Noname manuscript No.
(will be inserted by the editor)

Interactive Motion Photography from a Single Image

Okihide Teramoto · In Kyu Park · Takeo Igarashi

Received: date / Accepted: date

Abstract In this paper, we present an interactive and computational method to create a 'motion photograph' from a single image. A motion photograph is a still image in which moving objects are depicted as informative as well as motion-evocative, which is inspired by cartoon arts. Given an input image, which contains moving objects but captured as static, the user can add and edit various motion effects to the objects by a simple but efficient user interface. As a result, the edited object conveys an effective motion effect without blurring the object. The proposed system is user-friendly so that novice users are able to create motion photographs without any special skills. Furthermore, the system is extensible so that any new effect can be developed and plugged in. Experimental results and user study show that the proposed motion photography system produces a variety of interesting motion photographs. Compared with the general image editing tool (Photoshop), the proposed system creates high-quality motion photographs in significantly shorter time.

O. Teramoto
The University of Tokyo, Tokyo 113-0033, Japan
E-mail: tera1984@gmail.com

I. K. Park
School of Information and Communication Engineering,
Inha University, Incheon 402-751, Korea
E-mail: pik@inha.ac.kr

T. Igarashi
The University of Tokyo, Tokyo 113-0033, Japan
E-mail: takeo@acm.org

1 Introduction

During the last century, conventional photography has been the most efficient method to store visual information of a scene as it appears. Billions of beautiful scenes, natural phenomena, human activities, and scientific/archeological treasures are recorded in billions of photographs. Furthermore, professional photographers create artistic and creative effects on the photographs using camera control mechanisms, such as focal length, shutter speed, aperture, and ISO.

Over the last decade, digital cameras have been replacing conventional film cameras. Digital cameras have several advantages over conventional film cameras. First, the acquired photographs are digitally stored in computer memory without loss of sensed information. Consequently, they are easily copied and shared by other devices. Second, digital postprocessing can be performed using recent image processing and computer graphics methods. Adobe's Photoshop [10] is a powerful tool that is used in popular for this purpose, *i.e.* enhancing the quality, as well as editing the photograph content interactively.

However, although the optical, mechanical, and electronic capabilities of digital cameras have been constantly improved, a spectrum of photographs exists that current cameras can never capture. In the recent research area of 'computational photography', people try to generate images computationally that a conventional camera can never capture.

In this paper, we address an interesting problem of how effectively the motion of an object can be represented in a still image. A conventional camera can take a picture of fast moving objects. However, they can hardly convey the proper information of the object itself and its motion together. For example, the acquired

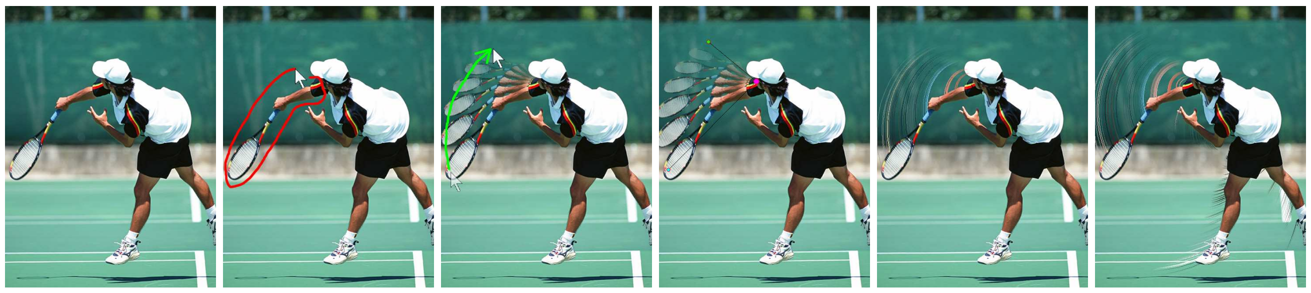


Fig. 1 Overall procedure for interactive motion photography. (a) Input image. (b) User interface for part segmentation. (c) Motion editing. (d)∼(f) Motion photographs with different effects.

photos are usually either static (when the shutter speed is short enough) or blurred. That is, conventional photographs of fast moving object are either (1) informative but not motion-evocative or (2) motion-evocative but not informative [5]. Note that the object can rarely be identified due to the severe motion blur in the latter case.

Conversely, cartoon artists deliver motion feeling efficiently by employing *motion depiction* techniques. In this paper, we adapt the basic idea of motion depiction and apply it to the real image as shown in Figure 1. We refer to this as *motion photography.* Unlike previous work [14,9,16,13,4,15,1] in which the input is either 3D keyframe animation or a video sequence, a single still image is used as input. An informative, as well as evocative, motion photograph is created in real-time using the proposed interactive method that is highly user-friendly and efficient. Our system is able to generate impressive and dynamic motion effects in a still image. In addition, we extend the basic idea to multiview images. By correctly transferring the motion effect of an image to others in the multiview image set using camera matrices and homography, it is possible to generate multiview motion photographs that are consistent to each other in terms of motion evocativeness from different viewpoints.

This paper is organized as follows. In the following two sections, previous related work and the overall structure of the proposed system are introduced. Subsequently, the proposed interactive motion photography is described in detail. Then the experimental results are shown and discussed. The last section concludes the paper.

