Pan-and-Tilt Self-Portrait System Using Gesture Interface

Shaowei Chu, Fan Zhang, Naye Ji, Zhefan Jin and Ruifang Pan
School of New Media
Zhejiang University of Media and Communications
Hangzhou, China
chu@zjicm.edu.cn

***Abstract*—Digital cameras are widely used in desktop and notebook PCs. Taking self-portraits is one of the important function of such cameras, which allows users to capture memories, create art, and improve photography techniques. A desktop environment with a large display and a pan-and-tilt camera provides users with a good area for exploring more angles and postures while taking self-portraits. However, most of the existing camera interfaces of this type are limited to device-based systems (i.e., mouse and keyboard) that prevent users from efficiently controlling the camera while taking self-portraits. This study proposes a vision-based system equipped with a gesture interface that control a pan-and-tilt camera for taking self-portraits. This interface uses gestures, particularly slight hand movements (i.e., sweeps, circles, and waves), to control the pan, tilt, and shutter functions of the camera. The gesture-recognition achieved good efficiency in performance (less than 2ms) and the recognition rate (0.9 on average in lighting conditions range 100 - 200). Experimental results indicate that the proposed system effectively controls the options in a self-portrait camera, this approach provides significantly higher satisfaction, particularly in terms of the intuitive motion gestures, freedom, and enjoyment, than when using a hand-held remote control or a conventional mouse-based interface. The proposed system is a promising technique for taking self-portraits in a desktop environment.**

Keywords—digital camera; self-portrait; gesture interface; user interface; user study

I. INTRODUCTION

The use of cameras on desktop or notebook PCs has become pervasive in recent years. These cameras are widely used for video chatting, videoconferencing, home video recording, gaming, and taking self-portraits [1, 2]. However, most of the existing camera interfaces still apply conventional input devices (i.e. mouse, or keyboard) to control camera functions, thereby limiting the efficiency and freedom of a user to control the camera. For example, users need to move back and forth relative to the camera to trigger the camera shutter, change the composition, and prepare their postures.

The advances in vision-based techniques, such as face detection, motion capture, and gesture recognition, provide multiple options for users to control the cameras. Successful applications, includes those reported by [1], who proposed the use of head motions to control the view of a PTZ camera during video chatting; in addition, [3] introduced the use of motions to make selections when playing a camera-based game, whereas [4] suggested the use of fingertip tracking to make interactions in an augmented reality application. However, only a few studies have investigated the use of a pan-and-tilt camera to take self-portraits. Taking self-portraits is an important function in camera applications, and the desktop environment provides a good area for application of a pan-and-tilt camera because the computer display allows a user to see and instantly check the preview. Moreover, the distance, composition, and angles of the camera are adjustable, allowing the user to explore different angles and obtain creative shots.

This paper presents a self-portrait system that consists of a pan-and-tilt camera in a desktop environment and employ a vision-based gesture interface. We introduce intuitive motion gestures (e.g., sweeping, circling, and waving) to control the pan-and-tilt, and shutter trigger functions allowing the user to take self-portraits. The gesture recognition technique is an extension of the previous work [2] published in Human-Computer Interaction International in 2011, introducing the finger detection and motion-based gestures to control a pan-and-tilt camera. In this work, we explore more flexible motion gestures and particularly considering both recognition performance, and as well as lighting robustness using optical-flow estimation. We also conduct two formal experiments to investigate the performance and user satisfaction of the system.

In the first experiment, we asked participants to perform specific gestures for a number of times, and we recorded the correctness of the recognition. The experiment also tested the interface under different lighting conditions. We found a 0.91 accuracy of gesture recognition under the light intensity range of 100-200.

The second experiment evaluated user satisfaction, and we asked the participants to use the motion interface, as well as a remote control and mouse interface, to take self-portraits. We determined that the proposed motion interface was more intuitive and offered more freedom and enjoyment compared with the conventional remote control and mouse interface. The proposed interface is deemed to be a promising technique for the self-portraits.

