

Design of a motion-based gestural menu-selection interface for a self-portrait camera

Shaowei Chu · Jiro Tanaka

Received: 14 August 2013 / Accepted: 12 June 2014 / Published online: 27 June 2014
© Springer-Verlag London 2014

Abstract Self-portraits allow users to capture memories, create art, and advance their photography skills. However, most existing camera interfaces are limited in that they do not support life-size previews, deviceless remote control, and real-time control over important camera functions. In this paper, we describe a new self-portrait camera system and develop a gesture interface for self-portraits. This self-portrait camera system supports life-size projection of a preview as well as a motion-based gesture system to select menu options to control camera functions including the shutter trigger, aperture size, shutter speed, and color balance. We experimentally evaluated the gesture-recognition accuracy and examined the effectiveness of the system compared with a hand-held remote control. The results suggest that the gesture-based interface is effective for controlling self-portrait camera options and improves the user experience when taking self-portraits. The gesture interface is expected to be useful in developing next-generation self-portrait cameras.

Keywords Digital camera · Gesture user interface · Motion gestures · Image processing · Human computer interaction

1 Introduction

Self-portraits became popular with the advent of the digital camera [23], which helps people to capture memories,

create art, advance photography skills, and interface with social network services [1–3, 9]. However, conventional methods of controlling the camera while taking self-portraits, such as use of a self-timer, can be tiresome because the user has to run back and forth to prepare and then pose for the shot, as well as time-consuming and frustrating because a number of shots may be required to obtain a satisfactory portrait [1, 2]. The use of a handheld remote control may mitigate these problems; however, this requires an additional device, which occupies the hand and limits freedom in terms of the possible postures that one can assume, perhaps resulting in unnatural postures [1, 9].

Recent advances in vision-based gesture interfaces offer an advanced technique to enable remote interactions. A number of commercially available cameras, including the Sony Party Shot [8] and Casio TRYX [5], exploit vision-based self-portrait options, such as smile recognition and motion detection. However, these interfaces lack function controls and only support a single shutter trigger. Other important functions of the camera, including the aperture size, shutter speed, and color balance, are desirable for obtaining high-quality images. Because the user must check the pose when preparing the self-portrait, a large display showing a life-size preview is also helpful to improve the user experience.

Since the use of gestures for taking self-portraits was first proposed [14–17], gesture-based approaches for self-portraits have attracted much attention [19]. We believe that gestures can offer an effective, intuitive, and non-intrusive method of taking self-portraits. Without holding any device, the user can be free to concentrate on preparing postures. However, further investigation of the effectiveness and usability of such a gesture interface to control the functions of a self-portrait camera is still required.

Here, we describe the development of a new self-portrait camera system that provides the capability to project a life-

S. Chu (✉) · J. Tanaka
Department of Computer Science, University of Tsukuba,
Tsukuba, Ibaraki 305-8573, Japan
e-mail: chushaowei2010@gmail.com

J. Tanaka
e-mail: jiro@cs.tsukuba.ac.jp

size preview, recognize gestures, and use these gestures to interact with the camera to take self-portraits. In the proposed system, a digital single-lens reflex (DSLR) camera was used to capture video and analyze the motion of the images to recognize gestures in real time. Based on these gestures, a menu-selection interface was developed to control the camera functions. The preview of the camera and the graphical user interface (GUI) were rendered, and life-size images were projected on a wall, allowing the user to see a life-size preview of the self-portrait. To interact with the camera, the user can use a 2D directional gesture to control a pie-menu interface, which was also displayed using the projector. The shutter trigger, aperture size, shutter speed, International Organization for Standardization (ISO) setting, and color balance could be controlled.

The remainder of the paper is organized as follows. The design of the system is outlined in Sect. 2, the implementation is described in Sect. 3, and the accuracy of the recognition of gestures is demonstrated in Sect. 4. In Sect. 5, an experiment to evaluate the system is described, and Sect. 6 summarizes the results in relation to existing work. Section 7 concludes the paper.

2 System proposal and design

2.1 System proposal

A self-portrait camera system should help an individual user to take self-portraits effectively and should allow the user to prepare a desired pose. DSLR cameras typically offer the user control over numerous parameters, including instant preview, aperture size, shutter speed, ISO setting, and color balance [10]. The user must manually verify that these settings are satisfactory, and a skillful photographer may carefully control all of these factors to achieve a high-quality image.

Unlike applications such as television or video game controllers, it is not desirable for the person using a self-portrait camera to hold a device or wear a sensor because the freedom to prepare poses is desirable, and the remote control or sensor should not be included in the image. Although the user may be able to conceal the remote control, possibly with the assistance of a timer, this will divert attention from the pose and/or require that the user move and again prepare the pose.

We propose a self-portrait system that uses a gesture-based interface to control the camera. We believe that this system will enhance the efficiency of and user satisfaction with taking self-portraits. The target environment for the system is indoors, as this allows for manual configuration of the lighting and background, which professional studio portrait photographers usually exploit [6].

