Interacting with a Self-portrait Camera Using Gestures

Graduate School of Systems and Information Engineering

University of Tsukuba

July 2013

Shaowei Chu

Abstract

Most existing digital camera user interfaces place little emphasis on self-portrait options. Therefore, it is not always easy to take self-portraits using conventional user interfaces. For example, use of a self-timer, can be tiresome (having to run back and forth to prepare and then pose for the shot), time consuming. Use of a handheld remote control may be a better choice, but the additional device occupies the hand, which limits freedom in terms of the possible postures one can assume and often results in unnatural postures.

Therefore, it is important to develop a user-friendly interface that allows an individual to take portrait shots both effectively and efficiently. In this research we present the design of new, vision-based gesture interfaces for interacting with a self-portrait camera. Three different types of gesture interfaces were described: 1) *hand gesture interface*, 2) *motion-based hand gesture interface* and 3) *head gesture interface.*

Hand Gesture Interface - In this study, we presented a new approach of interaction technique for taking self-portraits. The proposed self-portrait camera prototype utilized a camera with a pan & tilt platform to take pictures and a large size of display to show the camera preview. A finger detection technique was used to enable user to trigger camera shutter by using hand gestures. Besides, a novel hand directional movement gesture recognition technique and its interface were proposed to control the camera's pan & tilt. We evaluated the system performance, accuracy of gesture recognition, and user experience in the experiments. The results showed that the hand gesture interface is effective and useful for taking self-portraits.

Motion-based Hand Gesture Interface - In this study, we proposed the motion-based hand gestures, and aimed at developing a rich interface for controlling a professional digital single lens reflection (DSLR) camera. Three types of motion-based hand gestures: *waving*, *eight-direction selection* and *circling* gestures were introduced. They provided a good functional mappings scalability for interface design and offered a complete solution for controlling many functions of a DSLR camera, such as shutter speed, aperture, ISO, white balance, etc. In experiments, we evaluated the effectiveness and user satisfaction. The results showed motion gestures were able to provide us a way to develop a rich gesture interface for control a DSLR camera. In addition, The gesture interface was more useful than existing camera interfaces, such as self-timer.

Head Gesture Interface - In the third study, we proposed a technique that uses head nodding, shaking and mouth-opening gestures as the interface. Intuitive nodding and head-shaking gestures control the camera zoom in/out on the face, and a mouth-opening gesture triggers the camera to take a picture. The advantages of head gestures are that they can work well with small displays, and are suitable as the basis of a zooming interface. Because the face is unlikely to move outside of the field of view of the camera, as the face is the focus and most important region in a self-portrait, it is always available for performing gestures and mapping to camera control functions. We conducted an experiment to examine its usefulness (effectiveness, efficiency, and satisfaction) and compared it to a remote control. The results showed that users were able to use the gesture interface to interact with a camera effectively and felt satisfied with the technique for taking self-portraits. In addition, the gesture interface had a better satisfaction evaluation than the handheld remote control.

According to the study, important results confirmed that the gesture interface could be an effective interaction technique for developing a self-portrait camera. Such interfaces have a number of advantages: 1) they provide remote control capability that allows a user to interact with a camera while he is posing in front of it; 2) they are so effective and intuitive, that the user is able to perform just a wave of the hand to take a self-portrait shot; 3) the gesture interface is non-intrusive, that is without holding any devices, the user can feel free and concentrate on preparing postures.

Acknowledgments

I am deeply grateful to Professor Jiro Tanaka, my thesis supervisor, for his many valuable suggestions, precise directions, and kind encouragement. His optimistic and humorous personality deeply affected me, made me think positively, and gave me the confidence to adhere to the completion of this long-term research. His constant support, and the many resources and research equipment that he provided me were invaluable to conduct the studies.

I also greatly thank Assistant Professor Simona Vasilache for her English proofs and kind suggestions. As well, Associate Professor Shin Takahashi, Associate Professor Kazuo Misue and Associate Professor Buntarou Shizuki for their kind advice and suggestions regarding this research. I would like to thank Professor Yukio Fukui, Professor Kazuhiro Fukui, Associate Professor Tomohiro Haraikawa, and the other members of my thesis committee, for their many useful comments on this research.

I am also grateful to all the members of the IPLAB, Interactive Programming Laboratory, University of Tsukuba, for giving me many opportunities to discuss my research with them.

Finally, I extend my sincere gratitude to my parents.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Since the introduction of the first digital camera in 1988, the digital camera quickly swept over the photographic industry [1]. It provides many advantages: immediate image review, fast operating speed, competitive high quality images, and lower cost, etc. [2], [3]. That made the digital camera spread rapidly to the consumer market. With the rapid growth of digital cameras, the volume of self-portrait images is growing rapidly [4]. In general, taking self-portraits is becoming popular, particularly among the young [5]. Different from traditional photography, taking self-portraits has several benefits: it is fun, stimulates our creativity, logs our life in vivid photos, advances our photography skills, and presents us to society in social network services, such as Flickr, Facebook, etc. [6], [7].

However, when people want to take self-portraits using the available digital cameras, they have to face many problems. Since the design of interaction approaches for self-portrait options is limited, it is usually difficult to use the conventional ways to take self-portraits. For example, use of a contact-based self-timer, where people have to run back and forth to prepare and then pose for the shot, can be tiresome, time consuming, and frustrating as many shots may be needed to obtain a satisfactory portrait [8], [9]. Use of a handheld remote control may be a better choice, but the additional device occupies the hand, which limits freedom in terms of the possible postures people can assume and often results in unnatural postures [8], [10]. Conventional interface designs pay only modest attention to user interaction, and largely do not consider user-friendly ways for taking self-portraits.

Therefore, we feel that it is important to design a new interaction technique for self-portrait camera, and develop a new user-friendly interface that enables the user to interact with cameras and take self-portraits both effectively and efficiently.

1.1 Goal and approach

The first goal of this study is to design a remote control interface that provide the user with a way to interact with the camera when he/she is preparing the postures in front of the lens.

Our second goal is to implement a real-time interaction technique that enables the user to get instant response and visual feedback when interacting with the camera.

The third goal is to develop a more user-friendly interface, of which the usability should be better than conventional techniques, such as self-timer, hand held remote control.

Consequently, according to these goals, we propose a novel vision-based gesture interface for interacting with a self-portrait camera. The gestures offer several advantages: 1) they provide remote control capability, we can develop a real-time image-processing algorithm to make the camera see and responde to the user's gestures; 2) they are effective and intuitive, so that the user is able to perform just a wave of the hand to take a self-portrait shot; 3) the gesture interface is non-intrusive, that is, without holding any devices, the user can feel free and concentrate on preparing postures.

We implemented the applications in three different studies to prove the idea. And in the system development, we will follow several design principles. First, to maintain the intuitiveness, on the design of the interfaces we will not use any sensors or markers, but will apply pure vision-based gesture recognition techniques. Second, we will design the appropriate real-time visual feedback with gesture motions, which may enhance the user experience. Third, we will keep the interfaces simple and easy to use.

1.2 Dissertation organization

The presented dissertation is structured as follows. Chapter 1 is an introduction that defines the scope of the dissertation. Chapter 2 presents the study of using hand gestures for interacting with a self-portrait camera. Chapter 3 explains the usage of motion-based hand gestures to develop a rich interface for interacting with a professional DSLR camera. Chapter 4 describes the third study of developing a head gesture interface for interacting with a self-portrait camera that works with zooming functions. Chapter 5 concludes the dissertation and presents future work.

Chapter 2

Hand gesture interface

This chapter describes the design of a self-portrait camera, our proposed prototype of self-portrait system and the hand gesture interface for interaction [11]. The design and architecture of the system are presented, including detailed descriptions of the gesture recognition algorithm. We conducted an experiment to investigate the usability factors of gesture interface when people interacting with the self-portrait camera. We discussed some of the successes and limitations of our approach. Finally, we propose future directions, not only for this specific project, but also for similar projects that aim at developing vision-based gesture applications.

Our themes are: design a new interaction approach for self-portrait camera that helps user to take self-portraits effectively; propose gesture recognition techniques and develop a hand gesture interface for interacting with the self-portrait camera.

2.1 Designing a self-portrait camera

Soon, advancements in camera design will making cameras more interactive, responsive, and accessible to users, with particular emphasis on interactive approaches that smooth the process of taking self-portraits. Different from traditional photography shooting techniques, taking self-portraits have specific requirements, which guide us to consider a new approach of interaction technique of self-portrait camera.

Front facing LCD screen - A self-portrait camera must be able to be used by a lone individual, so ideally it will have a front facing LCD screen so that the user can see his/her preview and any GUI toolkits accompanied with gestures. The LCD screen size is a tradeoff between preview accuracy and camera mobility. A mobile camera should be equipped with a small front facing screen, makes it more portable for indoor and outdoor use. This design trend is evidenced in recent commercial digital cameras, such as the Samsung dual-view camera (see Figure 2.1 a)) and many other cameras with flip-out screens. In contrast, a more professional self-portrait camera should be equipped with a larger LCD front screen or the ability to output the preview to a TV or external display. Although this kind of camera can only be used indoors, the larger size will provide a clearer preview and help the user prepare the self-portrait. Many good portraits are captured indoors: users can manually configure the lighting conditions and fix colored backgrounds; because they do not feel self-conscious when posing, they can take as many photographs as they need [9], [10]. Another possibility is to embed a projector to the camera, which provides both mobility and larger size of preview. Recent, such kind of feature is evidenced in camera industry, such as Nikon COOLPIX S1200pj camera [12] (see Figure 2.1 b)), with a built-in projector that can project photos and movies onto any flat surface up to 60 inches diagonally.

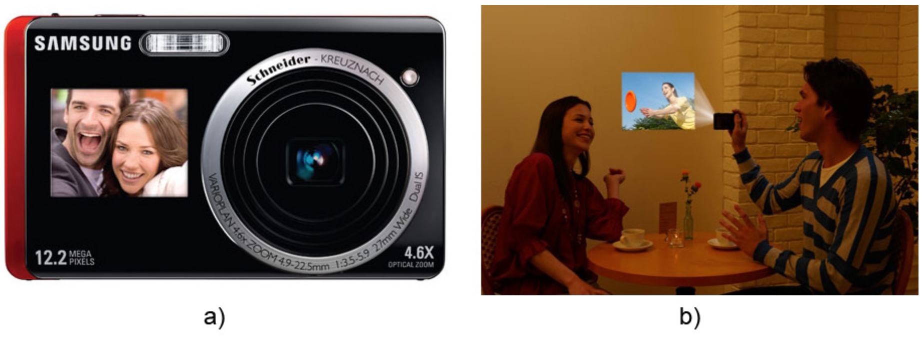

Figure 2.1: Types of cameras. a) Front facing LCD screen camera. b) Projector camera.

Computational smart camera - The digital camera should have computational features such as an open architecture design, or a way to let developers implement customized programs, similar to how smart phones enable developers to release third-party applications. Unfortunately, the camera industry currently has a closed architecture and does not allow modifications to embedded software. This is likely to change in the future, and may become a major trend in next-generation digital cameras. One previous study has already introduced a complete solution for a computational camera [13]; it would be possible to incorporate any image-processing algorithm into this kind of camera, such as an algorithm to recognize user gestures. In fact, this trend of study is becoming popular. With a custom modified firmware, e.g., Canon Hack Development Kit [14] and Magic Lantern Firmware [15], developers can implement customized programs in camera to enhance the shooting options or develop new interfaces. The success applications exist both in photography industry [16] and academic [17]. Furthermore, recently camera manufactures, such as Canon [18], also releasing their camera software development kit (SDK) to support programming customized applications.

Gesture interface - The gesture interface is proved an effective way for remote control and interacts in a ubiquitous environment [19], [20]. A great deal of research within the field of human-computer interaction has focused on gestural interfaces and their applications [19], [21], [22], [23], [24]. Many research prototypes have used sensors [25], [26] or hand held remote controls [27], [28] for gesture interaction. But the additional devices make it intrusive, that user has to wear the sensors or held the devices. However, in self-portrait scenario, an important desire is non-intrusive for user, since that user is eager to prepare various postures, and explore creative possibilities, the additional devices may frustrate user to feel free and concentrate on preparing. Recently, the 3D cameras, such as Microsoft Kinect, with the depth image capturing function, which simplified the gesture recognition and got big success in computer game [29]. However, the special kind of camera combined many devices, one infrared camera and one infrared projector, which made it very wasteful of electricity and heavy weight. It is difficult to be a mobile device.

Therefore, we believe the vision-based gesture recognition offers a better choice. Since the vision-based gesture recognition do not require user to wear sensors or hold a remote controller. Besides, the gesture recognition can be done in software manner by using image processing techniques, which has a lot of literature in human computer interaction. In our long time survey we found that, the hand gestures that combined with small motions should be the better choice for developing gesture interface for interacting with a self-portrait camera. Because that, the small motions are less fatigue and will not disturb user's attentions. But one the other side, the large motion gestures or body gestures may frustrate user when preparing specific posture, and result much fatigue. Therefore, in our study, the recognition of small motions of hand is the objective in developing gesture interfaces.

Based on the above requirements and considering the design principles presented in Chapter 1, we propose our first prototype of self-portrait camera that works in an indoor environment.

2.2 Prototype system

The Figure 2.2 shows the overview of the proposed prototype system. The camera we used is Logitech Orbit AF Camera, capturing 640 x 480 resolution video with 30 FPS with 189-degree field of pan and 102-degree field of tilt. A 30 inches large size display was used to give a clear live view to user. A desktop PC is used for doing the image processing, gesture recognition.

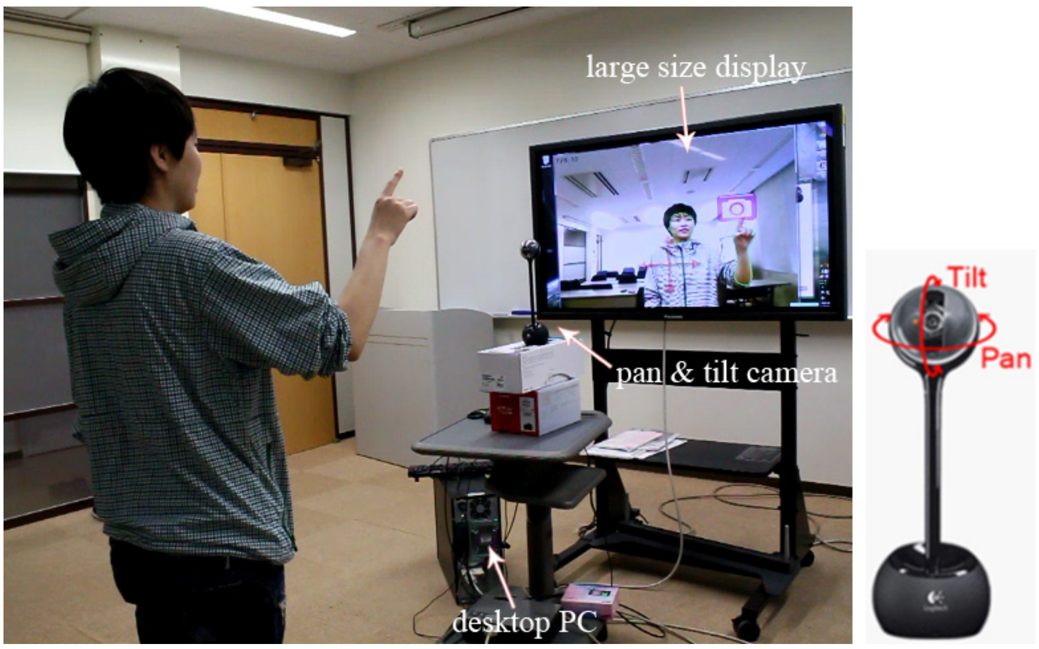

Figure 2.2: Overview of self-portrait camera system.

The novel aspects in this design are follows. First, the idea of using gestures to interact with the camera without physical contact with camera. Second, a pan & tilt camera is used that provides automatic adjusting the camera view angle. User can control the view angle to make photographic composition without physical moves when preparing postures. Third, a large size display gives preview of camera, and shows the graphic user interface and visual feedback according to gesture motions. User can check the instant result and status of the system which greatly enhanced the user experience during interactions.

To interact with the system and take self-portrait shots, user is able to perform gestures and stretch out a finger to activate the shutter control by air touch the hover button. And perform directional movement gestures within the region of cross motion interface to control camera pan & tilt. The preview and graphical user interface of the camera is shown in Figure 2.3. Note that, our system can highlight the skin-color region with colored contour, and mark the detected fingers with circles. These visual feedbacks can make users quickly understand the system and familiar with the interaction procedure, which is important to enhance the user experience when user exploring the gesture interaction system reported in study [30].