II. SELF-PORTRAIT APPROACHES

Hand-held camera, is the most popular approach for taking self-portrait, in which the camera is held as far as possible when taking photographs. Another recent technique is the use of a selfie stick that can extend the length of the arm of the user when taking wide-angle photographs [5]. This approach is

978-1-5090-5507-4/17/$31.00 ©2017 IEEE
ICIS 2017, May 24-26, 2017, Wuhan, China

IEEE computer society

highly popular technique among tourists [6]. In fact, many cameras are designed so that the lens and screen are located on the same side of the camera or the lens is rotatable (e.g., Canon 70D, Casio TR70, Oppo N3); hence, users adopting this technique can see themselves when taking self-portraits. Although this method can be enjoyable and can capture moments, the pictures taken are not ideal because they are enerally of poor quality; the face of the subject may be distorted or the user's hand may occupy a certain portion of the image; in addition, keeping the hand steady when taking photographs in usually difficult [7].

Robot camera, applies smart techniques (e.g., face detection and gesture, motions, or voice commands), such that the camera automatically capture images. The successful application of this technique includes Sony Party-shot camera and the camera described in [8], which uses the face/smile detection to automatically trigger the camera shutter. Moreover, the studies [9] and [10] both introduced face angle detection and portraits ranking technique that guides users to take good portraits. The studies [2] and [11] also proposed the detection of user fingertip and gesture motions for controlling the camera shutter, whereas [12] introduced the camera zooming interface, in which the camera zooms in/out while detecting the nodding and shaking motions of the users' head. Furthermore, [9] proposed an automatic shutter to shoot jump shots based on the user's jump trajectory. These studies provide users with multiple options when taking self-portraits; however, a few of them reported the application of gesture interface using a pan-and-tilt camera on a desktop environment.

Kiosk, is an approach in which the user can achieve more freedom given that the camera does not need to be held because it is fixed in an appropriate position and is equipped with a touch screen to set up the camera functions. This technique is a popular self-portrait approach and is extensively used in commercial products [13] and in studies on the social factors of self-portraits [14], [15], [16].

The present work proposed a kiosk-like approach, although we introduce gesture interface to as a means of controlling the camera. The system is available in a desktop environment, particularly at home which provides the user with a more relaxed environment and the freedom to explore various portrait options. We explored the motion detection approach that recognize gestures only on motion information without identifying the features of hand. This improved recognition robustness well with different lighting conditions

III. System Proposal and Gesture Interface

The main goal of the system is to achieve an effective vision-based gesture interface that recognizes distinct gestures to enable the user to control the pan, tilt, and shutter of the camera. We must consider both intuitiveness of the gestures and the recognition rate, and ensure that they are comfortable to perform.

A. Design of the Gesture Interface

The specific objectives and implement strategies to achieve the design goal of this work are:

1) To define a specific area for performing gestures.

2) The use of small movements of the hand.

3) To provide recognition of distinct gestures to control pan, tilt, and shutter.

4) To design a GUI that offers real-time visual feedback on the gestures.

To achieve objective (1), we exploit face-detection techniques, and adopt a near-face area to recognize gestures [17], as shown in Figure 1. We explored the appropriate area that is comfortable for users to raise their right hand to the right side of their face and perform gestures. The area does not overlap with the user's body, and so will not be triggered by movements of the body. The size of the area was calculated based on the size and position of the face, where the size is defined as 1.5 times the width of the face, and a distance of 1.0 time the width of the face from the area of the face. This limits the size of the gesture-recognition area, which facilitates rapid processing.

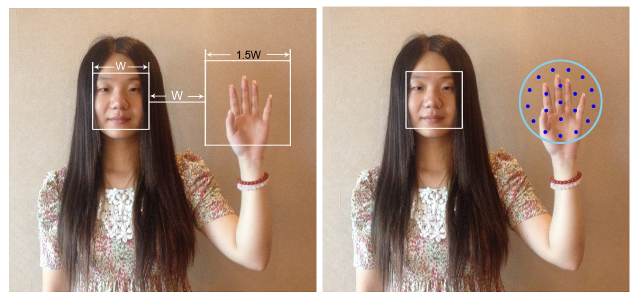

Fig. 1. Area for gesture recognition.

To achieve objective (2), we defined the gestures to be performed using the hand over the interface area, to limit the large motions and fatigue of the forearm.

