-2022].

[25] Jaeha Kim, Chang-Gyu Lee, Yeongmi Kim, and Jeha Ryu. Construction of a haptic-enabled broadcasting system based on the mpeg-v standard. *Signal Processing: Image Communication*, 28(2):151–161, 2013.

[26] Daniel Marfil, Fernando Boronat, Juan González, and Almanzor Sapena. Integration of multi-sensorial effects in synchronised immersive hybrid tv scenarios. *IEEE Access*, 10:79071–79089, 2022.

[27] Koichi Hirota, Seichiro Ebisawa, Tomohiro Amemiya, and Yasushi Ikei. A system for creating the content for a multi-sensory theater. In *Proceedings of the International Conference on Virtual and Mixed Reality*, pages 151–157. Springer, 2011.

[28] Koichi Hirota, Seichiro Ebisawa, Tomohiro Amemiya, and Yasushi Ikei. A theater for viewing and editing multi-sensory content. In *2011 IEEE International Symposium on VR Innovation*, pages 239–244. IEEE, 2011.

[29] João D. P. Sardo, João A. R. Pereira, Ricardo J. M. Veiga, Jorge Semião, Pedro J. S. Cardoso, and João M. F. Rodrigues. Multisensorial portable

device for augmented reality experiences in museums. *International Journal of Education and Learning Systems*, 3:60–69, 2018.

[30] João D. P. Sardo, João D. P. Pereira, Ricardo J. M. Veiga, Jorge Semião, Pedro J. S. Cardoso, and João M. F. Rodrigues. A portable device for five sense augmented reality experiences in museums. *WSEAS Transactions on Environment and Development*, 14:347–362, 2018.

[31] Nimesha Ranasinghe, Pravar Jain, Nguyen Thi Ngoc Tram, Koon Chuan Raymond Koh, David Tolley, Shienny Karwita, Lin Lien-Ya, Yan Liangkun, Kala Shamaiah, Chow Eason Wai Tung, Ching Chiuan Yen, and Ellen Yi-Luen Do. Season traveller: Multisensory narration for enhancing the virtual reality experience. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018. 577.

[32] Miguel Melo, Guilherme Gonçalves, Pedro Monteiro, Hugo Coelho, José Vasconcelos-Raposo, and Maximino Bessa. Do multisensory stimuli benefit the virtual reality experience? a systematic review. *IEEE Transactions on Visualization and Computer Graphics*, 28(2):1428–1442, 2020.

[33] Yuki Kon, Takuto Nakamura, and Hiroyuki Kajimoto. Hangerover: Hmd-embedded haptics display with hanger reflex. In *ACM SIGGRAPH 2017 Emerging Technologies*. ACM, 2017.

[34] Jan Gugenheimer, Dennis Wolf, Eythor R. Eiriksson, Pattie Maes, and Enrico Rukzio. Gyrovr: Simulating inertia in virtual reality using head worn flywheels. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, page 227–232. ACM, 2016.

[35] Chi Wang, Da-Yuan Huang, Shuo-wen Hsu, Chu-En Hou, Yeu-Luen Chiu, Ruei-Che Chang, Jo-Yu Lo, and Bing-Yu Chen. Masque: Exploring lateral

skin stretch feedback on the face with head-mounted displays. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, page 439–451. ACM, 2019.

[36] Takayuki Kameoka, Yuki Kon, Takuto Nakamura, and Hiroyuki Kajimoto. Haptopus: Haptic vr experience using suction mechanism embedded in head-mounted display. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*, page 154–156. ACM, 2018.

[37] Celso A. Saibel Santos, Almerindo N. Rehem Neto, and Estevao B. Saleme. An event-driven approach for integrating multi-sensory effects to interactive environments. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 981–986. IEEE, 2015.

[38] Fabien Danieau, Julien Fleureau, Philippe Guillotel, Nicolas Mollet, Anatole Lécuyer, and Marc Christie. Hapseat: Producing motion sensation with multiple force-feedback devices embedded in a seat. In *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology*, pages 69–76. ACM, 2012.

[39] Fabien Danieau, Jérémie Bernon, Julien Fleureau, Philippe Guillotel, Nicolas Mollet, Marc Christie, and Anatole Lécuyer. H-studio: An authoring tool for adding haptic and motion effects to audiovisual content. In *Proceedings of the Adjunct Publication of the 26th Annual ACM Symposium on User Interface Software and Technology*, pages 83–84. ACM, 2013.

