Camera Space Synthesis of Motion Effects Emphasizing a Moving Object in 4D films

Sangyoon Han[*] POSTECH

Gyeore Yun[†] POSTECH

Seungmoon Choi[‡] POSTECH

2021 IEEE Virtual Reality and 3D User Interfaces (VR) | 978-1-6654-1838-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/VR50410.2021.00093

Abstract

Four-dimensional (4D) films, which provide special physical effects to the audience with audiovisual stimuli, are gaining more popularity and acceptance. One of the most frequent 4D effects is the object-based motion effect, which refers to the vestibular stimulus generated by a motion chair to emphasize a moving object of interest, e.g., the flying iron man, displayed on the screen. In this paper, we present an algorithm for synthesizing convincing object-based motion effects automatically from a given object motion trajectory. While previous approaches use the 2D object position on the screen as input, our method takes the 3D position and orientation of the object in the camera space and computes its *motion proxy* that reflects both the object translation and rotation, as well as its size to the viewers' eyes. The 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. The motion proxy is fed to a motion cueing algorithm (MCA) that computes the command using a washout filter or model predictive control. The most appropriate MCA for our purpose is selected from six candidates by a user study. We also consider the effects of visual perception by incorporating two types of motion field equations into the computation of the visually perceived velocity. The results of a user study indicate that our algorithm can generate compelling object-based motion effects that better enhance the 4D film viewing experience than the previous methods.

Index Terms: Information systems—Information systems applications—Multimedia information systems—Multimedia content creation; Computing methodologies—Computer graphics—Graphics systems and interfaces—Virtual reality

1 Introduction

The four-dimensional (4D) platform is an immersive entertainment system that presents various physical effects, such as motion, vibration, wind, water, and scent, with audiovisual content to improve viewers' multimedia experiences [21]. Recently, 4D platforms are widely used in different applications, e.g., 4D rides, 4D films, and virtual reality (VR) games. Several 4D system manufacturers, such as D-BOX, MediaMation, and CJ 4DPlex, have actively developed technologies, applications, and contents. For example, CJ 4DPlex has opened over 700 4D theaters in 70 countries as of 2019 [6].

The production of 4D effects, however, still relies on manual authoring. This costly process prevents the rapid and broader dissemination of 4D content. For facilitation, a few groups addressed the need for automated algorithms that create 4D effects from audiovisual data [21, 22, 31]. In the taxonomy of 4D effects established by Lee et al. [21], motion effects used in 4D films are classified into four classes according to the audiovisual grounds used for design. Among them, motion effects that emphasize the motion of an object of interest in the scene, e.g., the flying iron man, are most frequently used (Fig. 1). For automatic synthesis of such *object-based motion effects*, we propose an algorithm that uses the full 3D translation and rotational motion of an object in the camera space as the basis of synthesis. Human vestibular and visual perception models are explicitly integrated into the synthesis algorithm.

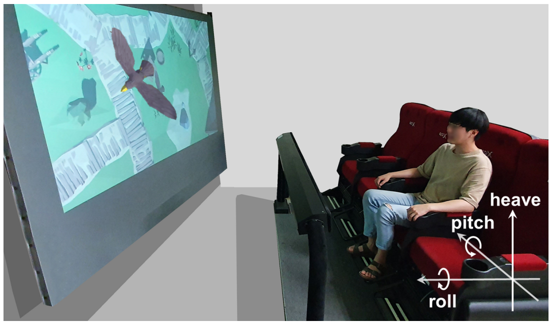


Figure 1: Person watching an object-moving movie with motion effects.

Our new synthesis algorithm is featured with the following advantages: 1) The full six-degree of freedom (DoF) motion of the object of interest in the camera space is transformed to an object motion effect, which minimizes the loss of information during the conversion, on the basis of *motion proxy*—a new concept abstracting the translational and rotational information of the object into one point (Section 3); 2) The effects of human visual perception as to the object size (Section 4) and position (Section 6) are investigated, modeled, and integrated into the synthesis framework; 3) Two types of conventional motion cueing algorithms (MCAs), washout filters and model predictive control (MPC), that were developed originally for motion effect computation in flight and vehicle simulations, are adapted to our purpose of audience experience enhancement while blending the physical constraints of the motion chair and the human vestibular perception into the formulation (Section 5); and 4) The entire algorithm (Section 7) is evaluated by a user study in comparison to the current state of the art, which demonstrates clear benefits of our method in terms of viewer experience (Section 8). Conclusions, current limitations and future work, are also presented (Section 9).