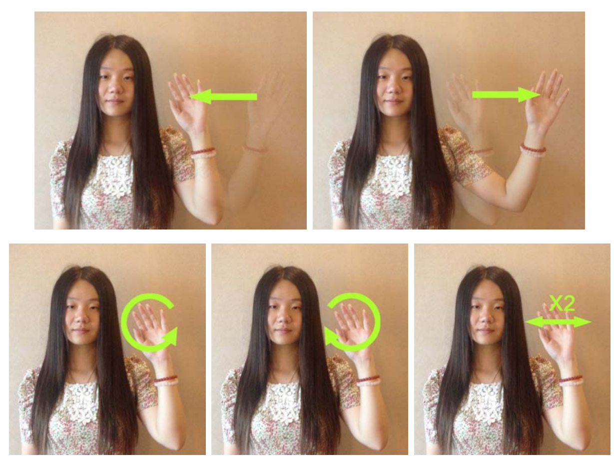

Fig. 2. The sweep left/right, circling CCW/CW and waving gestures.

To achieve objective (3), we implemented three types of gestures, i.e., sweeping to the left/right, circling CCW/CW, and waving to map to panning left/right, tilting up/down, and triggering the shutter, see Figure 2. In the user experiment at Section 5, it proved these gestures were intuitive to control a pan-tilt camera.

To achieve objective (4), we designed a GUI with visual feedback on the area for gesture recognition. This area is shown using a circle, the inside of which corresponds to the tracking points that are used for gesture recognition. When motion occurs, the tracking points move from the base position. Once the strength of the motion exceeds the threshold, an arrow appears, starting at the center and pointing to the edge; this arrow indicates the direction of the tracked motion, as shown in Figure 3, 4. The number shown below the circle indicates which gesture has been recognized.

B. Detecting Motion of the Hand

We designed the system to respond to 20 tracking points on the circle (see Figure 3, 4, 5). Each tracking point was calculated by applying Lucas-Kanade optical flow [18] to subsequent pairs of frames, and the motion of the tracked point was summed to determine the overall motion. If the strength of the motion exceeds a threshold, it is mapped to a gesture.

C. Sweep Left/Right Gesture

This gesture is shown in Figure 3, and can be performed from left to right or from right to left. During the period of time whereby the hand moves, the motion is calculated from the tracking points in the region. When the user's hand leaves this region, the resulting motion will be summed to calculate the mean direction of the gesture, which for the left/right sweep gesture will be motion to the left or right, respectively.

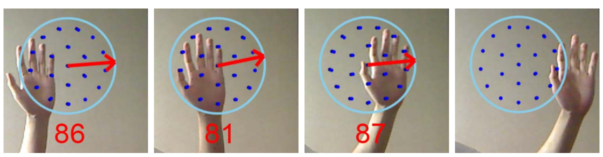

Fig. 3. The gesture whereby the user sweeps a hand to the right. During the time when the user's hand passes through the region the arrow indicates the instantaneous motion of the hand, where the 0˚ at 12 o'clock.

D. CCW/CW Circle Gesture

This gesture is shown in Figure 4. To recognize the gesture, the direction of motion in consecutive frames is monitored, and the change in this direction over time is recorded. A full 360˚ motion corresponds to a circling gesture. The accumulated angle is initially zero; a cumulative change in the direction of motion that exceeds 45˚ corresponds to the start of the gesture.

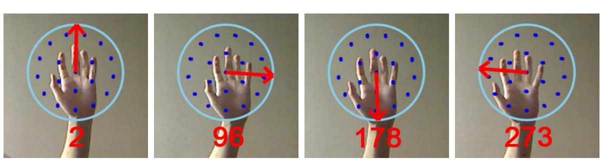

Fig. 4. A circling gesture in the CW direction.

E. Waving Gesture

This gesture is shown in Figure 5. The waving gesture can be performed by moving hand left/right repeatedly over the area. To recognize this gesture, we detect whether the direction of motion of the hand has reversed over two subsequent frames. A reversal corresponds to a change in direction of at least 135˚. Each such reversal of the direction of motion is considered an instance of waving. We set a four times of waving as a successful waving gesture command.

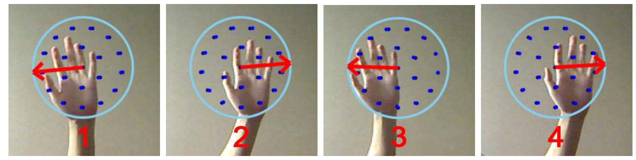

Fig. 5. The waving gestures, four repetitions of a waving.

F. Mapping Gestures to Camera Controls

The gestures were mapped to the controls of the camera as follows: a sweep to the left pans left by 3˚; a sweep to the right pans right by 3˚; a full 360˚ circling CCW tilts the camera up by 3˚; circling CW by 360˚ tilts the camera down 3˚; waving triggers the shutter. Table 1 summarizes this mapping.