[40] Misha Sra, Abhinandan Jain, and Pattie Maes. Adding proprioceptive feedback to virtual reality experiences using galvanic vestibular stimulation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019. 675.

[41] Yeongmi Kim, Jongeun Cha, Jeha Ryu, and Ian Oakley. A tactile glove design and authoring system for immersive multimedia. *IEEE MultiMedia*, 17(3):34–45, 2010.

[42] Sang-Kyun Kim. Authoring multisensorial content. *Signal Processing: Image Communication*, 28(2):162–167, 2013.

[43] Markus Waltl, Benjamin Rainer, Christian Timmerer, and Hermann Hellwagner. An end-to-end tool chain for sensory experience based on mpeg-v. *Signal Processing: Image Communication*, 28(2):136–150, 2013.

[44] Yuhao Zhoul, Makarand Tapaswi, and Sanja Fidler. Now you shake me: Towards automatic 4d cinema. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7425–7434. IEEE, 2018.

[45] Thomhert S. Siadari, Mikyong Han, and Hyunjin Yoon. 4d effect video classification with shot-aware frame selection and deep neural networks. In *2017 IEEE International Conference on Computer Vision Workshops*, pages 1148–1155. IEEE, 2017.

[46] Raphael Abreu, Douglas Mattos, Joel A. F. dos Santos, and Débora C. Muchaluat-Saade. Semi-automatic synchronization of sensory effects in mulsemedia authoring tools. In *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*, pages 201–208. ACM, 2019.

[47] Raphael Silva de Abreu, Douglas Mattos, Joel dos Santos, Gheorghita Ghinea, and Débora Christina Muchaluat-Saade. Toward content-driven intelligent authoring of mulsemedia applications. *IEEE MultiMedia*, 28(1):7–16, 2020.

[48] Byounghyun Yoo, Moohyun Cha, and Soonhung Han. A framework for a multi-sensory vr effect system with motional display. In *2005 International Conference on Cyberworlds*, pages 237–244. IEEE, 2005.

[49] Ali Israr, Siyan Zhao, Kaitlyn Schwalje, Roberta Klatzky, and Jill Lehman. Feel effects: Enriching storytelling with haptic feedback. *ACM Transactions on Applied Perception*, 11(3), 2014.

[50] Hasti Seifi, Kailun Zhang, and Karon E. MacLean. VibViz: Organizing, visualizing and navigating vibration libraries. In *Proceedings of the IEEE World Haptics Conference*, pages 254–259. IEEE, 2015.

[51] Suchul Shin, Byounghyun Yoo, and Soonhung Han. A framework for automatic creation of motion effects from theatrical motion pictures. *Multimedia Systems*, 20(3):327–346, 2014.

[52] Jongman Seo, Sunung Mun, Jaebong Lee, and Seungmoon Choi. Substituting motion effects with vibrotactile effects for 4d experiences. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* ACM, 2018.

[53] Beomsu Lim, Sangyoon Han, and Seungmoon Choi. Image-based texture styling for motion effect rendering. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology.* ACM, 2021.

[54] Eunsu Goh, Daeyeol Kim, Suyeong Oh, and Chae-Bong Sohn. Automatic effect generation method for 4d films. *International Journal of Computing and Digital Systems*, 9(2):281–288, 2020.

[55] Guangxin Zhao, Zhaobo Wang, Xiaoming Chen, Zhicheng Lu, Yuk Ying Chung, and Li Haisheng. Video2force: Experiencing object motion in video with dynamic force feedback based on bio-inspired sensing and processing.

In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops*, pages 858–859. IEEE, 2022.

[56] Jaebong Lee and Seungmoon Choi. Real-time perception-level translation from audio signals to vibrotactile effects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2567–2576. ACM, 2013.

[57] Yasushi Ikei, Shunki Kato, Kohei Komase, Shogo Imao, Sho Sakurai, Tomohiro Amemiya, Michiteru Kitazaki, and Koichi Hirota. Vestibulohaptic passive stimulation for a walking sensation. In *Proceedings of the IEEE Virtual Reality*, pages 185–186. IEEE, 2016.

[58] Koichi Shimizu, Vibol Yem, Kentaro Yamaoka, Gaku Sueta, Tomohiro Amemiya, Michiteru Kitazaki, and Yasushi Ikei. Rendering of virtual walking sensation by a vestibular display. In *Human Interface and the Management of Information. Information in Intelligent Systems*, volume 11570, pages 36–46. Springer, 2019.