2 Related Work

The 4D platform is a form of commercially successful mulsemedia [8], improving users' multimedia experiences by providing various sensory effects, such as vestibular [26], vibrotactile [5, 15], olfactory [27], and wind [25], with audiovisual content. With the growth of the 4D market, this field is gaining popularity also in academia. Hirota et al. [14] presented a multisensory theater with olfactory, wind, and pneumatic displays and a content editing tool using musical instrument digital interface (MIDI). Danieau et al. [10] designed a seat stimulating the user's two hands and head using force-feedback devices. They also introduced the concept of haptic cinematography and the associated taxonomy [9]. Sra et al. [32] improved VR experience by providing proprioceptive feedback using galvanic vestibular stimulation, rather than a motion chair.

[*] e-mail: han0209@postech.ac.kr

[†] e-mail: ykre0827@postech.ac.kr

[‡] e-mail: choism@postech.ac.kr

2642-5254/21/$31.00 ©2021 IEEE
DOI 10.1109/VR50410.2021.00093

Among the various 4D effects, motion effects are most frequently used for their effectiveness in improving immersiveness [21]. The motion platforms and related control algorithms were originally developed for flight simulations in the early 1970s [28]. For motion control, washout filters had been regarded as providing an optimal tradeoff between perceptual quality and algorithmic simplicity [26]. Recently, control algorithms based on MPC were developed for driving simulations. The MPC algorithms that include a computational model of vestibular perception [29] allow less perceptual error than the washout filters [3, 7, 35].

In industry, 4D designers use in-house authoring programs to design 4D effects, similar to authoring tools for 4D broadcasting [17, 18, 37]. These tools have similar interfaces and provide multiple timelines, each of which is for one kind of 4D effect. While useful, 4D designers must annotate all video segments that will match the 4D effects. To reduce such efforts, some studies proposed automatic annotation methods [1, 39]. They use neural networks trained on the audiovisual information to detect video segments for which 4D effects will be provided and classify the corresponding 4D effects. However, it remains the designers' responsibility to determine the specific properties of 4D effects, such as intensity and direction over time; the design process is still quite time-consuming.

Automatic generation of 4D effects is essential to accelerate the proliferation of 4D applications. This is a growing area, but there exist some notable attempts. Shin et al. [31] was the first in proposing a synthesis framework for creating motion effects that emphasize camera motion for point-of-view (POV) shots. The camera trajectory is estimated from sequential images and then used to compute the command to the motion chair using a washout filter. Lee et al. [21] developed much more efficient synthesis algorithms for camera-based motion effects with quasi-real time performance. According to user studies, their algorithms produce very convincing results comparable to those manually crafted by 4D experts. These algorithms were adapted to provide automatic vibrotactile effects using multiple actuators in a chair as an inexpensive substitute of camera motion effects [30]. Additionally, Lee and Choi [20] proposed an audio-to-vibrotactile translation algorithm for selective audio-tactile feedback (e.g., not responding to background music) by relating loudness and roughness between sound and touch.

Lee et al. [22] presented the first automatic synthesis algorithm that converts the 2D trajectory of object motion projected to the screen to the motion command of two DoFs (roll and pitch in Fig. 1). An object of interest is tracked by a computer vision algorithm, and its 2D screen velocity is fed to a washout filter. This strategy, named *viewer-centered rendering*, attempts to match the chair motion to the shift of the audience's visual attention. It results in plausible motion effects for many 4D scenes. The authors also report some limitations, such as "Only roll and pitch motions are used. Heave is essential for better motion effects." and "There were no motion effects for the spin of objects." Some of them can be resolved only by considering the object motion in the camera space.

3 Problem Formulation

As shown in Fig. 2, an 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 $\mathbf{m} = (roll, pitch, heave)^T$ (Fig. 1). Then, our research problem is formulated to finding 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.

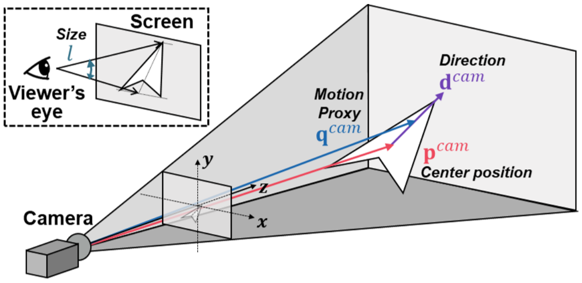


Figure 2: A moving object and its proxy in the 3D camera space.