TABLE I. MAPPING OF GESTURES TO CAMERA CONTROLS.

Gestures	Camera functions
Sweep to left	Pan left
Sweep to right	Pan right
Circling CCW	Tilt up
Circling CW	Tilt down
Waving	Shutter trigger

G. Implementation

The prototype system employed a Logitech Quickcam Orbit AF webcam, which provides panning through 189˚, and a range of tilts of 102˚. The resolution was 800 x 600 pixels, and the video rate was 30 frames per second (FPS). The image sequence was sent to a notebook PC via a universal serial bus (USB) connection. The PC had a 15” display and an Intel i7-3537U processor.

The software was implemented in C++. The real-time image-processing algorithms were based on OpenCV [19] and the graphics rendering was carried out using Microsoft Direct2D. The gesture recognition time was less than 2 ms, where the interface area was scaled to 120 x 120 pixels. The face detection was implemented in boosted cascade [20].

IV. EXPERIMENT 1: PERFORMANCE OF THE INTERFACE

We first investigated the performance on the three gestures in isolation and then looked at combinations of all three gestures to examine conflicts in gesture recognition. Finally, we investigated the robustness of gesture recognition in the three different lighting conditions.

A. Participants

There were 8 university students (5 male, 3 female), with a mean age of 21.1 years and a standard deviation of $\sigma = 0.9$ years. Each participant was paid ± 5 to participate the experiment.

B. Conditions

The participants sat on chair that was placed 1 m from the camera. The face corresponded to approximately 120 x 120 pixels. The interface was shown on the right side of the face, and participants were asked to use their right hand to perform the gestures. The participants tried to familiar with the gesture interface in about 5 minutes. The prototype system for the experiment is shown in Figure 6.

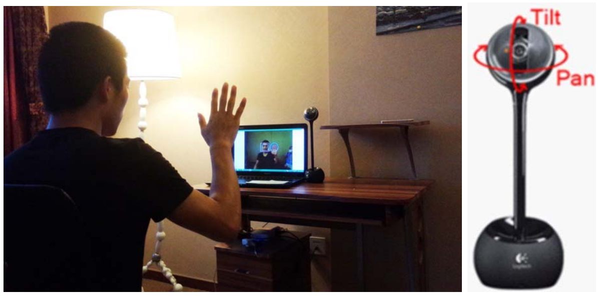

Fig. 6. The self-portrait system, including a pan-and-tilt camera and notebook computer. Gestures are used to control camera functions, such as pan, tilt and shutter trigger.

C. Procedure

Four test sessions were arranged to test the accuracy of Sweeping left/right, Circling CCW/CW, Waving gestures, and as well as combining the three gestures together. Participants were required to perform specific gestures under the guidance of the GUI shown on the screen, see Figure 7. Participants needed to perform the specific gesture again while the gesture is not recognized by the system. With each gesture five times, giving a total of 8 participants x 5 = 40 gestures for each set of tests.

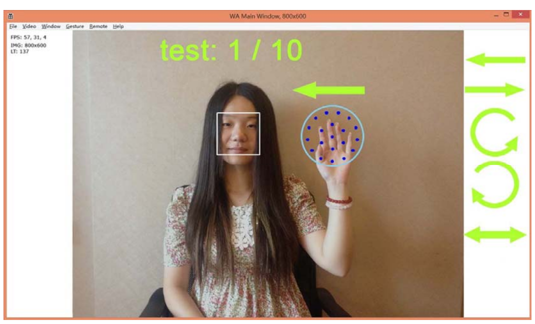

Fig. 7. The user tests and the GUI, which was used to aid participants in performing the gestures. The right side of the figure shows the icons that represent sweeps left/right, circling CCW/CW, and waving.

D. Results

Sweeping Left/Right: All the participants finished performing correct gestures, giving the precision was 1.0. While a total of 10 attempted gestures were not recognized, giving recall rates of 0.87 and 0.91 for moving to the left and right, respectively. The total time to finish the test was 25.7 s (σ = 2.2), giving a time to perform each gesture of 25.7 / 10 = 2.6 s.

Circling CCW/CW: The precision was 1.0, and 8 gestures were not recognized, giving recalls of 0.91 and 0.89 for the CCW and CW gestures, respectively. The time to finish the test was 27.6 s (σ = 1.7), giving a time to perform each gesture of 27.6 / 10 = 2.8 s.

