Motion Effects Synthesis for 4D Films

Jaebong Lee, *Student Member, IEEE*, Bohyung Han, *Member, IEEE*, and Seungmoon Choi, *Member, IEEE*

Abstract—4D film is an immersive entertainment system that presents various physical effects with a film in order to enhance viewers' experiences. Despite the recent emergence of 4D theaters, production of 4D effects relies on manual authoring. In this paper, we present algorithms that synthesize three classes of motion effects from the audiovisual content of a film. The first class of motion effects is those responding to fast camera motion to enhance the immersiveness of point-of-view shots, delivering fast and dynamic vestibular feedback. The second class moves viewers as closely as possible to the trajectory of slowly moving camera. Such motion provides an illusional effect of observing the scene from a distance while moving slowly within the scene. For these two classes, our algorithms compute the relative camera motion and then map it to a motion command to the 4D chair using appropriate motion mapping algorithms. The last class is for special effects, such as explosions, and our algorithm uses sound for the synthesis of impulses and vibrations. We assessed the subjective quality of our algorithms by user experiments, and results indicated that our algorithms can provide compelling motion effects.

Index Terms—4D film, multi-sensory theater, motion simulator, motion cueing, motion effects, synthesis, automatic generation

1 Introduction

4D film refers to an immersive entertainment system that presents various physical effects, such as motion, vibration, wind, water, and scent, with a 2D or 3D film. Evolving from popular attractions in amusement parks, 4D films are now produced from regular films and screened in 4D theaters. D-BOX, a major Canadian manufacturer of 4D motion chairs, has 94 theaters with its motion systems in the United States alone as of 2014. Another manufacturer based in Korea, CJ 4DPLEX, has opened 97 4D theaters in 23 countries and releases more than 50 4D films in a year [1].

This recent growth of 4D films has elevated the needs for efficient production methods of 4D effects. However, 4D effects designers still *manually* create all effects using in-house authoring software. As a result, 4D effects production remains highly labor-intensive. For example, three 4D effects designers need to work for about 16 days to make a 4D film out of one regular film (see Section 2.2). In this situation, automated algorithms that synthesize 4D effects from regular films by analyzing their video and audio streams, at least for partial clips, can improve productivity to the great extent. Such algorithms are also expected to allow 4D effects designers to spend more effort on artistic and creative tasks, thereby leading to better 4D films. Further, automatic algorithms may be able to pioneer new 4D applications, e.g., live broadcasting of F1 racing or World Cup with real-time motion effects. In this paper, we introduce our progress on this imperative topic of *automatic 4D effects synthesis*, with emphasis upon motion effects (see Fig. 1).

1.1 Related Work

4D systems are designed by integrating various display technologies developed for virtual reality, such as vestibular (motion) [2], vibrotactile [3], olfactory [4], and wind displays [5]. However, rigorous academic research on such 4D systems has not been very active, with only a few exceptions. For instance, Hirota et al. [6] presented a multi-sensory theater with olfactory, wind, and pneumatic displays, along with a content editing tool that used MIDI interfaces. There has also been growing interest in 4D broadcasting systems that extend MPEG-V by adding haptic feedback [7].

Among the various 4D effects, motion effects are generally regarded as the most effective in improving immersiveness, so they are used most frequently (see Section 2.1). Motion simulators and their control algorithms were first developed as early as in 1970s for flight simulation [8]. They began to be adopted for commercial entertainment from the mid-1980s [9]. In the meanwhile, many motion control algorithms have been proposed, but a general consensus is that a classical washout filter [2] provides the best trade-off between perceptual quality and algorithmic simplicity. For instance, a motion display platform by Yoo et al. [10] uses a washout filter to compose main motion commands, and then superimposes vibrations as needed to the motion commands for further enhancement.

In industry, in-house authoring programs are used to design 4D effects. Several research groups also developed authoring tools for 4D broadcasting [11], [12], [13]. All of them have similar interfaces providing multiple timelines, each of which is for an individual 4D effect that 4D designers need to create. This manual design process is time-consuming by nature, although it can be facilitated to some extent by useful user interface components such as a library of reusable 4D patterns.

• *The authors are with the Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), 77 Cheongam-ro. Nam-gu., Pohang, Gyeongbuk, Korea. E-mail: {novaever, bhhan, choism}@postech.ac.kr.*

Manuscript received 4 Nov. 2014; revised 10 Oct. 2015; accepted 25 Nov. 2015. Date of publication 10 Dec. 2015; date of current version 7 Sept. 2016. Recommended for acceptance by X. Tong.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2015.2507591

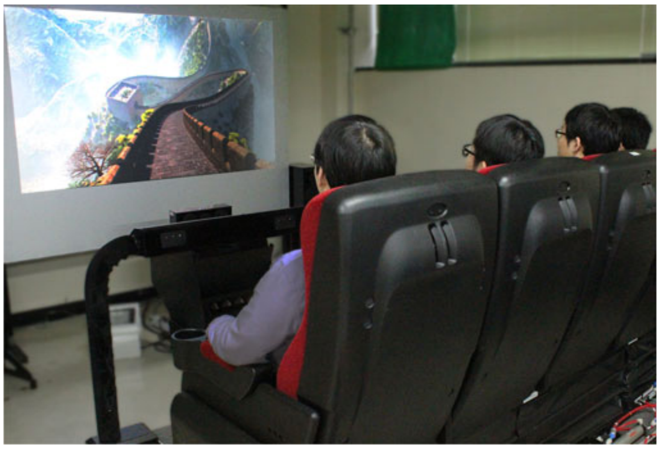


Fig. 1. Audience watching a 4D film with motion effects synthesized by our algorithms. See videos in supplemental material, available online.

Automatic generation of 4D effects is indispensable for the further growth of 4D films and associated applications. Research in this direction is still at infancy, but it has seen increasing endeavors. For example, our research group proposed an audio-to-vibrotactile translation algorithm [14]. For crossmodal conversion, this algorithm relates two perceptual variables, loudness and roughness, between sound and touch instead of physical signals, which allows for the design of more intuitive and selective conversion models. We also developed a tactile chair system that emphasizes visually salient regions on the screen by vibrotactile feedback that is spatially mapped onto the viewer's back [15]. The visually salient areas are identified by a real-time GPU-accelerated algorithm. Further, Lee and Han proposed an early algorithm that determines the posture of motion platform based on block matching between two video frames [16].

A notable progress has been made very recently by Shin et al. [17], who presented a framework that shared the same motivation with our present work. Their framework relies on Boujou (a commercial matchmover; Vicon Motion Systems) to estimate the 3D trajectory of camera in the world coordinate frame from sequential 2D images. The camera position is then numerically differentiated once and twice to obtain angular velocity and linear acceleration, respectively. These variables are fed to classical washout filters to make motion commands of a motion chair. To suppress the noise amplified by the numerical differentiation, they proposed a total variation-based noise reduction technique, which is a non-causal filter that finds an optimal balance between fitting error and jerk (the derivative of acceleration). They also project the direction of gravity progressively to the 3D scene for gravity rendering. The initial gravity direction can be estimated from the first image if the image contains prominent vertical lines, or it needs to be set by the user. They presented many motion effects synthesized by their framework, but reported no user studies that would allow for the objective assessment of perceptual quality.

Another important attempts have been made by Danieau et al.. They designed a low-cost motion chair, which stimulates the viewer's two hands and head using three force-feedback devices, and evaluated the quality of experience (QoE) of that approach [18]. They also introduced a concept of haptic cinematography and an associated taxonomy [19]. As a proof of concept, they designed haptic rendering models for several camera effects (e.g., zoom-in and tilting) that are frequently used in cinematography based on the assumption that camera motion is known, along with an evaluation of their QoE.

Also see [20], [21] for a more general review on the enhancement of user experience by means of haptic feedback provided along with audiovisual content.

1.2 Paper Overview

The long-term aim of our research is to contribute to 4D film production by providing autonomous synthesis algorithms of 4D effects. Among others, this work concentrates on motion effects, which are most frequently used in 4D films. This specific goal was identified through surveys on 4D effects and their current production system (Section 2). The surveys enabled us to classify motion effects into six classes and summarize common design strategies for each class, on which our synthesis algorithms are based.

Our algorithms assume that 4D effects designers segment a film into a number of audiovisual streams and determine the 4D effects that will be used for each segment (a procedure called scene breakdown; Section 2.2). Our algorithms process the audiovisual content of each segment to synthesize a motion effect of the designated type. Such motion effects should be optimized to improving the viewers' experiences, such as immersiveness and fun, not necessarily to achieving physically faithful effects.

Three classes of motion effects that are frequently used in 4D films are currently supported. For the first two classes, camera motion is restored from visual scenes using computer vision techniques (Section 3.1). The estimated camera motion is used to synthesize motion effects that respond to fast and abrupt camera motion (Section 3.2). These motion effects improve the immersiveness of point-of-view (POV) shots by delivering dynamic vestibular feedback, like 4D rides in amusement parks. When the camera moves very slowly in a long shot, our motion synthesis algorithm converts it to gentle and continuous movements of a motion chair (Section 3.3). Such effects make viewers feel as if they were looking at the scene in the viewpoint of a slowly moving camera, thereby improving presence—the feeling of being there. Both algorithms are designed for the limited workspace of motion chairs. The last class is for special effects in a film, such as explosions and collisions (Section 3.4). We exploit sound for the synthesis of impulses and vibrations on the basis of our prior perception-level detection algorithm [14]. The viability of our synthesis algorithms are demonstrated by various example plots and videos (Section 4). Last, we report the methods and results of two user experiments, one with general users and the other with 4D experts, carried out to assess the perceptual quality of our algorithms (Section 5). The results indicate that our synthesis algorithms can produce convincing motion effects, even comparable to those manually crafted by 4D effects designers in some cases.

To our knowledge, the contributions of our work are threefold: 1) the comprehensive survey and analysis on 4D effects, 2) the three algorithms that synthesize motion effects in a similar way to the practice of 4D effects designers, and

TABLE 1
Ten Regular 4D Films Used in the Survey

Title	Release Year
Percy Jackson: Sea of Monsters	2013
Gravity	2013
Thor: The Dark World	2013
Ender's Game	2013
Frozen	2013
The Hobbit: The Desolation of Smaug	2013
Need for Speed	2014
Captain America: The Winter Soldier	2014
The Amazing Spider-Man 2	2014
X-Men: Days of Future Past	2014

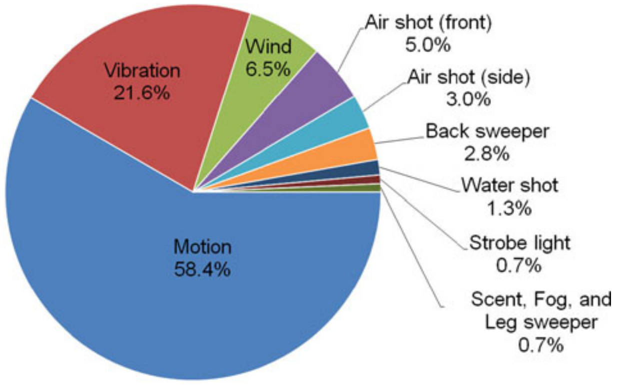


Fig. 2. Frequencies of 4D effects appearing in regular 4D films.