2.2 System design

To achieve the goal of taking self-portrait effectively, the system should achieve the following:

1. The system must be capable of showing the user a clear instant preview of the camera in real time.
2. The system must be capable of showing feedback to the gestures that the user performs.
3. Only simple gestures that can be easily memorized yet provide control over many functions are to be used to control the camera.
4. The gesture-recognition technique must be robust and function under a broad range of lighting conditions.

To achieve objective (1), a projector was used to show a preview in real time in the form of a life-sized image. The user can see the image clearly in a what you see is what you get manner and can check the posture(s) of the subject(s) while preparing the portrait. To achieve objective (2), the display shows the feedback to the user while the user is performing gestures. This real-time visual feedback is important and can facilitate more rapid learning of the gesture interfaces [18].

Another important factor is the type of gesture used in the interaction. To achieve objective (3), we propose motion-based 2D directional movement gestures. These are simple gestures that can be performed easily and memorized quickly, yet that provide control over numerous functions of the DSLR camera.

Robustness under varied lighting conditions is also important, particularly because the lighting conditions effectively change when the parameters of DSLR camera are changed. To achieve objective (4), rather than use the image color information to recognize the gestures, we propose to use motion information to recognize the gestures, and recognizing the shape of hands is not required.

The appearance of the system is shown in Fig. 1. The camera preview and user interface are projected on the wall surface. Demonstration movies that illustrate the concept can be found here: <http://www.iplab.cs.tsukuba.ac.jp/~chushaowei/puc/>.

3 System implementation

A Canon 60D DSLR camera was used. The resolution was $1,056 \times 704$ pixels, with a video rate of 30 frames per second (FPS), and the Canon software development kit (SDK) was used to interface with the device [4]. The image sequence was sent to a notebook personal computer (PC) via a USB cable. The processing was carried out using the PC, and a rendered preview of camera and GUI was sent to the large display via a video graphics array (VGA) cable.



Fig. 1 The prototype system, which includes a DSLR camera, a projector, and a notebook PC. The user can use gestures to control the system to take self-portraits

The software and the real-time image processing were implemented in C++, and the graphics rendering was carried out using Microsoft DirectX2D.

3.1 Recognition of 2D directional movement gestures

To control the camera to take self-portraits, we used the recognition of simple gestures involving 2D directional movement, whereby the user moves the hand over a short distance in a particular direction, as shown in Fig. 2. This is simple and easy to memorize, yet it can provide a rich control interface, appropriate for use in developing a menu-selection interface.

The gesture-recognition design goals were as follows:

1. The gesture must be combined with a strong pattern of motions that are robust against a dynamic background.
2. The gesture must be performed within a specific time period.
3. The distance of the hand movement should be small so that it is easy to perform.

To achieve objective (1) we used the Lucas–Kanade optical-flow algorithm [11], which was implemented using OpenCV [7] to estimate three point regions in an ordered line in a specific direction. The algorithm calculates the motion based on differences in successive images on the feature points and is robust to color or lighting conditions [21]. Figure 2 shows a recognition scenario whereby, when a user moves the hand from the left to right, the motion is initially detected by the first point region, and then, according to the movement of the hand, the second point region detects the subsequent motion. If the third point region detects the appropriate motion, the gesture is recognized. The motion at each point must be in the same direction and within the $\pm 20^\circ$ error range. In the example case shown in the figure, the gesture is a movement to the

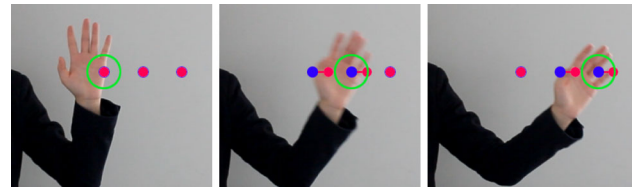


Fig. 2 Gesture recognition. The three point regions are used to estimate the hand motions

right, and frames must be detected from left to right one after the other. This strong pattern of motion excludes unwanted motions generated by accident.

The green circle around a point region in Fig. 2 represents the status of the recognition process. It will move from the first point to the third point according to the gesture motion, and it will either stay or move back to the first tracking point if no motion is detected for a given period of time. This design is intended to achieve objective (2) and to provide visual feedback to the user regarding the recognition status. Examination of the appropriate timing of gesture recognition and the appropriate distant of the hand motion aims to achieve objective (3). We conducted a pilot study, detailed in Sect. 3.3, to select appropriate parameters describing the recognition strategy.

3.2 The pie-menu interface

Using the directional movement gestures, we designed a pie-menu interface, which is an effective method of organizing function options [20, 22]. In the graphical pie-menu interface shown in Fig. 3, the icons clockwise from the left are designated west, northwest, north, northeast, east, southeast, south, and southwest.

The menu interface was designed to pop up in a location where the user can conveniently raise their right hand. The interface position is automatically updated based on the location of the face in the image; it is located to the right of the face, with a horizontal offset of two times the width of the face and a vertical offset equal to the height offset, as shown in Fig. 4. The interface appears once a face is detected and the user remains still for 1 s. If the head moves, the interface disappears.