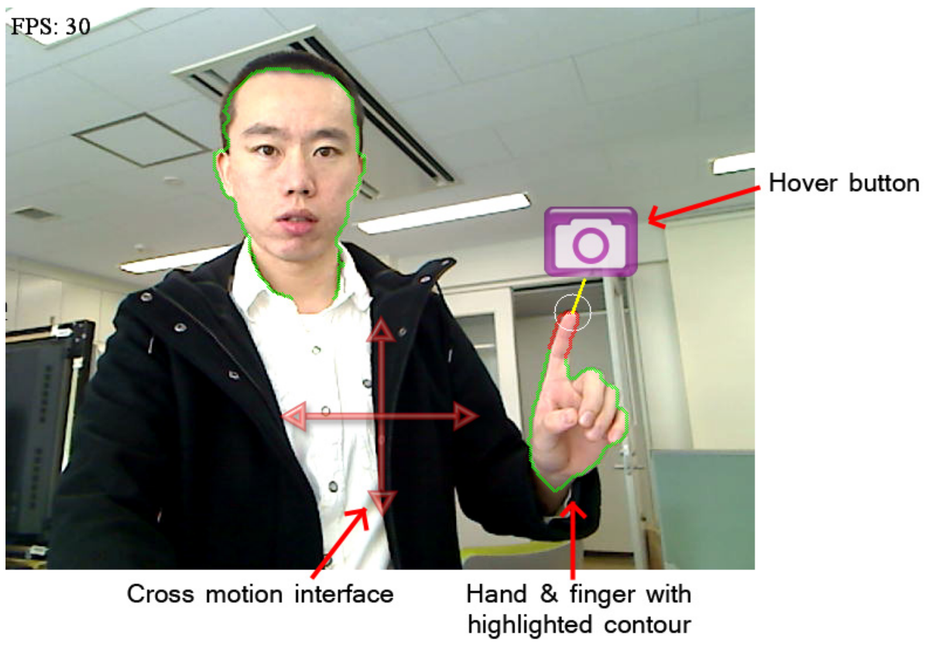

Figure 2.3: Preview of the camera and user interface.

2.3 Gesture recognition and user interface

We propose two types of gesture recognition: finger detection and 2D directional movement. Corresponding to the types of gestures, two types of user interfaces were developed: a near-field touch button and a cross-motion interface.

2.3.1 Hand-finger detection and user interface

Hand and finger detection was implemented by using image processing algorithms on each video frame. A model-based [31] detection approach was applied to recognize hand gestures. The recognition procedure contains three main tasks: 1) separate the hand region from the background of the image, 2) exclude the noise of separate skin-color regions, and 3) detect the fingertips from the overall hand shape. An example of this procedure can be found in Figure 2.4.

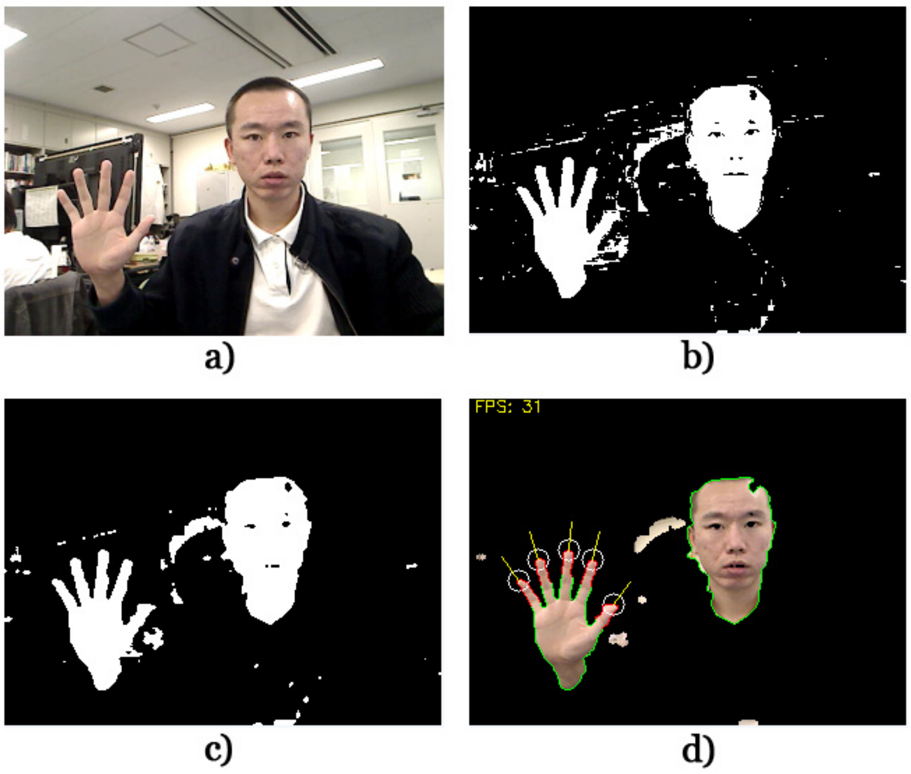

Figure 2.4: Hand-fingertips detection procedure. a) Original image. b) Skin-color segment, c) Median filtered image. d) Detected fingertips with color marker.

For the first task, we re-scaled the original input image size from 640 x 480 pixels to 320 x 240 pixels to increase the processing speed. Then, a skin-color filtering algorithm was applied to separate the different skin tone regions. Although many techniques are available for skin-color filtering [32], [33], [34], [35], we compared the robustness of the techniques in a variety of complex backgrounds using tests of their performance of and accuracy in hand and finger detection. After testing indoors under daylight incandescent and fluorescent lamp

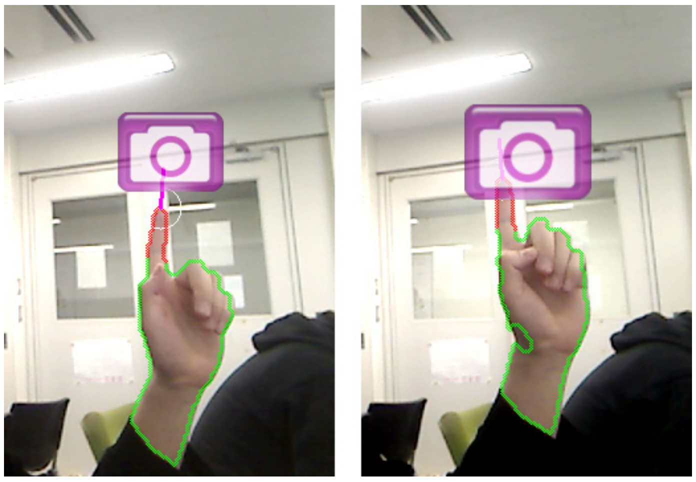
Figure 2.5: Interacting with the hover button interface.

lighting conditions, we chose to use the [35] algorithm for the segmentation processing of hands. This method of skin-color filtering provided good robustness against complex backgrounds, linear compute complexity, and competitive accuracy of segmentation results. In Figure 2.4, the two images on top are the original input image (left) and the skin-color filtered result (right). The image was captured indoors using a fluorescent lamp.

In the second task, we excluded noise from the skin-color filtering. Figure 2.4 b shows the results of the skin-color filtering; many trivial regions can be seen in the background, and the edges of the hands and face are sharp. Given that the shape of the hand should have smooth edges, a number of mathematical morphology techniques were tested as an edge filter, including eroding and dilating algorithms [36]. We finally settled on median filtering [37] as the best result for our scenario. Figure 1c shows the median filter processed result, with 3 x 3 kernel, from the segmented skin-color regions.

The final task identified the fingertips. First, we apply a single-connected components contour finding algorithm [38] to locate hand contours. We assumed that the hand should have a larger region than the fingers. Thus, small regions (contour length less than 600) were excluded, leaving only the larger area of skin-color regions as hand candidates.

Second, based on the obtained contours, the fingertips were detected using a k-curvature algorithm [31] that detects the sharpest parts of the contours. Figure 2.4 d shows the final result of an identified hand and fingertips. During the process of fingertip detection, the face region, which is skin-color segmented unexpectedly, a finding that may be useful for face detection in future research.

After the hand and fingers were successfully detected, the design of a floating button interface (hover button) for hand interaction and function mappings using a camera was straightforward (Figure 2.5). Once the finger moves inside the button, the button will enlarge to indicate the capability of interaction. After the finger stops and remains motionless within the button for 1 second, a function command will be triggered.

In the prototype system, the hover button was mapped to the function of a self-timer trigger. After the timer function is triggered, a count-down number is displayed (Figure 2.6). During the count, the user is able to pose. When the timer reaches 1, a portrait photo will be taken automatically.

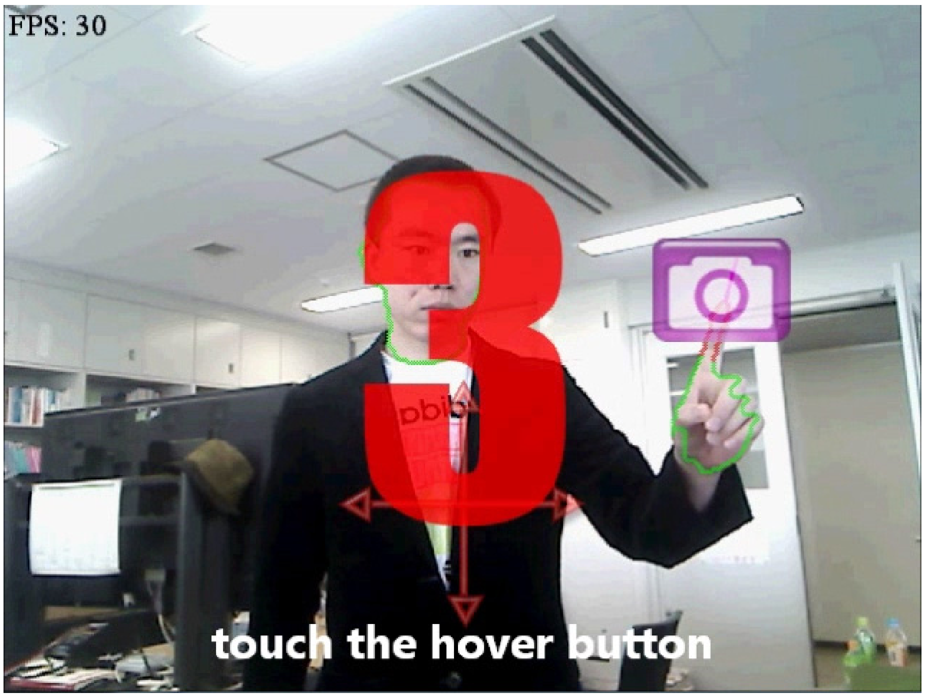

Figure 2.6: The timer will be triggered by the "air touch" of the hover button. After the timer counts down to 1, a picture will be taken.

2.3.2 Directional movement detection and user interface

In this section we introduce an original hand-gesture recognition technique that, by arranging a group of tracking points in a specific region, can detect hand motion and gestures. This enables recognition of 2D directional movements (i.e., up, down, left, and right; Figure 2.7). When the hand moves across the interface in a specific period of time, the interface will report the direction of the hand movement. The four arrows of the interface indicate the direction.

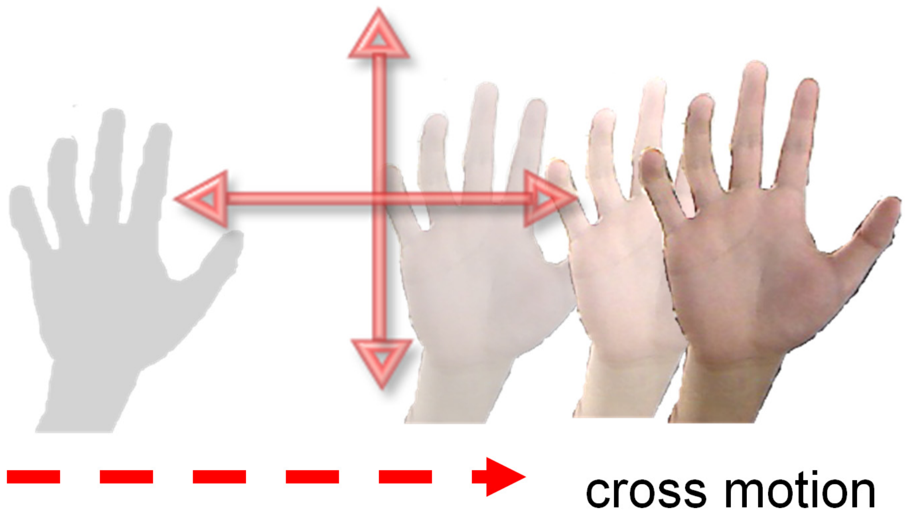

Figure 2.7: The cross-motion interface for recognition of 2D directional hand gestures.

Rather than identifying the shape of a hand, this technique recognizes only motion. The reason for this technique is that a fast-moving hand may blur the image, making fingertip detection difficult.

Gesture recognition is achieved by arranging 29 tracking points inside the interface region (Figure 2.8 a). Each of the tracking points is used for estimating motion using Lucas-Kanade optical-flow tracking [39]. Otical-flow tracking estimates the motion of two consecutive images from a video sequence and records the results of each tracking point. Displacement of the tracking points between two consecutive images indicates the amount of estimated motion.

Figure 2.8 b portrays a hand moving from left to right inside the region of tracking points. Each tracking point indicates a displacement from the previous to the current frame. Due to the noise generated by optical-flow tracking, we excluded small motions when considering displacement length and calculated only the mean value of the directions of the group tracking points. Figure 2.8 c shows that the motion in the current frame is heading to the right.

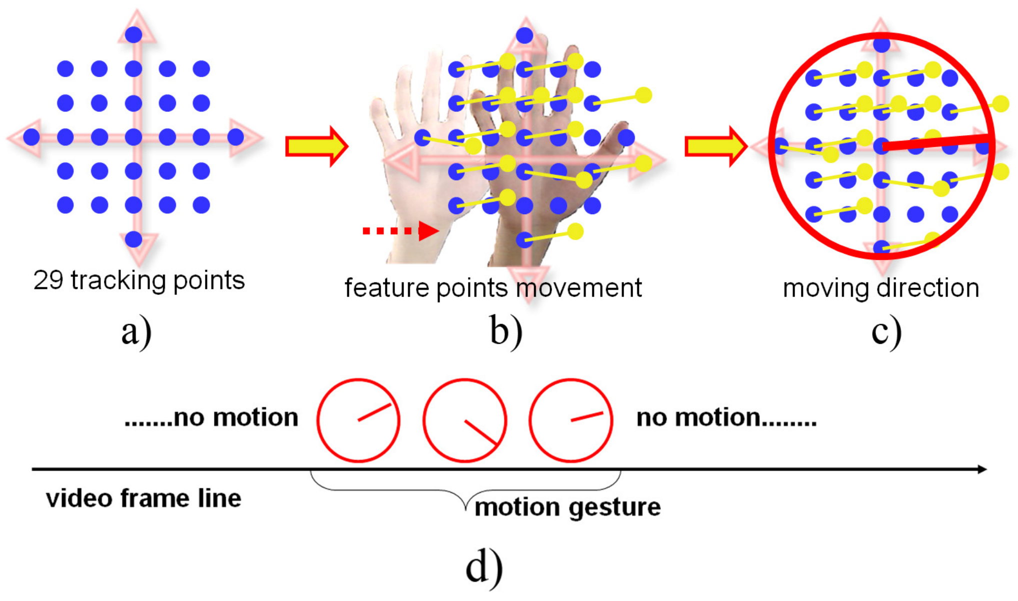

Figure 2.8: Recognition of cross-motion gestures.

When a hand gesture crosses the interface, it may generate several motion directions in the time lines of the video. Figure 2.8 d shows a typical example of the motions generated by a gesture to the right. We can see the pattern of motions in the video time line: the initial motion, the detection of several motions, and then no motion. The detected motions may contain less accurate directional estimates, but the main direction can be identified; the example shows movement to the right.

Time is an important restriction on gesture recognition. To perform a single cross-motion gesture, a moving hand usually crosses the tracking region in less than one second. Thus, we set the time for the first motion detected to exclude motions generated at one second. This helps to exclude slow movements generated by casual user motions and background noise.

At the same, a single gesture may generate more than two motions in the video time line; if a hand moves quickly across the tracking points, it may generate only one or two motions. In this case, it is considered noise and will not be recognized as a gesture. In other words, we only recognize the gestures that cross the tracking points within 1 second and generate more than two motions in the video time line. This restriction excludes motion noise in the background and improves recognition reliability. The four directional movements generate a similar pattern of data, and thus the recognition method is same.

Figure 2.9 shows a real scenario of a cross-motion gesture. Once the hand moves inside the tracking-point region, each tracking point will indicate the detected motion. A line, with a circle around it, is drawn from the center of the tracking points, indicating the motion direction. The example shows a right cross-motion gesture.

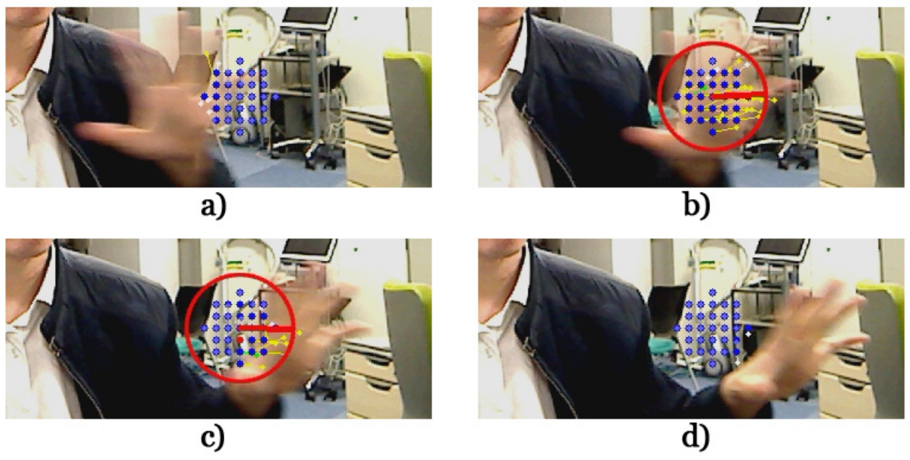

Figure 2.9: The procedure for motion-gesture recognition and key frames in the time line. a) Frame t, no motion detected. b) Frame t + 1, motion orientation detected, with the clock and arrow indicating the motion is rightward. c) Frame t + 2, motion is rightward. d) Frame t + 3, no motion detected.