Waving: The precision was 1.0 for waving. In total, 3 gestures failed to be recognized, giving recall rates of 0.93. The time to finish the test was 15.5 s (σ = 1.8), giving a time of 3.1 s per gesture.

The detail results are shown in Table 2.

TABLE II. RESULTS OF THE GESTURE RECOGNITION.

Gestures	Correct	Wrong	Lost	Precise	Recall	Time(s)
Sweep to left	40	0	6	1.0	0.87	25.7 (σ =2.2)
Sweep to right	40	0	4	1.0	0.91	
Circling CCW	40	0	4	1.0	0.91	27.6 (σ =1.7)
Circling Cw	40	0	5	1.0	0.89	
Waving	40	0	3	1.0	0.93	15.5 (σ=1.8)

Combinations of gestures: In this session we tested combine all the gesture together to evaluate the conflicts of the gestures. Participants were required to perform each gesture twice, and the order of gestures was randomized. In additional, we also investigated the recognition under different lighting conditions, dark (light = 100 ± 10 units) and bright (200 ± 10 units), where the lightness was calculated based on the red, green, and blue (RGB) color space for hue, saturation, and lightness (HSL), with lighting in the range 0-255 units, see Figure 8.

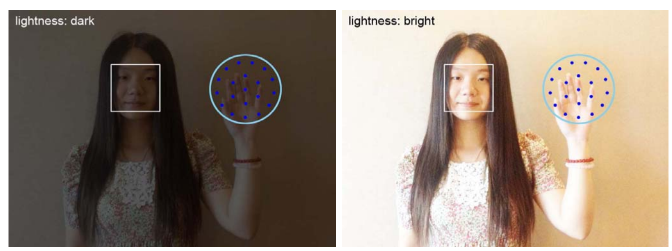

Fig. 8. Gesture performance under different lighting conditions. Left: light = 100 ± 10 units; Right: light = 200 ± 10 units.

TABLE III. THE GESTURES PERFORMANCE, WHERE LIGHTNESS = 150 / 100 / 200 ± 10.

Gestures	Correct	Wrong	Lost	Precise	Recall	Time(s)
Sweep to left	16/16/16	0/0/0	3/4/3	1.0/1.0/ 1.0	0.84/0.80 /0.84	29.1 (σ =1.7)
Sweep to right	16/16/16	0/0/0	2/3/2	1.0/1.0/ 1.0/	0.89/0.84 /0.89	
Circling CCW	16/16/16	0/0/0	1/1/1	1.0/1.0/ 1.0/	0.94/0.94 /0.94	22.7 (σ = 1.9
Circling Cw	16/16/16	0/0/0	2/0/2	1.0/1.0/ 1.0	0.89/1.0/ 0.89	
Waving	16/16/16	0/0/0	1/1/0	1.0/1.0/ 1.0/	0.94/0.94 /1.0	22.5 (σ =1.0)

Table 3 lists the results in three lighting conditions. The average precision of recognition was 1.0, and the average recall was 0.9 in normal lighting condition. It was 1.0 in dark conditions and in bright conditions, with average recall rates of 0.9 and 0.91, respectively. The experiment shows no obvious conflicts among the gestures. The time to finish the test in three conditions was 24.8 s on average, giving a time of 2.5 s to perform each gesture. The one-way analysis of variance (ANOVA) of the recall result in these three lighting conditions, shown no significant effects, $F(2, 12) = 0.04$ with $p = 0.96$. Therefore, we conclude the overall gesture recognition accuracy was 0.9 in the lightness range 100-200.

V. Experiment 2: User Satisfaction Survey

In the second experiment we investigated the user satisfaction of the motion interface, particularly on the intuitiveness, efficiency, freedom, enjoyment, etc., which are important in taking self-portraits. The experiment compared the motion interface with two conventional techniques: a remote-control system and a conventional mouse-based interface.

A. Participants

12 university students, 6 males and 6 females, were invited to participate. The average age was 21.6 years ($\sigma = 1.8$ years). Each participant was paid $5 to participate the experiment.

B. Conditions

The participants were asked to take self-portraits using the three different interfaces: the gesture interface, the remote control, and the conventional mouse interface (see Figure 9). When using the gesture interface, the GUI was displayed to the right side of the user. Participants used the sweep left/right, circle CCW/CW, and wave LR/UD gestures to control the camera to pan left/right, tilt up/down, and trigger the shutter, respectively. With the remote control, the D-pad directions were mapped to pan left/right and tilt up/down, and the A button was used to trigger the shutter (see the left panel of Figure 9).