Lee et al. [22] 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 Fig. 2 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 set our research goal as: *designing 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:

$$\mathbf{m} = w_T(l)(\mathbf{p}^{cam} - \mathbf{p}_0^{cam}) + w_R(l)(\mathbf{d}^{cam} - \mathbf{d}_0^{cam}), \tag{1}$$

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

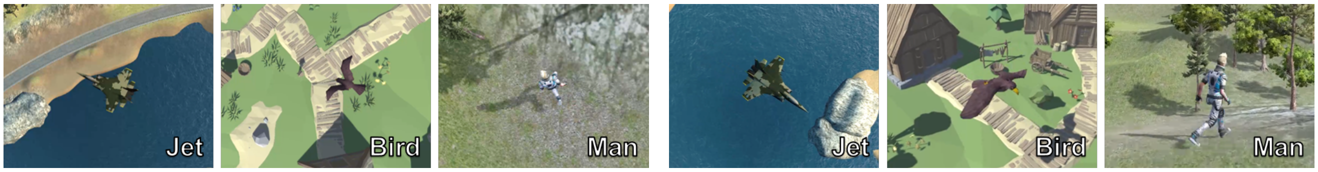
$$\mathbf{q}^{cam} = \mathbf{p}^{cam} + (w_R(l)/w_T(l))\mathbf{d}^{cam}, \tag{2}$$

$\mathbf{q}^{cam}$ stands for the point that is shifted from the object center by $w_R(l)/w_T(l)$ in the frontal direction (Fig. 2). $\mathbf{q}^{cam}$ 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 (1) can be simplified to

$$\mathbf{m} = w_T(l)(\mathbf{q}^{cam} - \mathbf{q}_0^{cam}). \tag{3}$$

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.

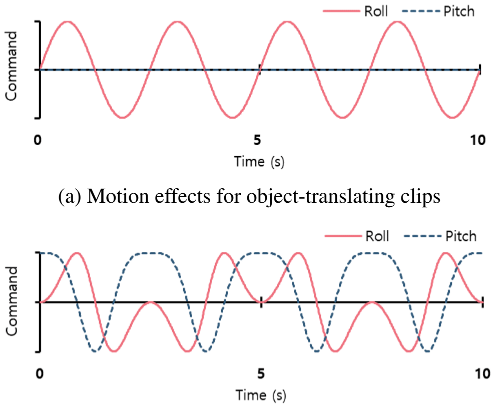
In this work, we assume that the center position $\mathbf{p}^{cam}$, the frontal direction $\mathbf{d}^{cam}$, and the object size l are given, and then focus on motion effect synthesis using them. This information is readily available in computer-generated scenes, such as animation films and VR games. For regular 4D films, the information can be extracted from the image sequence using recent computer vision techniques, such as motion estimation [16], depth estimation [4], and object detection [23]. We leave an integration of such automatic feature extraction as future work and focus on motion generation.



(a) Top-down camera view

(b) Other camera views

Figure 3: Three video clips used in user experiments. See an accompanied video for different motions of the objects and cameras.



(a) Motion effects for object-translating clips

(b) Motion effects for object-rotating clips

Figure 4: Motion effects used in the motion scaling experiment.

4 Optimal Motion Scale Rules

To compute the motion proxy $\mathbf{q}^{\mathbf{cam}}$ by (2) and the motion command $\mathbf{q}^{\mathbf{cam}}$ by (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. All experiments reported in this paper were approved by the Institutional Review Board at POSTECH (PIRB-2019-R018). They involved different participant groups.

4.1 Methods

The motion chair (4DX, CJ 4DPLEX; Fig. 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.).

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*; Fig. 3a), 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:

$$\mathbf{p}^{cam}(t) = (\sin(0.8\pi t), 0, 0)^T, \tag{4}$$

which has the 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 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:

$$\theta(t) = \pi(\cos(0.4\pi t) - 0.5), \tag{5}$$

which has the 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:

$$\mathbf{d}^{cam}(t) = (\cos(\theta(t)), \sin(\theta(t)), 0)^T. \tag{6}$$

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 (Fig. 4):

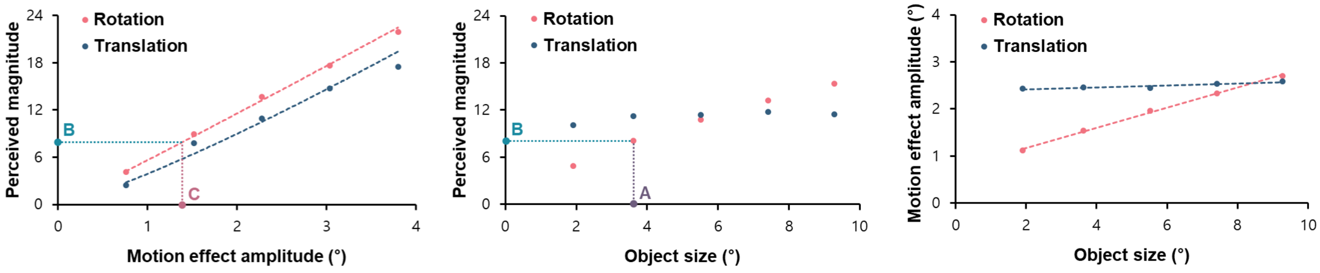
$$\begin{aligned} \mathbf{m}_T(t) &= A\,\mathbf{p}^{cam}(t) = (A\sin(0.8\pi t), 0, 0)^T, \\ \mathbf{m}_R(t) &= A\,\mathbf{d}^{cam}(t) = (A\cos(\theta(t)), A\sin(\theta(t)), 0)^T, \end{aligned} \tag{7}$$