[59] Tomohiro Amemiya, Michiteru Kitazaki, and Yasushi Ikei. Pseudo-sensation of walking generated by passive whole-body motions in heave and yaw directions. *IEEE Transactions on Haptics*, 13(1):80–86, 2020.

[60] Hyoseung Lee, Seungjae Oh, and Seungmoon Choi. Data-driven rendering of motion effects for walking sensations in different gaits. *IEEE Transactions on Haptics*, 2022. On-Line Access.

[61] Sangyoon Han, Jiwan Lee, Gyeore Yun, Sung H. Han, and Seungmoon Choi. Motion effects: Perceptual space and synthesis for specific perceptual properties. *IEEE Transactions on Haptics*, 2022. On-Line Access.

[62] Russell V. Parrish, James E. Dieudonne, Roland L. Bowles, and Dennis J. Martin Jr. Coordinated adaptive washout for motion simulators. *Journal of Aircraft*, 12(1):44–50, 1975.

[63] Lloyd D. Reid and Meyer A. Nahon. Flight simulation motion-base drive algorithms: Part 1. developing and testing equations. *UTIAS Report, No. 296*, 1985.

[64] MA Nahon, LD Reid, and J Kirdeikis. Adaptive simulator motion software with supervisory control. *Journal of Guidance, Control, and Dynamics*, 15(2):376–383, 1992.

[65] LD Reid and Meyer A Nahon. Flight simulation motion-base drive algorithms.: Part 2, selecting the system parameters. *UTIAS report*, (N307), 1986.

[66] Eduardo F. Camacho and Carlos Bordons Alba. *Model predictive control.* Springer Science & Business Media, 2013.

[67] D Cleij, J Venrooij, P Pretto, M Katliar, Heinrich H Bülthoff, D Steffen, FW Hoffmeyer, and H-P Schöner. Comparison between filter-and optimization-based motion cueing algorithms for driving simulation. *Transportation Research Part F: Traffic Psychology and Behaviour*, 61:53–68, 2019.

[68] J Venrooij, P Pretto, M Katliar, SAE Nooij, A Nesti, M Lächele, KN de Winkel, D Cleij, and HH Bülthoff. Perception-based motion cueing: Validation in driving simulation. In *Proc. DSC*, pages 153–161, 2015.

[69] Sergio Casas, Ricardo Olanda, and Nilanjan Dey. Motion cueing algorithms: a review: algorithms, evaluation and tuning. *International Journal of Virtual and Augmented Reality (IJVAR)*, 1(1):90–106, 2017.

[70] Nikhil JI Garrett and Matt Best. Driving simulator motion cueing algorithms–a survey of the state of the art. 2010.

[71] Ibrahim Kajo, Aamir Saeed Malik, and Nidal Kamel. Motion estimation of crowd flow using optical flow techniques: A review. In *2015 9th International Conference on Signal Processing and Communication Systems*, pages 1–9. IEEE, 2015.

[72] Julie Chang and Gordon Wetzstein. Deep optics for monocular depth estimation and 3d object detection. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 10192–10201. IEEE, 2019.

[73] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3917–3926. IEEE, 2019.

[74] Jozef J. Zwislocki and David A. Goodman. Absolute scaling of sensory magnitudes: A validation. *Perception & Psychophysics*, 28(1):28–38, 1980.

[75] David H. Krantz. A theory of magnitude estimation and cross-modality matching. *Journal of Mathematical Psychology*, 9(2):168–199, 1972.

[76] Joseph C. Stevens and Lawrence E. Marks. Cross-modality matching functions generated by magnitude estimation. *Perception & Psychophysics*, 27(5):379–389, 1980.

[77] Sung H. Han, Maengkee Song, and Jiyoung Kwahk. A systematic method for analyzing magnitude estimation data. *International Journal of Industrial Ergonomics*, 23(5-6):513–524, 1999.

[78] Stanley S. Stevens. On the psychophysical law. *Psychological Review*, 64(3):153–181, 1957.

[79] William S. Cleveland and Clive Loader. Smoothing by local regression: Principles and methods. In *Statistical theory and computational aspects of smoothing*, pages 10–49. Springer, 1996.

[80] Karl Duncker. Über induzierte bewegung. *Psychologische Forschung*, 12(1):180–259, 1929.

[81] Wikipedia contributors. Motion field — Wikipedia, the free encyclopedia, 2020. [Online; accessed 3-September-2020].