Fig. 9. The remote control and the conventional mouse-based interface.

The mouse interface, which was provided by Logitech Webcam Software, was used with the GUI so that arrows were placed at the right side of the software window, and the four arrows over the disc represented the pan left/right and tilt up/down actions (see the right panel of Figure 9). The button below the center preview area was used to trigger the shutter.

The participants were asked to navigate the camera pan and tilt prior to taking a picture with a simple portrait pose suitable for an identity card. To avoid bias in the experiment, the order of using the three interfaces was permuted, so that 3 x 2 = 6 orders were used in total; the participants were effectively split into six groups of two.

C. Procedure

The participants were invited to use the system, and the experimenter introduced the participants to the functionality of the interfaces and explained how to control the camera using the gestures, the remote control, and the mouse. Then the participants were invited to sit in a chair 1 m from the camera, and were given the opportunity to familiarize themselves with the system, which took 3-5 minutes. Then they were instructed to take self-portraits using each interface. The sequence of actions was to pan left, pan right, tilt up, tilt down, and trigger the shutter. After the shutter was triggered, a 3 s timer counted down while the participants prepared the pose.

During the experiment, the time taken was recorded. After the participants completed these tasks, they were allowed to use the system freely, and then asked to complete a 5-point Likert scale questionnaire (i.e., strongly disagree = 1, disagree = 2, neutral = 3, agree = 4 and strongly agree = 5). The study lasted approximately 15 minutes per participant.

D. Results

Time: The average time taken using the three techniques was as follows: 10.1 s ($\sigma = 1.3$) for the gesture interface, 3.1 s ($\sigma = 0.6$) for the remote control, and 6.0 s ($\sigma = 1.0$) for the mouse interface, and they have significant difference $F(2, 33) = 162$, $p < 0.001$. The post hoc tests also shown the significant difference among the three techniques ($p < 0.001$, $p < 0.001$, and $p < 0.001$).

Subjective preference: the participants broadly agreed with usefulness of the Pan-and-Tilt function (scored as 4.3), the helpfulness of GUI feedback (scored as 4.8), the intuitiveness of the gestures (scored as 4.1, 4.6, and 4.8) and the appropriate of the control function mapping (scored as 4.6, 4.3, 4.6). The result shown in Figure 10.

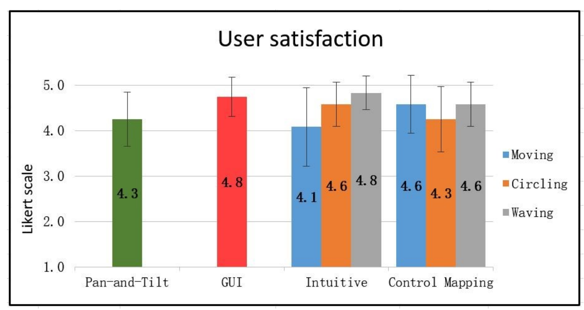

Fig. 10. The mean scores for questions..

When asked to compare the three techniques, most participants rated the gesture interface as best technique, the overall satisfaction score was 4.8, better than remote control 3.9, and Mouse 2.6. Particularly, participants assessed higher score with the gesture interface on intuitiveness, freedom, and enjoyment, the score were 4.7, 4.9, and 5.0, respectively. The ANOVA test shown a significant effect among the three techniques ($p < 0.05$), the gesture interface was better than remote controller and mouse, the detail comparison results shown in Table 4.

TABLE IV. RESULTS OF THE USER SATISFACTION ON THREE INTERACTION TECHNIQUES

	Gesture	Remote	Mouse	P	P(between groups)
Overall	4.8(0.2)	3.9(0.3)	2.6(0.6)	<0.05	<0.05, <0.05, <0.05
Intuitiveness	4.7(0.2)	4.1(0.4)	2.6(0.4)	<0.05	=0.02, <0.05, <0.05
Efficiency	4.3(0.6)	4.5(0.5)	2.9(0.6)	<0.05	**=0.40,** <0.05, <0.05
Freedom	4.9(0.1)	3.5(1.0)	2.9(0.6)	<0.05	<0.05, <0.05, **=0.13**
Enjoyment	5.0(0.0)	3.4(0.4)	2.3(0.8)	<0.05	<0.05, <0.05, <0.05

Though the gesture interface took longer time to complete the test task, there was no significant difference in user evaluation. Participants assessed 4.3, 4.5, and 2.9 with the three techniques, the ANOVA test shown the gesture interface and remote controller were no significant difference (p = 0.4).