2 Previous work

Traditional research focused on generating non-photorealistic motion effects [14,12,9,16,15] or artificial motion blur [2] for an extracted frame of 3D animation. They differ from our work since they need keyframe-based 3D animation as their input. However, they suggested practical use of speed-lines and repeated contours to achieve the motion effects in a still image.

A seminal paper by Cutting [5] provides an intensive survey of representing motion in a still image. He introduced traditional methods, such as dynamic balance, stroboscopic sequence, forward lean, photographic blur, speed-lines, and panning that have been popular in many historic paintings, artistic photographs, and cartoons. He addressed historical issues in motion representation and summarized the advantages and disadvantages of each method.

Kim and Essa [13] proposed a non-photorealistic method to illustrate motion effects on a selected frame in an input video sequence. They developed motion effects, such as time-lapse ghost, temporal-flare, particle-effect, and speed-lines. The main difference to our work is that they need a video sequence and do not support an interactive authoring mechanism.

Bennett and McMillan [1] proposed a content-aware video reduction method for a time-lapse video. Unlike uniform sampling of video frames where objects are flickering constantly, the frames are sampled adaptively based on the amount of active motion. The trails of objects in motion are added on the frames in the shortened video to convey the original motion information (usually speed and direction). As a result, they effectively reduce hours of time-lapse video into a few seconds of short video, whilst preserving important motion correctly. This is similar to our work in that motion is depicted in a non-photorealistic manner using ghost trails. However, they need a video sequence as input, and do not support a motion effect editing mechanism or various kinds of motion effects, as in this work. Their main objective is to reduce the time-lapse video computationally, not the interactive motion photography.

Freeman *et al.* [6] proposed an image processing method of changing the phase of image pixels using steerable filters. The static image is perceived as moving in a specified direction by changing the phase continuously. However, only horizontal or vertical direction of small motion is perceived due to the nature of the filter. Moreover, it is hard to maintain the original intensity and color of the input image.

Chi *et al.* [3] proposed an interesting method to illustrate a motion field using a visual illusion phenomenon called 'Rotating Snake'. The user can perceive the flow of a motion field even though the pattern itself is static. However, it is hard to control the speed of motion and even impossible to simulate the illusion whilst maintaining the original pixel color.

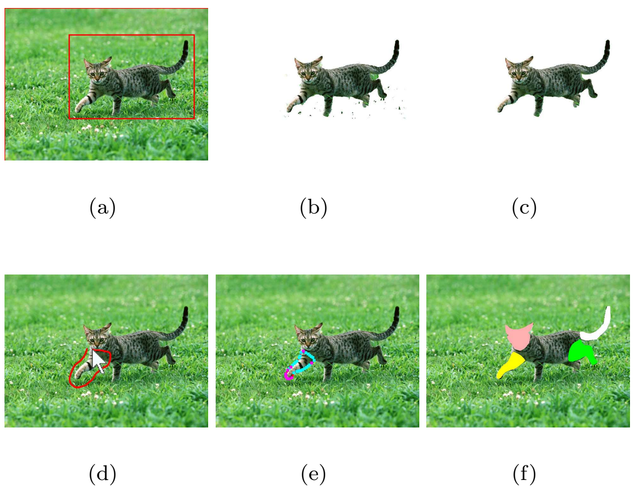


Fig. 2 Interactive segmentation. (a) Input for GrabCut method (b) Initial result of GrabCut. (c) Refined result of GrabCut. (d) Input for interactive Snake method. (e) Result of segmenting the cat's leg. (f) Result of multi-parts segmentation.

3 System Overview

The proposed system consists of three components: Interactive segmentation, interactive motion depiction, and motion transfer (in case of a multiview motion photograph). Given an input image, the object of interest is first segmented from the background. It is assumed that the input image does not have motion blur, *i.e.* the input image is informative but not motion-evocative. Depending on the motion effect we want to depict, the object can be segmented partially (*e.g.* leg or arm of a player) or as a whole object. After segmentation, the selected motion effect is rendered using the proposed user interface. The details of the motion effect (*e.g.* motion direction, speed, and effect types) are easily edited on-the-fly with only a few mouse clicks. Optionally, when the input is a multiview image set, the editing motion in a single image generates the corresponding motion effects for other images simultaneously, based on the homography between images. The details will be described in the following section.

4 Interactive segmentation

Before applying a motion effect, the target object or object part is segmented using a simple user interface. Since the main contribution of this paper is not the segmentation technique, we employ a couple of existing standard algorithms (active snakes [11] and *i.e. GrabCut* [17]) and modify them to interactive ones. The implemented system is highly extensible. Any other segmentation algorithms can be implemented and substituted for existing ones.

If a whole (rigid) object is going to be segmented, GrabCut is more efficient, since only the bounding rectangle needs to be specified. An example of segmentation using GrabCut is shown in Figure 2 (a)∼(c). Conversely, if the object has articulated motion, partial segments are usually extracted to apply a motion effect segment-by-segment. In this case, the active snake method is more flexible since the initial bounding region can be arbitrary shaped.