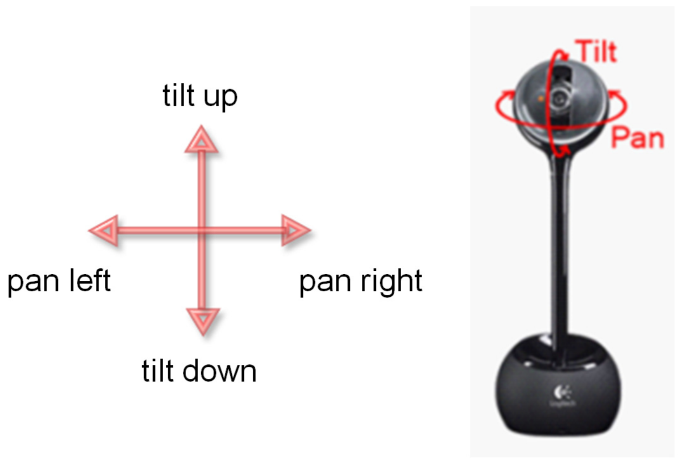

Figure 2.10: The function mappings of pan and tilt control.

Finally, in our self-portrait prototype system, the four directional movement gestures were mapped to the camera's pan and tilt functions. A gesture adjusts the pan or tilt by 4 degrees (Figure 2.10).

2.4 Experiments

So far, we explained the design and development of the new self-portrait prototype system. In this section, we conducted an experiment to evaluate the user satisfaction of the proposed system, and test the performance of the gesture recognition. A typical setup of the prototype system can be seen in Figure 2.2.

2.4.1 User satisfaction

We showed the prototype system to 10 participants, and record their impressions and satisfactions. First of all, the idea of gesture interaction with a self-portrait camera got positive evaluations. All of the participants believed the gesture interface is a better technique for taking self-portraits than conventional approaches. We will give the review of conventional approaches for taking self-portrait in Section 2.7.1. One participant suggested voice recognition technique that for interacting with camera. Secondly, the pan & tilt function that for adjusting camera view angle got fair evaluations from participants. Participants believed it can be useful, but it depends on specific scenarios, such as some inconvenient situations. In a general case, it may not so useful. Finally, the large size display shows preview and visual feedbacks with gesture motions, which reported useful from participants. It is convenient to check the status of postures, and greatly improved the user experience of interaction. The precise of hand and fingertip tracking gave deep impression to participants.

In additional, our prototype system was introduced to various anonymous audiences. We showed the system to many university students at the event of laboratory Open House. And we demonstrated the system to many senior high school students also. The feedbacks from such audiences are also positive. The system can quickly be understood and users were able to explore the system without difficulty.

Table 2.1: The performance of hand finger detection, video resolution at 320 x 240

	Process time (millisecond)
Skin-color segmentation	0.6
Mean Filter	3.7
Finding Contour	0.5
Fingertip Detection	0.7
Total	5.5

2.4.2 Performance of gesture recognition

In this section we introduce the performance evaluaction and accuracy of the gesture recognition. The performance experiment was conducted on a desktop PC with an Intel Core 2 Quad Q8300 CPU 2.5GHz. And we invited 10 participants (23 - 29 years old) to participate in the experiment to test the accuracy of gesture recognition.

Table 2.1 shows the process time of hand and fingertip detection. The result shows that the algorithm runs at good performance, it takes 5.5 milliseconds on average to complete the detection. In another word, the recognition of hand and fingertip is able to run in real-time manner with about 180 FPS.

The recognition of hand directional movement gestures is also showing good performance. It takes 3 milliseconds on average for recognition, in the case that, image size is set at 80 x 80, and tracking 29 points for estimating motions.

In a user test, we evaluated the accuracy of gesture recognition by asking participants perform certain times of gestures. We also tested the distant aspect regard to recognition result. The test is conducted in an indoor environment with daylight condition. Two recognitions, hand finger detection and hand directional movement gesture, were evaluated.

For testing the hand fingertip detection, we asked the participants to stretch out the arm and open palm to camera, meanwhile, examiner checked whether the hand can be detected. We asked participants start from 0.5 meter and then move further from camera until the hand and fingertip detection fail. From the user test, we found the algorithm can works well as far until to 2 meters. Therefore, we concluded the algorithm is able to detect hand and fingertip at a distance from camera to user at 0.5 meter to 2 meters.

For testing hand directional movement gesture, we asked users to perform the gestures in four directions for 5 times respectively. And we saw whether they were correctly recognized.

Table 2.2: Accuracy of directional movement gesture (0.5 meter)

	LEFT	RIGHT	UP	DOWN	NONE
LEFT	90%				10%
RIGHT		84%		3%	13%
UP	7%		90%		3%
DOWN		4%		90%	6%

Table 2.3: Accuracy of directional movement gesture (1.2 meter)

	LEFT	RIGHT	UP	DOWN	NONE
LEFT	80%			2%	18%
RIGHT		82%			18%
UP			86%	3%	11%
DOWN			2%	83%	15%

Table 2.4: Accuracy of directional movement gesture (2.0 meter)

	LEFT	RIGHT	UP	DOWN	NONE
LEFT	52%				48%
RIGHT		64%			46%
UP			85%	1%	14%
DOWN			3%	88%	9%

The three different distances, 0.5, 1.2 and 2.0 meters, that user stood from camera were examined. The results are shown in Tables 2.2, 2.3 and 2.4. Left column of the tables represent user performed gestures; corresponding row is result of recognized gestures.

From the result represented in the tables, we can found that the distance factor greatly impact the recognition. The recognition works at good accuracy from 0.5 meter to 1.2 meters, which achieved above 80% recognition rate.

2.5 Summary and discussions

We have developed a prototype of self-portrait camera. We proposed the gesture interface that for interacting with camera to take self-portrait shots, and used a large size display to show the camera preview and visual feedbacks with gesture motions. Two types of gesture interfaces, the hover button and the cross motion interface, were implemented. In the experiment we found the system got high evaluations of user satisfaction, the gesture interface was useful and effective, and the proposed gesture recognition technique had achieved a good performance.

However, there are several limitations exist. First, our prototype system requires a large size display, which makes the system can only work indoors. Second, the lighting sensitivity is a main weakness for skin-color based hand detection. For example, the hand fingertip detection will fail on the people have dark tone skin-color. Third, the gesture interface can only work when user is standing within 2 meters from the camera.

The future work should be to improve the accuracy of gesture recognition, and explore more reliable gesture recognition techniques. The potential technique may the motion-based gesture recognition, which robust to lighting and color conditions [39]. In addition, to implement the gesture interface on a real digital camera, such as a DSLR camera. Finally, the gesture interfaces that can work outdoors with small or no screen cameras can also be the next consideration.

2.6 Usage scenarios and potential applications

Our proposed prototype system may possible to be applied into many usage scenarios. For example, in an interactive Photo booth, users use gestures to interact with camera to take self-portrait shots. Different from traditional applications, such as Purikura [5], our prototype can provide gesture-based remote control, which makes user do not need to physically touch a screen to set parameters. In addition, our prototype system uses a webcam and a display, which is easy to be deployed by one at home, user can use the system to explore many possible postures, find his/her best styles for shooting. Or in public place, develop a gesture control self-portraits taking application combining with interactive advertisement [40].

Another usage example is that, we can apply the gesture interface into video chat system. When user chatting with friends in front of the camera, he can leave away from camera, and he does not need to use mouse or keyboard to control the pan & tilt of the camera, but uses gestures to make interactions with the camera.

The proposed gesture recognition technology can provide us to develop many vision-based applications. For example, the augmented reality (AR), which views of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or GPS data. One of challenges in AR is the interaction. For example, in the study of head-mounted display (HMD), many researches proposed the techniques mix the camera captured video with computer-generated virtual information [41], [42], [43]. But when developing the user interfaces, they still used conventional techniques, such as marker [21], sensor glove [44], which make it non-intuitive and un-convenience to use. Instead, in our proposed techniques, we do not require user to wear any markers, sensors, or additional hand held devices, which provides an intuitive interaction approach for interacting design for AR. Particularly, recent, the smart phone with built-in camera, which makes the mobile AR become popular [45], [46], the vision-based gesture interface provides a better choice to develop user interfaces for AR.

Our proposed hand finger detection technique can detect not only fingertips, but also the finger directions, see Figure 2.11. Thus, the possible user interface design can use the hand finger direction as interaction considerations. The finger direction can do distance pointing as described in [47], which is able to design a pop-up menu interface, and use finger direction to activate menu items, see Figure 2.11 a. In addition, the two hand gesture interaction can also be developed, such as Figure 2.11 b shows, to use the hand fingers to make zoom function on an image.

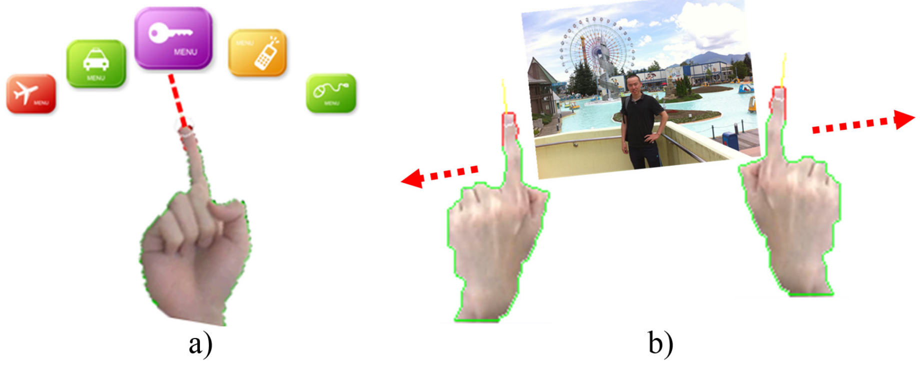

Figure 2.11: The potential user interface design by using hand finger interaction. a) use finger direction as interact approach, b) two hands interface that use to fingers to manipulate of an image.

Finally, the proposed motion-based gesture recognition is novel and has many possibilities to recognize many more gestures. In Chapter 3, we will use this technique to recognize three different types of motion gesture, and design a rich gesture interface that for controlling a professional DSLR camera.

2.7 Related work

This section describes some interacting approaches that for taking self-portraits, and also reviewed the related gesture recognition techniques that regard to our proposed method.

2.7.1 The conventional interaction approaches

We are interested in self-portrait photography. Taking self-portraits has several benefits [6]: 1) With self-portrait, people have the luxury of being able to work at their own pace, in a safe environment. One is able to explore self-portrait only for her/him-self, which gives people the freedom to experiment with lighting, posing, post-processing and so much more, without the pressures of facing to others. 2) The creativity and originality play an important role in taking self-portraits, which stimulates people to open mind and explore unique form of self-portrait and new creative possibilities. 3) The self-portraits can capture our memories. Soon after, we can once again see our past happy days. 4) Taking self-portraits is one of the best ways to advance our photography skills.

However, the conventional approaches do not satisfy the self-portrait scenario [8], [10].

Asking for help from others. One can ask someone else to take a photograph, but this may lead to unsatisfactory and/or disappointing results. Asking others for help also does not take advantage of skills that one may have developed. That is, a person may pay a great deal of money to purchase a quality camera and may study photography for some time to learn the requisite skills, only to hand their camera to someone with poor skills to obtain a portrait. The quality will depend in part on the skills of the person taking the picture. One may also feel shy or embarrassed to ask another to take the photograph, and there is typically little chance to practice postures or take multiple pictures to see which one is best. Finally, others may not always be around or willing to take a picture. For these reasons, we believe that self-portraits are better when taken by one-self.

The long arm. Most people have taken this kind of self-portrait, which involves holding the camera as far away as possible to take the photograph. This is a very popular technique with tourists. In fact, several camera manufacturers now produce cameras with two LCD screens, one of which is in front so that users using this technique can see themselves when taking self-portraits. Although this method can be fun and records a moment even when there is no one else around to take the picture, such pictures are not ideal as they are generally of poor quality; they may distort the face due to the short-range shots (see Figure 2.11), and it is difficult to keep the hand steady [10].

Figure 2.12: The long arm self-portrait. Can only shoot pictures in short-range.

In contrast, a more promising self-portrait camera should be put on any flat and stable surface. The user can stand a distance from the camera, relax, and perform intuitive gestures to interact with the camera.

The self-timer. Almost all cameras now have a self-timer option. Of course, the user will not be holding the camera when the picture is taken, so a tripod or a steady surface such as a shelf or table is required. In addition, preparing for the shot typically entails positioning the camera while behind it, perhaps using placeholders such as a potted plant or other object as stand-ins for the user, pressing the button, quickly getting into position, and then waiting for the picture to be taken, see Figure 2.13. This method is time consuming and can be frustrating, i.e., ensuring the correct focus and good composition, as it may require many attempts to obtain a satisfactory portrait [8], [9]. In addition, the user may find it difficult to find the appropriate zoom and then take a picture while (s)he keeps touching the camera to make zooming adjustments.

A better choice would be for the user to be able to interact with the camera remotely while in front of the lens, as in our prototype.

Remote control. Using a remote shutter release controller, such as a smartphone, it is possible to pose and click without having to run back and forth to the camera (see Figure

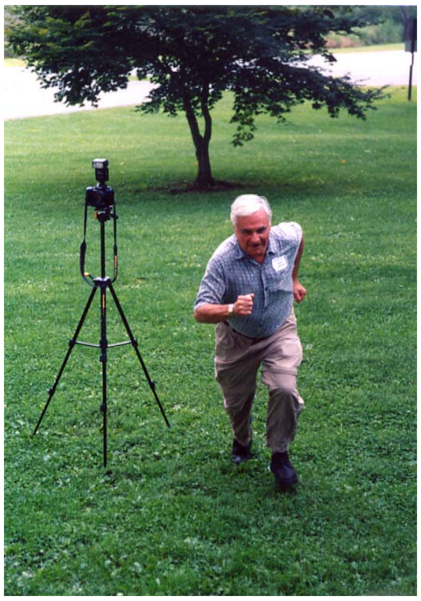

Figure 2.13: Use a self-timer interface to take self-portraits.

2.14). The camera will also be able to focus correctly as the user will already be in front of the lens [10]. However, if the hands are going to be in the shot, it will be necessary to conceal the handheld remote control [8], [10]. Indeed, the additional device occupying the hand often results in unnatural postures in final photographs.

Our proposed gesture interface can provide remote control without requiring any devices to be held. The user is able to perform intuitive head gestures to trigger camera functions.

Vision-based gesture interface. This technique involves the application of complex image-processing algorithms to the camera live view image sequence and the detection of specific features, actions, and human gestures as control commands to interact with the camera. Successful applications, such as Sony Camera Robot [48], use face-detection techniques to locate people and take pictures automatically. The Casio Motion Shutter [49] also enables users to take pictures using hand motions. However, the lack of design of interfaces that for taking self-portraits frustrate people to explore many of possibilities. For example, the small size of front facing screen makes user difficult to capture the detail of postures when they stand in front and facing to camera. In contrast, by equip with a larger size front facing screen or the ability to output the preview to a TV or external display that, will provide a clearer preview and help the user prepare the self-portrait. Although this kind of camera can only be used indoors, in fact, many good portraits are captured

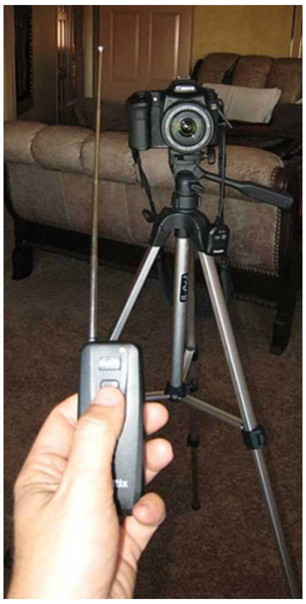

Figure 2.14: Use a remote control to take self-portraits.

indoors: users can manually configure the lighting conditions and fix colored backgrounds; because they do not feel self-conscious when posing, they can take as many photographs as they need.

Our proposed system also uses a vision-based gesture interface. However, our system is novel in that it uses intuitive hand gestures to interact with the camera. We have implemented two important control functions: shutter trigger and pan & tilt operating, by using gestures.

2.7.2 The gesture recognition techniques

Many innovative techniques have been proposed in the previous studies to deal with the difficulties in computer vision to control the devices from a distance. Vision based hand gesture recognition is believed to be an effective technique [19], [50]. Therefore, many systems have been proposed.