The GUI was designed to be animated according to the gesture performed. As Fig. 3 shows, once the user moves the hand toward the east and the first tracking point is detected, the east icon will be up-scaled in size by 120 %. If the user performs the gesture and the third tracking point detects the motions, then a scaling animation will be started to scale the icon to 200 % of its original size for 1 s. This graphical feedback is used to prompt that the gesture has been recognized and that the selected action is triggered.

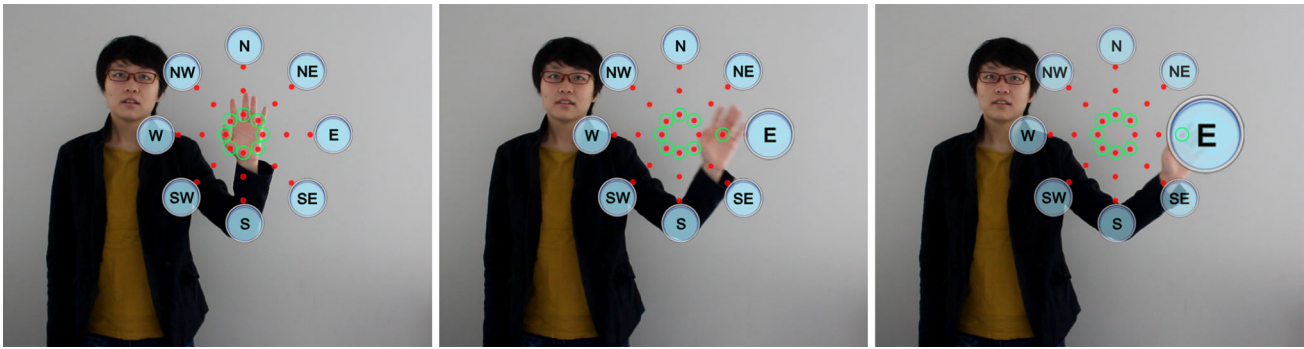


Fig. 3 The menu interface with the GUI for feedback. The *green circle* moves with hand motion, and when hand reaches second tracking point, the icon is *scaled* by 120 %. When hand reaches the third tracking point, the action is activated (color figure online)

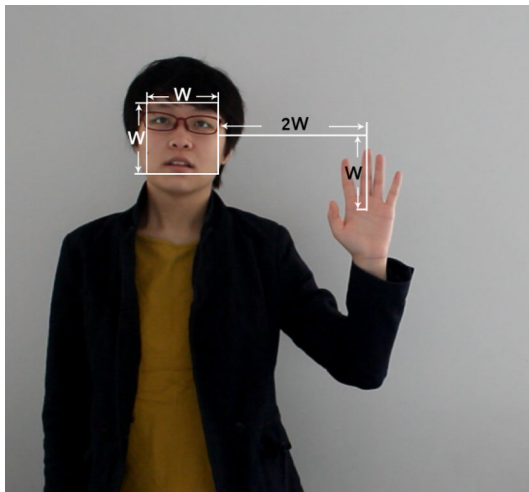


Fig. 4 The interface position in relation to the face position

3.3 Pilot study

We examined the appropriate radius of the pie-menu interface (i.e., the distance of hand motion) and the timing for performing gestures. These parameters are critical to designing an effective gesture interface, as the gesture interfaces are inherently error prone. The computational performance was also evaluated.

Four students majoring in Human–Computer Interaction were invited to join the design process. They were asked to stand in front of the system so that the face region occupied approximately 100×100 pixels and perform the gestures to interact with the pie-menu interface. Three different radii of the pie-menu interface were used: $1.2\times$, $1.4\times$ and $1.6\times$ in relation to the face size. Each person was asked to perform each gesture 24 times. The accuracy of gesture performance and the time taken to perform the gesture were recorded.

A summary of the results is shown in Fig. 5. The main observations are as follows. First, at $1.4\times$, the pie-menu

interface achieved the best accuracy of 0.95. The average time to perform gestures was 659 ms, with a SD of $\sigma = 31$ ms. At $1.2\times$, it was more difficult to be accurate due to the interface, and the accuracy was 0.78. When the interface was $1.6\times$, the user had to move the hand farther, which led to an increased time for performing gestures. Based on these observations, we chose to design the pie-menu interface with a radius of $1.4\times$ the face size. Furthermore, we restrict the time for performing gestures to 700 ms; hence, the user must perform the gesture within this time to exclude spurious motion from the background or any other unwanted motion generated by the user.

After determining the radius of the interface, we then calculated the size that the image should be for use in image processing and gesture recognition to achieve good performance while not adversely affecting the accuracy or requiring a long time to process the data. We tested the recognition accuracy with four different interface sizes ranging from 300×300 to 120×120 pixels. The results are shown in Fig. 6. From these results, we conclude that 240×240 pixels provided the best trade-off between accuracy and process time.