The interactive snake method works as follows. When the user draws strokes, the list of pixels are stored as vertices as shown in Figure 2 (d). They are represented by the mass-spring model, in which neighboring vertices are connected to each other by springs. During the iteration, the region shrinks to the direction of reducing the cost function defined by the difference of the pixel color between the consecutive steps of the movement (Figure 2 (e)). When the energy is sufficiently low, the vertex moves to the next coordinate. After convergence, the user can correct the position of the vertices manually for fine tuning. The user can lock some vertices to prevent them from moving; this is useful if the user wants to cut a part of the foreground object (*e.g.* the leg of the cat in Figure 2). In this case, the target region might straddle the border between the background and the foreground of the entire object. The part-by-part segmentation can be performed by consecutively applying the proposed procedure ((Figure 2 (f)).

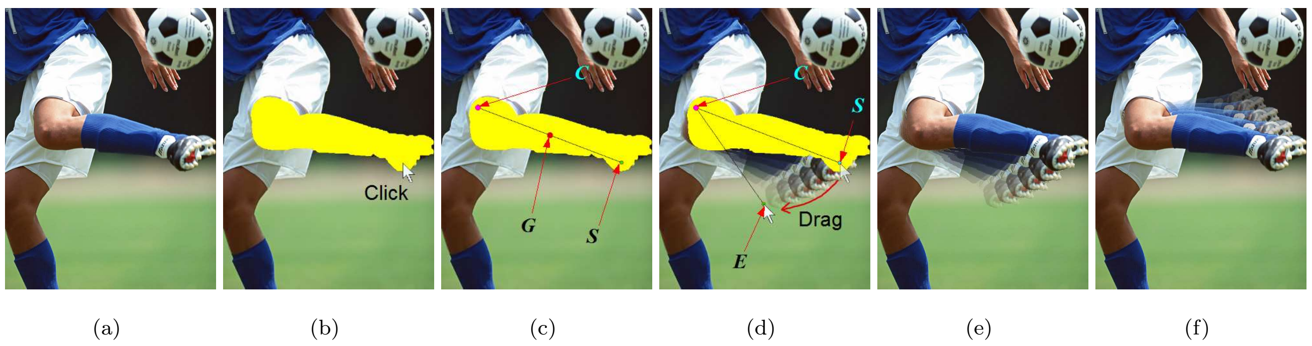


Fig. 3 Interactive motion depiction using motion anchor. (a) Input image. (b) The user clicks the anchor point $\boldsymbol{S}$ (after segmentation). (c) Computing the center of mass $\boldsymbol{G}$ and the rotation center $\boldsymbol{C}$. (d) Generating time-lapse ghosts by dragging the anchor point to $\boldsymbol{E}$. (e) Final result. (f) Another result showing different motion direction.

5 Interactive Motion Depiction

In this section, we describe the main user-interfaces and algorithms used to generate the target motion photograph. The proposed system supports four types of foreground motion effects (*i.e.* rigid time-lapse ghosts[1], scaled time-lapse ghosts, linear and curved speed-lines), two types of background motion effect (*i.e.* linear and radial panning), in which the basic UI is common to all. The effect can be applied to multiview motion photography. This will be described at the latter part of this section.

5.1 Foreground Motion Effect

The user manipulates the selected moving part of a foreground object interactively using drag-and-drop. The system produces the motion effects and renders it on the motion photograph in real-time. The user is able to control the degree of motion effect interactively (*e.g.* the number of ghosts in the time-lapse effect and the density of speed-lines) simply by scrolling the mouse wheel. In addition, the user can edit the existing motion effects and even mix additional motion effects.

5.1.1 Motion Anchor

We define the *motion anchor* first in order to formulate the motion. The motion anchor is a set of points for interactive manipulation of motion effects. It consists of three points, *i.e.* the center of rotation $\boldsymbol{C}$, the motion start $\boldsymbol{S}$ and end points $\boldsymbol{E}$ over which the user drags the object. The proposed UI is very simple, such that all the motion effects are created and edited by the motion anchor.

First, when the user clicks the motion start point $\boldsymbol{S}$, the system determines which segment is selected (as shown in Figure 3 (b)). Then, it computes the center of mass $\boldsymbol{G}$ of the selected segment and the center of rotation $\boldsymbol{C}$ as the symmetric point of $\boldsymbol{S}$ about $\boldsymbol{G}$ (Figure 3 (c)) which is defined by

$$\boldsymbol{C} = \boldsymbol{G} + \overrightarrow{\boldsymbol{SG}}. \tag{1}$$

After that, the user continues dragging $\boldsymbol{S}$ to the motion end point $\boldsymbol{E}$ (Figure 3 (d)). While the user drags the object, the selected motion effect is rendered in real-time (Figure 3 (e)).

5.1.2 Rigid Time-Lapse Ghosts

The time-lapse ghost effect is easily generated by copying and pasting the selected segment based on the motion anchor described in the previous section,. More specifically, the system transforms the pixel $\boldsymbol{p}_s$ in the segment to the target position $\boldsymbol{p}_t$ using the following steps.