[82] Thomas Mergner and Wolfgang Becker. Perception of horizontal self-rotation: Multisensory and cognitive aspects. In *Perception and Control of Self-motion*, pages 219–263. Lawrence Erlbaum Hillsdale, NJ, 1990.

[83] Bruce E. Goldstein and James Brockmole. *Sensation and perception.* Cengage Learning, 2016.

[84] Claude Bonnet. Thresholds of motion perception. In *Tutorials on Motion Perception*, pages 41–79. Springer, 1982.

[85] Hans Wallach. Eye movement and motion perception. In *Tutorials on motion perception*, pages 1–18. Springer, 1982.

[86] Sungkil Lee, Gerard Jounghyun Kim, and Seungmoon Choi. Real-time tracking of visually attended objects in virtual environments and its application to lod. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):6–19, 2008.

[87] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[88] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. Xnect: Real-time multi-person 3d motion capture with a single rgb camera. *Acm Transactions On Graphics*, 39(4), 2020.

[89] Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *2010 IEEE Intelligent Vehicles Symposium*, pages 486–492. IEEE, 2010.

[90] Tomáš Souček and Jakub Lokoč. Transnet v2: an effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020.

[91] Alok Singh, Dalton Meitei Thounaojam, and Saptarshi Chakraborty. A novel automatic shot boundary detection algorithm: robust to illumination and motion effect. *Signal, Image and Video Processing*, 14(4):645–653, 2020.

[92] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *CVPR*, 2020.

[93] Yukun Su, Jingliang Deng, Ruizhou Sun, Guosheng Lin, and Qingyao Wu. A unified transformer framework for group-based segmentation: Co-segmentation, co-saliency detection and video salient object detection. *IEEE Transactions on Multimedia*, 2023.

[94] William S Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836, 1979.

[95] A LuNežič, Tomáš Vojíř, L Čehovin Zajc, Jiří Matas, and Matej Kristan. Discriminative correlation filter tracner with channel and spatial reliability. *International Journal of Computer Vision*, 126(7):671–688, 2018.

[96] YouTube. Avengers vs chitauri army - hulk punches thor - final battle scene - movie clip hd, 2019.

[97] YouTube. How to train your dragon 2 (2014) - the wingsuit scene (1/10) — movieclips, 2018.

[98] YouTube. [genshin story episode 1] the beginning of the story (opening), 2020.

[99] YouTube. Yuna kim's breathtaking performance to send in the clowns — music monday, 2019.

[100] YouTube. Last call for mr. paul, 2016.

[101] YouTube. Jennie - solo dance cover / cover by sol-e kim (mirror mode), 2018.

[102] YouTube. For honor in 60fps just hits different (ps5/4k) - 1v1 gameplay, 2022.

[103] YouTube. Hellish quart - longsword duel, 2021.

[104] YouTube. Spider-man remastered no way home suit (ps5) 4k 60fps hdr + ray tracing gameplay - (full game), 2021.

Curriculum Vitae

Name : Sangyoon Han

Education

2012. 3. – 2016. 8. Department of Computer Science and Engineering, Pohang University of Science and Technology (B.S.)

2016. 9. – 2023. 8. Department of Computer Science and Engineering, Pohang University of Science and Technology (Ph. D)

Affiliation

1. Interaction Lab., Department of Computer Science and Engineering, Pohang University of Science and Technology

Publications

International Journals

1. **Sangyoon Han***, Jiwan Lee*, Gyeore Yun*, Sung. H. Han, and Seungmoon Choi (*Co-first Author). Motion Effects: Perceptual Space and Synthesis for Specific Perceptual Properties. *IEEE Transactions on Haptics*, 15(3), pp. 626-637, 2022.

International Conferences

1. **Sangyoon Han**, Jaehyeok Ahn, and Seungmoon Choi. Generating Haptic Motion Effects for Scene Components to Improve 4D Experiences. In *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces* (IEEE VR), 2024 (*To be submitted*).

2. Jaejun Park, **Sangyoon Han**, and Seungmoon Choi. Merging Camera- and Object-based Motion Effects for Improved 4D Experiences. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality* (IEEE ISMAR), 2023 (*Submitted*).

3. Jaejun Park, Junwoo Kim, **Sangyoon Han**, Chaeyong Park, Junseok Park, and Seungmoon Choi. Information Transfer of Full-Body Vibrotactile Stimuli: An Initial Study with One to Three Sequential Vibrations. In *Proceedings of the IEEE World Haptics Conference* (WHC), 2023.