3.4 Interface design and function mapping

We used the 2D directional gesture-activated pie-menu interface to control a DSLR camera for self-portrait images, as shown in Fig. 7. We used the northeast, north, and northwest gestures for the main functions in the interface design, as these are more intuitive to perform. The main menu is shown in Fig. 7; the icons from the left to right represent aperture size, ISO setting, shutter trigger, color balance, and shutter speed. The icon images were designed to be the standard icon design in the DSLR camera manual. North was used for the shutter triggering function, which is the most commonly used command. The camera will run in auto-focus on the user's face once the shutter is triggered. It typically takes 3 s to focus, so we set a timer counting down from 5; when it reaches 1, the picture is taken.

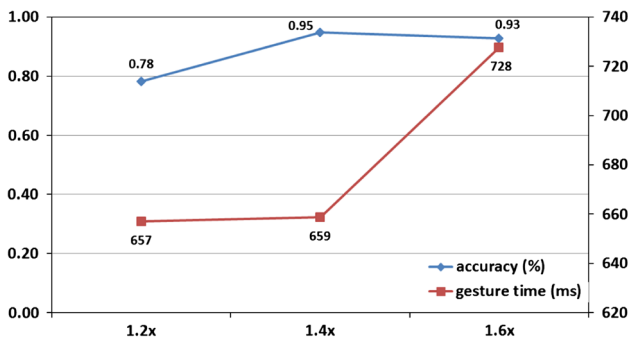


Fig. 5 The accuracy and gesture time

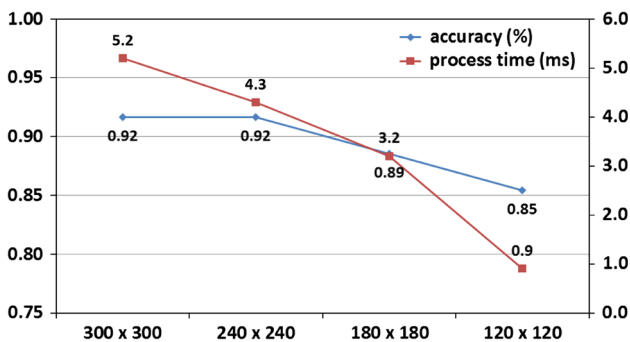


Fig. 6 The accuracy and process time for four different interface sizes

The menu interface has two hierarchy levels. Once an option is selected in the main menu interface, the aperture size, ISO setting, or shutter speed is selected, and the system switches to the value-adjust interface, as shown in the middle image of Fig. 7. The value-adjust interface includes plus and minus icons, which are mapped to the plus and minus functions. The North direction is used for the back function to return to the main menu. The color

balance function has six options, represented in Fig. 7: auto, daylight, shade, cloudy, tungsten lighting, and fluorescent lighting. The interface switches back to main menu after the user has selected an option.

4 Experiment 1: Gesture-recognition accuracy

We conducted an experiment to evaluate the recognition accuracy of the 2D directional gestures and the interface. We recruited eight participants (five male, three female), with an average age (mean, SD) of 25.5 years ($\sigma = 2.8$ years). The participants were asked to select options indicated by the instruction for the interface (see Fig. 8). The experimenter set the interface position to the center of the participant’s hand to allow gesture control by using a mouse. During each task, a destination direction icon was marked by a red circle, as shown in Fig. 8, and the participant was required to perform the gesture to select the destination direction. In each session, the participant was required to complete 24 gestures three times each. The order of the gesture selection was random.

4.1 Robustness under different lighting conditions

We tested the accuracy of the system under three different lighting conditions: dark (light = 100 ± 10), normal (150 ± 10), and bright (200 ± 10), as shown in Fig. 8. Here, 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 unit. Each participant was asked to complete one session of gesture input for each lighting condition.

The results are shown on the Tables 1, 2, and 3. The tables list the mean accuracy results for recognition of the

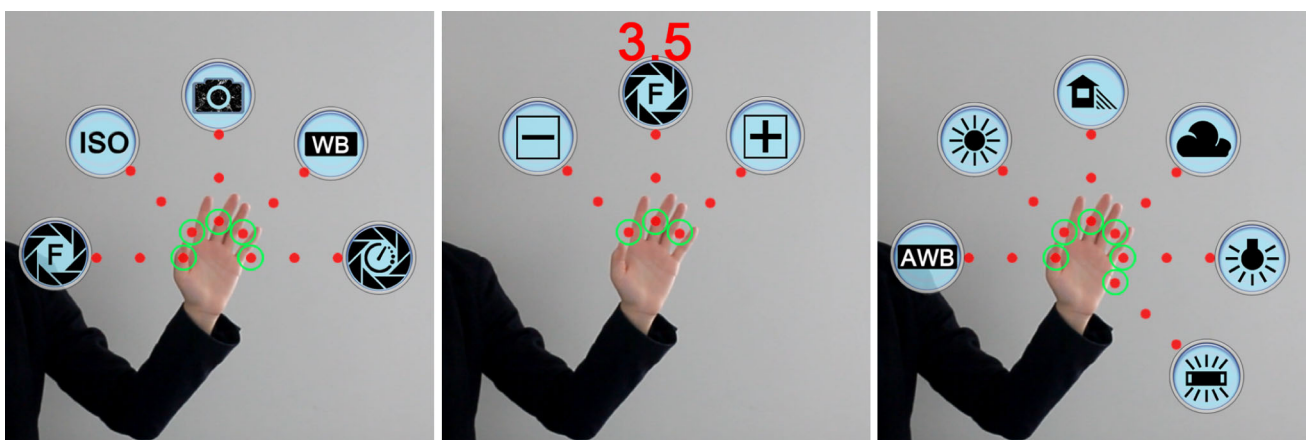


Fig. 7 The interface design and function mapping. Left the main menu interface; middle the parameter-adjust interface; right color balance interface