The transformation is defined efficiently only with the motion anchor. The system separates the translational and rotational components of transformation from the motion anchor. The transitional component is obtained by comparing the length of $\boldsymbol{CS}$ with $\boldsymbol{CE}$ and the rotational component is obtained by the angle between $\boldsymbol{CS}$ and $\boldsymbol{CE}$, respectively. When the system draws the i_{th} ghost among $n+1$ ghosts, the position of the transformed pixel $\boldsymbol{p}_t$ is computed by

$$\boldsymbol{p}_t = \mathbf{R}\left(\frac{i}{n}\angle \boldsymbol{SCE}\right)\mathbf{T}\left(\frac{i}{n}(|\boldsymbol{CE}| - |\boldsymbol{CS}|)\frac{\overrightarrow{\boldsymbol{CS}}}{|\boldsymbol{CS}|}\right)\boldsymbol{p}_s, \quad 0 \le i \le n \tag{2}$$

[1] 'Time-lapse' effect is similar to the stroboscopic image in the previous articles, as in [5].

where $\mathbf{R}$ and $\mathbf{T}$ denote the rotation around $\boldsymbol{C}$ and the translation parallel to $\overrightarrow{\boldsymbol{CS}}$, respectively.

The ghosts fade out according to the gradually reducing alpha value to depict the time-elapsing effect. This is as given by

$$\alpha_i = \left(\frac{n-i}{n}\right)\alpha_0, \quad 0 \le i \le n \tag{3}$$

where α_0 is the alpha value of the original segment, usually 1.0 (opaque).

5.1.3 Scaled Time-Lapse Ghosts

The overall process of generating the scaled time-lapse ghosts is similar to that of generating the rigid time-lapse ghosts. This effect is better suited to (1) depict rotational motion of the partial segment(s) of the foreground object or (2) the object coming closer to / going farther from the viewer.

Instead of considering the translational component, the scaling component is taken into account in this case; the scaling term is represented by the ratio of $\boldsymbol{CE}$ and $\boldsymbol{CS}$. The transformation equation of i_{th} ghost is now given by

$$\boldsymbol{p}_t = \mathbf{R}\left(\frac{i}{n}\angle\boldsymbol{SCE}\right)\mathbf{S}\left(\frac{|\boldsymbol{CE}|}{|\boldsymbol{CS}|}^{\frac{i}{n}}\right)\boldsymbol{p}_s, \quad 0 \le i \le n \tag{4}$$

where $\mathbf{S}$ denotes the scaling around $\boldsymbol{C}$.

5.1.4 Linear Speed-Lines

Linear speed-lines are appropriate to depict the translational motion of a rigid object. The system first searches the pixels on the trailing-side boundary of the moving segment. Then, for each pixel, the system generates the random number between 0 to $|\boldsymbol{SE}|$, which is the length of individual speed-line starting from the pixel. The speed-lines are parallel to $\boldsymbol{SE}$.

5.1.5 Curved Speed-Lines

Similar to the scaled time-lapse ghosts, curved speed-lines are adequately represent rotational movement of an object or an object coming closer to / going farther from the viewer. The basis of this effect is the as same as the linear speed-lines, *i.e.* the pixels on the trailing-side boundary are stretched randomly. However, this effect draws curves from $\boldsymbol{p}_s$ to $\boldsymbol{p}_t$, using rotation and scaling as given by (4).

5.2 Background Motion Effect

Two types of background motion effect are developed, *i.e.* linear and radial panning. Similar to generating foreground motion effects, the user simply specifies a motion vector $\overrightarrow{\boldsymbol{V}}$ by dragging and dropping using the mouse button. If the starting point $\boldsymbol{S}$ of $\overrightarrow{\boldsymbol{V}}$ is on the foreground segmented region, the system generates the radial panning effect by blurring the background region radially. Conversely, if the starting point of the motion vector is on the background region, the system generates linear panning by blurring the background region in parallel.

The background blurring algorithm in both cases is simple. First, the system assigns a motion vector to each background pixel. In the case of linear panning, the vector is identical to $\overrightarrow{\boldsymbol{V}}$ for all the background pixels. Conversely, in case of radial panning, the vector assigned to each pixel has different radial direction, while the length is still $\left|\overrightarrow{\boldsymbol{V}}\right|$. In our implementation, the center of radian motion blur is $\boldsymbol{S}$, the starting point of $\overrightarrow{\boldsymbol{V}}$. Therefore, the motion vector $\overrightarrow{\boldsymbol{V_P}}$ assigned to a background pixel at $\boldsymbol{P}$ is given by

$$\overrightarrow{\boldsymbol{V_P}} = \left|\overrightarrow{\boldsymbol{V}}\right| \frac{\overrightarrow{\boldsymbol{SP}}}{|\boldsymbol{SP}|}. \tag{5}$$

After obtaining the motion vectors of background pixels, the new (blurred) color of a pixel at $\boldsymbol{X}$ is computed by averaging the color of background pixels that their motion vectors pass through $\boldsymbol{X}$.

5.3 Extension to Multiview Motion Depiction

We extend the proposed single-view motion photography to multiview motion photography. Multiview input is a set of images captured by multiple cameras at the same time but from different viewpoints. As is common in multiview stereo matching [7], it is assumed that the input images are already calibrated and therefore the homography between images is known too. Although any placement of the multiview camera is allowed under the assumption, we prefer concentric multiview cameras, since this gives more flexibility in realizing multiview motion effects.