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 [40]. 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 [19], and the relationships between two such variables follow the power law [33]. 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.



(a)

(b)

(c)

Figure 5: 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.

The rating data were independently standardized using the mean deviation standardization [13] to reduce the individual deviations.

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.

4.2 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 [34], 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.

Fig. 5a shows the two psychophysical magnitude functions for motion effects for object translation and rotation. The equations are:

$$\begin{aligned}\psi_T &= 3.8985A^{1.2039} \ \ (R^2 = 0.9737),\\ \psi_R &= 5.6332A^{1.0372} \ \ (R^2 = 0.9978),\end{aligned} \tag{8}$$

where A is the motion amplitude in (7). All the coefficients of determination (R^2) were higher than 0.97, indicating a very good fit.

Fig. 5b 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 Fig. 5a and 5b as follows: 1) select an object size (e.g., point A in Fig. 5b), 2) find the corresponding perceived magnitude of a motion effect that would best match the object size using Fig. 5b (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 Fig. 5a (point C).

Applying this procedure to all the data in the plots of Fig. 5b 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 Fig. 5c), such that

$$\begin{aligned}w_T(l) &= 0.0211\,s + 2.3730 \ \ (R^2 = 0.8578),\\ w_R(l) &= 0.2133\,s + 0.7474 \ \ (R^2 = 0.9973),\end{aligned} \tag{9}$$

where w_T and w_R are the scale factors in (1)—note that they have the same role with A in (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 (1) with the motion scaling rules in (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 (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 (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 (9) that the perceptual role of rotation becomes more important as the object size increases.

5 Motion Cueing Algorithms

Actual motion commands are constrained by the limited performance of the motion chair. The simple rule in (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) 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.

5.1 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.

1) *Washout filter with velocity input* (WV): WV (Fig. 6a) uses the motion velocity $\mathbf{v}$ as input. This is the same one used in Lee et al. [22], motivated by the fact that the object velocity is detected from visual information (i.e., optical flow). $\mathbf{v}$ 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.

2) *Washout filter with acceleration input* (WA): Here the input is the motion acceleration $\mathbf{a}$, 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 (Fig. 6b). WA and WV are similar, but different in that WA regards the initial velocity as 0.

3) *Washout filter with acceleration input and tilt coordination* (WAT): WAT is the combination of WA and tilt coordination (Fig. 6c). 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.

5.2 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.

1) *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

$$\begin{aligned} \mathbf{m} = \arg\min_{\mathbf{m}} \quad & \|\mathbf{v} - \mathbf{v_m}\|^2 \\ \text{subject to} \quad & \mathbf{m}(t) = \int_0^t \mathbf{v_m}(t')dt',\ |\mathbf{m}(t)| \le \mathbf{m}_{max}, \end{aligned} \tag{10}$$

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

2) *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 [29]:

$$\frac{\hat{a}}{a} = \frac{K(\tau_n s + 1)}{(\tau_S s + 1)(\tau_L s + 1)}, \tag{11}$$

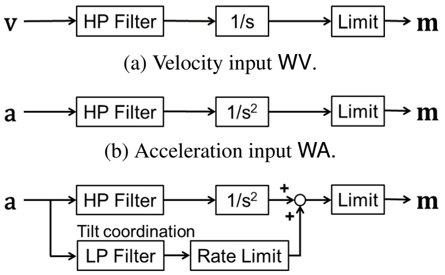
where K, τ_n, τ_S, and τ_L are the model coefficients of the otoliths (0.4, 13.2, 0.66, and 5.33, respectively), and s is a Laplacian variable. This transfer function has good responsiveness in 0.01–0.5 Hz and effectively works as a band-pass filter. Then MA solves the following optimization problem:

$$\begin{aligned} \mathbf{m} = \arg\min_{\mathbf{m}} \quad & \|\hat{\mathbf{a}} - \hat{\mathbf{a}}_\mathbf{m}\|^2 \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}a(t),\ \hat{a}(t) = \mathbf{C}\mathbf{x}(t), \\ & \dot{\mathbf{x}}_\mathbf{m}(t) = \mathbf{A}\mathbf{x}_\mathbf{m}(t) + \mathbf{B}a_\mathbf{m}(t),\ \hat{a}_\mathbf{m}(t) = \mathbf{C}\mathbf{x}_\mathbf{m}(t), \\ & \mathbf{m}(t) = \int_0^t \int_0^{t'} \mathbf{a_m}(t'')dt''dt',\ |\mathbf{m}(t)| \le \mathbf{m}_{max}, \end{aligned} \tag{12}$$