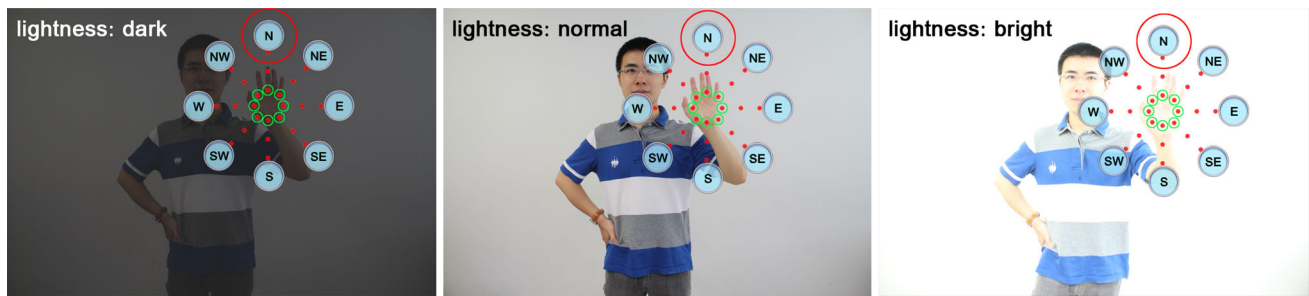


Fig. 8 Gesture accuracy under the three different lighting conditions

eight directions; the leftmost column indicates gesture inputs, and the corresponding row shows the recognition result. The average accuracy on three different conditions are: 0.79, 0.91, and 0.85. In the dark condition, the optical-flow tracking on the point region easier to generate wrong result, which lead to a lower accuracy. This is also observed in very bright condition. And the failure cases occurred for several reasons: first, the user’s forearm or body sometimes conflicted with the hand motions, particularly in the West and Northeast direction. Second, rapid hand movement sometimes produced an incorrect result or no result. In cases with no result, the user often tried to move his/her hand back to the center of the interface and perform the gesture again, but overshoot the center and moved in the opposite direction, causing an incorrect result.

Statistically, we calculated the one-way analysis of variance (ANOVA) on the result, and it showed no significant effect on the three conditions, $F(2, 21) = 3.11, p = 0.07$. Therefore, we conclude the gesture-recognition accuracy is about 0.85 on average from the lightness arrange from 100 to 200 unit.

4.2 Accuracy of the camera at different distances

Another goal of the gesture recognition is to detect the gestures in a full-body portrait case. In such a situation, the camera takes full-body shots, which can be achieved by adjusting the camera lens to zoom out or by increasing the distance of the pose from the camera. Therefore, we also expect that the recognition can detect gestures made by a hand of the size appropriate to the image size. In this experiment, the participant moved away from the camera so that the whole body was in the field of view of the camera, as shown in Fig. 9.

The recognition accuracy is shown in Table 4; the accuracy was 0.89 on average. ANOVA comparing these data with the results for upper body only show no significant effect $F(1,141) = 0.39, p = 0.54$. We conclude that the accuracy of our proposed technique for gesture recognition using the full body and just the upper body was 0.86.

Table 1 Gesture recognition for lightness = 100 ± 10

	W	NW	N	NE	E	SE	S	SW
W	0.75	0.08	0.08	0.04		0.04		
NW	0.04	0.71	0.17	0.08				
N			0.92				0.08	
NE		0.04	0.29	0.58	0.04	0.04		
E			0.04	0.08	0.79		0.08	
SE		0.04		0.04	0.04	0.83		0.04
S			0.08			0.04	0.92	
SW	0.04	0.04	0.04	0.08				0.79

The leftmost column indicates required gesture input, and the corresponding row shows the frequency of the gesture recognition

Table 2 Gesture recognition for lightness = 150 ± 10

	W	NW	N	NE	E	SE	S	SW
W	0.92	0.08						
NW		0.88	0.12					
N		0.04	0.92			0.04		
NE			0.17	0.75	0.04			0.04
E				0.08	0.92			
SE			0.04			0.92	0.04	
S							1.0	
SW								1.0

The leftmost column indicates required gesture input, and the corresponding row shows the frequency of the gesture recognition

Table 3 Gesture recognition for lightness = 200 ± 10

	W	NW	N	NE	E	SE	S	SW
W	0.67	0.25	0.04			0.04		
NW		0.83	0.13	0.04				
N			0.96				0.04	
NE	0.04		0.21	0.75				
E			0.08	0.08	0.83			
SE	0.04			0.04		0.83	0.04	0.04
S							1.0	
SW				0.04			0.04	0.92

The leftmost column indicates required gesture input, and the corresponding row shows the frequency of the gesture recognition