VI. DISCUSSION AND FUTURE WORK

The results obtained in Experiment 1 showed good performance of the gesture recognition, the accuracy rate was 0.9 in lightness range 100-200. Interestingly, the recall rate (0.87) when using the sweep left/right gesture was lower than that when using the circling CCW/CW and waving gestures, in which the recalls were 0.92 and 0.94, respectively. This revealed that gestures with repeated motions tended to achieve better performance, and the strong pattern of the gestures could exclude noise motions, thereby improving the robustness of the recognition. On the other hand, the single actions (e.g., sweeping left/right) achieved fair results because the participants must perform the gesture again when not initially recognized, which yielded low recall rates. This finding suggests that use of gestures with repeated motions (e.g., circling CCW/CW and waving) than sign actions (e.g., sweeping left/right) is better in designing gesture interfaces.

The main effectiveness of gesture control for self-portraits has been also investigated in previous work [17]. In that work the single directional actions were employed and the hand moving distance was defined at 1.4x the face size. Those experimental results showed a 0.85 of gesture-recognition accuracy, and a better user experience than remote control. Although the results found in [11] are relevant to the Experiment 2, the study has only included single motion actions without considered the repeated motions. However, results observed in Table 3 indicate that the repeated motions are more intuitive and achieved better recognition rate. Moreover, the times of the repeated gesture motions (e.g. the times of waves and circles) can be counted in present technique. This opens the possibility for the design of interfaces that take advantage of the parameter adjustment channel. If the times of motions can be counted, this times could be used to design parameter control interfaces. More work is needed to establish this.

The self-portrait system in a desktop environment proposed in [2] recognized the hand shape based on a skin-color algorithm using fingertip information to make function controls. However, the gesture recognition was sensitive to the lighting conditions. In the present study, we used a motion tracking technique which is robust to changes in the lighting conditions. Interestingly, using head motions to control the camera for self-portrait applications has also proposed [12]. That system worked with a smaller front-facing screen and used head gestures to control the zoom and shutter trigger. However, the system did not provide functions to control other options, such pan-and-tilt that adjust shooting angles.

The gesture interface was more intuitive and provided more freedom and enjoyment compared with conventional remote control and mouse interfaces. However, the difference in terms of efficiency was not significant. The performance and error proneness of the gesture interface prolonged the completion of the experimental tasks. Nonetheless, these glitches lack a considerable effect on the interaction efficiency of the camera according to the users.

Freedom (i.e., device-less hands-free interaction) is one of important factors when preparing various postures. Most of the participants mentioned that using the gesture interface eliminated the need to conceal the remote control or to move backward while taking pictures. They preferred the waving gesture to control the camera shutter, which allowed them to concentrate on the preparation for a specific posture. Thus, the efficiency of the gesture interface offers advantages, particularly when taking multiple shots.

This work examined the performance of gesture interface in taking self-portraits in a desktop environment. Future studies must examine the adoption of the gesture interface in professional cameras, or the interaction with smartphone camera in indoor/outdoor scenarios. In addition, considering the different postures, the pose-assistance interface must be further analyzed to help users prepare their poses, and thus improve the quality of their portraits.

VII. CONCLUSION

The main objective of this work was to design an effective vision-based gesture interface that recognizes distinct gestures to enable the user to control the pan, tilt, and shutter of the camera. To this goal, we investigated the motion gestures (i.e., sweeps, circles, and waves) and used sparse optical-flow to recognize gestures in an estimated region in image to improve the performance. Experimental results showed the accuracy of gesture-recognition was 0.9 in the lightness range 100-200, the processing performance was less than 2ms. In addition, the user experiments indicated a significantly higher satisfaction, particularly in terms of the intuitive motion gestures, freedom, and enjoyment, while using gestures than when using a hand-held remote control or a conventional mouse-based interface.

The outcome of this study has shown that the repeated motion gestures motions (e.g., circling CCW/CW and waving) achieved better performance than single action gestures. Moreover, the counted times of the repeated gesture motions (e.g. the times of waves and circles) opens the possibility for the design of interfaces that adjusting linear values.