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 (11) for each component, given by

$$\mathbf{A} = \begin{pmatrix} -\frac{1}{\tau_L} - \frac{1}{\tau_S} & 1 \\ -\frac{1}{\tau_L \tau_S} & 0 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} \frac{K\tau_n}{\tau_L \tau_S} \\ \frac{K}{\tau_L \tau_S} \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 1 & 0 \end{pmatrix}. \tag{13}$$

Note that the lines 2–3 in (12) and the state space model in (13) are applied to each of the 1D component in $\mathbf{a}$ and $\mathbf{a_m}$.



(a) Velocity input WV.

(b) Acceleration input WA.

(c) Acceleration input with tilt coordination WAT.

Figure 6: Three washout filters.

3) *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 (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.

5.3 User Study

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.

5.3.1 Methods

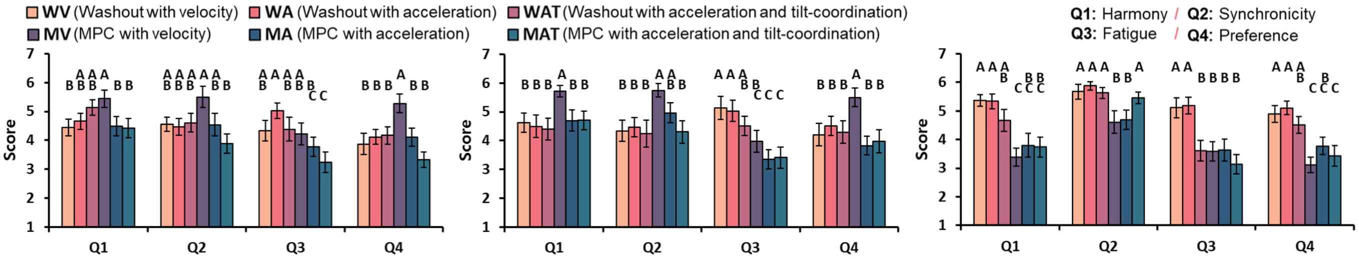
We made three 20-s video clips of moving objects, one for each of the three scenes (*Jet*, *Bird*, and *Man* in Fig. 3a), using Unity3D. The camera orientations were fixed to the top-down view. The 2D horizontal positions of the objects and the cameras were made using Perlin noise with different amplitudes 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.[1]

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 8.

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. Did the motion effects match the object motion? (Harmony); Q2. Were the motion effects and the object motion synchronized in time? (Synchronicity); Q3. Did the motion effects make you tired? (Fatigue); Q4. Did you like the motion effects? (Preference). 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.

[1] Plots for the object and camera motions are available in the supplemental material for further illustration.



(a) Jet

(b) Bird

(c) Man

Figure 7: 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.

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.

5.3.2 Results

Experimental results are shown in Fig. 7. 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 Fig. 7.[2]

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.

6 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 [11] and motion field equations [38].

Object motion can be perceived in two ways: subject-relative and object-relative perception [11]. 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 [24], we designed our algorithm to support both modes so that designers can select a mode that generates more plausible effects.

6.1 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 [12]. 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 $\omega^{cam} = (\omega_x, \omega_y, \omega_z)^T$ (Fig. 8a). 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 (14)-(16) hold [38]:

$$\mathbf{v}^{cam} = -\mathbf{t}^{cam} - \omega^{cam} \times \mathbf{q}^{cam},\ \mathbf{q}^{img} = \frac{f}{q_z^{cam}} \mathbf{q}^{cam},\ \text{and } \mathbf{v}^{img} = \frac{d}{dt} \mathbf{q}^{img}, \tag{14}$$

which gives:

$$\mathbf{v}^{img} = \frac{1}{f} \mathbf{G} \omega^{cam} + \frac{1}{q_z^{cam}} \mathbf{H} \mathbf{t}^{cam}, \tag{15}$$

where

$$\mathbf{G} = \begin{pmatrix} q_x^{img} q_y^{img} & -(q_x^{img})^2 - f^2 & f q_y^{img} \\ (q_y^{img})^2 + f^2 & -q_x^{img} q_y^{img} & -f q_x^{img} \end{pmatrix} \text{ and} \\ \mathbf{H} = \begin{pmatrix} -f & 0 & q_x^{img} \\ 0 & -f & q_y^{img} \end{pmatrix}. \tag{16}$$