SixthSense [21] is one of the best examples presented the concept of gesture-based interactions. It used a webcam and attached it on the user's head to capture video and recognize gestures. A small size projector is worn on the user's chest, it enabled to project the graphic contents and user interfaces on any places. It diverted the PC from the conventional desktop environment to concept of ubiquitous, which greatly affected both the academic and general public. However, their gesture recognition technique and gesture interfaces required user to wear several colored marker on the fingertips. This makes it intrusive and un-convenience to use.

In our proposed gesture recognition, it is non-intrusive. It does not require user to wear any markers. We used a skin-color based hand segment technique and our algorithm can recognize fingers, which is used for interaction.

Chen [23] presented an optical flow with MoSIFT appearance features, to recognize gestures for controlling TV operations. The MoSIFT was computationally expensive, the authors' implementation was based on parallel processing with multi-core processors to improve the recognize performance. However, the results showed that it still required quiet long processing time. It took about 2.5 seconds between an activity action and a result response. Lenman [51] showed a study of using gestures to interact with pie and marking menus and it used as a remote controller to control electronic appliances in a home environment, such as TV sets and DVD players. The system recognized the hand poses based on a combination of multi-scale color feature detection, view-based hierarchical hand models and particle filtering. The hand poses can be detected, and then their movements can be tracked. The problems of this approach are that the performance is slow and it cannot get good detection accuracy on ant complex dynamic background. Moreover the user needs to keep the hand pose still and face the camera for several seconds to activate an operation.

In our implemented gesture recognition algorithm, it runs at 5.5 milliseconds of detecting hand fingertips and 3 milliseconds of recognizing hand movement gestures.

In [31], two downward-pointing cameras are attached above a planar surface, and a stereo hand tracking system provides the 3D positions of a user's fingertips on and above the plane. It proposed the hand and fingertip detection method. We used a similar technique to detect fingertips, but our algorithm can segment hand from any dynamic background, and only require one camera.

In late 2010, Microsoft released 3D depth sensing camera, the Kinect. This novel device provides easier segmentation of human body. However, it combines several components, such as an infrared projector and an infrared sensor, and it can only work well in an indoor environment. But in our proposal, we want to pursuit a pure vision based method that is easily embedded into current version of digital cameras.

So far, although there are many gesture recognition techniques, these are not appropriate for developing a user interface for interact with a self-portrait camera. The originality of our proposed gesture interface is that, it can recognize gestures in real-time on any dynamic background within 2 meters distance from camera, which provides a good approach for developing a gesture interface for self-portrait cameras.

Chapter 3

Motion-based hand gesture interface

This chapter introduces an enhanced version, compared to last chapter work, of hand gesture interface for interacting with a self-portrait camera. We used a professional DSLR camera for taking pictures, and projector for showing the preview. We will introduce the new motion-based gesture recognition technique. And describe the development of a rich interface that for controlling various functions of camera. A prototype system, see Figure 3.1, was implemented and used for evaluate the usability of the proposed gesture-based interaction. Inspired by Nikon projector camera [12], we used a projector to show the camera live view. User can see the instant preview and perform gestures to make interaction with the camera to take self-portrait photographs. The prototype system was deployed in an indoor environment, since many good portraits are captured indoors: users can manually configure the lighting conditions and fix colored backgrounds; and because they do not feel self-conscious when posing, they can take as many photographs as they need [9], [10]; additionally a larger size display can be used, which provides a clearer preview and help the user prepare the self-portrait.

The new proposal in this study is that, three types of motion gesture (i.e., *waving*, *eight-direction selection*, and *circling*) were introduced for interacting with the self-portrait camera. By combining the three gestures we can develop rich gesture interface and controlling both digital and analog parameters. The new proposed gesture interface can provide control of various essential functions of the camera: the aperture, shutter speed, ISO, white balance, shutter trigger, etc. This gives a complete solution for gesture-based control of camera interaction.

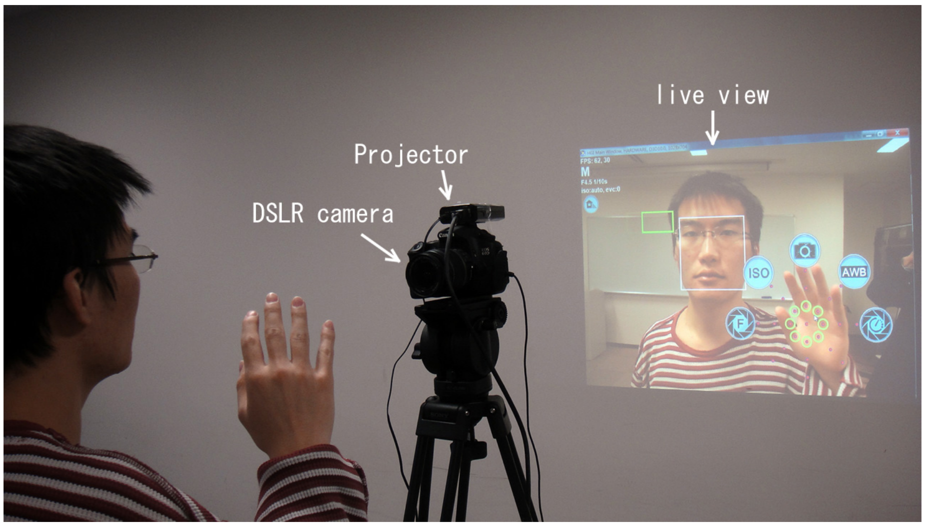

Figure 3.1: The self-portrait system prototype includes a DSLR camera mounted on a tripod and a projector.

We conducted two experiments to determine the feasibility and performance of the proposed technique. In the first experiment we evaluated the usability factors of the gesture interface that for taking self-portraits and compared it with a traditional self-timer technique. In the second experiment we assessed the accuracy and performance of the proposed gesture recognition technique. The experiment results showed that the gesture interface has high user satisfaction and it is superior to the traditional self-timer technique. In addition, the proposed recognition technique achieved about 80% accuracy of detecting motion gestures.

3.1 Motivations of motion-based gestures

The main challenges in this work are: 1) implement a gesture recognition method that robust to lighting and color conditions, 2) develop a gesture interface that provides rich interaction with a professional DSLR camera.

The robustness of gesture recognition that against lighting and color conditions is critical when developing gesture interfaces for controlling a DSLR camera. Since, when camera parameters changed, such as aperture, shutter speed, ISO, white balance, etc., which will great affect the brightness in pictures. The color sensitive method, such as skin-color segmentation, may fail to segment the shape of hand and recognize gestures. See Figure 3.2. Therefore, we must find the lighting robust method to recognize gestures.

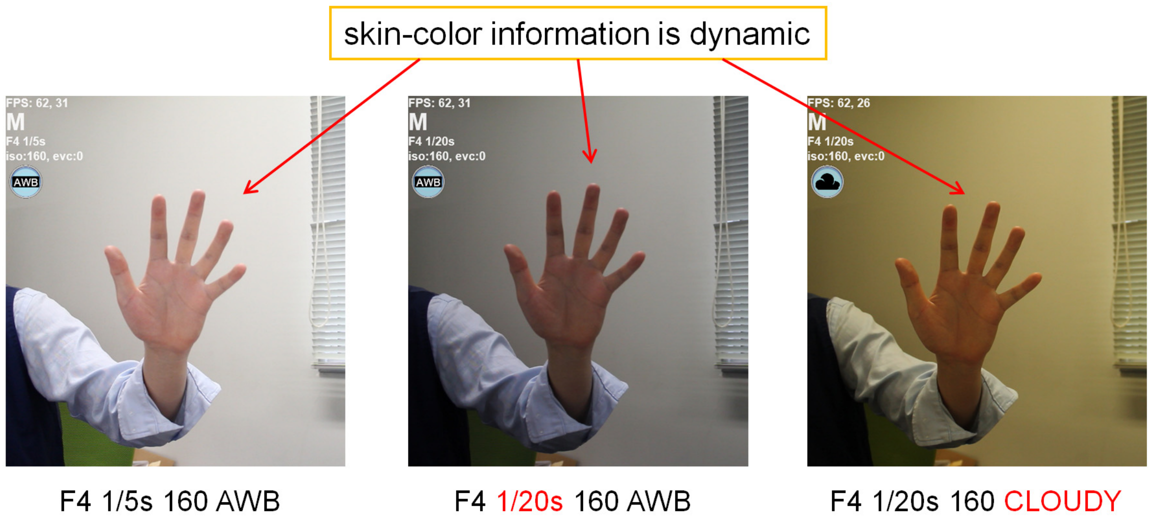

Figure 3.2: The skin-color information is dynamic.

Figure 3.3: The DSLR camera parameter setting interface.

The rich interaction means that, we try to define many functions to map various function controls of a professional DSLR camera. In general, there are many parameters, aperture, shutter speed, ISO, white balance, etc., user can set and find the best setting to take shot. See Figure 3.3. However, when taking self-portrait user can only judge those settings on stand-in, not him/herself. But by using gesture interface, user have chance to modify this settings and check the instant result when he/she is stand in the right position in front of camera. In this study we challenge the possibility of developing the gesture-based interface can offer same capability to control such many camera functions.

In the following section, we reviewed the gesture recognition techniques, and introduce the solutions that fulfill the above requirements. We will also discuss the advantages of our proposed motion-based gesture recognition method.

3.1.1 The gesture recognition techniques

Commonly used vision-based hand gesture recognition methods can be classified into two groups: model-based methods and motion-based methods [19], [51].

In the model-based approach, the standard procedure of gesture recognition [19], [50] combines several tasks: initialization, tracking, pose estimation, and recognition. Initialization captures prior knowledge of a specific configuration, such as color pattern, to distinguish the shape of the hand, which is then used to constrain tracking and pose estimation. In the final recognition step, actions are distinguished as behaviors performed by the user in one or more frames. Many successful applications have been developed: Wilson [52] developed a background subtraction method to detect a pinching gesture above a tabletop; SixthSense [21] uses a color marker attached to the hand for tracking the fingers, and a pointing technique to recognize actions. Other studies have used color information to distinguish hand models and have applied comparison algorithms to determine the three-dimensional position of the hand stored in the dataset [41], [53], [54]. Other studies have reported static position detection using template matching to recognize hands [55].

The disadvantage of the model-based approach is that it often requires a predefined configuration in initialization, such as a pre-calibrated environment [52], specific color lighting conditions [53], [54], the need to attach markers or wear gloves [21], [54], or restricted hand position during interaction [55]. The computational complexity is also a significant issue in this kind of approach. In our self-portrait camera scenario, color lighting conditions are often dynamic, and wearing markers or gloves can results in unnatural portraits.

Therefore, we used a motion-based recognition approach. In this kind of technique, the recognition procedure combines a motion measurement algorithm to determine image differences between two consecutive frames, and a pattern recognition method to distinguish motion actions. It ignores initialization and position estimation procedures, and therefore allows more freedom in hand motion than model-based approaches. Moreover, it is particularly robust to different color lighting conditions.

Many research prototypes have used this approach, applying a sparse Lucas-Kanade (LK) or dense Horn-Schunck (HS) optical flow measurement [37], [56] to detect motion, and a Support Vector Machine (SVM) or AdaBoost to classify human actions [23], [57], [58], [59]. Due to the considerable computational complexity, some of these studies applied GPU speed-up algorithms to achieve real-time application. However, performance was still poor. Also, use of the machine learning approach to classify gestures restricts the recognition results in digital outputs, reducing the recognition rate and limiting the set of gestures that can be classified.

Instead of measuring motions from a large set of tracking points, we developed a method that involves arranging various layouts of a small set of tracking points on a specific region to detect hand motion. Thus, our approach uses a manually restrained timing algorithm to recognize gestures and track each step of motions in the recognition procedure. The method requires less computational complexity and no pre-training, and enables both digital and analog gesture actions as outputs.

3.2 Motion gesture recognition

3.2.1 The proposed gestures

The innovative point of this study is we focused on three different types of motion gestures, and combined them to develop rich gesture interfaces.

Waving gesture - This gesture involves raising the hand and moving it from side to side. It is commonly used to attract attention at a distance, so it is intuitively useful for mapping to a wake-up interaction function for computer systems [60]. It is also a good clue for the system to recover the hand region in the image. Our interface design is based on the assumption that the user interface is around the user's hand and can be manipulated by small hand motions. Thus, recognition of waving gestures and recovery of the accurate region of the waving hand is important to recognition of the other two types of gesture.

Eight-direction selection gesture - This gesture involves raising the hand and moving it from one position in a specific direction over a given distance. This gesture is similar to that described in a previous study [25]. However, in our work, the directions can be: Left (LT), Up-Left (UL), Up (UP), Up-Right (UR), Right (RT), Down-Right (DR), Down (DW), Down-Left (DL); these eight gesture inputs make it easier to develop a menu selection user interface and offered good accuracy of function mapping [20].

Circling gesture - This gesture involves raising the hand and moving it in circles over a region, in the clockwise or counter-clockwise direction, which was studied in [25], [61]. The circling gesture can provide analog data input with direction angle (or moved circles) per frame. This advantage allows a user interface to control linear values.

The three types of gesture have specific features and are connected logically. First, waving gesture recognition not only provides a startup action but also allows recovery of the region of the hand in the frame. The latter outcome is beneficial to narrow gesture recognition to a small region around the hand, thereby enabling greater detail regarding hand motions in subsequent gesture recognition procedures. The *eight-direction selection* gesture and circling gesture are highly dependent on the results of the recognized waving hand region: detection is carried out only around the region identified by the initial hand-waving gesture.

Second, the *eight-direction selection* gesture provides only eight selection choices. These are appropriate for developing a menu selection interface, but permit only a limited number of digital inputs and cannot provide a complete interface solution to control many parameters. For example, it is not possible to adjust linear pattern values using this type of gesture recognition. Therefore, we also introduced the *circling* gesture, which provides analog output and uses clockwise and counter-clockwise motions to distinguish between positive and negative adjustment. This type of gesture recognition can be integrated into an interface for value adjustment. Thus, it is possible to select an option by *eight-direction selection* and then adjust the parameters with *circling* gestures. These operations can be repeated, enabling rich interfaces to control many parameters and functions.

Third, all three types of gesture are intuitive and easy to learn, and our experiments demonstrated that subjects understood them quickly after simple explanations.

3.2.2 The motion estimation technique

The proposed gestures are recognized by using motion estimation based on a standard Lucas-Kanade optical flow tracker [39]. An optical flow tracker is an algorithm that estimates the velocity of movement for a given set of points on a grayscale image, using various images. A typical method for estimating the movement of a point is to calculate derivatives of pixel intensity at each point, and then determine the motions within a window centered at that point in another image. The window is an integral window in which a similarity function is performed to search for an optimal candidate as the estimated moved point. During the process, each point is calculated independently of the others. The Lucas-Kanade implementation uses spatial intensity gradient information to direct the search for the position that yields the best match for tracking points. This method applies an image pyramids technique to maximize search speed, and matches both small and large movements of tracking points, enabling rapid and accurate estimation of optical flow. The Lucas-Kanade method of optical flow tracking has been widely used in various motion tracking and real-time applications. One advantage of optical flow tracking is that it calculates the derivatives of pixel intensity from nearby pixels, and does not rely only on the color information of one pixel; this makes it less sensitive to image noise and brightness [39], [56].

The novel aspect of our proposed technique is that, a set of tracking points are defined at a specific region on an image frame, and the optical flow at these points is calculated repeatedly in sequences of image frames to analyze the gesture motions. This method can greatly reduce the compute complexity and do not require any pre-training. The following sub-sections describe three patterns of layout for recognizing three types of gesture.

3.2.3 Waving gesture recognition

To recognize waving gesture, we can apply a matrix layout of tracking points on full-size images (see Figure 3.4) to detect hand motion. The current system uses $15 \times 7 = 105$ points arranged at a resolution of 360×240 pixels. Each point is used to detect motions separately. The waving motion pattern is detected by the motion displacement of a point, determined by optical flow, from a direction angle θ to its semi-opposing direction θ'; length is larger than 1, because a lower value indicates a very slow speed of motion and is therefore excluded. The span angle between θ and θ' should be larger than 120°; this indicates a successful transform of wave motion: the point's wave *transforms* value plus 1. If no transform is detected over a specified period (in this case, 500 ms), the *transforms* value is reset to 0. *transforms* values should be set at 4 or greater to filter out unnecessary motion, and to identify candidate waving gesture points.

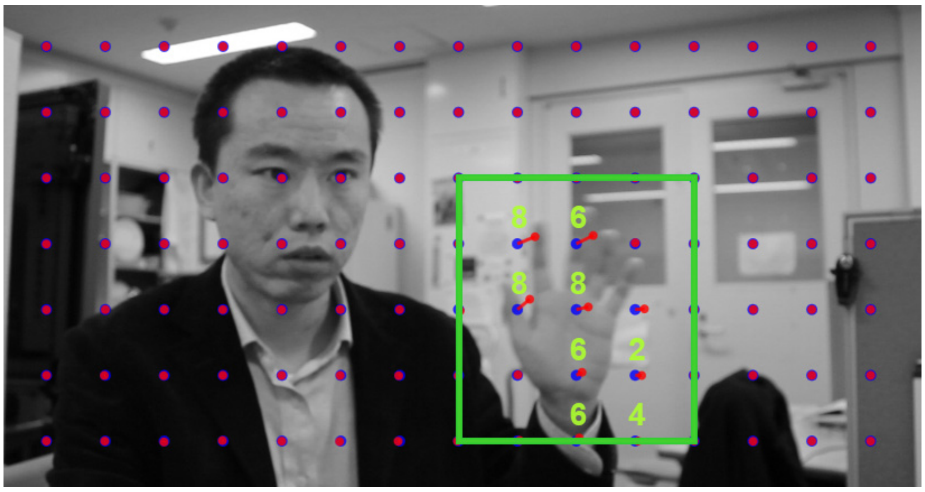

Figure 3.4: Matrix of tracking points for detecting a waving gesture.