3) the thorough performance evaluations including user experiments.

2 Surveys on 4D Effects

Detailed guidelines for 4D effects designs are difficult to establish. After all, it is a creative artistic process that greatly depends on the expertise, experience, and preference of human designers. Under this circumstance, the first and foremost step of our research had to be defining the goals and requirements of algorithms that would afford the best benefits. To this end, we carried out two surveys, and results are described in this section.

2.1 Classification of 4D Effects

Our first survey was to classify the types and design methods of 4D effects provided in 4D films and observe their frequencies of use. At present, two kinds of 4D films are popular. One is *4D rides*, in which most scenes consist of POV shots for the best 4D experience. 4D effects are provided intensively, but in a short running time (less than 10 minutes) to prevent viewers' fatigue. The other type is regular films to which 4D effects are added after production. Such *regular 4D films* are much longer and present 4D effects more sparsely.

For this survey, we watched 10 regular 4D films at 4DX theaters (CJ 4DPLEX) and eight 4D rides at various places, and then analyzed the 4D effects displayed with the films. Regular 4D films are generally made for action and adventure genres, and our selections shown in Table 1 are an adequate representative of regular 4D films.

The 10 regular 4D films included a total of 2,278 4D effects. We counted the number of times with which each 4D effect was presented, and their frequencies of use are visualized in Fig. 2. Motion effects were most frequently used (58.4 percent), while vibration effects ranked at the second (21.6 percent). The use of other effects was sporadic. These results indicate that motion effects deserve the highest priority for automatic synthesis, followed by vibration effects.

We further classify motion effects into six classes according to the grounds of motion effects. Typical examples of the six classes are also provided in Fig. 3.

Camera Motion Class (C). Motion effects are generated by following the camera motion. It has two subclasses: *fast camera motion class* (CF) and *slow camera motion class* (CS). CF effects are fast and abrupt, and deliver dynamic motion effects for POV shots. They are frequently used in 4D rides and films with racing scenes. CS effects present gently and continuously moving sensations when a slowly moving camera shoots landscape. Viewers perceive an illusional effect that they observe the scene from a distance while moving together with the camera. Effect durations are generally long (up to 30 s).

Object Motion Class (O). Motion effects track the motion of a character or an object of interest, also in two subclasses: *continuous object motion class* (OC) and *discrete object motion class* (OD). OC effects target an object that moves continuously for a relatively long period of time, e.g., in running, chasing, driving, and flying scenes. OD effects represent short, discrete motions and are often used in fighting scenes. OD effects are very short (less than 1 s), while OC effects are usually longer (2-5 s).

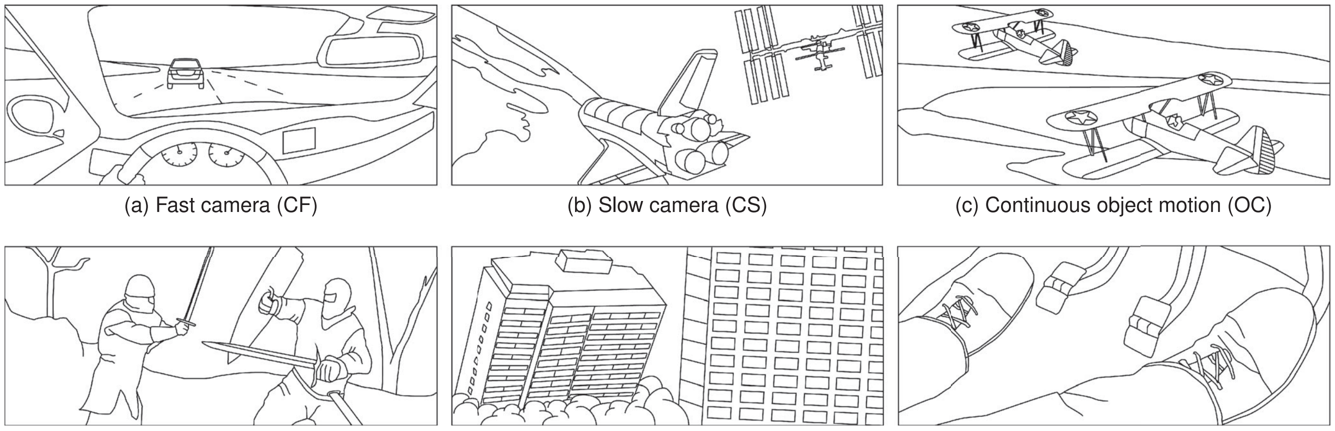
There can be an ambiguity between C and O effects when the camera and objects move simultaneously. In such cases, 4D effects designers determine which class of motion effects to use based on their subjective judgment of which motion is more dominant, also with contextual consideration.

Impulse and Vibration Class (V). Motion effects in this class offer responses to impacts or vibrations in the scene, such as gun fire, explosion, or vehicle vibration. V effects are very short in time but they occasionally last for a few seconds, e.g., when expressing ambient vibrations inside a vehicle.

OD effects can also be related to impacts. The key difference between OD and V effects is that OD effects produce clear directional cues while V effects are usually omnidirectional.

Context Class (T). Certain motion effects are created based on the designers' understanding of the context of events. Current images or sounds do not provide direct clues. An example is when a scene shows only a driver turning the steering wheel to left, motion feedback is also exerted to left to simulate the expected movement of the car. T effects are relatively short, similar to O effects.

Fig. 4 shows the relative frequencies with which the six classes of motion effects appeared in the 18 4D films. The



(a) Fast camera (CF) (b) Slow camera (CS) (c) Continuous object motion (OC)

(d) Discrete object motion (OD) (e) Impulse and vibration (V) (f) Context (T)

Fig. 3. Examples of six motion effect classes. (a) Motion effects are generated following fast and abrupt camera motion aligned with the driver. (b) A motion chair follows slow camera motion to provide an illusion that the viewer is looking at the earth and the spacecraft while floating at the camera position. (c) A motion chair follows the continuous movement of the biplane. (d) When a knight attacks an opponent with a sword, a motion chair produces short, discrete feedback tracking the movement of the sword. (e) When the building collapses, a motion chair invokes a strong, rough vibration. (f) When the driver hits the accelerator of a car, a motion chair tilts backward to provide the sensation of sudden acceleration.

4D rides relied on only CF and V motion effects. All classes of motion effects were observed in the regular 4D films. It is noted that even though O effects were observed more frequently than C effects, the latter has longer durations as described earlier. The actual playing time of C effects is comparable to or even exceeds that of O effects.

2.2 4D Effects Production

The second survey was an expert interview with three experienced 4D effects designers who worked for CJ 4DPLEX, in order to learn the general production procedure of 4D films and common design heuristics. They had two to four years of experience, and had produced more than 100 4D films.

According to the interviewees, 4D effects production is divided into three stages: scene breakdown, editing, and revision. In scene breakdown (also called pre-production), designers make an overall plan with the goal of maximizing immersiveness while avoiding possible 4D sickness. They segment the film and assign appropriate 4D effects into the segments. In the editing stage, the designers create detailed 4D effects according to the production plan, which involves numerous trials and errors. Last, the designers evaluate the entire 4D film and carefully revise the 4D effects for a final release. On average, 4D effects production takes 16 days (2, 10, and 4 days for the three stages) by three designers for regular 4D films and 12 days (1, 10, and 1 day) by one designer for 4D rides.

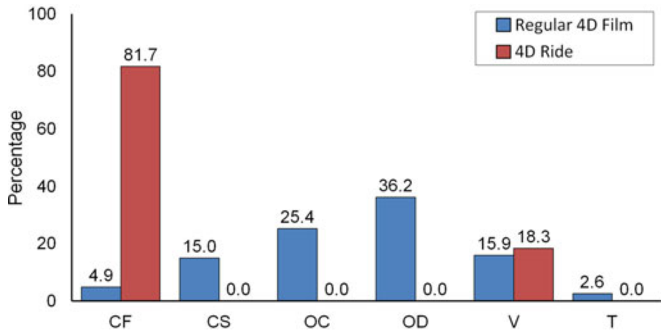


Fig. 4. Frequencies of the six classes of motion effects.

Motion chairs have a number of kinematic and dynamic constraints, e.g., in the degree of freedom (DOF) and the movement range, maximum velocity, and maximum acceleration in each independent axis. Therefore, motion effects need to be optimized to provide the best perceptual effects under the constraints of the chair. To this end, according to the interview, designers greatly rely on their contextual understanding of a film and artistic instincts, instead of using simple patterned motion effects. As such, the interviewees had difficulty in expressing their tactics and methods of motion effect design precisely in language. In our position, however, we needed rules of thumb that are adequate for automation. Hence, we inferred common design methods from the results of our motion effects analysis of the 18 4D films and presented them to the interviewees for their feedback. After long discussion, the interviewees confirmed that the following is generally accepted practice.