It is very hard to control the motion direction in depth dimension. Therefore, the proposed multiview motion depiction system focuses on the perpendicular motion to the optical axis of each image. Motion effects are added by (1) selecting an appropriate view and (2) editing motion the effect using motion anchor, as in the case of single-view motion depiction. The procedure is performed segment-by-segment iteratively. As soon as a

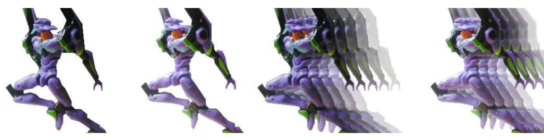


Fig. 4 An example of stereoscopic motion photography. (a) Input left image. (b) Input right image. (c) Left motion photography. (d) Right motion photography in which the ghosts are transferred from (c).

motion effect is added on a particular image, it is transferred to every other image in the multiview, yielding a multiview motion photograph.

The scale of transferred ghost should be computed correctly to transfer the depicted ghost correctly. The motion in the other view(s) has an out-of-plane component, causing the depth changes. In our approach, the scale of the transferred ghost is computed as follows. Initially it is assumed that the distance between the camera and object is the same as the width of the image. Then, the system calculates the depth of projected $\boldsymbol{S}$ (*i.e.* z_S) and projected $\boldsymbol{E}$ (*i.e.* z_E). The scaling factor between the first and last ghost is now the ratio of the depth values ($= \frac{z_S}{z_E}$). Finally, the scaling factors of in-between ghosts are computed as the linear interpolation of 1 and this factor.

Preliminary test on multiview motion photography is done for stereoscopic views where camera matrices are known. For input stereo pair images shown in Figure 4 (a) and (b), the time-lapse ghost effect is applied to the left image and the corresponding ghosts are transferred to the right image as shown in Figure 4 (c) and (d). When the observer carefully overlaps the left and right images in his/her left and right eyes, respectively, the object and the ghosts are felt as 3D motion photographs.

6 Experimental Results

In order to evaluate the performance of the proposed motion photography system, we performed intensive experiments for a variety of input images. All the source code is written in Java (JDK 1.5.0) on Windows Vista.

In addition to this, we perform a user study to evaluate the efficiency subjectively. Each of the experiments are described in detail in the following sections.

6.1 Examples of Motion Photographs

In Figure 5, different motion effects are illustrated for a few input images, in which the observer feels differently in terms of speed and direction of motion. For example, speed-lines and background panning effect in the horse-racing example would deliver faster motion feeling than the time-lapse ghost effect. The time-lapse ghost effect in the airplane example looks like the plane is landing, while the background panning one looks like taking off.

More examples of creating interesting motion photographs are shown in Figure 6 and Figure **??**. Figure 6 (i) ∼(k) show the results in which more than two motion effects are applied simultaneously.

6.2 User Study

Since a single user cannot evaluate the system convincingly enough, we perform a user study to evaluate the efficiency subjectively. In our user study, six subjects are selected. Four of them are novices with little or no experience with Photoshop nor other paint tools, while other two are Photoshop experts. Each subject is given five-minute instruction session on the proposed system. The novices are given additional five minutes of instruction on the related tools in Photoshop. We split them into two groups such that each group has one expert and two novices.

Each group is asked to depict similar motion effects shown in Figure 7 within no more than 10 minutes for each method. Actual elapsed time is recorded for each individual subject. One group uses Photoshop first and the proposed system next, while the other does vice versa. Verbal explanation is provided in addition to showing the sketched direction. Therefore, subjects know what the target (*i.e.* foreground) object is and how the subject should be depicted.

In the experiment, the segmentation is done beforehand and the mask image is provided as a separate file (proposed system) and copied to a layer (Photoshop). Note that user interface for segmentation is not the goal of this paper. When using Photoshop, subjects are recommended to use special tools (like copy-and-paste, finger tool, and radial-blur filter, etc). However, subjects are also allowed to use other tools (*e.g.* wind filter).

After the test, the result of one group is shown to the other group and vice versa. Each subject is then asked to vote his preference if he/she judges based on the visual quality. An example of results by a particular subject is shown in Figure 8.

The result shows that the average elapsed time using the proposed system is much shorter than the elapsed