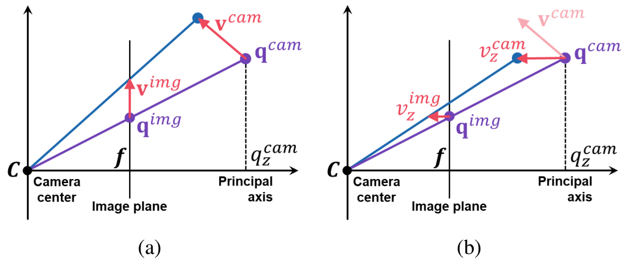
Denoting the subject-relative velocity vector of the object by $\mathbf{v}^{sub} = (v_x^{sub}, v_y^{sub}, v_z^{sub})^T$, we can obtain the depth component v_z by scaling down v_z^{cam} based on the distances from the camera to the image plane and to the object (Fig. 8b) by

$$v_z^{sub} = f \frac{v_z^{cam}}{q_z^{cam}} \tag{17}$$

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

$$\mathbf{v}^{sub} = \frac{1}{f} \mathbf{G}' \omega^{cam} + \frac{1}{q_z^{cam}} \mathbf{H}' \mathbf{t}^{cam}, \tag{18}$$

[2] The detailed results of all statistical tests are provided in the supplemental material to respect the page length requirement.



(a)

(b)

Figure 8: Motion vectors to determine the subject-relative velocity. (a) Horizontal and vertical components. (b) Depth component.

where

$$\mathbf{G}' = \begin{pmatrix} & \mathbf{G} & \\ -q_y^{img} & q_x^{img} & 0 \end{pmatrix} \text{ and } \mathbf{H}' = \begin{pmatrix} & \mathbf{H} & \\ 0 & 0 & -f \end{pmatrix}. \qquad (19)$$

6.2 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 by Duncker [11]: 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

$$\mathbf{v}^{obj} = \mathbf{v}^{sub} - \mathbf{v}_{bg}^{sub} = \mathbf{H}' \left(\frac{1}{q_z^{cam}} \mathbf{t}^{cam} - \frac{1}{Z_{bg}} \mathbf{t}_{bg}^{cam} \right), \qquad (20)$$

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 (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.

7 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 object motion effects, as depicted in Fig. 9. 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 (2). The two scale factors for translation and rotation, $w_T(l)$ and $w_R(l)$, are determined by (9) based on the experimental results of Section 4. 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 (18) or in the object-relative mode using (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 a MCA.

8 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 [21, 22].

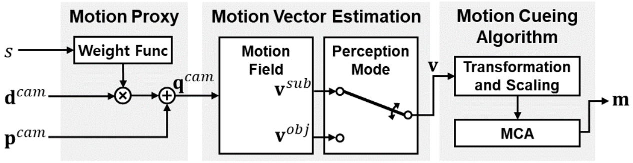


Figure 9: Flow of our synthesis algorithm for object motion effects.

8.1 Methods

Using Unity3D, we made three 20-s video clips each showing the three moving objects of *Jet*, *Bird*, and *Man* in Fig. 3b. 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 5.3. 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.