- When a motion chair does not support the full six DOFs, motion in a missing direction is substituted by motion in the most similar direction.
- For class C motion effects, designers move a chair to the same direction of camera motion to align the chair's motion to viewers' viewpoint, which is critical for immersiveness. The only exception is for CF effects during acceleration or deceleration, where the chair is moved to the opposite direction to render inertia. Also, providing onset cues at precise timing is of great importance for CF effects.
- To design CS effects, designers move a chair gently and continuously by tracking the camera motion while making a full use of the chair's motion range. If the chair can no longer follow the camera motion due to the chair's motion range limit, the chair is restored to the opposite direction and then pushed again to the original direction. This slow swing motion is effective in improving presence and so is standard in CS effects.
- Class O effects are designed in a similar way to CF effects. In most cases, the movement of a character or an object of interest is transferred to the audience in

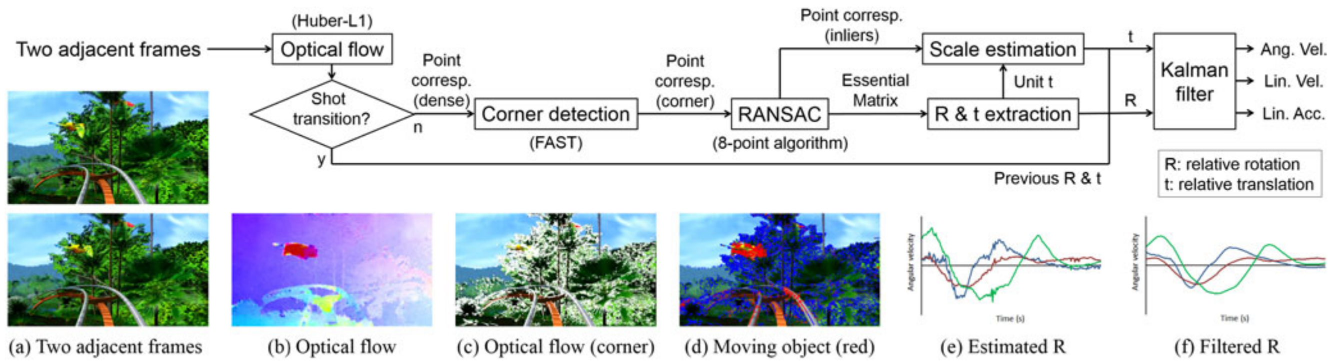


Fig. 5. Overview of our camera motion estimation algorithm and an example output of each step. Images (a) to (f) are example outputs taken from Amazon (ride film; Simuline Inc.). (a) to (d) are results obtained from the 607th and 608th frames. In (b), hue and brightness represent the direction and length of optical flow, respectively. In (c) and (d), flow vectors are represented by lines. In (d), blue lines represent inliers and red lines denote outliers. (e) and (f) are angular velocities estimated from the 510th to the 692th frames. Blue, red, and green lines represent angular velocities in roll, pitch, and yaw, respectively.

the direction displayed on the screen, i.e., in the third-person viewpoint. For example, if a car moves to right on the screen, then a chair is tilted to right, regardless of the direction to which the driver has turned the steering wheel. It is also possible that first-person experience, e.g., identifying the viewer as the car driver and tilting the chair to the direction of steering wheel rotation, is more adequate. Which method to use is decided by designers' understanding of the context.

- Class V effects are well correlated to sound effects, e.g., those for gunfire, explosion, and engine. Hence, sound is a good source for V effects design.

2.3 Summary

The survey results shed light on elucidating the specific needs for synthesis algorithms of 4D effects, as well as the goals and requirements of the algorithms. Among the three stages of 4D effects production, it is evident that the editing stage can be the best beneficiary of computational algorithms; human intelligence and judgments are vital in the other two stages.

After scene breakdown, a film is partitioned to a large number of segments along with the types of 4D effects to be used in each segment. If designers can apply a synthesis algorithm to each segment and then revise the output 4D effects, it will significantly improve the productivity of 4D effects production. Our synthesis algorithms are based on this use scenario.

Among the many types of 4D effects, motion effects are most frequently used, so they must be the first aim. Among the six classes of motion effects we defined, all the classes seem eligible for automatic synthesis, except class T effects that highly depend on the context. Class C effects and O effects are of similar use statistics, but C effects are more straightforward for algorithmic synthesis owing to the simplicity of the design strategy involved. O effects are likely to require more user interventions for algorithms' behavior.

Based on these findings, we decided to set our first target to the synthesis of 4D motion effects for class CF, CS, and V. These classes cover almost all effects of 4D rides and about 36 percent of motion effects for regular 4D films (Fig. 4).

3 Synthesis Algorithms

Our synthesis algorithms assume that audiovisual segments from a film are given, as well as the class (CF, CS, or V) of motion effects to make. Our algorithms for CF and CS effects use video as a source. They include a camera motion estimation method based on the epipolar constraint between two frames (Section 3.1). The estimated camera motion is used to compute motion commands for both CF and CS effects, but with different motion mapping algorithms (Sections 3.2 and 3.3). Our synthesis algorithm for V effects makes use of sound (Section 3.4).

3.1 Camera Motion Estimation

Relative camera motion between two consecutive frames is estimated using the epipolar constraint. We employ optical flow to find corresponding points and the normalized 8-point algorithm to compute a fundamental matrix. Camera motion parameters are extracted from the essential matrix [22]. The pipeline of our camera motion estimation algorithm is illustrated in Fig. 5.

Our camera estimation algorithm is optimized to synthesizing plausible motion effects for viewers, not to reconstructing physically-exact camera motion. It is easily implementable and widely applicable since the algorithm is well-established and does not require strong assumptions such as known camera intrinsics.

3.1.1 Identifying Corresponding Points

We employ optical flow to obtain correspondences between two frames because it typically leads to better estimation of fundamental matrix than sparse matching algorithms [23]. Although optical flow may have trouble in handling large displacement, this problem is not critical in our algorithm since corresponding points are computed only between two adjacent frames.

We use a variational optical flow estimation algorithm based on an anisotropic Huber-L1 regularization [24], called Huber-L1 optical flow hereafter, to obtain correspondence points. Huber-L1 optical flow is fast and shows reasonable performance in public benchmark [25].

For two input images I_0 and I_1, the Huber-L1 optical flow at a point $\mathbf{x}$ on a rectangular region $\Omega \in \mathbb{R}^2$ is defined as

$$\min_{\mathbf{u}}\left\{\int_{\Omega}|\nabla u_1(\mathbf{x})|_{\epsilon}+|\nabla u_2(\mathbf{x})|_{\epsilon}+\lambda|\rho(\mathbf{u}(\mathbf{x}))|d\mathbf{x}\right\}, \qquad (1)$$

where $\mathbf{u}(\mathbf{x}) = [u_1(\mathbf{x})\ u_2(\mathbf{x})]^{\mathsf{T}}$ denotes a 2D flow field. Its regularization terms are based on the Huber-L1 norms of motion gradients given by

$$|\nabla u_d|_{\epsilon}=\begin{cases}\frac{|\nabla u_d|^2}{2\epsilon} & \text{if } |\nabla u_d|\le\epsilon\\ |\nabla u_d|-\frac{\epsilon}{2} & \text{otherwise}\end{cases},\quad \text{for } \epsilon>0. \qquad (2)$$

The data term $|\rho(\mathbf{u}(\mathbf{x}))|$ is derived from the linearized brightness constancy constraint, such that

$$|\rho(\mathbf{u}(\mathbf{x}))| \equiv |\mathbf{u}(\mathbf{x})^{\mathsf{T}}\nabla I_1(\mathbf{x})+I_1(\mathbf{x})-I_0(\mathbf{x})|. \qquad (3)$$

where λ controls balance between the regularization and the data term.

This Huber-L1 algorithm well preserves not only discontinuities in object boundaries but also details inside an object. The staircasing effects caused by L1 norm are alleviated by the quadratic case of Huber norm while discontinuity-preserving property of L1 norm still holds at motion boundary with $|\nabla u_d| > \epsilon$. If the mean difference of optical flows between two frames is larger than a predefined threshold, we assume that a shot transition is detected. In this case, the observed motion is disregarded, and Kalman filter is employed to smooth motion.

Optical flow estimation is often unreliable in smooth and textureless areas (e.g., sky or road), and using too many corresponding points increases processing time significantly without practical benefit in the computation of fundamental matrix. Therefore, we select only the corner points extracted by a high-speed corner detection algorithm, FAST [26], which rejects a large number of non-corner points very quickly (Fig. 5c). FAST classifies a point as a corner if the point is brighter or darker than the majority of its neighborhood pixels by a certain threshold. The threshold is adaptively determined in our system to maintain a sufficient number of corner points.

3.1.2 Computing Robust Camera Motion Parameters

Once the correspondence points between frames are obtained, the fundamental matrix is estimated by RANSAC combined with the normalized 8-point algorithm [22]. Camera motion is estimated accurately when objects in the scene are stationary. RANSAC copes with a large portion of outliers such as moving objects effectively based on the property that the corresponding points from moving objects violate the epipolar constraint, as demonstrated in Fig. 5d.

To restore a camera motion, the fundamental matrix is converted to the essential matrix using the following definition:

$$\mathbf{E}=\mathbf{K}^{\mathsf{T}}\mathbf{F}\mathbf{K},\quad \mathbf{K}=\begin{pmatrix} f & 0 & w/2\\ 0 & f & h/2\\ 0 & 0 & 1\end{pmatrix}, \qquad (4)$$

where $\mathbf{K}$ is a camera calibration matrix parameterized by image width and height (w, h) and a predefined focal length f. Since the true focal length is different from f, a restored camera motion is also different from the true camera motion by a constant scale factor. This scale difference is handled by constant gains (c and b) in the motion mapping step (Sections 3.2 and 3.3). When the focal length changes between two frames, the restored camera motion might be different from the true camera motion beyond a constant factor. However, such distortion is negligible unless the focal length change is significant. Moreover, the focal length change is partly absorbed into extrinsic camera parameters and then applied to synthesizing appropriate motion effects.

Without loss of generality, a camera projection matrix of the previous frame is assumed to be $\mathbf{P}_1 = [\mathbf{I}|\mathbf{0}]$. The projection matrix of the current frame $\mathbf{P}_2 = [\mathbf{R}|\mathbf{t}]$ is estimated by a singular value decomposition of the essential matrix, $\mathbf{E} = \mathbf{U}\Sigma\mathbf{V}^{\mathsf{T}}$. Then, the rotation and translation matrices are given by

$$\mathbf{R}=\mathbf{U}\mathbf{W}\mathbf{V}^{\mathsf{T}} \text{ or } \mathbf{U}\mathbf{W}^{\mathsf{T}}\mathbf{V}^{\mathsf{T}} \quad\text{and}\quad \mathbf{t}=\mathbf{u}_3 \text{ or } -\mathbf{u}_3, \qquad (5)$$

where

$$\mathbf{W}=\begin{pmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 1\end{pmatrix} \text{ and } \mathbf{U}=[\mathbf{u}_1\ \mathbf{u}_2\ \mathbf{u}_3]. \qquad (6)$$

We select the best from the possible four combinations of $\mathbf{R}$ and $\mathbf{t}$ in terms of reconstruction errors.

The scale s of camera translation should also be estimated since the estimated translation $\mathbf{t}$ is a unit vector due to its inherent scale ambiguity. For a pair of corresponding points $(\mathbf{x}_i^1, \mathbf{x}_i^2)$, s is proportional to their distance $\|\mathbf{x}_i^2 - \mathbf{x}_i^1\|$ under the assumption that the 3D structure of the scene remains unchanged between two adjacent frames and there is no rotation. Based on this observation, $\mathbf{x}_i^1$ in $\|\mathbf{x}_i^2 - \mathbf{x}_i^1\|$ is replaced by $[\mathbf{R}|\mathbf{0}]\mathbf{X}_i$ to cancel out the distance originated from the rotation of camera, where $\mathbf{X}_i$ is a 3D point triangulated from $(\mathbf{x}_i^1, \mathbf{x}_i^2)$. Therefore, the scale of translation can be approximated by

$$s\approx\frac{\sum_i^N\|\mathbf{x}_i^2-[\mathbf{R}|\mathbf{0}]\mathbf{X}_i\|}{N}, \qquad (7)$$

where N is the number of reconstructed points. This provides sufficiently accurate solutions for our application unless scene changes are very abrupt.