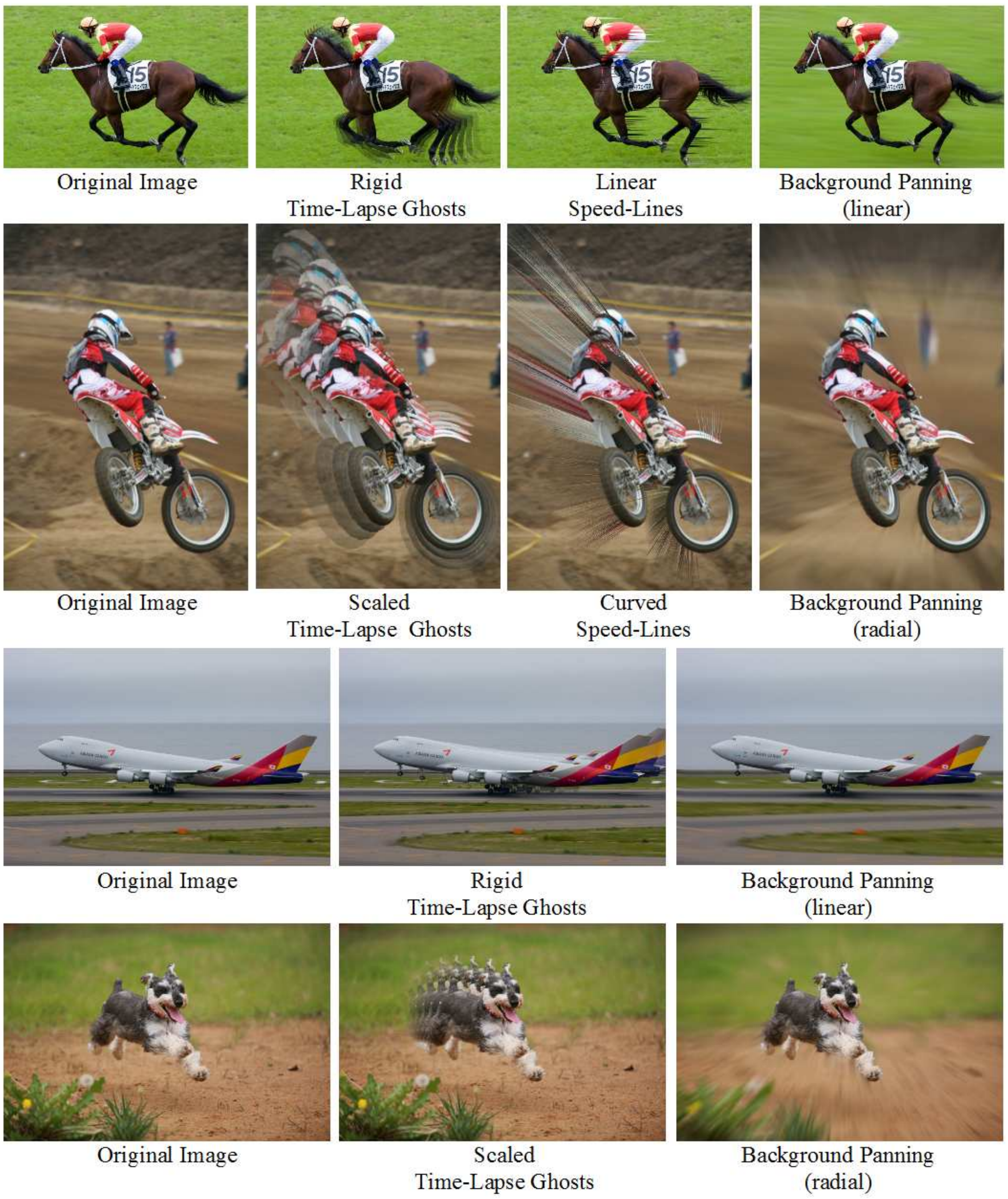


Fig. 5 The result of created motion photographs.

time using Photoshop as shown in Figure 9. Especially for time-lapse ghost and speed-lines, it is more than five times faster to use the proposed system. Since Photoshop supports general radial-blur tool, the relative performance to depict background radial panning is better than the former cases. However, it still takes 3 times longer than when they use the proposed system. As shown in Figure 10, Photoshop experts create similar-quality motion photographs no mater what system they used. However, novice users create quite better motion photographs using the proposed system, which is one of the strength of the proposed system. Note that this is mainly due to the easiness of user interface.

Figure 11 shows the result of the subjects' voting to the question "which result image is better?". More subjects prefer the result of the proposed system. Even though the difference is not dominant for the result using speed-line and the background panning, it is evident that they spend quite less time to achieve better motion photographs.