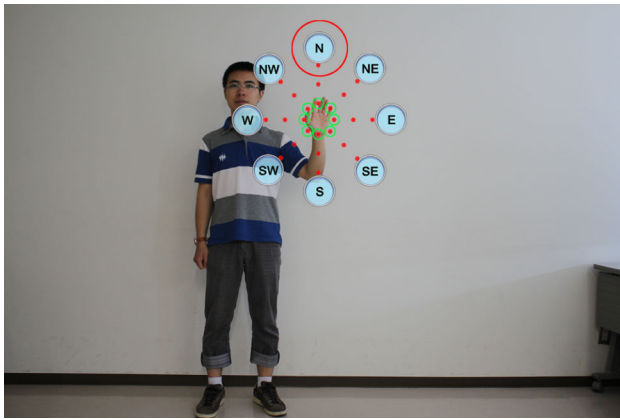


Fig. 9 Full-body gesture recognition

Table 4 Gesture-recognition accuracy in the full-body case, where the face size was 50 × 50 pixels

	W	NW	N	NE	E	SE	S	SW
W	0.79	0.13	0.08					
NW		0.92	0.04			0.04		
N		0.04	0.92				0.04	
NE			0.29	0.71				
E			0.04		0.92	0.04		
SE		0.04				0.96		
S			0.04				0.96	
SW			0.04				0.04	0.92

The leftmost column indicates required gesture input, and the corresponding row shows the frequency of gesture recognition

5 Experiment 2: User experience

In this experiment, we evaluated users’ experience of the system. The effectiveness in facilitating user satisfaction with taking self-portraits and a comparison with a conventional remote control interface were evaluated.

5.1 Participants

We recruited 12 graduate students in the University of Tsukuba (six male, six female), with an average age of 25.6 years ($\sigma = 3.1$). All of the participants were right-handed.

5.2 Conditions

We asked participants to take self-portraits using the gesture-recognition interface and a Nintendo Wiimote remote control interface. The GUI of the remote control interface was the same as that of the gesture interface, but the Wiimote controller was used to navigate the functions, as shown in Fig. 10. The D-pad directions on the controller were mapped to the options selection on the menu, and the A button was



Fig. 10 Nintendo Wiimote remote control and GUI



Fig. 11 Pose for the self-portrait

used to confirm and trigger the selected action. The selected option on the menu was marked with a red circle.

The participant was asked to navigate and set a few of parameters before taking a picture. A specific pose for taking a self-portrait is required: placing hands on the waistline, see Fig. 11. This pose is a very common pose easily be prepared by both male and female. And to avoid bias in the experiment, half of the participants used the handheld remote control first, and the other half used the head gesture interface first.

5.3 Procedure

The participants were invited to use the system, and the experimenter introduced participants to the interface functions and explained how to control the camera using gestures or the remote control. The participants were then invited to stand in front of the system at a distance of 1.5 meters and were given the opportunity to familiarize

themselves with the system. They were then instructed to take self-portraits using both the gesture interface and the remote control.

Because it is difficult to account for personal preferences, we asked the participants to select and set each of the parameters of the camera in order. First, the aperture was selected and adjusted one increment. Next, the ISO setting was selected and adjusted one increment. Then, the color balance was selected and an option was selected. Then, the shutter speed was selected and adjusted one increment. Finally, the shutter trigger was selected, and the 5-second timer counted down while the camera auto-focused on the user's face as they prepared the pose.

The participants were asked to use the gesture interface and the remote control to complete the same task. During the experiment, we recorded the time taken and number of actions required to complete the task, and in addition, the behavior of the users was observed as they used the system. After the participants completed these tasks, they were asked to complete a questionnaire.

5.4 Questionnaire

We designed a questionnaire to determine whether the gesture interface was effective and whether the participants enjoyed using the interface to take self-portraits. The questions were as follows:

- Q1 Do you think the life-size display of an instant portrait preview was useful for monitoring the pose?
- Q2 Were the gestures easy to perform and memorize, and was it easy to control the camera functions?
- Q3 Do you think the GUI and the visual feedback for the gestures were effective?
- Q4 Was the gesture interface intuitive?
- Q5 Was the remote control interface intuitive?
- Q6 Did the gesture interface provide freedom to prepare the pose and take self-portraits?
- Q7 Did the remote control provide freedom to prepare the pose and take self-portraits?
- Q8 Did you enjoy using the gesture interface to take self-portraits?
- Q9 Did you enjoy using the remote control to take self-portraits?

We assigned scores according to a 5-point Likert scale: strongly disagree = 1, disagree = 2, neutral = 3, agree = 4, strongly agree = 5.

5.5 Results

We computed the average score for each question among the 12 participants. Table 5 shows the average results for

Table 5 The rating from participants for question 1–3 ($N = 12$)

	Mean	SD
Q1 (life-size)	4.5	1.0
Q2 (easy)	4.4	0.3
Q3 (GUI)	3.9	0.6

the first three questions. The participants broadly agreed with these questions.

Question 1 assessed the use of the large display. Most participants reported that the large display was effective and that they are able to observe their poses well. Four participants complained about the mobility of the system, since it uses a projector, making it impossible to work outdoors.