Once the relative geometric configuration of camera between two frames is determined, it is straightforward to compute camera motion parameters—angular velocity, linear velocity, and linear acceleration, which are smoothed by Kalman filter (Fig. 5f). These output variables are used for the synthesis of CF and CS effects.

3.2 Synthesis of Fast Camera Motion Effects

Our synthesis algorithm for CF effects, Algorithm CF, includes the camera motion estimation described in Section 3.1 and a motion command synthesis. In what follows, ω, v, and a represent angular velocity, linear velocity, and linear acceleration, respectively. Subscripts, r, p, y, g, s, and h denote motion in roll, pitch, yaw, surge, sway, and heave, respectively, of a motion chair, as shown in Fig. 6a. $\mathbf{p} = [p_r, p_p, p_h]^{\mathsf{T}}$ is the final command to a motion chair. Our motion mapping method for Algorithm CF is depicted in a block diagram shown in Fig. 7.

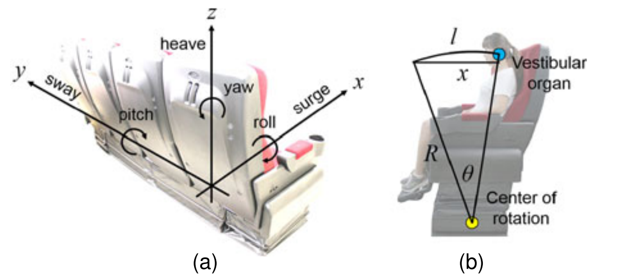


Fig. 6. (a) Definition of 6-DOF motions. (b) Linear-to-angular approximation (surge to pitch). For small θ, $x \approx l = R\theta$, so $x \propto \theta$.

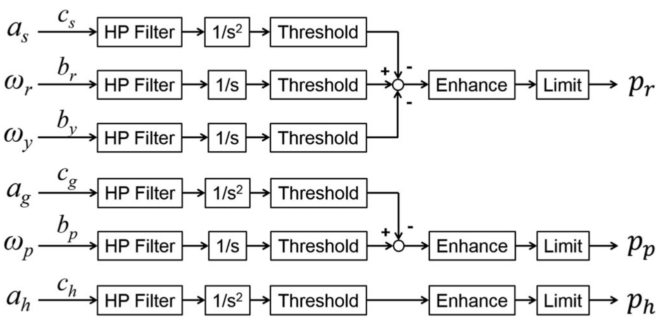


Fig. 7. Motion synthesis algorithm for CF effects.

For fast camera motion, we provide *vestibular feedback* to improve 4D experience. The human vestibular system responds to only acceleration, so not to linear velocity, but can also sense angular velocity through the linear acceleration occurring in rotational motion [27]. Thus, we use the three angular velocities and three linear accelerations of the camera as the input.

Physically exact reconstruction of camera motion using a motion chair is impossible due to two constraints: a motion chair has a limited workspace and often does not support the full six DOFs. For the former, the motion chair needs to return to its origin after generating one effect so that it can be ready for the next motion. For the latter, motion in the unsupported directions should be replaced by motion in the available directions. These two are the major requirements that need to be carefully addressed in motion synthesis.

For the first requirement, we use a washout filter: a de facto standard in motion simulators that has a proven convergence property to the initial state [2]. It is a high-pass filter, and it renders high-frequency onset cues while removing the low-frequency motion energy that tends to push a motion chair to its workspace limit. We use a first-order Butterworth filter with a cutoff frequency of 1.0 Hz for ω and a second-order Butterworth filter with a cutoff at 2.5 Hz for a.

It should be clarified that we do not consider tilt coordination, a technique for simulating sustained acceleration, such as gravity and centrifugal force, by tilting a motion chair for a relative long time. Tilt coordination is included in the original washout filter and commonly used for motion simulators with training purposes. Our decision is due to several reasons. First, tilt coordination is generally implemented using low-pass filters, but this technique is not very effective with general motion chairs for 4D films because of their much smaller workspace (8-14 degrees) than those of flight or driving simulators (usually greater than 40 degrees). Second, if sustained acceleration is provided with more abrupt CF effects, it is generally masked and not clearly perceptible to viewers. Third, we did not find instances in which only sustained acceleration was predominant out of the 2,278 4D effects we analyzed (Section 2). Last, including tilt coordination makes motion synthesis algorithms and associated gain tuning more complicated.

For the second requirement, we use a motion substitution rule that was designed based on the fact that an angular motion constrained at the center of rotation of a motion chair incurs a linear motion of the human vestibular system. As illustrated in Fig. 6b, a rotation in pitch makes a linear translation in surge, and their relationship is linearly proportional if the rotation is small. Since increasing pitch corresponds increasing surge in our coordinate system (see Fig. 6a and the path from a_g to p_p in Fig. 7), we can generally add the surge of camera to the pitch for motion mapping. For Algorithm CF, however, we subtract surge motion from pitch motion (Fig. 7) in order to render the inertia induced by acceleration/deceleration in the viewer's frontal direction. Similarly, increasing roll corresponds to decreasing sway in our coordinate system (Fig. 6a), sway motion is subtracted from roll motion (see the path from a_s to p_r in Fig. 7). Further, yaw is also linked to roll (see the path from ω_y to p_r in Fig. 7); the rotation of a volumetric object in yaw usually accompanies a linear change in sway (e.g., when a car makes left or right turn), which is mapped to roll in our substitution rule. Last, camera heave is mapped to chair heave (see the path from a_h to p_h in Fig. 7). This mapping is tailored to three DOF motion chairs that support roll, pitch, and heave motions, which is the most common configuration in 4D theaters [28]. Expert designers also use the same or similar rules (Section 2.2).

Scale factors b and c in Fig. 7 take care of the unit and range differences between ω and a. They also work as gains for motion effects. Soft thresholding is applied to remove small noise during the stationary state.

We further enhance the combined motion commands for roll, pitch, and heave to obtain more dynamic effects by the following nonlinear amplification rule:

$$\hat{m} = \operatorname{sgn}(m)\alpha\left(\frac{\|m\|}{\alpha}\right)^{\beta}, \quad (0 < \beta \leq 1), \tag{8}$$

where m is a motion command, $\hat{m}$ is an enhanced motion command, α is the maximum displacement or angle of that motion, and β is a gain for motion enhancement. Decreasing β makes motion effects more dynamic and fun deviating from the estimated camera motion, while using $\beta = 1$ renders the original motion as it is. According to our experience, the range of β for perceptually-best motion effects is 0.7-0.9 for most 4D films.

In the last step, we limit the motion commands not to exceed the maximum displacement and velocity of the motion chair for safety.

3.3 Synthesis of Slow Camera Motion Effects

The goal of our synthesis algorithm for CS effects, Algorithm CS, is to re-create slow camera motion using a motion chair to provide viewers with an illusion of observing the scene from a distance while slowly moving or floating at the camera position within the space of the scene. To this end,

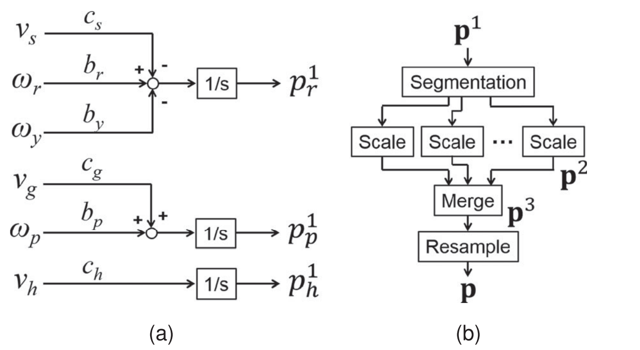


Fig. 8. Motion synthesis algorithm for CS effects. (a) Motion mapping and conversion. (b) Segmentation and scaling.

Algorithm CS transforms continuous camera motion, possibly to one direction for an extended period of time, to swinging chair motion staying within the workspace of the chair, as is done by 4D effects designers (Section 2.2).

After camera motion estimation, Algorithm CS proceeds in two phases. The first phase shown in Fig. 8a computes a 3D camera trajectory $\mathbf{p}^1$ from the linear and angular velocities of camera motion. Using velocities as input enables constant velocity motion, which is not possible with the washout filters for Algorithm CF in the linear directions. Here, surge, sway, and yaw camera motions are substituted by pitch and roll motions as in Algorithm CF. Surge motion is now added to pitch motion since CS effects do not require inertia rendering. Then $\mathbf{p}^1$ is obtained by integration. $\mathbf{p}^1$ is computed for all frames.

Unlike washout filters, the operations in the first phase do not have the convergence property to the initial state. Thus, $\mathbf{p}^1$ may exceed the workspace of the motion chair. We handle this problem by mimicking the 4D designers' strategy in the second phase. For this purpose, $\mathbf{p}^1$ is further segmented and scaled as shown in Fig. 8b. $\mathbf{p}^1$ is divided to many segments using either equi-time or equi-distance segmentation. In the equi-time segmentation, all motion segments have the same duration. In the equi-distance segmentation, all motion segments are enclosed within a sphere of the same radius. In the first and last segments, the motion chair should start from and come back to the origin, respectively, so only a half of the workspace is usable. They are made by using a half duration or a half radius of the enclosing sphere compared to the other segments.

Then, each segment of $\mathbf{p}^1$ is scaled to fit into the workspace of the motion chair. For $\mathbf{p}^1(i)$ $(i_{k-1} \le i \le i_k)$ in the kth segment, the scaled motion command is computed by

$$\mathbf{p}^2(i) = \frac{\mathbf{p}^1(i) - \mathbf{c}^*}{\max\limits_{i \in [i_{k-1}, i_k]} \left\| \mathbf{p}^1(i) - \mathbf{c}^* \right\|} \circ \boldsymbol{\alpha}, \tag{9}$$

where $\boldsymbol{\alpha} = \left[\alpha_r, \alpha_p, \alpha_h\right]^\mathsf{T}$ is the maximum motion range and $\circ$ denotes the element-wise product operator, and

$$\mathbf{c}^* = \begin{cases} \mathbf{p}^1(1) & \text{the first segment} \\ \mathbf{p}^1(N) & \text{the last segment} \\ \underset{\mathbf{c}}{\operatorname{argmin}} \left(\max\limits_{i \in [i_{k-1}, i_k]} \left\| \mathbf{p}^1(i) - \mathbf{c} \right\| \right) & \text{otherwise,} \end{cases} \tag{10}$$

where N is the total number of elements in $\mathbf{p}^1$. The last case of (10) computes the center of the minimum enclosing sphere of the segment.