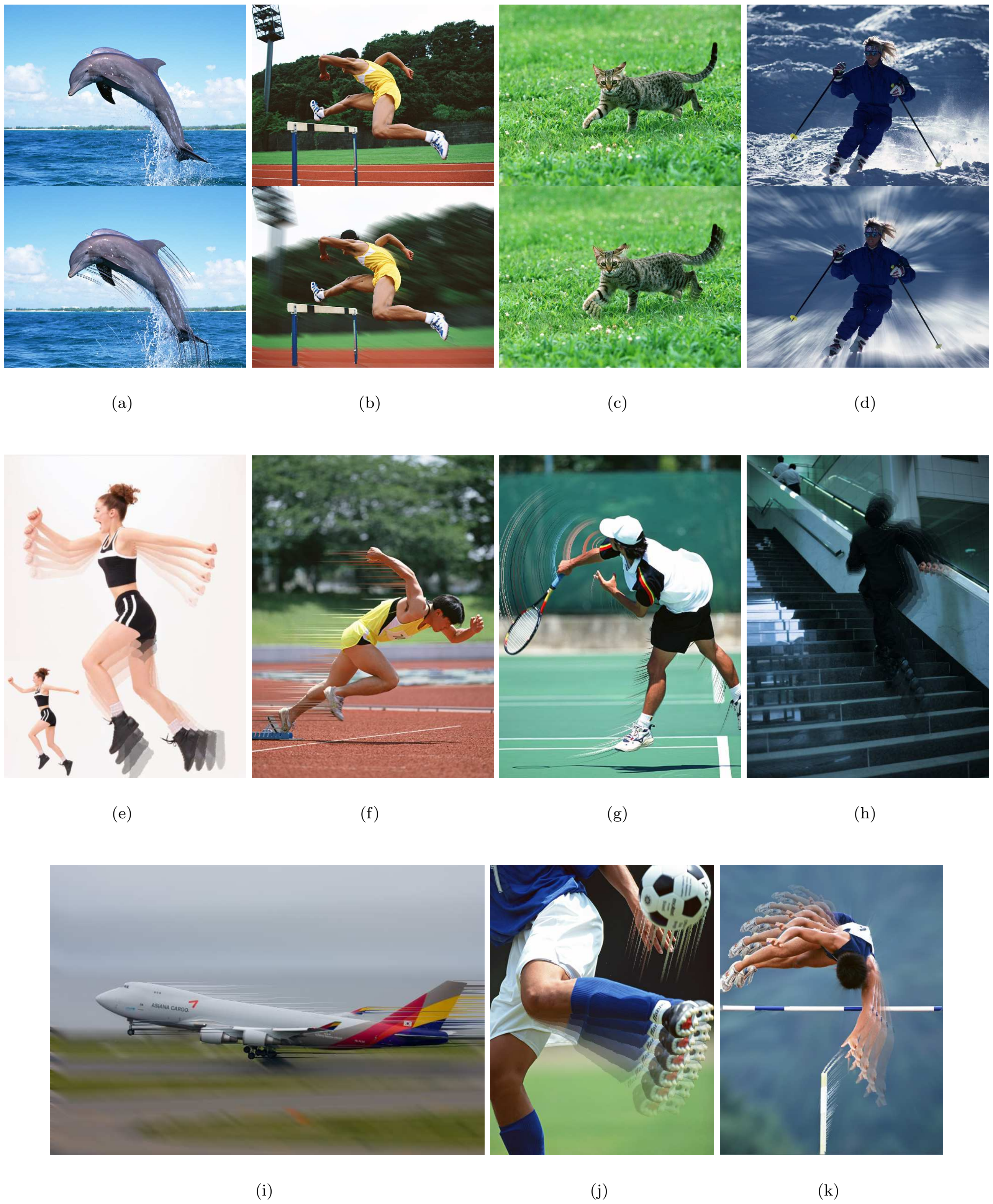
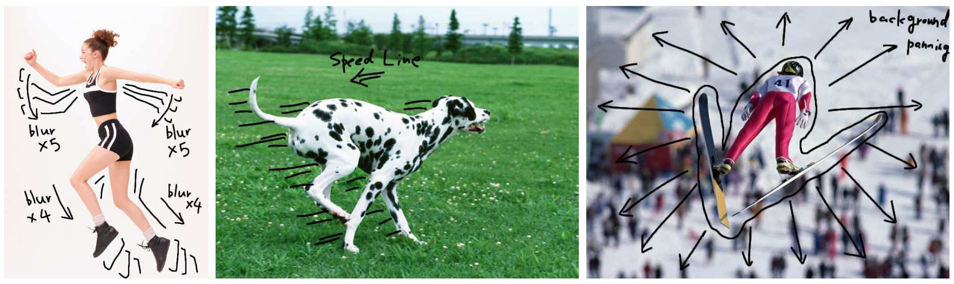
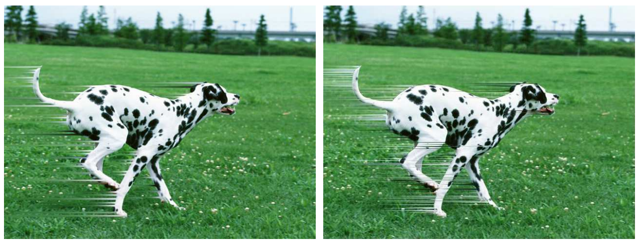


Fig. 6 More results of created motion photographs. (i)∼(k) show multi-effect results.



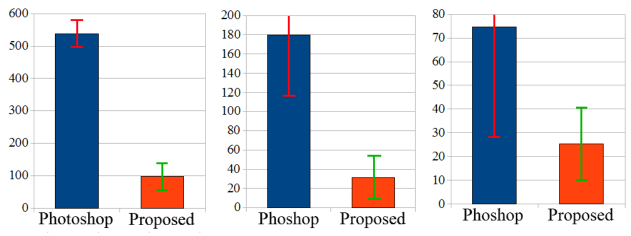
(a) (b) (c)

Fig. 7 Visual instructions used in the user study to evaluate (a) time-lapse ghost (Test1), (b) speed-Lines (Test2), and (c) background panning (Test3).



(a) (b)

Fig. 8 Example of the results depicted by a subject. (a) Using Photoshop. (b) Using proposed system.



(a) (b) (c)

Fig. 9 Average elapsed time (in seconds) to make result image. (a) Test1. (b) Test2. (c) Test3.

After the user test, we obtain valuable comments from the subjects. They agree that the proposed system is intuitive and automatic as an easy-to-use integration of useful tools for generating motion photographs. Many subjects suggest to provide the proposed system as a plugin of Photoshop. However, they wanted more function to our system (*e.g.* "Undo"). There is another argument that one region having two or more motion anchor is not practicable. Most of negative comments are technical or implementation-related issues. We will reflect them in the revised version of the proposed system in the near future.