In general, when a waving gesture occurs, the waving hand will occupy a region on the image and several neighboring points will detect motion simultaneously. To group the neighboring points and estimate the hand region, we can apply an algorithm to merge the neighboring points into rectangles, and then assign the merged rectangles as the final result of waving gesture recognition (see Figure 3.4). This process involves several steps. First, a rectangle is placed around each candidate point in the matrix. The size of the rectangle is the same as the horizontal and vertical distance of two neighboring points. In the second step, the intersected rectangles are merged together to union rectangles. Finally, among the merged rectangles, the one containing the maximum *transforms* value is assigned as the dominant recognized waving gesture result.

Our purpose is not only to detect a waving gesture but also to recover the hand region in the image, which plays a key role in the two subsequent gesture recognition procedures.

3.2.4 Eight-direction selection gesture recognition

The eight-direction selection gesture can be identified after the waving motion is identified. In the eight-direction selection gesture, the hand moves from the recognized hand region in a specific direction: Left, Up-Left, UP, Up-Right, Right, Down-Right, Down, or Down-Left (see Figure 3.5).

To recognize the gesture, 24 tracking points are separated into eight directions with a radial-shape layout (see Figure 3.5) to enable detection of hand motions. For each direction,

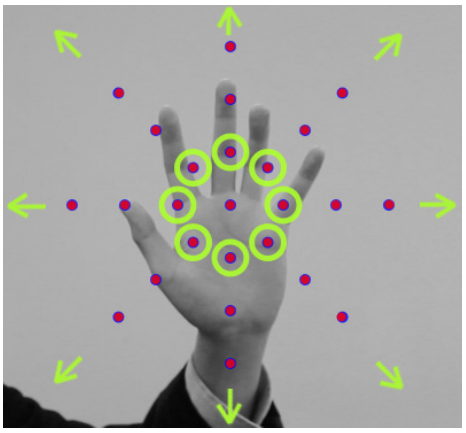

Figure 3.5: A radial-shaped layout of tracking points for detecting eight-direction selection gestures.

three points are organized as a set and arranged in a radial line outward from the center. The detection of a moving gesture in a specific direction involves several steps. Using the UP direction as an example, each of the three points detects directional motion: length as determined by optical flow measurement must be larger than 1, and angle θ must be approximately $^\circ$ ($\pm 22.5^\circ$). If the three points detect this motion within a specified period (in this case, 500 ms), the direction selection gesture is assumed to have been successfully detected. Similar processes are involved in detecting gestures in other directions.

Note that the system only recognizes one motion from the center outward at a time, and excludes any motion from the outside into the center. The recognition results are obtained as a digital output corresponding to the direction of the gesture.

3.2.5 Circling gesture recognition

In the *circling* gesture, the hand moves in circles over the region that has already been identified as the waving hand region.

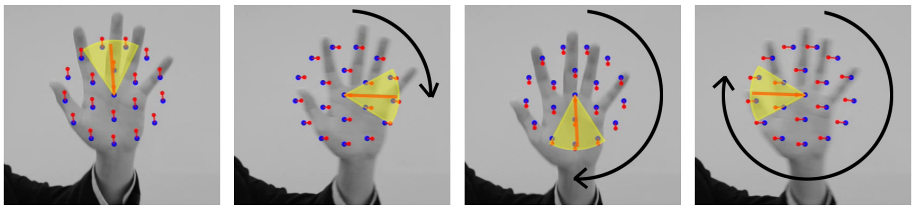

Figure 3.6: A circular layout of tracking points for detecting circling gestures.

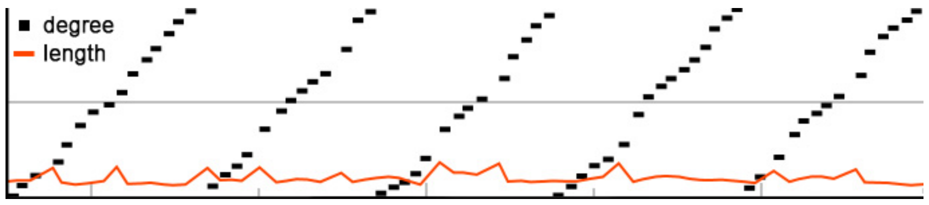

Figure 3.7: Two analog data parameters per frame time line.

To recognize the gesture, 20 tracking points are arranged in a circular pattern (Figure 3.6). The motion of the set of points is calculated according to two mean values: *direction angle* and *length*. Figure 3.7 presents the variation diagrams for these two factors for a clockwise circling motion pattern. The moved circling angle is calculated by accumulating the shifted direction angle (difference in direction angle between two consecutive frames), and by incorporating the timing factor in each frame. If the direction angle in consecutive frames has a clockwise pattern, this indicates a clockwise movement angle. In contrast, a counter-clockwise motion is detected as a counter-clockwise movement angle. If no motion is detected, the moved angle value is reset to 0.

This layout of tracking points can also recognize a waving gesture in which the hand moves from side to side over the point cloud.

3.3 Gesture user interfaces and application

We designed two user interfaces for controlling a DSLR camera, by incorporating the gestures the system recognizes as well as digital and analog information. One important consideration of interface design is to show appropriate visual feedback to users when they perform gestures, which can improve the user experience [30].

3.3.1 Mode switching interface

The *mode switching interface* is a pie-like menu that pops up on the screen around the user's hand once a waving gesture is detected. The menu can provide a maximum of eight selection choices: LT, UL, UP, UR, RT, DR, DW, and DL. The user can interact with the system using the *eight-direction selection* gestures.

We mapped five of these items (LT, UL, UP, UR, RT) to five camera functions: shutter trigger, white balance aperture, ISO, and shutter speed (see Figure 3.8, left). When the shutter trigger function (UP item) is selected, a timer appears and counts down from 5 to 1; during this time the camera autofocuses on the user's face and the user prepares his/her pose. Once the timer reaches 1, the camera takes a photograph automatically. When the white balance (UR) function is selected, another set of six function icons pops up. As shown in Figure 3.8, right, the white balance icons from left to right are: auto white balance, daylight, shade, cloudy, tungsten, and fluorescent.

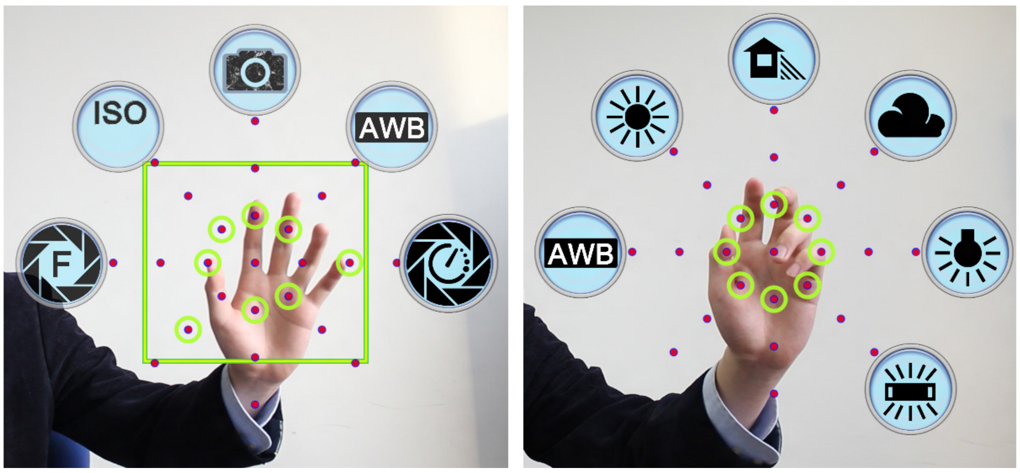

Figure 3.8: Mode switching interface. Left: Main menu of the interface. Right: The six white balance icons.

The other three icons represent aperture, ISO, and shutter speed, which are linear pattern values. We designed a value adjusting interface to modify such linear parameters.

3.3.2 Value adjusting interface

This interface was designed based on *circling* gestures. The user can perform gestures within the region of points to adjust a parameter value linearly.

However, the values of aperture, ISO, and shutter speed are not pure linear data, but are sequential. For example, the aperture has a sequence of values: 3.5, 4, 4.5, 5, 5.6, 6.3, 7.1, etc., in which each increase in value represents double the volume of light to the photoreceptor in the camera. The ISO and shutter speed parameters have similar values sequences. Thus, to modify such sequence patterns of values, the user completes a full circling motion, moving the hand 360°, to move to the next step. A complete clockwise circling motion increases the value by one step, while a complete counter-clockwise circling motion decreases the value by one step. Changing the aperture, ISO, and shutter speed involves similar mechanisms. Figure 3.9 shows the interface for three cases when adjusting these parameters.

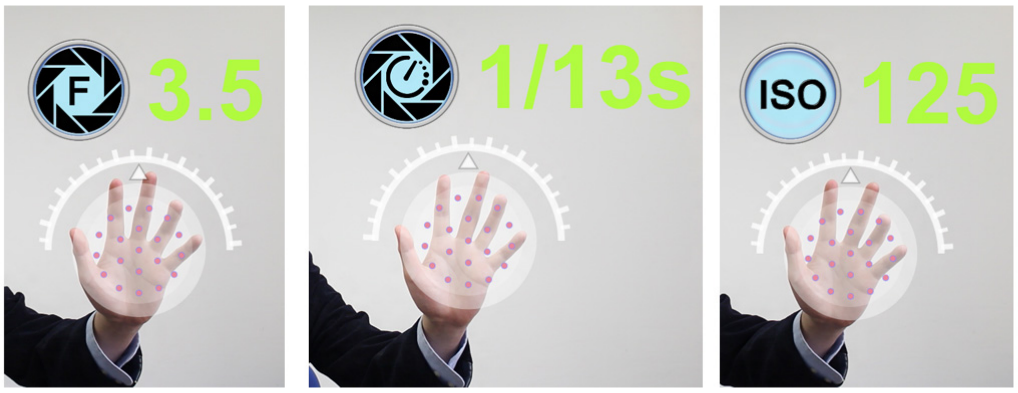

Figure 3.9: Value adjusting interface. Perform circling gesture inside the tracking point circle to adjust the value.

After setting the desired value/s, the user performs a *waving* gesture within the region of points to return to the mode switching interface. The value adjusting interface then disappears, and the mode switching interface pops up. Many of the camera settings can be configured by switching between the two interfaces.

3.4 Implementation

The current implementation uses the OpenCV library [62], which provides Lucas-Kanade optical flow tracking. We used a multi-core PC, Intel Core i3 2.4 GHz CPU, and dedicated one thread for *waving* gesture detection, and one thread for both the *eight-direction selection* gesture and *circling* gesture recognition. The control signal and data exchange between the DSLR camera and computer were provided by a Canon SDK [18]. The preview and GUI were rendered using Microsoft Direct2D [63]. The entire program was written in C++.

Table 3.1 summarizes the performance of the algorithm. The *waving* gesture recognition uses a frame image with a resolution of 360 $\times$ 240 pixels and 105 tracking points. On average, motion estimation and gesture recognition takes 8.3 ms (120 FPS). The other two processing tasks, *eight-direction selection* gesture recognition (360 $\times$ 360 pixels) and *circling* gesture recognition (160 $\times$ 160 pixels), were conducted using a single thread and average speeds were 15.7 ms (63 FPS) and 11.1 ms (90 FPS), respectively. The Canon

60D camera provides 30 FPS with a preview video stream resolution of 1056 × 704 pixels, and the gesture recognition processing performance is more than sufficient to support a real-time application.

Table 3.1: Gesture recognition performance

	Process time (millisecond)
Waving gesture	8.3
Eight-direction selection gesture	15.7
Circling gesture	3.1

The apparatus of the prototype system included a Canon 60D DSLR camera, a projector, a tripod, and a stand to hold a ThinkPad X201i notebook PC. The camera is connected to the notebook PC by a USB to enable processing, and the projector to serve as the camera viewfinder. Figure 3.1 shows an overview of the system arrangement.

3.5 Experiment 1: Evaluating the Gesture Interface

We conducted an experiment to evaluate the feasibility of using the gesture interface for taking self-portraits. In the experiment, we compared the conventional self-timer technique with the proposed gesture interface. We did not compare gesture interface with a hand hold remote control, since in our pilot study and the study in next chapter we found the additional device held in hand will distract participants' attention and make them not free to prepare postures. In fact, the hand held remote control is not popular and very few people actually bought it. The experiment was designed to measure the efficiency of the shooting procedure, user satisfaction with the resulting portraits, input difficulty (ease of use) of the interface, and user satisfaction regarding the two techniques. The results were based on observation and a questionnaire with scores ranked on a five-point Likert scale (1: strongly disagree c 5: strongly agree) after the experiment.

3.5.1 Apparatus

The apparatus of the proposed gesture-based system was described in the previous section and the arrangement is shown in Figure 3.1.

A similar arrangement was used in the self-timer scenario, but another item was used as a stand-in to allow the camera to autofocus, because the DSLR camera has no autofocus function without the user touching the shutter button. This means that the user must press the shutter button halfway to autofocus, push it down completely to trigger the self-timer, and then run to the front of the camera and pose. Because the user has no opportunity to stand in the correct position and face the camera to allow it to autofocus, we use a method popular in self-portrait photography that positioned a picture board to act as a stand-in while the user was setting the camera functions. The arrangement of the self-timer system is shown in Figure 3.10.

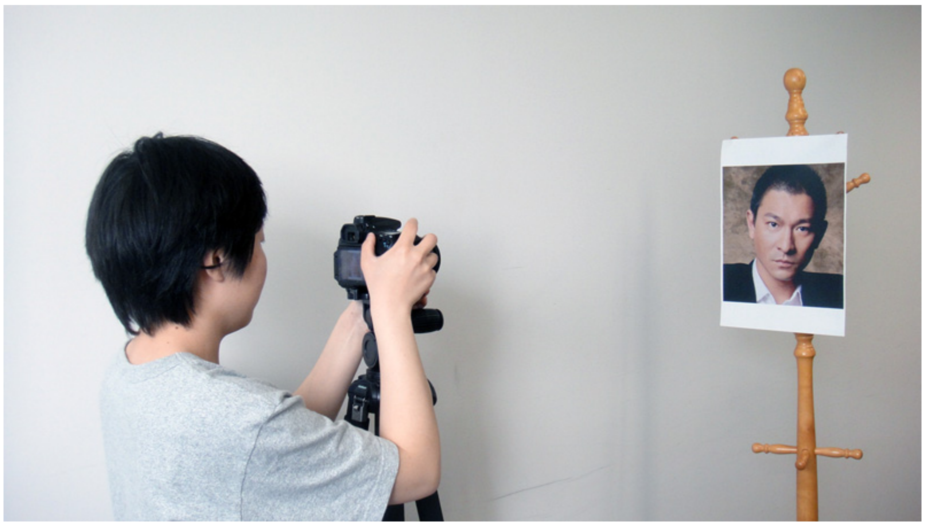

Figure 3.10: The self-timer scenario in which a picture board was used as a stand-in to allow the camera to autofocus.

3.5.2 Participants

We invited 11 subjects (5 women, 6 men), ranging in age from 23-29 years (mean: 26.1), participated in the experiment.

3.5.3 Task and procedure

Participants were given a simple introduction to the system. The author demonstrated in person how to take two self-portrait photographs using the two different techniques.

In the self-timer scenario, participants positioned the picture board in the desired position as a stand-in. Then, they went to the camera and pressed the shutter button half-way to trigger the autofocus on the stand-in. Next, the shutter button was depressed completely to trigger the 10-s self-timer countdown. The participants then quickly ran to the stand-in board, removed it, and put themselves in its place. Once the self-timer reached the end of the countdown, the camera released the shutter and took a photograph.

In the gesture interface scenario, participants stood in front of the camera with their upper bodies in view of the camera. They performed a waving gesture to wake up the system, and the mode switching interface popped up around the waving hand in the preview. Next they moved their hand in the UP direction to select shutter trigger function, activating a 5-s timer countdown. The camera autofocused on the participant's face and then took a photograph.

Participants were asked to use both techniques. This preliminary test simulated a simple task, triggering the camera shutter, which is the most commonly used procedure when taking portraits.

In the second phase of testing, we evaluated and compared the two techniques in much greater detail. Participants were asked to set many camera parameters (aperture, shutter speed, ISO, and white balance, etc.) using the traditional button-based interface and the proposed gesture interface. Participants were permitted to experiment with the two techniques and to take many self-portrait photographs until they were familiar with and understood the two interaction approaches. This test was designed to assess the ease of use of each interface, and user preference for the two techniques when controlling many camera functions.

After each test, participants were asked to complete a questionnaire.

3.5.4 Results

In the preliminary test, it took about 18 s for participants to complete a self-portrait shot using the self-timer. In contrast, the gesture interface took about 10s. Most participants preferred the gesture interface, and no significant differences appeared in user satisfaction with the resulting portraits between the two techniques. Figure 3.11 presents the results.