All scaled segments of $\mathbf{p}^2$ are linked together to $\mathbf{p}^3$ by inserting linearly interpolated points between adjacent segments. C^1 discontinuities in $\mathbf{p}^3$ are not perceptually salient owing to slow chair motion. $\mathbf{p}^3$ is then resampled to ensure that the final motion command $\mathbf{p}$ has the same duration as $\mathbf{p}^1$ before segmentation. An example demonstrating the motion synthesis of Algorithm CS is given in Fig. 11.

Since our motion scaling is distance-to-distance, the equi-distance segmentation preserves the camera velocity more accurately. In the equi-time segmentation, the distance scale from camera to motion chair varies segment by segment, providing continuously moving sensations with less velocity variations. 4D effects designers can choose one of the two segmentation methods suitable to their needs.

It is noted that Algorithm CF and similar previous motion effects rendering algorithm [17] can also provide slowly moving sensations for slow camera motion. However, Algorithm CS synthesizes fundamentally different motion effects. For example, when the camera moves in one direction at a constant velocity, Algorithm CS moves the chair in the same direction at the same velocity (although scaled), but Algorithm CF make zero output due to zero accelerations.

3.4 Synthesis of Impulse and Vibration Effects

The goal of this algorithm, Algorithm V, is to synthesize class V motion effects that respond to the events that involve impulses and vibrations such as gunfire, collision, explosion, and engine vibration. Such events are generally accompanied with sound effects that are perceived rougher and louder than ambient sound or background music. Therefore, we can detect these events by analyzing the perceptual characteristics, roughness and loudness, of sound. The key advantage of this perception-level approach is high detection performance (approximately 90 percent hit rate and 20 percent false alarm rate) for a wide variety of films, even with no need for prior learning. Further details can be found in our previous work [14].

We first compute auditory loudness L and roughness R of sound using the computational models presented in [14]. Then the intensity I of a motion effect is determined by

$$I = c\sqrt{L}R^2 - o, \tag{11}$$

for a scaling constant c and an offset o. Motion commands are triggered only when $I > 0$, that is, when the sound is sufficiently rough and loud.

The original algorithm in [14] was designed for vibrotactile feedback, and its vibration synthesis algorithm is not applicable to motion chairs that have a greatly lower bandwidth of movement. Hence, we use the following motion command tailored to motion chairs. If $I > 0$ for less than 0.4 s, a sine wave with a 2.5 Hz frequency (0.4 s period) is rendered for 0.2 s. If $I > 0$ for a longer period, irregular motion commands are generated using the Perlin noise. These motion commands are also designed to start from and end at zero to provide the convergence property. The magnitude of these motion effects is proportional to I and is determined considering the workspace of the motion chair.

TABLE 2
Parameter Values for FlowLib v3.0

Parameter	Value
model	INTERPOLATE_FAST_HL1
iters	5
warps	2
scale_factor	0.9

3.5 Default Parameters and Implementation Issues

We present the default parameters of our motion effects synthesis algorithms to help reproduction by readers. All motion effects used in the experiments reported in Sections 4 and 5 were generated using the default parameters.

Most parameters for camera motion estimation can be used without tuning. We tested FlowLib v3.0 [24] and Liu's code [29] for optical flow estimation. Although both implementations with their default parameters synthesize good motion effects, we further tuned several parameters of FlowLib v3.0 to speed up computation, as shown in Table 2. All input images were scaled to 640-pixel wide, and the focal length f was fixed to 640.

Scale factors b and c for Algorithm CF (Fig. 7) and Algorithm CS (Fig. 8a) are summarized in Table 3. In Algorithm CF, the b and c gains are tuned while considering the workspace volume. In their default values, b_p and c_g are larger than the other corresponding gains to utilize the larger limit of pitch of our motion chair (pitch: ± 7 degrees; roll: ± 4 degrees; heave: ± 4 cm). Thresholds for noise removal in the steady state were 0.05 for angular velocities and 0.15 for linear accelerations. β is the single design parameter of Algorithm CF that 4D designers control to make motion effects more dynamic (smaller β) or less dynamic (larger β). Its default value is 0.8.

In Algorithm CS, 4D designers determine the segmentation method and the length of segments for each scene. For example, we used the equi-time segmentation in all the experiments we report, and the length of segments varied from 10 to 35 s. The default gains of b and those of c should be the same, respectively, since the motion trajectories of Algorithm CS already make use of the full workspace. The only exception is c_h for heave motion. We use a much smaller value to reflect the fact that human sensitivity to heave motion is much inferior to that for surge or sway, but using high c_h decreases the usable range of roll and pitch motion in our motion chair. This is particularly important for slow motion.

In Algorithm V, the scaling constant c was 0.05, and the offset o was 2.0.

For motion chairs with a higher DOF, the motion substitution rule described in Section 3.2 needs to be revised for the additional axes, but it is generally straightforward. For example, for a four-DOF chair with roll, pitch, yaw, and heave, the yaw velocity ω_y in Fig. 7 is no longer fed to roll motion p_r, but to yaw motion exclusively.

4 Results

The performance of our synthesis algorithms can be best demonstrated by the videos available in supplementary material, which can be found on the Computer Society

TABLE 3
Default b and c for Algorithm CF and CS

Algorithm	b_r	b_p	b_y	c_g	c_s	c_h
CF	1,800	2,200	1,800	150	120	120
CS	400	400	400	1	1	0.1

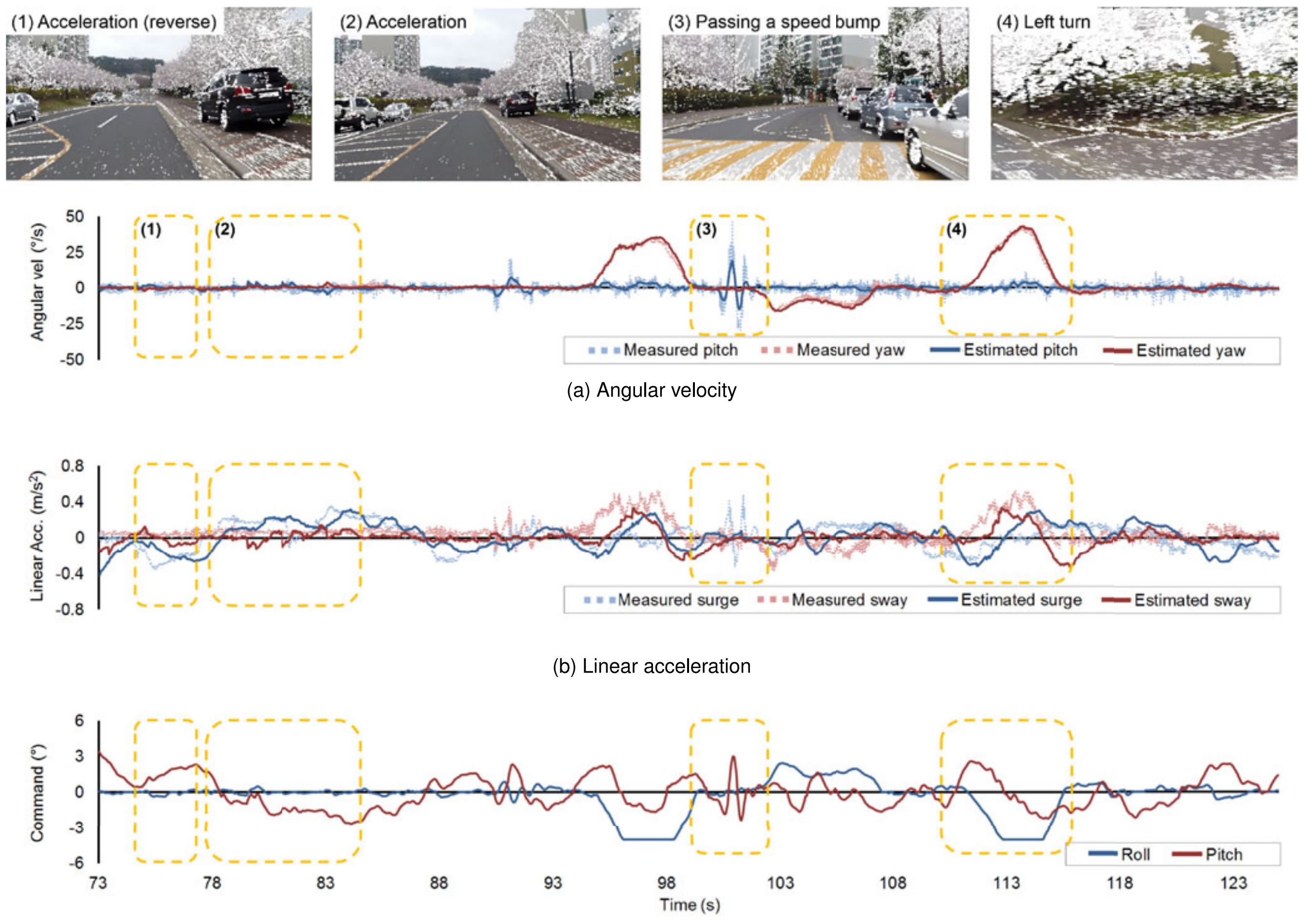
Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2015.2507591. In addition, we present sample results for each algorithm in this section, followed by two user studies in the next section.

4.1 Camera Motion Estimation and Motion Effects

Fig. 9 presents the results of camera motion estimation and CF effects synthesis using a video we recorded in a car driving on a paved city road. The angular velocities and linear accelerations of the camera were measured using an additional motion sensor (CruizCore XA3300; Microinfinity Corp.) for ground-truth generation. Fig. 9a clearly illustrates that our method estimates the angular velocities of camera with high accuracy and almost no delays. According to Fig. 9b, the estimated linear accelerations well reflect the general trends of the true linear accelerations, but with some errors and delays. This is expected since camera translation estimation is inherently less accurate than rotation estimation [30] and acceleration estimation requires differentiation that amplifies noise. This is one of the primary reasons for our motion synthesis algorithms to use all the angular velocities and linear accelerations for robust motion synthesis. Synthesized motion commands in Fig. 9c express the camera motion faithfully: (1) car acceleration (on the reverse) $\rightarrow$ chair pitch increase (leaning forward), (2) car acceleration $\rightarrow$ chair pitch decrease (leaning backward), (3) car passing a speed bump $\rightarrow$ an abrupt and short oscillation in chair pitch, and (4) car left turn $\rightarrow$ chair roll decrease (tilting left), and car deceleration before a curve $\rightarrow$ chair pitch increase and car acceleration after the curve $\rightarrow$ chair pitch decrease.