Question 2 assessed the ease of performing gestures to control the camera functions. Most participants agreed that the gestures were easy to perform, memorize, and use. One participant suggested that using finger movements for the interactions would induce less fatigue. Four participants reported initial confusion about performing gestures, as they thought they should perform the gestures at the region corresponding to the icons.

Question 3 assessed the GUI and visual feedback of the interface design. Although the participants mostly agreed that the GUI design was effective, two participants commented that the speed of the animation was too fast and that it would be better if the GUI did not overlap with the subject's body.

Questions 4–7 assessed user satisfaction with the new system compared with the remote control. The results are shown in Table 6. The ANOVA revealed a significant difference between the two techniques, with the gesture interface scoring significantly higher than the remote control.

Questions 4 and 5 assessed the intuitiveness of the two techniques. The gesture interface was reported to be more intuitive because the participants were not required to hold a device. One participant suggested using the foot to make interactions, as this would totally free the hands.

Questions 6 and 7 assessed the freedom to prepare poses. Because the freedom to relax and prepare a pose is important in taking self-portraits, the design of the interactive system must take this into account. Most participants reported that they felt free when using the gesture interface, as they did not need to consider concealing the remote control when taking pictures. We also observed participants' behavior when using the system, particularly when preparing poses. When using the remote control, participants had to conceal the remote controller and prepare the specific pose during the timer countdown. Because the camera must auto-focus on the user's face, the user cannot move to put the remote control down, so participants tossed the remote control to a soft chair near the camera. This behavior distracted their

Table 6 Comparison of questionnaire data for the gesture-based interface and the remote control ($N = 12$)

	Gesture interface		Remote control		<i>p</i>
	Mean	SD	Mean	SD	
Q4, Q5 (intuitive)	4.6	0.5	3.3	1.0	<0.05
Q6, Q7 (freedom)	4.4	0.4	2.8	1.1	<0.05
Q8, Q9 (satisfaction)	4.7	0.4	3.2	1.1	<0.05

Table 7 The number of actions and time taken to complete the task using the gesture interface and the remote control ($N = 12$)

	Gesture interface		Remote control		<i>p</i>
	Mean	SD	Mean	SD	
Number of actions	13.4	2.3	24.5	3.4	<0.05
Total time	26.2 s	11.1	23.6 s	4.6	0.03

attention and led to motion-blurred pictures. However, using the gesture interface, motion-blurred images were not generated, and participants felt free to relax and prepare the pose. Although the gesture interface provided more freedom than the remote control did, it also required users to place their hand in a specific position to perform the gestures, leading to limitations that were noted by the participants.

Questions 8 and 9 assessed overall user satisfaction. The participants agreed that the gesture interface was an appropriate technique for controlling the camera.

Finally, we report the time taken and number of actions performed by participants during the experiments as shown in Table 7. The gesture interface required fewer actions to complete the task, as each gesture was able to trigger an action. However, a longer time was required to complete the task owing to the increased time necessary to complete the gesture compared with using the remote control, as well as the increased error rate of the gesture interface.

6 Discussion

Chu and Tanaka [14] proposed the use of hand gestures to control cameras for taking self-portraits. They used a large display and recognized the hand shape based on a skin-color algorithm using fingertip information. However, the system used a web camera, and the gesture recognition was sensitive to the lighting conditions. In the present study, we used a professional DSLR camera; with this camera, changes in parameters such as the aperture size, shutter speed, and color balance affect the light level of the image. Therefore, our technique is robust to changes in the lighting conditions. Additionally, our interface provides numerous function controls, many of which are essential options for the camera.

Chu and Tanaka [15, 16] head interface was able to control the camera for self-portrait applications. 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. Our system provides a rich interface to control many important camera functions. Furthermore, our proposed hand interface can achieve similar efficiency to that of a remote control.

Reports relating to self-portrait imaging are sparse. A number of articles have discussed similar approaches [1, 2, 9], such as using a long arm to hold the camera to take the photograph and using a self-timer. Camera manufacturers also produce cameras with two liquid crystal display (LCD) screens or a rotating frame, including the Samsung dual-view camera and Casio TR150, and front-facing screens allow the users to see themselves when taking self-portraits. Although this method can be fun and can record the moment, such pictures are generally of poor quality and may distort the face due to the proximity of the lens. Furthermore, it is difficult to maintain a steady hand.

Sony’s Party Shot pan-and-tilt camera uses face and smile detection to track users and takes pictures automatically; however, it lacks interactivity and does not provide any control capabilities. In contrast, our system provides a higher level of control, allowing the user to set numerous functions using gestures.

Although the Microsoft Kinect depth sensor camera provides an easier way to recognize hand gestures, we wish to develop a pure software technique for a 2D camera that can be implemented on existing cameras and that does not require additional hardware sensors for depth information. A number of investigations have examined motion-based gesture recognition [12, 13, 24] using a 2D camera. Generally, however, the performance is slow, as they estimate motion based on a large area region with many tracking points. In our work, we only detect gestures in one section of the image corresponding to the hand position, the relative position of which was determined using face recognition. Furthermore, we manually arranged limited tracking points to recognize the motion gestures. These techniques greatly reduced the computational complexity of the problem, resulting in faster gesture recognition.