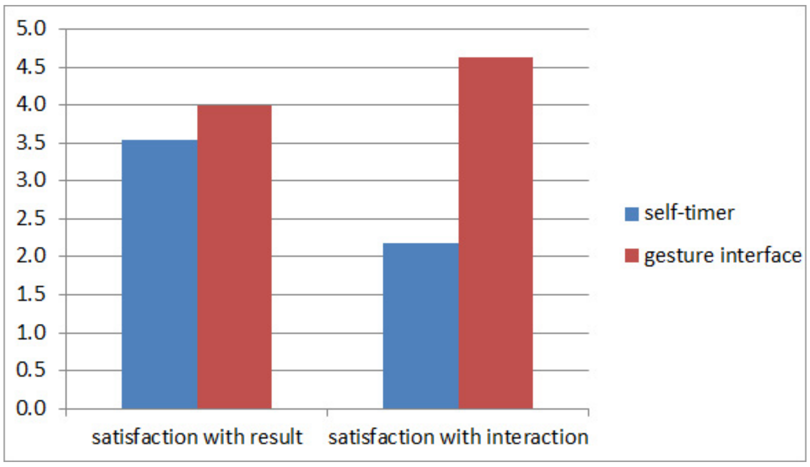

Figure 3.11: Results of the preliminary test.

In the second phase of testing, participants reported slightly less difficulty in camera parameter input and adjustment when using the gesture interface. Participants preferred the gesture interface for controlling the camera functions compared to the button-based interface. Figure 3.12 presents the results.

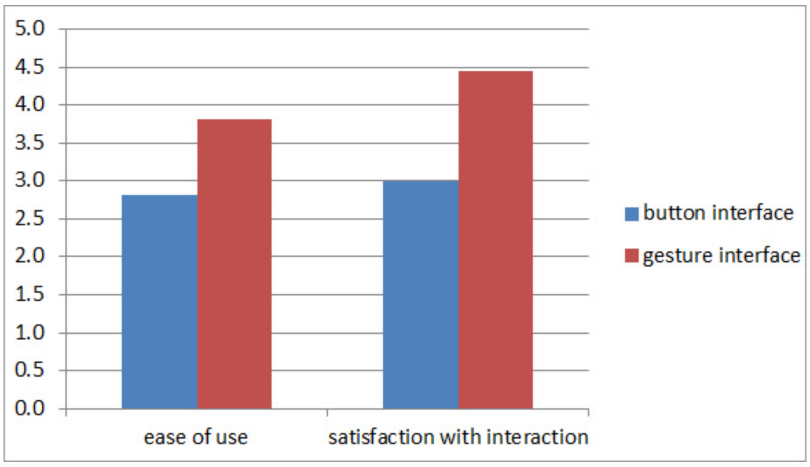

Figure 3.12: Results of the second test.

3.6 Experiment 2: Accuracy of Gesture Recognition

This experiment was designed to evaluate the results of gesture recognition, and to assess the recognition rate of the three types of gesture.

The apparatus used for the experiment was described in Section 3.4. A total of 10 participants (3 women, 7 men), ranging in age from 22-28 years (mean: 25.2) were recruited for this experiment.

3.6.1 Task and procedure

In the *waving* gesture test, participants stood a given distance from the camera with the upper body in view of the camera. They were then asked to perform the waving gesture three times. Once a waving gesture was detected, visual feedback appeared on the preview screen to notify the participant. During the test, we observed and recorded the sensitivity and speed of the gesture recognition.

In the *circling* gesture test, participants were asked to perform the circling gesture for a while to familiarize themselves with the actions. On the screen the circling angle of motion will be showed on top of hand. Then, the participants were asked to perform a full circling gesture, i.e. 360°, for 10 times, and we recorded the errors of the recognition. Because this gesture is analog, we also observed the sensitivity and user experience of the gesture recognition result during the test.

A specific interface was developed for the *eight-direction selection* gesture test (Figure 3.13) with eight icons representing each of the eight directions (LT, UL, UP, UR, RT, DR, DW, DL). Participants were asked to select an icon indicated by a red circle marker. After they selected a direction icon, regardless of whether it was correct, the interface disappeared and then reappeared for the next test trial. Each participant completed 24 test trials (three trials for each direction) presented in a random order.

3.6.2 Results

In the *waving* gesture test, the gesture was usually detected when the participant waved his/her hand four times from side to side within 2.5 meters from camera. It could detect the waving hands in the image from size 62 × 73 pixels to 930 × 511 pixels. A single waving gesture takes about 2 s to perform. The gesture was not be detected if the participant waved at a slow speed (i.e., the four side to side movements lasted longer than 2 s).

During the *circling* gesture test, we found that the recognition was very sensitive to hand motions, the visual feedback of the interface can accurate report the circling angle to user. However, participants feel difficult to set the value of their expected in the range of

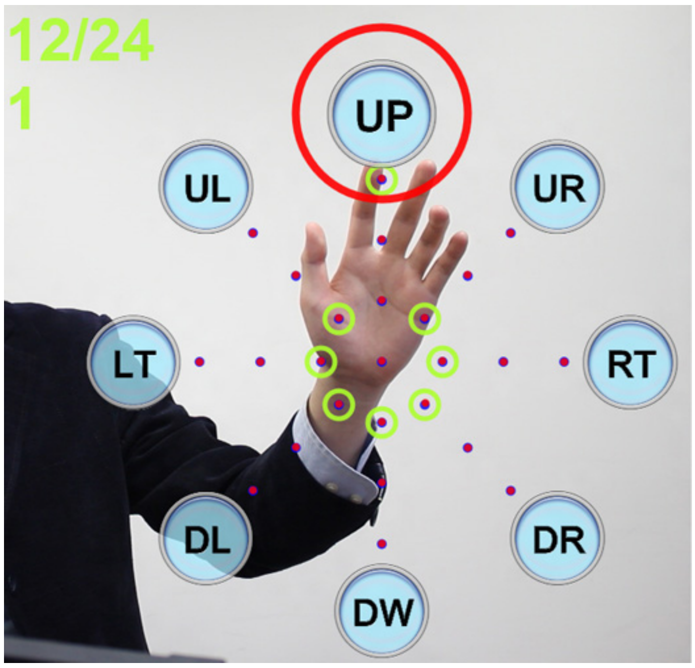

Figure 3.13: The eight-direction selection gesture experiment.

360°. The full circling gesture test showed 138° error on average. But participants felt no difficult to set values using a full circling motion to adjust one step of value change.

We collected the results of 240 test trials (30 trials for each direction) from the *eight-direction selection* gesture test. Table 3.2 lists the mean accuracy results for recognition of the eight directions; the left-most column indicates gesture inputs, and the corresponding row shows the recognition result.

As shown in Table 3.2, the mean accuracy of the eight directions was 81%, and the accuracy for each direction exceeded 70%. The upper directions, LT, UL, UP, UR, RT, had better accuracy, while DW had the best result at 97%. A few cases of failure were observed; these occurred for several reasons. First, the user's forearm sometimes conflicted with the hand motions. 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 overshot the center and moved in the opposite direction, causing an incorrect result. Because incorrect actions can occur, the system is designed to allow the user to perform a waving gesture to cancel the selection and then perform the selection gesture again.

Table 3.2: Accuracy of eight-direction selection gesture

	LT	UL	UP	UR	RT	DR	DW	DL
LT	0.80	0.03	0.03		0.03			0.10
UL		0.83	0.13				0.03	
UP		0.07	0.83				0.10	
UR	0.10		0.03	0.80		0.07		
RT	0.03		0.10	0.03	0.80	0.03		
DR		0.10	0.10	0.03	0.03	0.70	0.03	
DW			0.03				0.97	
DL			0.07	0.07		0.03	0.07	0.77

3.7 Summary and discussions

In this study we proposed three types of motion gestures, i.e. *waving*, *eight-direction selection*, and *circling*, which offered us to developing a rich gesture interface to control many of functions of a professional DSLR camera.

The proposed *waving* gesture recognition is accurate at estimating the waving hand position. During the experiments, although it may be less accurate for estimating the hand size, the participants did not report any significant deviation in the recognized hand position. Future research will improve the accuracy by adding more tracking points with a dense matrix layout, but this must be balanced with the associated decrease in performance.

In the *eight-direction selection* gesture recognition and mode switching interface tests, participants said they preferred using gestures in the upper semicircle directions (LT, UL, UP, UR, RT), which were more convenient to reach than the lower directions (DL, DW, and DR). Therefore, the interface should be designed with the most frequently used functions arranged on the upper semicircle.

In the *circling* gesture and interface tests, which simulated the analog input, users found it difficult to stop the motion at a particular value. Thus, we designed the interface to use a full circling gesture (360°) to increase or decrease the value, to make it easy to stop at a particular value.

Participants were positive about the idea of a vision-based gesture interface for controlling a self-portrait camera. Most of the participants agreed it can become the next generation of interactive technology of the camera. However, they were frustrated by the imperfect gesture recognition rate.

In this study we found: 1) the motion gestures are robust to light and color conditions. The changing of the parameters of a DSLR camera have greatly effect on the lighting condition in image, but the proposed gesture recognition can still work well. 2) the variety of motion gestures provided us to develop rich gesture interfaces. 3) the gesture interface is more useful than conventional button-based interface, such as self-timer.

In the future, we will test different motion estimation algorithm and implement better recognition algorithm. Some suggestions referred that to use sound feedback to the user's selection, which may improve the user experience. With regard to the experiments, we are planning to add an additional setting with a remote control to get a detail comparative result of the gesture interface. The present study focused on interactions and ignored the results of the photographs. A recent paper [64] discussed a technique for selecting still candid portraits from video sequences. In future studies, we plan to develop a system to support video recording and continuous shooting, which would provide possibilities for a wider range of satisfying portraits.

3.8 Usage scenarios and potential applications

Comparing to Chapter 2, the study in this chapter proposed an enhanced version of hand gesture interface. But we focused on developing gesture interfaces for controlling a professional DSLR camera. This makes the possibility to deploy the system for studio photography. The Figure 3.14 shows a scenario, that photographer exploring the camera to take professional pictures. Usually, people have to consider many of settings, the lighting condition, camera aperture, white balance, etc., however, he have to run back and forth to check these settings. But by using a display (or a projector) to show the preview, and gesture interface to interact with camera, this can greatly improve the productiveness of self-portrait taking procedure.

In addition, we also proposed the new motion-based gestures, *waving*, *eight-direction selection* and *circling*, we proved these gestures can provide rich gesture interfaces. But it is not difficult to apply these gestures to develop interfaces and deploy to many vision-based control applications, such as mobile camera interaction, AR, HMD, etc. For example, gesture capturing that uses front camera in a mobile phone, which can provide non-contact

Figure 3.14: The studio photography scenario.

interactions. Some proposal, such as [65], [66], [67] used front camera captures finger motions and recognize gestures as digital inputs, but by using our proposed *circling* gesture, which can provide analog inputs. Many researches, such as [45], [46], [68], [69], introduced development of gesture interfaces for interact with virtual objects in AR, but it may provide rich interface if we apply our gestures, which provides both menu selection and linear value adjusting functions.

3.9 Related work

In study [23], authors presented the MoSIFT feature and tracking technique, it recognized gestures as commands to control a TV. However, such technique got a low performance of processing. Usually, it took 2.5 seconds between user gesture and response. In fact, it is a general problem in many motion-based recognition techniques. In study [59], authors proposed the motion action recognition by learning from mid-level features, and in [70], it used motion history gradients, but these studies also meet the challenge of performance. Bayazit [57] introduced using of GPU based processing to improve the performance; however, the performance is still low (about 20FPS). But in our proposed method, the performance is much improved, 8 ms, 15 ms, 3ms, for recognizing *waving*, *eight-direction selection*, and *circling*, respectively. Comparing to conventional methods we chose two main techniques to reduce the compute complexity: reduce the size of image for processing and avoid to use machine learning approach to recognize the gestures. In our proposed technique, we first estimate the hand region in the image by recognizing *waving* hands, and then restrict the processing region around the detected waving hand. In addition, we arranged a small set of tracking points for estimation the motions, which greatly reduced the computation complexity also. In short, our approach had achieved a better performance to compare with the conventional techniques.

In study [58], authors proposed the hand location estimating by using face detection. But, in our technique, it can detect hand gestures without face cue but based on the motion information only.

The study [71] proposed the motion estimation interfaces for VIDEOPLACE-like interaction. They developed two types of interfaces: button and movable button. The button function will be triggered once the motions are detected within the region of the button interface. However, the proposed technique only provide with limited scalability for the design of user interface. For example, user had to move him/herself to reach and interact with specific buttons. But, in our study, we proposed to use *waving* gesture as a startup action and the interfaces will pop-up around the user's waving hand. Thus, user did not need to move his/her body to make interactions.

Chapter 4

Head gesture interface

This chapter presents a head gesture interface for interacting with a self-portrait camera [72]. The intuitive *nodding* and *head-shaking* gestures are recognized and control the camera zoom in/out on the face, and a *mouth-opening* gesture triggers the camera to take a picture. The interface helps users to take self-portraits effectively and efficiently. We will explain the implementation of gesture recognition, and the design of user interface in detail manner. In the experiments, we evaluated its usability factors (effectiveness, efficiency, and satisfaction) and compared it to a hand held smartphone remote control in a user study. We also discussed the design goals, degree of completion and limitations of the proposed system. Finally, we proposed the future directions.

4.1 Motivations and design goals

The previous two chapters have suggested that a vision-based gesture interface may a better choice for developing a remote control interface for interacting with a self-portrait camera. It provides intuitive and convenience interactions. However, it is difficult to develop a zooming interface using hand gestures, as the user's hands may go outside the field of view of the camera, especially at high zoom values. (see Figure 4.1). Moreover, when using hand gestures a large size display is needed to show the live view and the graphical user interface. This limits the portability of the system and obstructs its practical use in outdoor environment.

Therefore, in this chapter we propose a new gesture interface that uses head *nodding*, *shaking* and *mouth-opening* gestures as the method for interacting. The head gestures have

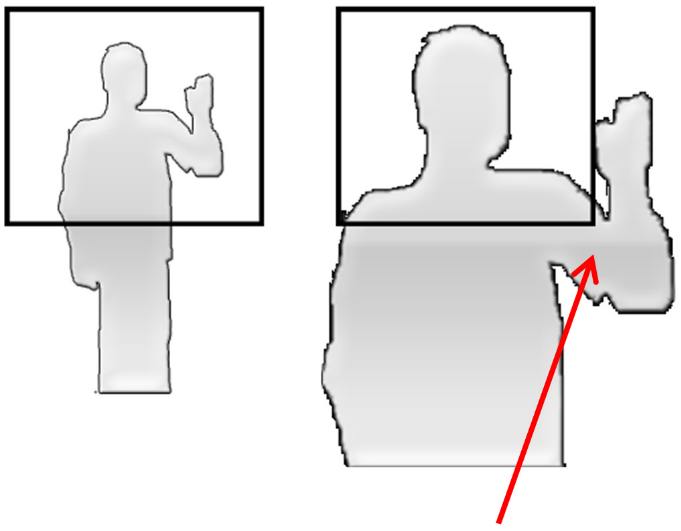
Figure 4.1: Hands may go outside the field of view of the camera.

several advantages, 1) it may work well with small front screen, 2) it is suitable as the basis of a zooming interface, because the face is unlikely to move outside the field of view; 3) as the face is the focus and most important region in a self-portrait, it is always available for gestures, which can be mapped to camera control functions.

Our main design goal in this chapter is to develop a vision-based head-gesture interface for controlling a digital camera to take self-portrait pictures effectively and efficiently. Considering the feedback from previous studies and surveys, we present a summary of the design goals of the current system.

Small size of frontal screen - Unlike conventional vision-based interfaces, which usually have a large display to provide visual feedback, we took mobility into consideration to design a system that would work well with a small screen. Currently, we use a 3.5-inch viewfinder and have introduced several strong gesture patterns that work well with little visual feedback.

Strong motion gesture patterns - Portrait shots do not capture motion, but rather focus on a static posture. Therefore, it is straightforward to use motion gestures as control commands. The gestures should have strong motion patterns that cannot easily be triggered by accident. Moreover, it is better to use small motions that do not disturb the user's eye contact with the lens when performing gestures. Thus, the user can still observe a preview of his/her posture and confirm the status of the camera.

Head gestures only - We believe that head gestures are the best choice for mapping to the zooming function of the camera. To maintain consistency, we did not combine head gestures with hand or body gestures.

Real-time processing - The image-processing procedure for detecting gestures should be fast (at least 30 frames per second, FPS) to guarantee that the system shows smooth video on the viewfinder and so the user can obtain instant feedback from the camera while performing gestures.

We will discuss the validity of our design in the Summary and discussions section on the basis of the results of the user study.

4.2 Proposed gesture interface

The major innovation of this work is that the self-portrait camera is responsive to head and mouth motion gestures when the user is in front of the lens. When the user faces the camera, the camera first detects the user's face and then tracks *nodding*, *head shaking*, and *mouth-opening* gestures. The user can perform these gestures to zoom in, zoom out, and trigger the shutter, respectively, to take a self-portrait photograph.