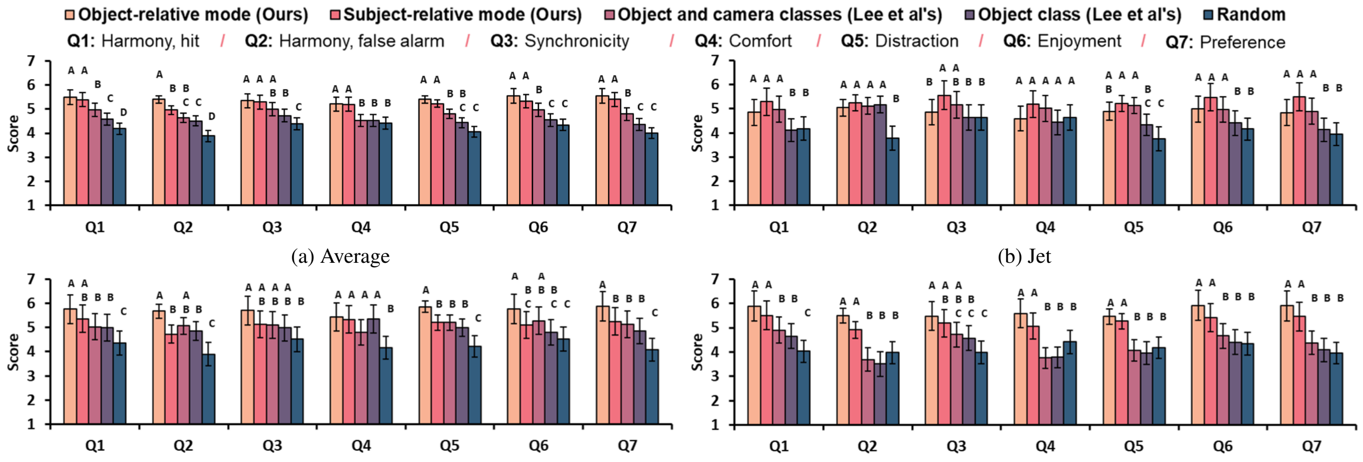
- Object-relative mode (OR): Motion effects synthesized by our algorithm (perception mode: object-relative, MCA: MV).
- Subject-relative mode (SR): Motion effects synthesized by our algorithm (perception mode: subject-relative, MCA: MV).
- Object and camera classes (OC): Addition of the two motion effects for object and camera classes generated by [21, 22].
- Object class (O): Object class motion effects synthesized by [22].
- Random (R): Randomly generated motion effects by Perlin noise.

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 were 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 [2]), 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. Did the motion effects follow the expected motion well? (Harmony, hit); Q2. Were the motion effects played even though you did not expect any motion? (Harmony, false alarm); Q3. Were the motion effects and the object motion well synchronized in time? (Synchronicity); Q4. Did the motion effects feel comfortable? (Comfort); Q5. Were the motion effects distracting from focusing on the object motion? (Distraction); Q6. Did you enjoy watching the video with the motion effects? (Enjoyment); Q7. Did you like the motion effects? (Preference); Q8. 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.

Twenty-four people (13 males and 11 females; 20–30 years old



(a) Average

(b) Jet

(c) Bird

(d) Man

Figure 10: 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.

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.

8.2 Results

The experimental results are shown in Fig. 10. 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 Fig. 10).

On average (Fig. 10a), 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* (Fig. 10d). Also, OR was more favored than SR in *Bird* (Fig. 10c). We speculate that this is because object motion is predominantly perceived through object-relative perception [36]. In contrast, SR was slightly more favored in *Jet* (Fig. 10b). *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 [21, 22] 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.

9 Conclusions and Future Work

In this paper, we have presented an automatic synthesis algorithm that generates object motion effects by abstracting the 6 DoF object motion and utilizing human vestibular and visual perception models. Our 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 can produce perceptually better motion effects than the current state of the art.

The contributions of our 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) Our 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.

Our algorithm has a potential for extenion 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 [4, 16, 23] 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.

Acknowledgments

This work was supported by the Samsung Research Funding & Incubation Center of Samsung Electronics (SRFC-IT1802-05).

References

[1] R. Abreu, D. Mattos, J. A. d. Santos, and D. C. Muchaluat-Saade. Semi-automatic synchronization of sensory effects in mulsemedia authoring tools. In *Proc. WebMedia*, pp. 201–208. ACM, 2019.

[2] C. Bonnet. Thresholds of motion perception. In *Tutorials on motion perception*, pp. 41–79. Springer, 1982.

[3] E. F. Camacho and C. B. Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[4] J. Chang and G. Wetzstein. Deep optics for monocular depth estimation and 3d object detection. In *Proc. ICCV*, pp. 10193–10202. IEEE, 2019.

[5] S. Choi and K. J. Kuchenbecker. Vibrotactile display: Perception, technology, and applications. *Proceedings of the IEEE*, 101(9):2093–2104, Sep 2012.

[6] CJ. Cgv 4dx surpasses 700 locations worldwide. `http://english.cj.net/cj_now/view.asp?bs_seq=14331&schBsTp=1&schTxt=`, 2019. [Online; accessed 26-July-2020].

[7] D. Cleij, J. Venrooij, P. Pretto, M. Katliar, H. H. Bülthoff, D. Steffen, F. 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, Feb 2019.

[8] A. Covaci, L. Zou, I. Tal, G.-M. Muntean, and G. Ghinea. Is multimedia multisensorial?-a review of mulsemedia systems. *ACM Computing Surveys*, 51(5):1–35, Sep 2018.

[9] F. Danieau, J. Fleureau, P. Guillotel, N. Mollet, M. Christie, and A. Lécuyer. Toward haptic cinematography: enhancing movie experiences with camera-based haptic effects. *IEEE MultiMedia*, 21(2):11–21, Apr–Jun 2014.