4. **Sangyoon Han**, Jaejun Park, and Seungmoon Choi. Generating Motion Effects for Multiple Articulated Bodies to Improve 4D Experiences. *ACM SIGCHI Conference on Human Factors in Computing Systems* (CHI), 2023.

5. Beomsu Lim, **Sangyoon Han**, and Seungmoon Choi. Image-Based Texture Styling for Motion Effect Rendering. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology* (VRST), Paper 20, 2021.

6. Gyeore Yun, Hyoseung Lee, **Sangyoon Han**, and Seungmoon Choi. Improving Viewing Experiences of First-Person Shooter Gameplays with Automatically-Generated Motion Effects. *ACM SIGCHI Conference on Human Factors in Computing Systems* (CHI), Paper 320, 2021.

7. **Sangyoon Han**, Gyeore Yun, Seungmoon Choi. Camera Space Synthesis of Motion Effects Emphasizing a Moving Object in 4D Films. In *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces* (IEEE VR), 2021.

8. **Sangyoon Han**, Amit Bhardwaj, Seungmoon Choi. Automatic Transfer of Musical Mood into Virtual Environments. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology* (VRST), 2018.

9. **Sangyoon Han**, Jaebong Lee, and Seungmoon Choi. Towards Automatic Synthesis of Motion Effects. In *Lecture Notes in Electrical Engineering* (AsiaHaptics 2018), 2018.

Posters and Demonstrations

1. **Sangyoon Han**, Sunung Mun, Jongman Seo, Jaebong Lee, and Seungmoon Choi. 4D Experiences Enabled by Automatic Synthesis of Motion and Vibrotactile Effects. Demonstrated in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI), 2018.

Domestic Conferences

1. Jaejun Park, Junwoo Kim, **Sangyoon Han**, Chaeyong Park, and Seungmoon Choi. Estimating Information Transfer for Consecutive Full Body Vibrotactile Stimuli. In *Extended Abstracts of HCI Korea*, 2023.

2. Dong Yeong Jeong, Sung H. Han, Seungmoon Choi, Mingyu Lee, Haewoo Kang, **Sangyoon Han**, Gyeore Yun, Hyo Seung Lee, Jiwan Lee, and Dajin Lee. Motion and Haptic Effects Classification of a 4D Cinema Seat. In *Proceedings of the Spring Conference of Ergonomics Society of Korea*, 2019.

3. **Sangyoon Han**, Hojin Lee, and Seungmoon Choi. Augmenting Realistic Experience with Motion Estimation Module in Virtual Reality. In *Extended Abstracts of HCI Korea*, 2019.

4. **Sangyoon Han**, Yongjae Yoo, and Seungmoon Choi. Gesture Interface and Vibration Feedback for Improvement of Office Environment. In *Proceedings of Korea Computer Congress* (KCC 2016), pp. 1405-1407, 2016.

Patents

1. Seungmoon Choi, and **Sangyoon Han**, "GENGERATING HAPTIC MOTION EFFECTS FOR MULTIPLE ARTICULATED BODIES FOR IMPROVED 4D EXPERIENCES," Domestic Patent. Appl. No. 10-2023-0049380, Appl. Date: 2023-04-14

2. Seungmoon Choi, Beosu Lim, and **Sangyoon Han**, "METHOD FOR GENERATING TEXTURE MOTION EFFECTS BASED ON IMAGES AND APPARATUS THEREOF," International Patent (US). Appl. No. 18/054,709, Appl. Date: 2022-11-11

3. Seungmoon Choi, **Sangyoon Han**, and Gyeore Yun, "METHOD AND DEVICE FOR TRANSFORMING DEGREE OF FREEDOM IN MULSEMEDIA SYSTEM," International Patent (US). Appl. No. 17/704,790, Appl. Date: 2022-03-25

4. Seungmoon Choi, **Sangyoon Han**, and Gyeore Yun, "METHOD AND APPARATUS FOR GENERATING MOTION EFFECTS BASED ON REDUCED OBJECT MOTION," Domestic Patent Appl. No. 10-2021-0055061, Appl. Date: 2021-04-28

5. Seungmoon Choi, **Sangyoon Han**, and Hojin Lee, "METHOD FOR GENERATING VIRTUAL ENVIRONMENT BASED ON USER MOVEMENT OR VEHICLE MOVEMENT AND APPARATUS FOR THE SAME," Domestic Patent Appl. No. 10-2017-0047498, Appl. Date: 2017-04-12