4.2.1 Overview

The prototype system (Figure 4.2) consists of a professional digital single-lens reflex (DSLR) camera, an iPhone, and a personal computer (PC, not shown in the figure). The DSLR camera is used to take pictures, and the iPhone is used as a frontal screen to enable users to see themselves in the live view video. A specific type of tripod or professional tripod can be used to hold the camera. In our current prototype system, a PC is connected to the camera via a USB cable to exchange data with the camera, processing images to detect gestures, and sending previews to the iPhone through a WiFi connection.

To interact with the system, the user faces the camera and performs the head and/or mouth gestures. The face and detected gestures will appear on the live view iPhone screen as a visual aid.

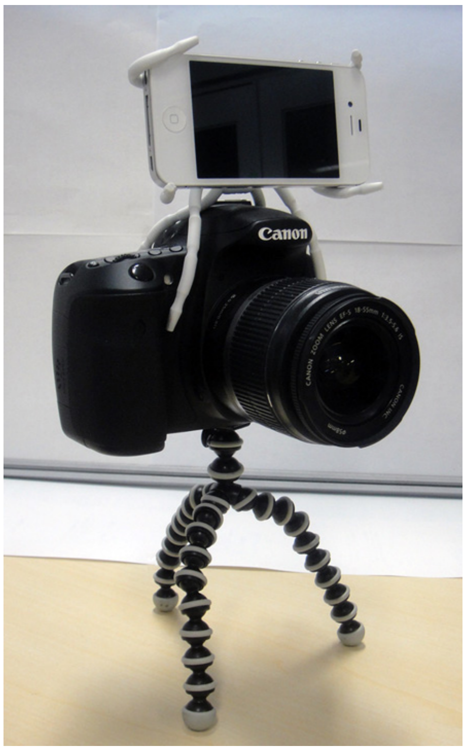

Figure 4.2: Prototype of self-portrait system that use head gestures to make interaction.

4.2.2 Introduced interaction gestures

We examined many possible head and facial gestures to identify intuitive motion gestures that could be mapped to two important camera functions, i.e., zooming and shutter trigger. We decided on motion gestures, as static head poses or facial expressions may be easily confused with a user's portrait postures and expressions. In addition, when a user is making a static pose for a photograph, motions can be easily distinguished as command functions. Another important consideration is that the gestures should not be easily triggered by accident. When preparing for a shot, the user may try different creative postures and expressions with lots of movement; therefore, the gestures should be markedly different from whimsical or general movements of the head. In addition, the gestures mapped to camera functions should be intuitive.

Based on the above considerations, we chose *nodding*, *head shaking*, and *mouth-opening* as candidate gestures for controlling camera functions. All three gestures are intuitive and common. Many studies have considered these types of gestures. In fact, *nodding* and *head shaking* gestures are widely utilized in mobile devices [73], [74], interactive dialog systems [75], [76], wheelchairs [77], [78], and human-robot conversation [79], [80], [81]. *Mouth-opening* gestures have also been studied [82], [83], [84], [85]. These gestures induce little fatigue and can be easily remembered by users. In our earlier implementation, we attempted to detect an *eyebrow-raising* gesture, which has also been discussed previously [83], [86]. However, in a user study, we found that this gesture cannot be recognized well in users with long bangs, especially women, and we therefore abandoned it. We also considered *clockwise/counterclockwise head-tilting* gestures, but experiments showed that it was impractical, as users easily lost eye contact with the screen when performing it, and it was difficult to recognize these gestures. Hence, we also abandoned this gesture.

In the following subsections, we describe our gesture-recognition technique using *nodding*, *head shaking*, and *mouth-opening* gestures.

4.3 Gesture-recognition technique

The proposed gesture-recognition technique is based on face-detection methods that have achieved great success [87], [88], [89]. We use face detection to detect the face region in each frame of the video. To recognize *nodding* and *head shaking* gestures, we first present a safe zone to restrict head motions to exclude unnecessary motions (Figure 4.3), and apply Lucas-Kanade optical flow tracking [39] to determine the two-dimensional (2D) motions of the head. To recognize the *mouth-opening* gesture, we first conduct empirical estimation to locate the mouth region of the face. The recognition of gestures is made under the assumption that a user's face is directly in front of the camera.

4.3.1 *Nodding* and *head shaking* recognition

To recognize the head gestures, we first define a safe zone as mentioned above. This zone, which is 40% larger than the face region, is initialized based on the face region in the image. Then, in the following frames, we check whether the face has moved out of the safe zone or not. If so, the zone is re-initialized and a new check will start; if not, the zone remains as it is and the system waits for the next frame and face region to check the head motion. When the face does not move out of the safe zone within a specific period (currently 500 ms), then it is assumed that the head is still and that it is a good time to recognize head gestures.

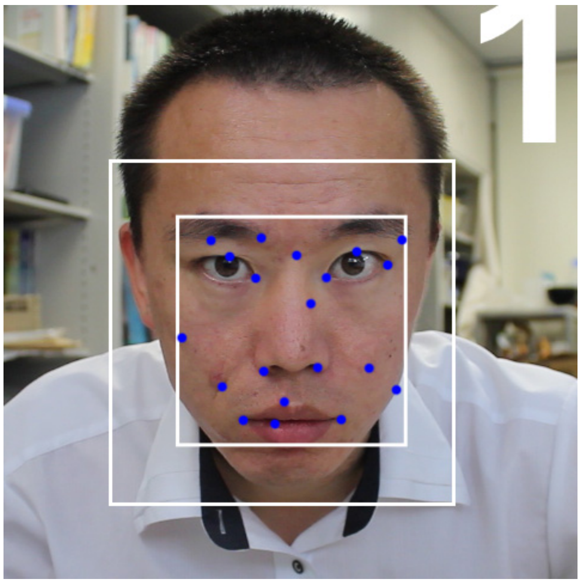

Figure 4.3: Face region (inside rectangle), safe zone (outer rectangle), and the feature points extracted are marked with filled circles.

Within the face region, we extract the image features for tracking head motion. Several feature-extraction algorithms have been reported previously [90], [91], but we chose a fast extraction method derived from the Hessian matrix, and selected the top 30 feature points as good features to track, as defined by Shi and Tomasi (S-T) [92]. After feature extraction (see Figure 4.3), the Lucas-Kanade [39] method optical flow measurement is conducted for tracking the motion of each feature point. This is one of most precise methods for tracking Shi and Tomasi features in images. The optical flow measurements are feature points in the current frame displacement from the previous frame. The length and direction of motion of each feature point can be determined by calculating each feature's displacement. We calculate the set of feature points within the face region, and calculate the mean length (speed) and direction of movement as the main parameters in each frame.

The general motion data patterns of *nodding* and *head shaking* are shown in Figure 4.4. These data were collected and recorded from one female graduate student. Important statistical information can be obtained from these data. First, in the *nodding* gesture (Figure 4.4a), the motion directions are between 90° and 180° during tilting of the head down, and 270° and 360° during tilting of the head up. In contrast, the directions of motion in the *head shaking* gesture (Figure 4.4b) are smooth and steady along the 45° and 225° lines. Second, the motion length data change periodically during the head gestures, but

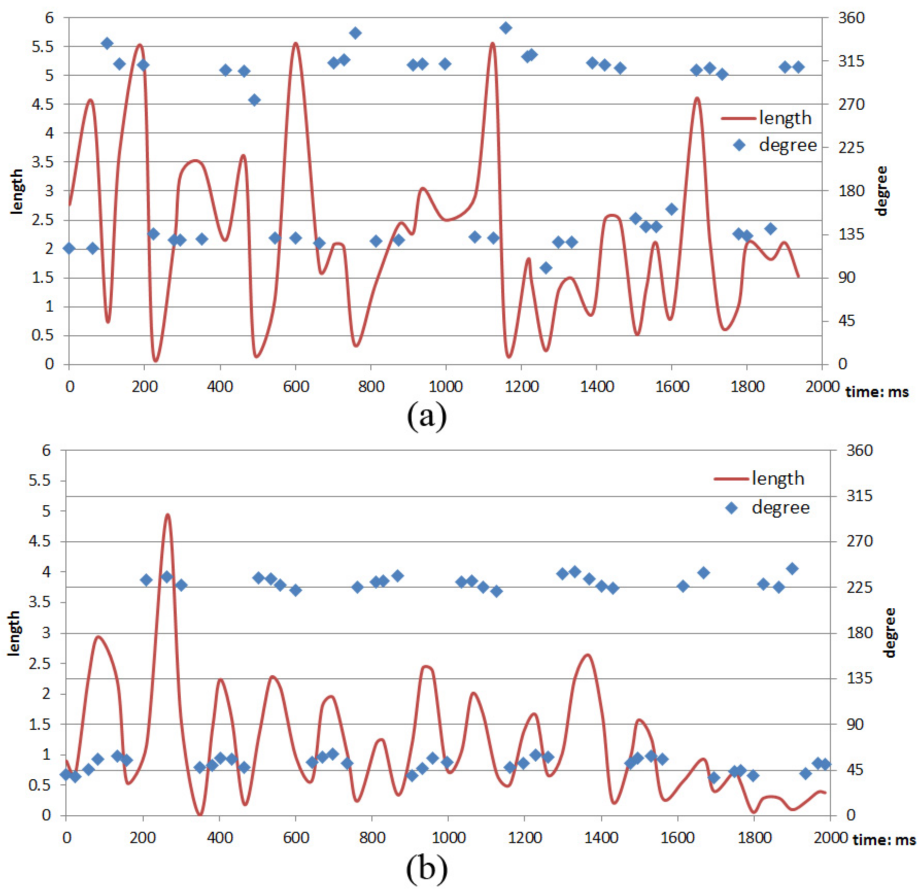

Figure 4.4: Nodding (a) and head shaking (b) gesture data.

remain below a peak value of 6. The time intervals of each action switch, i.e., up-down and left-right movements, were calculated as 122.2 ms and 137.5 ms, respectively. However, we found in a number of experiments that these intervals varied depending on the user. To make the recognition less restricted, we ultimately chose an interval threshold of 500 ms in our application.

Based on these data, we concluded that the shaking gesture is a steadier motion than the *nodding* gesture. Hence, we separated the motion region of moving recognition into four regions: right, 10-80° (70° span); left, 190-260° (70° span); up, 260-360° and 0-10° (110° span); and down, 80-190° (110° span). The length of motion must be larger than 0.5 and less than 6.0 in the recognition process.

The timing-based finite state machine (FSM) shown in Figure 4.5 was used to recognize the gestures. The figure shows an example of *head shaking* gesture recognition with a transition chart with two main states: Stationary 1 as the motionless state and Safe Zone when the face is inside the safe zone within a certain period. The Safe Zone state includes Stationary 2, left motion, and right motion. The transition from left to right or reverse transition adds one factor to the *shaking* count.

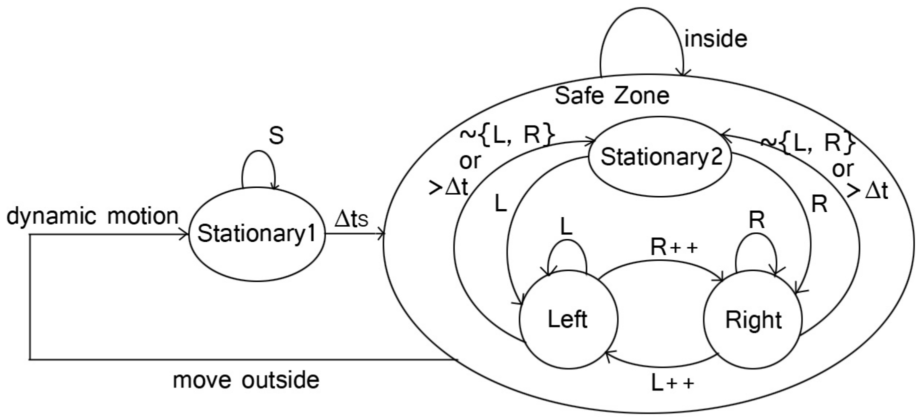

Figure 4.5: Finite state machine for recognizing the *head shaking* gesture.

Figure 4.6 shows an example of the gesture-recognition procedure in image sequences. In each image, the fan shape and the segment line at the middle of the face indicate the direction of motion. The count of nods and shakes recognized in each frame is displayed at the top left of the safe zone ("NOD" represents nodding, "SAK" represents *shaking*). The user can see the recognition states while performing gestures. This is a very useful visualization technique that allows the user to easily understand the gesture recognition process and to quickly become familiar with the gestures.

4.3.2 Recognition of the *mouth-opening* gesture

To recognize the *mouth-opening* gesture, we first estimated the mouth region related to the rectangle containing the face (Figure 4.7). Two subjects, one female and the other male, participated in the user study to examine the mouth region related to the face rectangle. We considered both closed and opened mouths, and tested the final recognition accuracy of the *mouth-opening* gesture. We concluded that based on a rectangle using the distances from the eyebrows down to the lower lip and between the outside edges of the two eyes, the mouth width was 50% of the face width and open-mouth height was 50% of the face height.

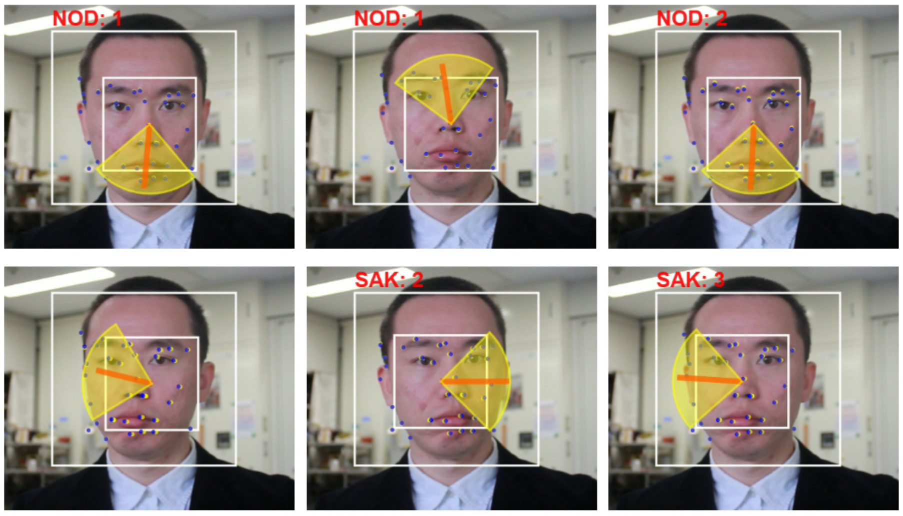

Figure 4.6: *Nodding* and *shaking* gesture motions. Images in the top row show two nods; those in the bottom row show four shakes.

In the second step, we manually arranged a matrix of 6 × 6 tracking points inside the region of the mouth, and applied Lucas-Kanade [39] optical flow to track the points' motions in each video frame (Figure 4.8). We did not apply dense Horn-Schunck [56] optical flow tracking as used in a previous study [82], as it has low performance. We used a history-recording and accumulation method to analyze the motion data, similar to the previous study [82]. Two 6 × 6 historical matrixes accompany mouth motions. One detects opening motions, defined as the Down Matrix, which records motions in the downward direction. The other detects closing motions, defined as the Up Matrix, which records motions in the upward direction. The two matrixes are shown together with the mouth image in Figure 4.8, with the Up Matrix on top and the Down Matrix on the bottom of the mouth image. The strength of detected motions is represented as dark blocks. The images from left to right represent three key frames of the *mouth-opening* gesture. The motion data are accumulated within 500 ms. When the accumulated data reach the defined threshold, a *mouth-opening* gesture can be detected.

The strength of matrix data is calculated in each image frame, and the pattern can be seen in the graph at the bottom of the figure. We concluded that when the mouth is opened, the Down Matrix peaks, and when it is closed, the Up Matrix peaks. Thus, the *mouth-opening* gesture can be recognized when such data patterns are detected.

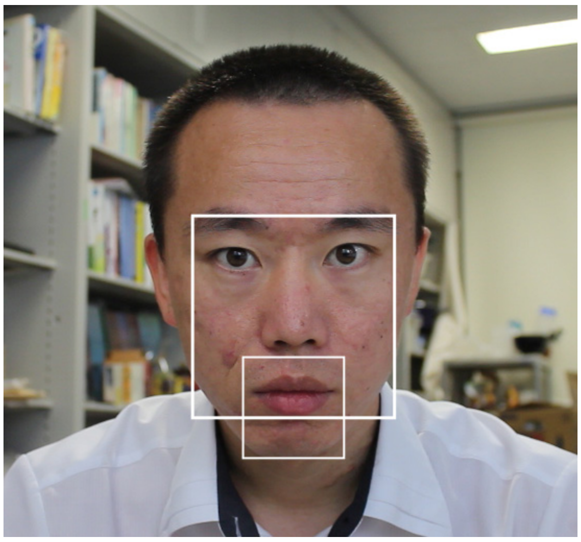

Figure 4.7: Estimated mouth region.

4.4 Function mappings

We used *continuous nodding* and *shaking* for zooming in and out, respectively. In a pilot study, we confirmed that these gestures are intuitive to users. In the earlier phase, we mapped *double nodding* gesture, the head nods twice and stops, to shutter trigger function [72]. However, we found it can easily be confused with *continuous nodding*. Therefore, we abandoned the *double nodding*, and mapped the *mouth-opening* gesture to the shutter trigger function instead. This gesture initiates autofocus on the user's face, and triggers the shutter timer to start countdown from 5 to 1, after which the camera takes the picture. If *head shaking* is performed during the countdown, it cancels it.