7 Conclusion

A gesture-based system allows users to control a camera for self-portrait applications in an indoor environment using gestures was developed and evaluated. The user was able to use an intuitive 2D directional gesture system to control the camera using a pie-menu interface. The menu interface offered control over the aperture size, shutter

speed, ISO setting, color balance, and the shutter trigger. To recognize the gestures, we used a motion-based recognition technique employing optical-flow tracking in real time to detect hand motions. To design the gesture interface, we conducted a pilot study to determine the appropriate parameters to optimize the recognition accuracy.

We evaluated the performance of the proposed gesture interface by testing its accuracy under different conditions, including different light levels and a range of distances between the user and the camera. We achieved 85 % accuracy on average, and the light condition did not significantly affect the accuracy. We then evaluated the usability of the gesture interface by comparing it with a remote control. The results showed that the users preferred our system. The motion-based gesture interface is expected to be useful in the development of next-generation self-portrait cameras.

In the future, we plan to improve the gesture-recognition accuracy, for example by using a hidden Markov model to train and estimate the recognition parameters. We also plan to use a pan-and-tilt platform or mobile robot to control the shooting angle. Furthermore, we plan to apply the motion gesture interface to other applications, including augmented reality, helmet-mounted display, and applications that use a camera as an input channel.

References

- 100 seriously cool self-portraits (and tips to shoot your own!). <http://photo.tutsplus.com/articles/inspiration/100-seriously-cool-self-portraits-and-tips-to-shoot-your-own/>
- 4 tips for taking gorgeous self-portrait and outfit photos. <http://www.shrimpsaladcircus.com/2012/01/self-portrait-outfit-photo-graphy-guide.html>
- 5 reasons why you should take a self portrait. <http://www.iheartfaces.com/2012/04/self-portraits-tutorial/>
- Canon digital camera software developers kit. http://usa.canon.com/cusa/consumer/standard_display/sdk_homepage
- Casio tryx camera. <http://www.casio-intl.com/asia-mea/en/dc/extr150/>
- Indoor portrait. <http://www.photoflex.com/pls/category/indoor-portrait>
- Open source computer vision library (opencv). <http://opencv.org/>
- Sony party shot. <http://www.sony.jp/cyber-shot/party-shot/>
- Taking a great self portrait with your camera. <http://www.squidoo.com/self-portrait-tips>
- Taking portraits using your digital slr. <http://www.dummies.com/how-to/content/taking-portraits-using-your-digital-sl.html>
- Baker S, Matthews I (2004) Lucas–Kanade 20 years on: a unifying framework. *Int J Comput Vis* 56(3):221–255
- Bayazit M, Couture-Beil A, Mori G (2009) Real-time motion-based gesture recognition using the GPU. In: IAPR conference on machine vision applications (MVA), pp 9–12
- Chen M, Mummert L, Pillai P, Hauptmann A, Sukthankar R (2010) Controlling your tv with gestures. In: MIR 2010: 11th ACM SIGMM international conference on multimedia information retrieval, pp 405–408
- Chu S, Tanaka J (2011) Hand gesture for taking self portrait. In: Proceedings of the 14th international conference on human–computer interaction: interaction techniques and environments—volume part II, pp 238–247
- Chu S, Tanaka J (2012) Head nod and shake gesture interface for a self-portrait camera. In: ACHI 2012, the fifth international conference on advances in computer–human Interactions, pp 112–117
- Chu S, Tanaka J (2013) Development of a head gesture interface for a self-portrait camera. *Trans Hum Interface Soc* 15(3):247–259
- Chu S, Tanaka J (2013) Interacting with a self-portrait camera using motion-based hand gestures. In: Proceedings of the 11th Asia-Pacific conference on computer–human interaction (APCHI2013), pp 93–101
- Eisenstein J, Mackay WE (2006) Interacting with communication appliances: an evaluation of two computer vision-based selection techniques. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 1111–1114
- Graham-Rowe D (2011) Say cheese! taking photos with a wave of the hand. *New Sci* (2817):25
- Guimbretière F, Nguyen C (2012) Bimanual marking menu for near surface interactions. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 825–828
- Molnár J, Chetverikov D, Fazekas S (2010) Illumination-robust variational optical flow using cross-correlation. *Comput Vis Image Underst* 114(10):1104–1114
- Ni T, Bowman DA, North C, McMahan RP (2011) Design and evaluation of freehand menu selection interfaces using tilt and pinch gestures. *Int J Hum Comput Stud* 69(9):551–562
- Okabe D, Ito M, Chipchase J, Shimizu A (2006) The social uses of purikura: photographing, modding, archiving, and sharing. In: Pervasive image capture and sharing workshop, ubiquitous computing conference, pp 2–5
- Zivkovic Z (2004) Optical-flow-driven gadgets for gaming user interface. In: 3rd international conference on entertainment computing C ICEC 2004, pp 90–100