We further investigated the accuracy of our camera motion estimation algorithm by comparing it with Boujou 5 that was used in [17]. This comparison focused on surge since it is the most difficult DOF to estimate using only monocular images. Recall that whereas the output of our algorithm is local velocities, that of Bonjou is global positions (after full 3D trajectory reconstruction). For fair comparison, we used the first 20 seconds of the driving video (Fig. 9) on the straight lane before a large orientation change. The velocities estimated by our algorithm were converted to positions using the cumulative trapezoidal numerical integration, and the positions computed by Boujou were converted to velocities using the symmetric derivative. Results are provided in Fig. 10, where the two methods showed very similar position and velocity profiles of surge. This is expected to some extent since Boujou in its first step includes essentially the same procedure of camera pose estimation in the local coordinate frame [22], [31].

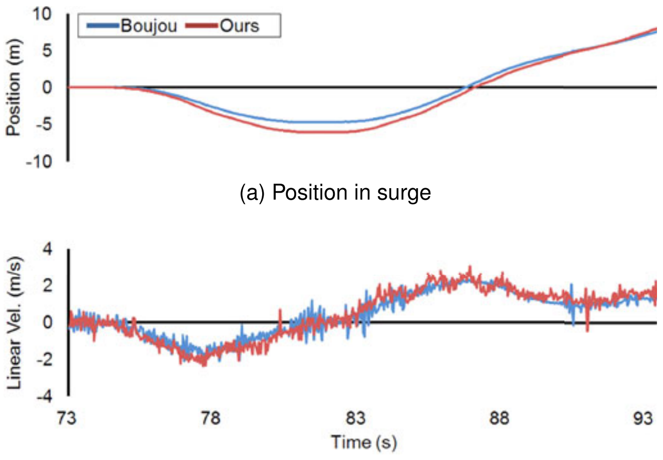
Fig. 11 illustrates an example result of Algorithm CS. The motion effects are generated for a scene in Frozen in which Elsa builds an ice castle. The motion chair is tilted forward slowly when the camera approaches to the castle (a red



(c) Motion effects generated by Algorithm CF

Fig. 9. A comparison between measured and estimated camera motions (a, b) and synthesized CF motion effects (c). Highlighted parts (1)-(4) are also shown in upper images. Note that the high-frequency noise in the measured camera motion is injected from ambient noise in the car.

segment in Fig. 11). Then, the chair is tilted backward because the camera faces up to the ceiling, and then it is tilted forward again when the camera looks down on the floor (a blue segment). The left and right roll motions describe the counter-clockwise and clockwise rotations of the camera, respectively. In this scene, the focal length also changes due to zooming. Since we used a fixed focal length model (see **K** in Section 3.1.2), the zooming is replaced with forward or backward translation, which in practice matches zoom-in or zoom-out well, respectively. As shown in Fig. 11b, the CS motion effects fully utilize the workspace of the motion chair.



(b) Velocity in surge

Fig. 10. Comparison of surge motions estimated by our camera motion estimation algorithm and Boujou 5 before smoothing or filtering. The same video used in Fig. 9 was used as input.

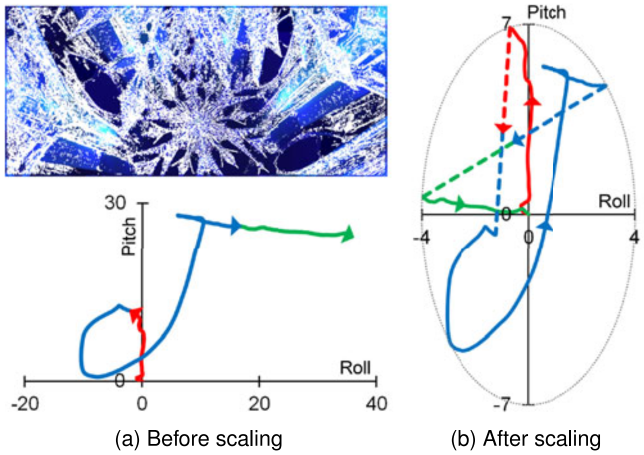


Fig. 11. Motion effects generated with Frozen by Algorithm CS with the equi-time segmentation. Red, blue, and green lines are for the first, second, and third segments, respectively. In (b), dashed lines are interpolated motions to connect the segments. A dotted ellipse represents the workspace of our motion chair.

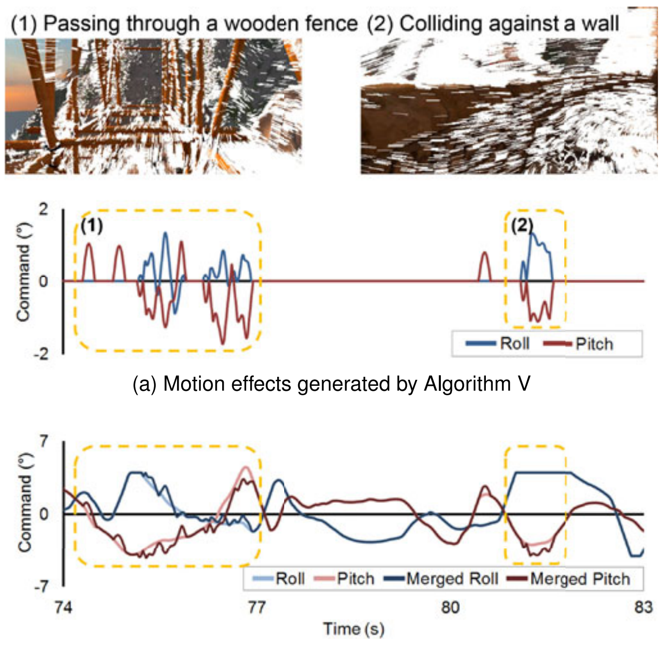


Fig. 12. Results of Algorithm V (Wall of China; ride film, Simuline Inc.).

Results of Algorithm V are demonstrated in Fig. 12. When a tricycle breaks through a wooden fence (Fig. 12(1)) and hits against a wall (Fig. 12(2)), irregular motion effects are generated by detecting special sound effects (Fig. 12a). Then the effects are merged with the motion effects synthesized by Algorithm CF, and the combined effects feel more realistic (Fig. 12b).

4.2 Computational Complexity

We measured the computational complexity of our synthesis algorithms on a commodity PC (second generation Intel i5-2400 3.10 GHz CPU, 4.00 GB RAM, and NVIDIA GeForce GTX 760 graphics card), and the results are summarized in Table 4. All input frames were resized to be 640-pixel wide, and optical flow is computed on NVIDIA CUDA. Most steps take constant times, except RANSAC and the computation of **R** and **t** in Section 3.1.2. The computational times of the latter two depend on the number of corner points.

The number of corner points can be adjusted by a threshold (Section 3.1.1). In our implementation, using 10,000 corner points allows robust camera motion estimation. In this case, our algorithms achieve approximately four frame/s, which is sufficient for our use scenario (Section 2.3). For example, for a film that has 24 frames per second, it takes only 3 min to synthesize a motion effect for a 30-s scene (usual motion effects are greatly shorter). A better use strategy is to execute the camera motion estimation that requires expensive computations during the scene breakdown stage. Then in the editing stage the designers can generate various motion effects using the motion mapping algorithms almost instantly (Table 4).

5 User Experiments

We carried out two user experiments to assess the perceptual quality of our motion effect synthesis algorithms, one with general users and the other with 4D experts.

TABLE 4
Execution Times (ms) for One Frame

# of corner points	< 5,000	< 15,000	< 30,000
Optical flow		103.892	
Cut detection		0.131	
Corner detection		0.503	
RANSAC	104.653	173.521	328.953
R & **t** extraction	2.196	11.701	29.381
Kalman filter		0.013	
CF motion mapping		0.032	
CS motion mapping		0.048	
Algorithm V		0.468	

5.1 Experiment I: General Users

The goal of Experiment I was to compare the immediate benefits that general users perceive from three different sets of motion effects: those randomly generated, synthesized by our algorithms, and manually designed by 4D experts.

5.1.1 Methods

The participants were eighteen students (13 male and 5 female; 18-32 years old) recruited from the authors' institution. None of them reported prior involvement in 4D effects production. Some participants had limited experiences of watching 4D films, eight times at most. Each participant was paid 10 USD after the experiment.

The motion chair (4DX; CJ 4DPLEX) used in the experiment had three DOFs for roll (± 4°), pitch (± 7°), and heave (± 4 cm). 3D images were projected onto a 94-inch screen by an EB-W16SK polarized 3D projector (Epson corp.). The participants wore passive 3D glasses during the experiment. Sound was played through NS-150 5-channel home theater speakers (Yamaha corp.).

Three 4D films were used in the experiment: Wall of China, Fairy Balloon Ride, and Frozen. The first two were ride films with running times of 3′59″ and 4′12″. For Frozen, a 3′36″ clip in which the heroine Elsa sings the main song "Let it go" was used.

The motion effects for Wall of China and Fairy Balloon Ride were synthesized using Algorithm CF. Algorithm V was also used for Wall of China. The motion effects for Frozen were generated by Algorithm CS. These automatic motion effects (AE) were compared with two other conditions, randomly-generated effects (RE) and manually-designed effects (ME). RE were made using Perlin noise so that they had similar strength and frequency to ME. RE served as a baseline in the experiment. ME for Wall of China and Fairy Balloon Ride were commercial ones that had been played at 4D attractions, and they were purchased from Simuline. ME for Frozen were provided by an anonymous vendor under the condition that its identity must remain undisclosed because of copyright. It is noted that the manual effects for Frozen included class O motion effects, as well as class C effects. This is in contrast to AE for Frozen that included only CS effects.

The experiment consisted of three sessions, one for each 4D film. In each session, one 4D film was presented three times with different sets of motion effects. The order of the 4D films and the motion effect sets was randomized for each participant. After each session, the participant took at

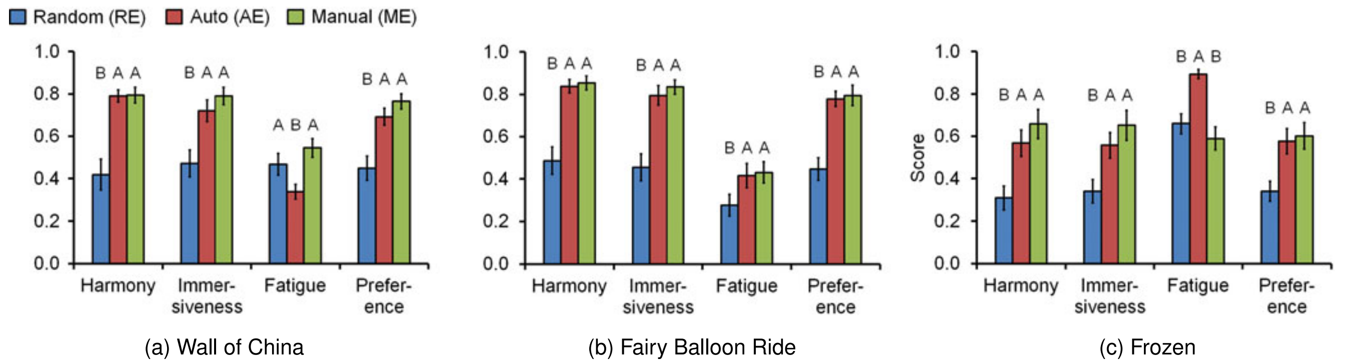


Fig. 13. Results of Experiment I with general users. Error bars represent standard errors. Note that the scale for fatigue is inverted so that the higher the fatigue score, the better the performance (less tiring), to be consistent with the others. The sets of motions effects marked with the same alphabets indicate that they did not show statistically significant differences by Tukey's HSD test.

least 5-min rest to prevent motion sickness and fatigue. The participant finished the experiment in one hour.