Furthermore, the adoption of the motion gestures in other applications, such as virtual reality, head mounted display, mobile phones are also possibly useful.

Acknowledgment

This paper is supported by National Natural Science Foundation of China (NSFC) (No. 61502415), Public Projects of Zhejiang Province (No. 2016C31G2240012, 2016C33196), and Natural Science Foundation of Zhejiang Province (No. LQ15F020005).

References

[1] K. Yamaguchi, T. Komuro and M. Ishikawa, "Ptz control with head tracking for video chat", CHI'09 Extended Abstracts on Human Factors in Computing Systems, **(2009)**, pp. 3919-3924.

[2] S. Chu and J. Tanaka, "Hand gesture for taking self portrait", 14th International Conference on Human-Computer Interaction, Human-Computer Interaction, Part II, **(2011)**, pp. 238-247.

[3] Z. Zivkovic, "Optical-flow-driven gadgets for gaming user interface", in Entertainment computing–icec 2004, pp. 90-100, Springer, 2004.

[4] T. Lee and H. Tobias, "Handy ar: Markerless inspection of augmented reality objects using fingertip tracking", Wearable Computers, 2007 11th IEEE International Symposium on, **(2007)**, pp. 83-90.

[5] T. P. Lee and W. Kosasih, "Exploring the distance, location, and style of "selfies" with the self-pano mobile application", in New contributions in information systems and technologies, pp. 875-884, Springer, 2015.

[6] T. P. Lee, N. Kabbany and C. K. Jun, "Self-portrait images for mobile application", in Advances in information systems and technologies, pp. 827-838, Springer, 2013.

[7] "Taking a great self portrait with your camera", **(2016)**, http://hubpages.com/art/self-portrait-tips.

[8] C.-C. Lo, S.-P. Huang, Y. Ren and Y.-C. Tseng, "Design of a self-portrait application with sensor-assisted guiding for smart devices", Communications and Networking (BlackSeaCom), 2013 First International Black Sea Conference on, **(2013)**, pp. 99-101.

[9] C. Garcia, J.-C. Bazin, M. Lancelle and M. Gross, "Automatic jumping photos on smartphones", Image Processing (ICIP), 2014 IEEE International Conference on, **(2014)**, pp. 2522-2526.

[10] Y. Mei-Chen and L. Hsiao-Wei, "Virtual portraitist: Aesthetic evaluation of selfies based on angle", Proc. Proceedings of the 22nd ACM international conference on Multimedia, Orlando, Florida, USA, **(2014)**, pp. 221-224

[11] S. Chu and J. Tanaka, "Interacting with a self-portrait camera using motion-based hand gestures", Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction, **(2013)**, pp. 93-101.

[12] S. Chu and J. Tanaka, "Head nod and shake gesture interface for a self-portrait camera", ACHI 2012, The Fifth International Conference on Advances in Computer-Human Interactions, **(2012)**, pp. 112-117.

[13] C. Dugan, S. Laumer, T. Erickson, W. Kellogg and W. Geyer, "The #selfiestation: Design and use of a kiosk for taking selfies in the enterprise", in Human-computer interaction – interact 2015, eds. J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque and M. Winckler, pp. 38-46, Springer International Publishing, 2015.

[14] T. H. H. Chua and L. Chang, "Follow me and like my beautiful selfies: Singapore teenage girls' engagement in self-presentation and peer comparison on social media", Computers in Human Behavior, vol. 55, no., **(2016)**, pp. 190-197.

[15] A. Jain, S. Maguluri, P. Shukla, P. Vijay and K. Sorathia, "Exploring tangible interactions for capturing self photographs", Proc. Proceedings of the India HCI 2014 Conference on Human Computer Interaction, New Delhi, India, **(2014)**, pp. 116-121

[16] D. Okabe, M. Ito, J. Chipchase and A. Shimizu, "The social uses of purikura: Photographing, modding, archiving, and sharing", Proc. PICS Workshop, UbiComp, **(2006)**, pp. 2-5.

[17] S. Chu and J. Tanaka, "Design of a motion-based gestural menu-selection interface for a self-portrait camera", Personal and Ubiquitous Computing, vol. 19, no. 2, **(2015)**, pp. 415-424.

[18] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", In Proceedings of the International Joint Conference on Artificial Intelligence, **(1981)**, pp. 674-679.

[19] "Opencv", **(2016)**, http://opencv.org/

[20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", vol. 1, no. 1, (2001), pp. 511-518