7 Conclusion and Future work

In this paper, we present an interactive and computational method of creating 'motion photography' from single image. By using the proposed system, the user can make high-quality motion photography very efficiently. Given an input image, which contains moving objects but captured as static, the user can add and edit various motion effects to the objects by a simple but efficient user interface. A variety of experimental result as well as user study shows that the proposed system creates motion photographs which is better in visual quality and faster to generate than using the convention general tool (*i.e.* Photoshop).

We consider extending this work to video summarization. Unlike schematic storyboard for video visualization [8], context-based extraction of important frames and illustrating motion in them efficiently will deliver effective method of video summarization. On the other hand, it is also planned that the multiview motion photography is investigated further to achieve 3D motion photography which is a challenging problem.

References

1. E. P. Bennett and L. McMillan. Computational time-lapse video. *ACM Transactions on Graphics*, 26(3):102:1–102:6, July 2007.
2. G. Brostow and I. Essa. Image-based motion blur for stop motion animation. In *ACM SIGGRAPH*, pages 561–566, August 2001.
3. M.-T. Chi, T.-Y. Lee, Y. Qu, and T.-T. Wong. Self-animating images: Illusory motion using repeated asymmetric patterns. *ACM Transactions on Graphics*, 27(3):62:1–62:8, August 2008.

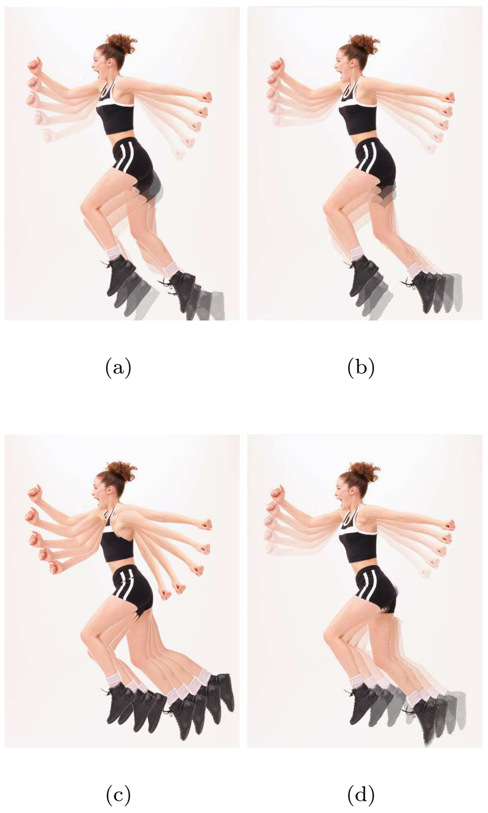


Fig. 10 The comparison of results by Photoshop experts and novices. An expert's result using (a) Photoshop and (b) proposed system. A novice's result using (c) Photoshop and (d) proposed system.

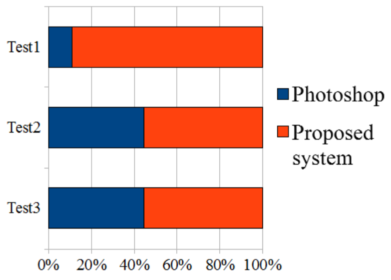


Fig. 11 Result of user preference voting.

4. J. P. Collomossea, D. Rowntreeb, and P.M. Halla. Rendering cartoon-style motion cues in post-production video. *Graphical Models*, 67(6):549–564, November 2005.
5. J. E. Cutting. Representing motion in a static image: Constraints and parallels in art, science, and popular culture. *Perception*, 31(10):1165–1193, October 2002.
6. W. T. Freeman, E. H. Adelson, and D. J. Heeger. Motion without movement. In *Proc. of ACM SIGGRAPH*, pages 27–30, July 1991.
7. Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
8. D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz. Schematic storyboarding for video visualization and editing. *ACM Transactions on Graphics*, 25(3):862–871, July 2006.
9. M. Haller, C. Hanl, and J. Diephuis. Non-photorealistic rendernig techniques for motion in computer games. *ACM Computers in Entertainment*, 2(4):11–11, October 2004.
10. Adobe Systems Inc. Adobe photoshop. http://www.adobe.com/products/photoshop.
11. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.
12. Yuya Kawagishi, Kazuhide Hatsuyama, and Kunio Kondo. Cartoon blur: Non-photorealistic motion blur. In *Proc. of Computer Graphics International*, pages 276–281, 2003.
13. B. Kim and I. Essa. Video-based nonphotorealistic and expressive illustration of motion. In *Proc. of Computer Graphics International*, pages 32–35, June 2005.
14. M. Masuch, S. Schlechtweg, and R. Schulz. Speedlines: Depicting motion in motionless pictures. In *ACM SIGGRAPH Abstracts and Applications*, page 227, August 1999.
15. T. Moriya, T. Takahashi, and A. Tanaka. Non-photorealistic rendering technique for depicting 3d objects motion. In *ACM SIGGRAPH Research Poseter*, number 114, July 2006.
16. S. Obayashi, K. Kondo, T. Konma, and K. Iwamoto. Non-photorealistic motion blur for 3d animation. In *ACM SIGGRAPH Sketches*, number 93, July 2005.
17. C. Rother, V. Kolmogorov, and A. Blake. “grabcut”: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, August 2004.