After watching each 4D film sitting in the 4D chair, the participants answered a questionnaire. It included the following four questions: Harmony—"Did the motion effects match to the film?"; Immersiveness—"Did the motion effects help you be immersed in the film?"; Fatigue—"Did the motion effects make you feel tired?"; and Preference—"Did you like the motion effects provided with the film?". All the questions were rated on a continuous scale 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, "very inharmonious" at the left end and "very harmonious" at the right end for harmony.

5.1.2 Results

Experimental results are shown in Fig. 13. We performed one-way ANOVA on each question using motion effect set as an independent variable for each 4D film. Motion effect set was statistically significant ($p < 0.05$) for all the subjective metrics and all the 4D films. Tukey's HSD tests were then used for post-hoc multiple comparisons, and their results are also represented in Fig. 13.

ME showed the best scores in all the metrics and the conditions, except fatigue for Frozen. AE generally received comparable scores to ME. Significant differences were observed in only fatigue for Wall of China and Frozen. In fact, ME felt more tiring than AE for Frozen. It is presumably because ME presented much more motion effects than AE in that particular condition. As mentioned earlier, ME for Frozen included both C and O motion effects, while AE provided only CS effects. Between AE and RE, AE obtained significantly higher scores in all the metrics and the conditions, with one exception in fatigue for Wall of China.

To gain further insights, Fig. 14 presents exemplar motion effects of RE, AE, and ME alongside the dominant camera motions annotated by a human viewer. It can be seen that the roll and pitch motion commands of RE are not correlated with the dominant camera motion, but those of both AE and ME express the dominant camera motion effectively. For example, left camera turn (labeled by L) led to decreasing chair roll (tilting left), right camera turn (R) to increasing chair roll (tilting right), camera descent (D) to increasing chair pitch (leaning forward), and camera ascent (A) to decreasing chair pitch (leaning backward). There are also some specific differences between AE and ME. For example, ME includes two small oscillations (Figs. 14(1) and (2)) inserted to express the impact caused by collisions to a snowball in the movie. Such effects are not generated by AE since Algorithm V is not able to detect collisions of weak sound effects. In addition, ME is generally smoother than AE. An obvious example is marked in Fig. 14(3); it appears that the expert designers made motion effects for the sudden ascent of a balloon much smoother than its actual value, probably for the concern of motion sickness.

In conclusion, in terms of perceptual quality to general users, the motion effects synthesized by our algorithms outperformed the random effects by large extent, and were comparable to the commercial manual effects, at least for the tested 4D films, although the manual effects included more thoughtful, subtle merits.

5.2 Experiment II: 4D Experts

Inspired by the promising results of Experiment I, Experiment II aimed at gauging the quality of motion effects

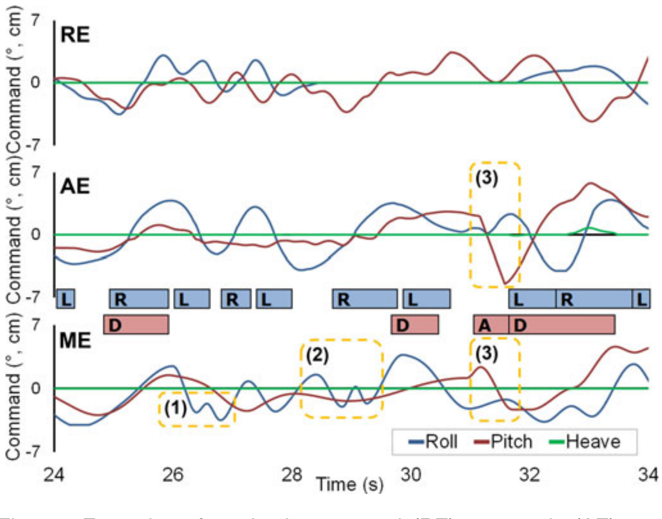
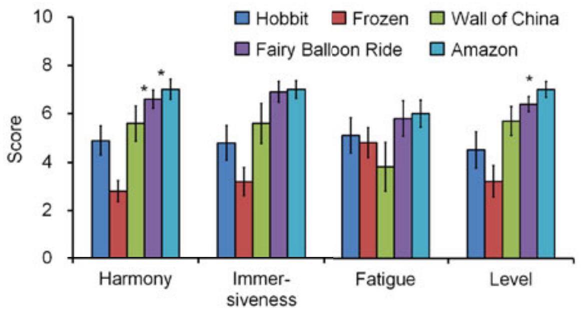
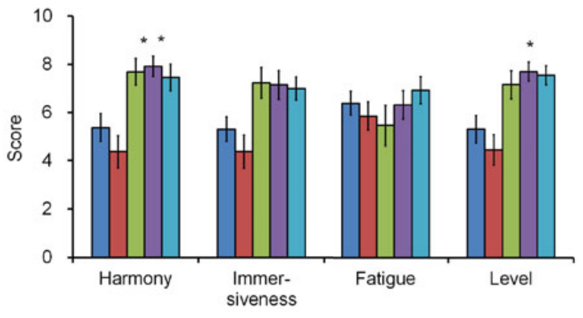


Fig. 14. Examples of randomly-generated (RE), automatic (AE), and manually-designed (ME) motion effects (Fairy Balloon Ride; a ride film from Simuline Inc.). Blue and red boxes represent the dominant camera motions that were manually annotated by a human viewer. The blue boxes indicate horizontal motions (L: left turn and R: right turn; related to roll), and the red boxes are for vertical motions (D: descent and A: ascent; related to pitch).



(a) Designers (DG)



(b) The others (OG)

Fig. 15. Results of the user study with 4D experts. Error bars represent standard errors. The scale for fatigue is inverted as in Fig. 13. Asterisks indicate that the results between DG and OG had statistically significant differences.

synthesized by our algorithms with the careful eyes of 4D effects experts and designers.

5.2.1 Methods

Twenty-three 4D experts (19 male and 4 female; 26-43 years old) who were working in CJ 4DPLEX participated voluntarily in this evaluation. Ten of them were responsible for 4D effects production, nine of them for software or hardware development, and the other four for planning. On average, the participants had 34 months of work experience in their current position. All of them had ample experience in making or evaluating 4D effects.

In addition to the three 4D films used in Experiment I, two more 4D films, The Hobbit: The Desolation of Smaug (45″; when Gandalf climbed a mountain to meet Radagast) and Amazon (4′11″), were used in this experiment. Algorithm CS was used to generate motion effects for the Hobbit, and Algorithm CF was used for Amazon. The motion effects for the other three 4D films were the same as those used in Experiment I.

Since our expert participants could distinguish details in motion effects, they were very likely to notice whether motions effects were manually designed or automatically synthesized. This can instill a substantial bias to blind comparisons. Therefore, in this experiment, the participants were asked to make absolute assessments on the quality of only automatic motion effects. Instead, we divided the participants into two groups, the designers group (DG) and the others (OG) for between-group comparisons. DG were expected to give lower scores than OG due to their higher standards.

The questionnaire was the same as that of Experiment I, except for the last question. It was replaced with: Level-"What was the overall level of the motion effects?". All questions were rated using a score between 0 and 10. The point 10 meant that the motion effects were in the same level as the final product designed by 4D experts. We also had interviews with the participants regarding the quality of our synthesized motion effects after the experiment. The experiment took about 45 min per participant.

5.2.2 Results

The results of the evaluation are shown in Fig. 15. Overall, the CF motion effects for the three 4D rides were generally well received; the scores of the four measures ranged from 3.8 to 7.0 with DG and from 5.5 to 8.0 with OG. The lowest scores were for fatigue. The scores for level were 5.7-7.0 with DG and 7.2-7.7 with OG. These scores, especially those of OG, can be considered as very high; people are generally reluctant to give the highest or lowest score in a questionnaire.

The CS motion effects for the two regular 4D films obtained lower yet moderate scores: 2.8-5.1 with DG and 4.4-6.4 with OG. The scores for level were 3.2-4.5 with DG and 4.5-5.3 with OG. Participants commented that the motion effects for Frozen and Hobbit received lower scores than the others because of the absence of class O effects for the motion of the main character. Since less O effects were expected in Hobbit due to its short playback time, its scores were higher than those of Frozen.

The scores between DG and OG were compared using a paired t-test ($\alpha = 0.05$) for each 4D film and each subjective metric. Despite the generally lower scores of DG, significant differences were found only in harmony for Wall of China and Fairy Balloon Ride and in level for Fairy Balloon Ride.

In the participant interviews, all experts agreed that our algorithms would be helpful for motion effects production. They also commented that the level of our synthesized motion effects was greatly higher than their initial expectation. Overall, the evaluation results with 4D experts are promising for the applicability of our synthesis algorithms to actual 4D effects production.

6 Discussion

Our work has been carried out independently of Shin et al. [17]. As such, there exist both similarities and dissimilarities between the two system. Most of all, our synthesis algorithms are tailored to the extensive surveys that we executed for a systematic classification of various 4D motion effects in actual use and their design practice in the industry. This approach is expected to improve immersiveness of 4D films most effectively and also the likelihood of our algorithms being adopted by 4D effects designers. In contrast, the framework in [17] is based more on the conventional principles of motion simulation for training purposes, in which the provision of physically exact sensory stimuli has the top priority. This appears to be the most fundamental underlying difference between the two systems.

For example, both systems provide motion effects synthesis algorithms for CF effects, but only our system considers CS and V effects. CF effects have been used in training simulators and 4D attractions for a long time, but the other two are relatively new effects and more frequently used than CF effects in regular 4D films (Fig. 4). As for CF effects, the two systems follow the same approach. However, in our system, the computation of camera velocity and acceleration is done directly from the relative camera motion in the local coordinate frame obtained from two consecutive images. In [17], it is done from the 3D camera position in the world frame estimated by an additional reconstruction step from the relative camera motions. This method resembles VR-based motion simulations where the 3D trajectories of all virtual objects are known, which enables the rendering of sustained low-frequency forces such as gravity and centrifugal force. Rendering these forces is important in the applications that require physically exact simulations such as flight simulators. However, the computational load inevitably becomes greatly higher, which significantly degrades usability as authoring software that calls for iterative design. Although our method is much more efficient, it cannot estimate fictitious forces applied to the camera, sacrificing gravity and centrifugal force rendering. This was a strategic decision made considering that scenes where only gravity or centrifugal force is dominant are scarce in 4D films. Moreover, the workspace of commercial motion chairs is quite limited and not suitable for clear tilt coordination effects, as described earlier in Section 3.2. This difference, and also the fact that all our algorithms are integrated into one optimized system while the system in [17] requires several independent software packages to be used in a series, seem to contribute to the substantial difference in synthesis efficiency; for example, our Algorithm CF is 50 times faster than [17] (a rough comparison between Section 4.2 of our paper and Section 6 of [17]).