[10] F. Danieau, J. Fleureau, P. Guillotel, N. Mollet, A. Lécuyer, and M. Christie. Hapseat: producing motion sensation with multiple force-feedback devices embedded in a seat. In *Proc. VRST*, pp. 69–76. ACM, 2012.

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

[12] E. B. Goldstein and J. Brockmole. *Sensation and perception*. Cengage Learning, 2016.

[13] S. H. Han, M. Song, and J. Kwahk. A systematic method for analyzing magnitude estimation data. *International Journal of Industrial Ergonomics*, 23(5-6):513–524, Mar 1999.

[14] K. Hirota, S. Ebisawa, T. Amemiya, and Y. Ikei. A system for creating the content for a multi-sensory theater. In *Proc. VMR*, pp. 151–157. Springer, 2011.

[15] A. Israr and I. Poupyrev. Tactile brush: drawing on skin with a tactile grid display. In *Proc. CHI*, pp. 2019–2028. ACM, 2011.

[16] I. Kajo, A. S. Malik, and N. Kamel. Motion estimation of crowd flow using optical flow techniques: A review. In *Proc. ICSPCS*, pp. 1–9. IEEE, 2015.

[17] S.-K. Kim. Authoring multisensorial content. *Signal Processing: Image Communication*, 28(2):162–167, Feb 2013.

[18] Y. Kim, J. Cha, J. Ryu, and I. Oakley. A tactile glove design and authoring system for immersive multimedia. *IEEE MultiMedia*, 17(3):34–45, Jul–Sep 2010.

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

[20] J. Lee and S. Choi. Real-time perception-level translation from audio signals to vibrotactile effects. In *Proc. CHI*, pp. 2567–2576. ACM, 2013.

[21] J. Lee, B. Han, and S. Choi. Motion effects synthesis for 4d films. *IEEE Transactions on Visualization and Computer Graphics*, 22(10):2300–2314, Dec 2015.

[22] J. Lee, B. Han, and S. Choi. Interactive motion effects design for a moving object in 4d films. In *Proc. VRST*, pp. 219–228. ACM, 2016.

[23] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang. A simple pooling-based design for real-time salient object detection. In *Proc. CVPR*, pp. 3917–3926. IEEE, 2019.

[24] T. Mergner and W. Becker. Perception of horizontal self-rotation: Multisensory and cognitive aspects. In *Perception and Control of Self-motion*, pp. 219–263. Lawrence Erlbaum Hillsdale, NJ, 1990.

[25] T. Moon and G. J. Kim. Design and evaluation of a wind display for virtual reality. In *Proc. VRST*, pp. 122–128. ACM, 2004.

[26] M. A. Nahon and L. D. Reid. Simulator motion-drive algorithms-a designer's perspective. *Journal of Guidance, Control, and Dynamics*, 13(2):356–362, 1990.

[27] T. Nakamoto and H. P. D. Minh. Improvement of olfactory display using solenoid valves. In *Proc. IEEE VR*, pp. 179–186. IEEE, 2007.

[28] R. V. Parrish, J. E. Dieudonne, R. L. Bowles, and D. J. Martin Jr. Coordinated adaptive washout for motion simulators. *Journal of Aircraft*, 12(1):44–50, Jan 1975.

[29] L. Reid and M. Nahon. Flight simulation motion-base drive algorithms: part 1. developing and testing equations. *UTIAS Report, No. 296*, 1985.

[30] J. Seo, S. Mun, J. Lee, and S. Choi. Substituting motion effects with vibrotactile effects for 4d experiences. In *Proc. CHI*, pp. 1–6. ACM, 2018.

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

[32] M. Sra, A. Jain, and P. Maes. Adding proprioceptive feedback to virtual reality experiences using galvanic vestibular stimulation. In *Proc. CHI*, pp. 1–14. ACM, 2019.

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

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

[35] J. Venrooij, P. Pretto, M. Katliar, S. Nooij, A. Nesti, M. Lächele, K. de Winkel, D. Cleij, and H. Bülthoff. Perception-based motion cueing: validation in driving simulation. In *Proc. DSC*, pp. 153–161, 2015.

[36] H. Wallach. Eye movement and motion perception. In *Tutorials on motion perception*, pp. 1–18. Springer, 1982.

[37] M. Waltl, B. Rainer, C. Timmerer, and H. Hellwagner. An end-to-end tool chain for sensory experience based on mpeg-v. *Signal Processing: Image Communication*, 28(2):136–150, Feb 2013.

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

[39] Y. Zhou, M. Tapaswi, and S. Fidler. Now you shake me: Towards automatic 4d cinema. In *Proc. CVPR*, pp. 7425–7434. IEEE, 2018.

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