In addition, our current synthesis algorithms have the following limitations: 1) Algorithm CF sometimes does not generate sufficiently strong motion commands for the acceleration/deceleration and the heave motion of camera. This stems from the inherent limitation in camera motion estimation that translation estimation is much less accurate than rotation estimation; 2) Algorithm CS offers only one option (linear interpolation) for connecting motion segments. More options (e.g., cubic interpolation) might be necessary for 4D designers' selection to consider the context; 3) Algorithm V also responds to rough human voice, although this can be avoided by careful film segmentation prior to effect design; and 4) The presence of very abrupt camera motion and significant lighting variation may substantially increase errors in optical flow, and such errors can inject perceivable false cues to motion effects, although such extreme occasions are rare in movies.

7 Conclusions

In this paper, we have presented synthesis algorithms that generate motion effects for 4D film. After categorizing motion effects into six classes based on surveys, synthesis algorithms for three classes of motion effects are described. Our algorithms estimate camera motion and then transform it to vestibular feedback using a washout filter for dynamic POV shots, or to chair motion using motion segmentation and scaling for slowly changing scenes. Impulses and vibrations are provided by the detection of special sound effects and associated modality conversion. Various motion effect examples made by our algorithms are provided using plots and videos. Moreover, user experiments with both general users and 4D experts have indicated that our algorithms are capable of synthesizing compelling motion effects in terms of perceptual quality.

As future work, we plan to improve the limitations of our algorithms. For example, we expect that using stereo images of 3D films can significantly increase the accuracy of camera translation estimation. We will also extend the coverage to class O effects, the remaining 61.7 percent motion effects of regular 4D films.

Acknowledgments

This work was supported by a research grant NRF-2013R1A2A2A01016907 from the National Research Foundation (NRF) of Korea. The authors greatly appreciate CJ 4DPLEX for their generous help and advices. They also thank Hojun Cha for developing a tool that generates random motion effects. Seungmoon Choi is the corresponding author.

References

[1] CJ 4DPLEX, "Absolute cinema experience, 4DX," Brochure.
[2] M. A. Nahon and L. D. Reid, "Simulator motion-drive algorithms: A designer's perspective," *J. Guidance, Control, Dynamics*, vol. 13, no. 2, pp. 356–362, 1990.
[3] A. Israr and I. Poupyrev, "Tactile brush: Drawing on skin with a tactile grid display," in *Proc. ACM Conf. Human Factors Comput. Syst.*, 2011, pp. 2019–2028.
[4] T. Nakamoto and H. P. D. Minh, "Improvement of olfactory display using solenoid valves," in *Proc. IEEE Virtual Reality*, 2007, pp. 179–186.
[5] T. Moon and G. J. Kim, "Design and evaluation of a wind display for virtual reality," in *Proc. ACM Symp. Virtual Reality Softw. Technolog*, 2004, pp. 122–128.
[6] K. Hirota, S. Ebisawa, T. Amemiya, and Y. Ikei, "A system for creating the content for a multi-sensory theater," in *Proc. Virtual Mixed Reality (Held as Part of HCI Int.)*, 2011, pp. 151–157.
[7] J. Kim, C.-G. Lee, Y. Kim, and J. Ryu, "Construction of a haptic-enabled broadcasting system based on the MPEG-V standard," *Signal Process.: Image Commun.*, vol. 28, no. 2, pp. 151–161, 2013.
[8] R. V. Parrish, J. E. Dieudonne, and R. L. Bowle, "Coordinated adaptive washout for motion simulators," *J. Aircraft*, vol. 12, no. 1, pp. 44–50, 1975.
[9] A. Yamashita, "Theaters of illusion: The continuing evolution of entertainment simulation," *ACM SIGGRAPH Comput. Graph.*, vol. 28, no. 2, pp. 142–144, 1994.
[10] B. Yoo, M. Cha, and S. Han, "A framework for a multi-sensory VR effect system with motional display," in *Proc. Int. Conf. Cyberworlds*, 2005, pp. 237–244.
[11] Y. Kim, J. Cha, J. Ryu, and I. Oakley, "A tactile glove design and authoring system for immersive multimedia," *IEEE MultiMedia*, vol. 17, no. 3, pp. 34–45, Jul. 2010.
[12] S.-K. Kim, "Authoring multisensorial content," *Signal Process.: Image Commun.*, vol. 28, no. 2, pp. 162–167, 2013.
[13] M. Waltl, B. Rainer, C. Timmerer, and H. Hellwagner, "An end-to-end tool chain for sensory experience based on MPEG-V," *Signal Process.: Image Commun.*, vol. 28, no. 2, pp. 136–150, 2013.
[14] J. Lee and S. Choi, "Real-time perception-level translation from audio signals to vibrotactile effects," in *Proc. ACM Conf. Human Factors Comput. Syst.*, 2013, pp. 2567–2576.
[15] M. Kim, S. Lee, and S. Choi, "Saliency-driven real-time video-to-tactile translation," *IEEE Trans. Haptics*, vol. 7, no. 3, pp. 394–404, Jul.-Sep. 2014.
[16] I. Lee and S. Han, "Estimating the 3d posture of 6dof platform from image sequences," in *Proc. Int. Symp. Adv. Intell. Syst.*, 2007, pp. 894–897.

[17] S. Shin, B. Yoo, and S. Han, "A framework for automatic creation of motion effects from theatrical motion pictures," *Multimedia Syst.*, vol. 20, pp. 327–346, 2013.

[18] F. Danieau, J. Fleureau, P. Guillotel, N. Mollet, M. Christie, and A. Lecuyer, "Hapseat: Producing motion sensation with multiple force-feedback devices embedded in a seat," in *Proc. ACM Symp. Virtual Real. Softw. Technol.*, 2012, pp. 69–76.

[19] F. Danieau, J. Fleureau, P. Guillotel, N. Mollet, M. Christie, and A. Lecuyer, "Toward haptic cinematography: Enhancing movie experience with haptic effects based on cinematographic camera motions," *IEEE MultiMedia*, vol. 21, no. 2, pp. 11–21, Apr.-Jun. 2013.

[20] F. Danieau, A. Lécuyer, P. Guillotel, J. Fleureau, N. Mollet, and M. Christie, "Enhancing audiovisual experience with haptic feedback: A survey on HAV," *IEEE Trans. Haptics*, vol. 6, no. 2, pp. 193–205, Apr.-Jun. 2013.

[21] S. Choi and K. J. Kuchenbecker, "Vibrotactile display: Perception, technology, and applications," *Proc. IEEE*, vol. 101, no. 9, pp. 2093–2104, Sep. 2013.

[22] R. Hartley and A. Zisserman, *Mutiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[23] M. Mainberger, A. Bruhn, and J. Weickert, "Is dense optic flow useful to compute the fundamental matrix?" in *Proc. Int. Conf. Image Anal. Recog.*, 2008, pp. 630–639.

[24] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, "Anisotropic huber-L1 optical flow," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 1–7.

[25] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vis.*, vol. 92, no. 1, pp. 1–31, 2011.

[26] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.

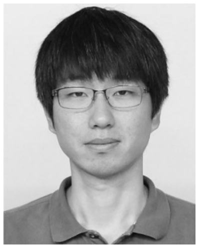
[27] F. E. Guedry, "Psychophysics of Vestibular Sensation," in *Vestibular System Part 2: Psychophysics, Applied Aspects and General Interpretations,* H. H. Kornhuber, Ed. New York, NY, USA: Springer, 1974, pp. 3–154.

[28] G. Whitehead, "The technological art of simulation," *SMPTE Motion Imaging*, vol. 110, pp. 39–42, 2001.

[29] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Instit. of Technol., Cambridge, MA, USA, May 2009.

[30] V. Rodehorst, M. Heinrichs, and O. Hellwich, "Evaluation of relative pose estimation methods for multi-camera setups," in *Proc. Int. Archives Photogrammetry, Remote Sens. Spatial Inf. Sci.*, 2008, pp. 135–140.

[31] D. P. Robertson and R. Cipolla, "Structure from motion," in *Practical Image Processing and Computer Vision*, M. Varga, Ed. Hoboken, NJ, USA: Wiley, 2009.



Jaebong Lee received the BS degree in mathematics with a minor in computer science and engineering at POSTECH in 2008 and the MS degree in computer science and engineering at POSTECH in 2010. Currently, he is working toward the PhD degree in the Department of Computer Science and Engineering at POSTECH. He was a software research engineer at LG Electronics Inc. in 2010 to 2012. His research interests include haptics, virtual reality, and computer vision. He is a student member of the IEEE.

Bohyung Han received the BS and MS degrees from the Department of Computer Engineering at Seoul National University, Korea, in 1997 and 2000, respectively, and the PhD degree from the Department of Computer Science at the University of Maryland, College Park, MD, in 2005. He was with the research staff at the Samsung Electronics Research and Development Center, Irvine, CA, and Mobileye Vision Technologies, Princeton, NJ. He is currently an associate professor with the Department of Computer Science and Engineering at POSTECH, Pohang, Korea. He served as an area chair in ACCV 2012 and WACV 2014, and an area chair and a demo chair in ACCV 2014. His current research interests include computer vision, machine learning, pattern recognition, and computer graphics. He is a member of the IEEE.

Seungmoon Choi received the BS and MS degrees in control and instrumentation engineering from Seoul National University in 1995 and 1997, respectively, and the PhD degree in electrical and computer engineering from Purdue University in 2003. He is a professor of computer science and engineering at the Pohang University of Science and Technology (POSTECH). He received a 2011 Early Career Award from IEEE Technical Committee on Haptics and several best paper awards from major international conferences. He was a co-chair of the IEEE Technical Committee on Haptics in 2009 to 2010. He serves/served in the editorial board of *IEEE Transactions on Haptics*, *Presence*, *Virtual Reality*, and *IEEE Robotics and Automation Letters*. He is the general co-chair of IEEE Haptics Symposium in 2014 and 2016 and was the program chair of IEEE World Haptics 2015. His research interests lie on haptic rendering and perception, both in kinesthetic and tactile aspects. His basic research has been applied to mobile devices, automobiles, virtual prototyping, and motion-based remote controllers. He is a member of the IEEE.