Regardless of how fast the user performs the gestures, zooming is set at a constant speed of 0.75X per second. We abandoned mapping faster speed gestures to faster zooming, as it made the users nervous and resulted in loss of accuracy.

4.5 Implementation

In the current implementation, we used a professional DSLR camera (Canon 60D) with an 18-55 mm lens. The camera offers a software development kit [18] for developers, including embedded hardware support for face detection. It runs at 30 FPS with 1056 ×

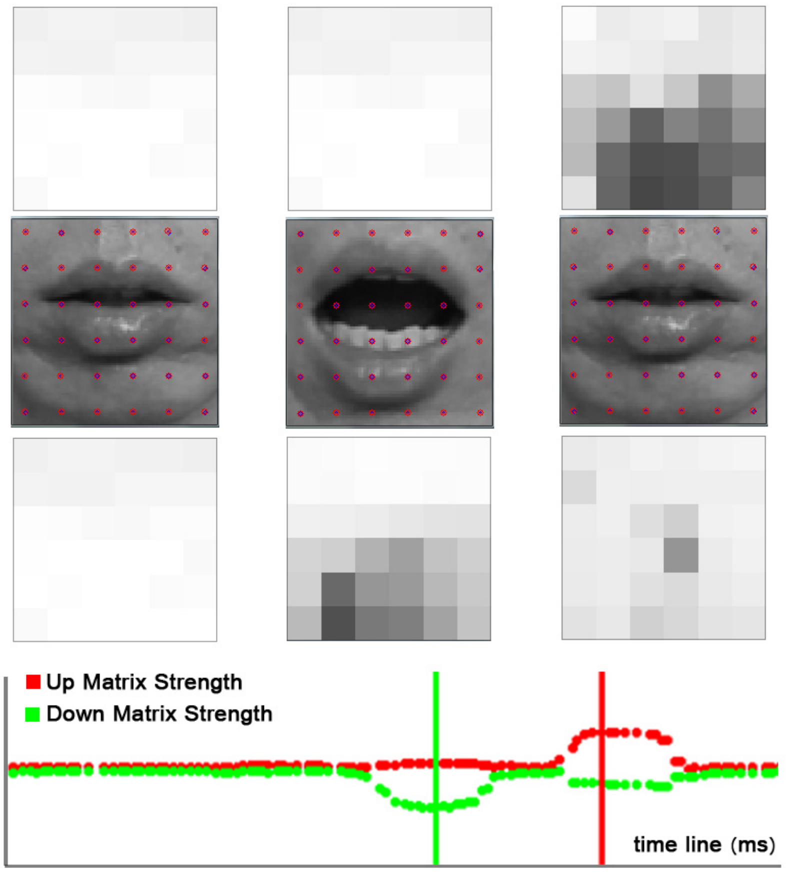

Figure 4.8: The mouth-opening gesture. The Up Matrix and Down Matrix are shown at the top and bottom of the mouth image, respectively. The graph shows the data strength of the two historical matrixes.

704 video image sequences for live view, and can detect faces at sizes from 58 × 58 to 513 × 513 in the image frame.

We developed an iPhone app and used the iPhone screen to show the camera preview. The iPhone preview app runs at 20 FPS with an image size of 480 × 320 received from the camera in real time. The iPhone can also be used as a handheld remote control. It provides the same functions as the head gesture interface, with zooming and shutter trigger. We used the iPhone as a handheld remote control for comparison with the proposed gesture interface in the user study.

A desktop PC with an Intel Core 2 Q8300 2.5-GHz CPU was used for image processing and gesture recognition. The camera was connected to the desktop PC via a USB cable, and the PC sent the preview to the iPhone through a WiFi connection.

We used the OpenCV library [62], which provides an implementation of Lucas-Kanade optical flow tracking to estimate head motions. The application program was written in C++.

The performance of the implemented gesture-recognition algorithm is summarized in Table 4.1. Gesture recognition had the same performance at different face sizes in the image, as we scaled the face to a standard size of 120 × 120.

Table 4.1: Performance of gesture recognition

	Process time (millisecond)
Nodding and shaking	1.2
Mouth-opening	0.5

The prototype system was constructed in our laboratory and used for the experiments described in the following section (Figure 4.9).

4.6 Experiments

A user study was performed to evaluate the usability of the proposed interface. Although usability evaluations have been discussed in many previous reports [93], [94], [95], [96], [97] we followed the ISO 9241 usability definition [98] and collected data on three distinct aspects of the proposed interface: effectiveness, efficiency, and user satisfaction. We also compared the gesture interface with a handheld remote control.

Effectiveness. This measures the accuracy and completeness with which users can achieve specified goals in particular environments. We arranged consecutive tasks to allow the user to perform the gestures many times to complete three types of shots: full-body, close-up, and upper-body shots. We evaluated the task completeness and the gesture recognition rate. We also compared the results to the handheld remote control when the users completed the same tasks.

Efficiency. This reflects the resources expended in relation to the accuracy and completeness of the goals achieved. We observed how many times participants actually performed the gestures to complete the three shots, and the time to completion.

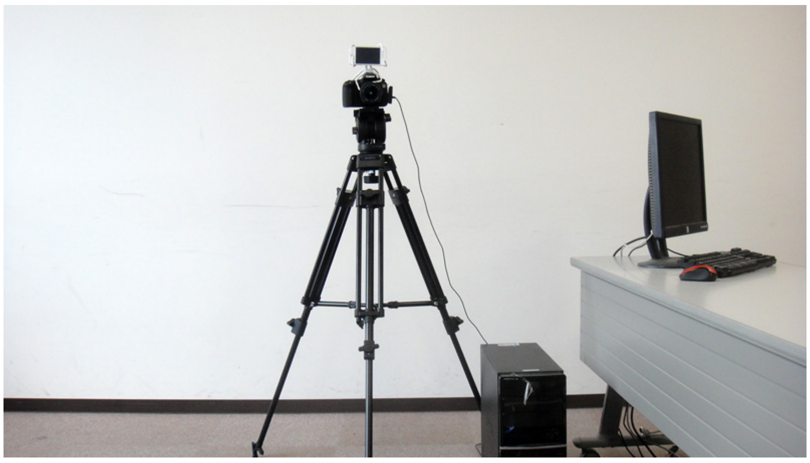

Figure 4.9: Experimental environment and apparatus.

User satisfaction. This is the comfort and acceptability of the system to its users and other people affected by its use. We used subjective assessment expressed on a five-point scale.

4.6.1 Apparatus

We set up the experiment indoors under standard daylight conditions. We did not investigate the robustness of the algorithm to lighting conditions, as it depends on the implementation of face detection and Lucas-Kanade optical flow, which have been examined in previous studies [99], [100]. The camera was placed on a tripod. The experimental setup is shown in Figure 4.9.

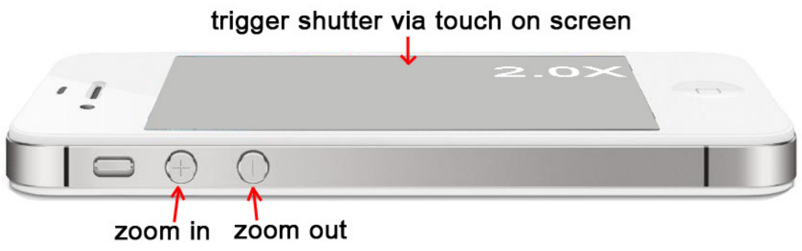

Figure 4.10: iPhone remote control: two buttons to control zooming, touch on the screen to take shots.

The iPhone was attached to the top of the camera. Two types of remote control were implemented: a button remote control and a touch remote control. The button remote

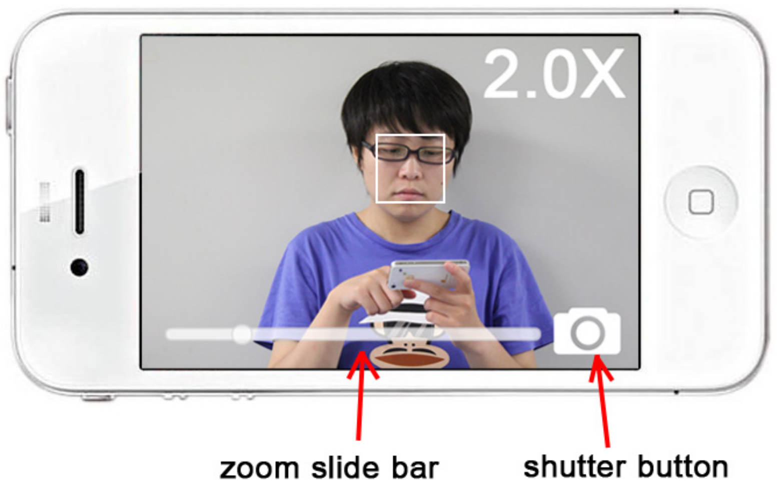

Figure 4.11: iPhone remote control. The screen shows the camera preview and any camera information.

control represents the traditional method, which provides users with eyes-free operation. Figure 4.10 shows the application where the two physical buttons of iPhone were used to control zooming, and where the touchscreen was used to trigger the camera shutter. On the other hand, the touch-based interface (see Figure 4.11) provides the live preview on screen and the graphical user interface. The graphical user interface on the touchscreen includes two distinct functions: a shutter button icon that triggers the camera to take a picture, and a zoom slide bar to control zooming.

4.6.2 Participants

We recruited 12 subjects (6 female, 6 male) aged 23-31 years (mean = 26.6, SD = 2.5) from different countries (one from a Western country, one black African subject, the rest Asian). Each participant was paid 1000 yen to participate in the study, which took 20-40 min. All participants were anonymous volunteers who saw our recruitment announcement in the international researchers' mailing list in the city of Tsukuba.

4.6.3 Task and procedure

Each experimental session involved one participant and consisted of four phases: 1) The experimenter introduced the system and allowed the participant to become familiar with it; 2) the participant completed consecutive tasks to measure the effectiveness and efficiency of the interface; 3) the subject completed a questionnaire as a subjective assessment of the system; and 4) user gestures were recorded to calculate the recognition accuracy and for future improvement.

In the first phase, the experimenter explained and demonstrated the two basic functions: zooming and shutter triggering. As this was likely to be the first time that the participants had experienced such an interface, each participant was shown how to use it and given time to become familiar with the system. The use of the iPhone remote control was introduced at the same time.

In the second phase, participants used both the gesture interface and a handheld remote control to complete three consecutive tasks, namely taking full-body, close-up, and upper-body portraits (Figure 4.12). The participants stood in the same position during the tests, but were allowed to explore the head gesture interface or handheld remote control to zoom and take several shots. The three types of photographs are illustrated in Figure 4.12 and summarized below.

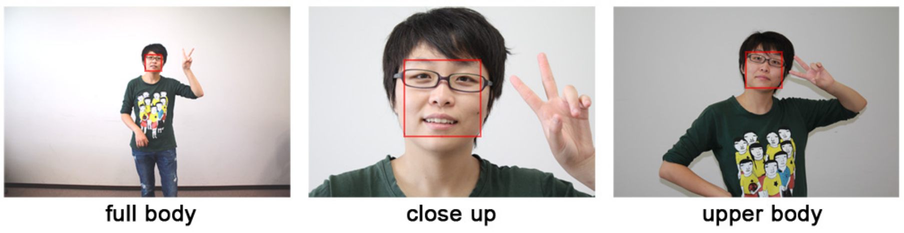

Figure 4.12: Three types of photograph: full-body, close-up, and upper body.

Full-body shot. No zoom required. Perform shutter trigger command after standing in the correct position (size of face region from 60 × 60 to 90 × 90 in the image).

Close-up shot. The participant zooms the camera from full-body to close-up (size of face region from 120 × 120 to 180 × 180 in the image) and takes the picture. This task tests the zooming in function.

Upper body shot. The participant zooms out from close-up to upper body (size of face region from 260 × 260 to 340 × 340 in the image) to take a picture. This task tests the zooming out function.

The two techniques (i.e., head gesture interface and using the remote control) were arranged under the same conditions as outlined above. 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.

The interaction gestures and time to completion of each of the three tasks were automatically recorded. The portraits were saved on the camera and PC.

In the third phase, the subjects completed a questionnaire as mentioned above, and in the fourth phase, the subjects participated in a separate experiment in which they performed the three types of gestures several times. The process was recorded as a video file for offline analysis of the accuracy of gesture recognition. Participation in this phase was voluntary.

4.6.4 Results of effectiveness and efficiency

We assessed three factors to determine the effectiveness and efficiency of the proposed gesture interface, i.e., task *completeness*, the *number of gestures* participants performed to complete the tasks, and the *recognition rate* of gestures. We also collected a subjective assessment of effectiveness of the gesture interface compared to the handheld remote control.

The task completeness results are shown in Table 4.2. All of the participants completed the three types of pictures using both the gesture interface and the handheld remote control. We concluded that the head gesture interface is effective for users to control the zooming function and take self-portraits.

Table 4.2: Task Completeness

	Full-body	*Close-up*	*Upper body*
Gesture interface	100%	100%	100%
Remote (button)	100%	100%	100%
Remote (touch)	100%	100%	100%

To evaluate the efficiency of the gesture interface we recorded the number of gestures performed by each of the 12 users during the experiment sessions. The results are shown in Table 4.3. The mean values were calculated and the standard deviations are shown in parentheses.

The full-body shot served as the starting point, for which the participants stood in a specified position and performed the mouth-opening gesture to take the first picture. Thus, there were no nods or head shakes. After this, participants performed the nodding gesture

Table 4.3: Performance for the three types of picture

	Nodding	*Shaking*	*Time*
Full-body to Close-up	10.9 (1.1)	0.8 (1.4)	7.6 s (1.2 s)
Close-up to Upper body	1.4 (1.9)	16.3 (1.7)	7.8 s (2.0 s)

to zoom in and take a close-up shot. This took 11.4 (SD = 1.1) nods and 7.6 s (SD = 1.4 s) on average to complete the task and take a shot. Sometimes, the participants had to perform the shaking gesture to zoom out slightly as an adjustment after zooming in. However, this was rare. The upper body shot took on average 16.8 (SD = 1.9) shaking gestures and 7.9 s (SD = 2.2 s) to complete. The final task, performing the shaking gesture to zoom out from close-up to upper body, took on average 16.8 (SD = 1.9) shakes and 7.9 s (SD = 2.2 s) to complete. In some cases, participants performed the nodding gesture to zoom back in as an adjustment.

A comparison with using the remote control with regards to time is shown in Figure 4.13 Users took about 2 times longer to complete the tasks when using gesture interface. This is generally due to the exactness of gesture recognition and the speed of zoom (0.75X per second). One-way analysis of variance (ANOVA) showed a significant effect ($p < 0.001$) of the three techniques both for close-up and upper body zoom tasks.

To examine gesture recognition accuracy, we collected 118 nodding, 126 shaking, and 20 mouth-opening gestures on AVI video samples from eight participants who agreed to allow video recording. We ran our application to recognize gestures on sample video to calculate the recognition rate. The results, summarized in Table 4.4, indicate good performance and recognition accuracy above 80% for nodding and shaking gestures. In addition, it achieved 100% accuracy recognizing mouth-opening gestures.

Table 4.4: Accuracy of gesture recognition

	Total	*Recognized*	*Missed*	*Accuracy*
Nodding	118	101	17	85.6%
Shaking	126	102	24	81.0%
Mouth-opening	20	20	0	100.0%

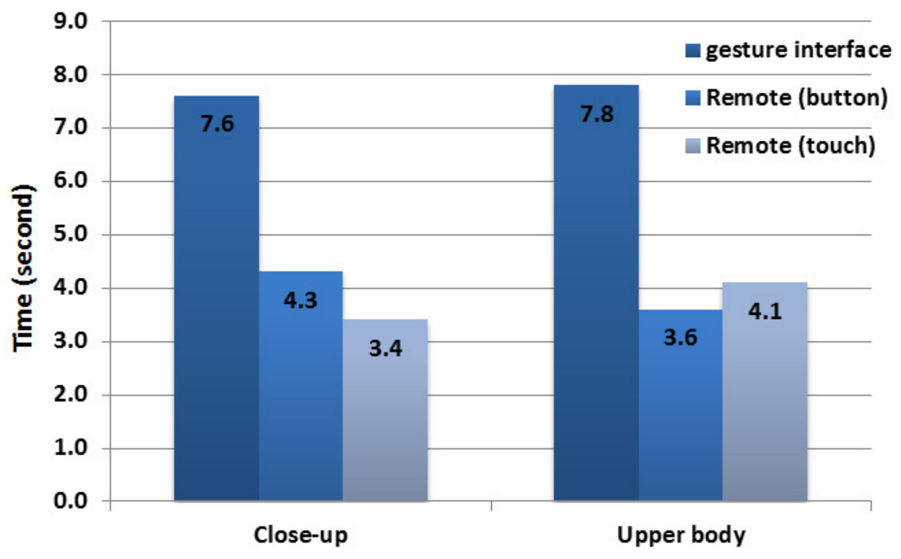

Figure 4.13: Time comparison of the three techniques.

Finally, participants were asked to rate their satisfaction with the effectiveness of the proposed gesture interface compared to the handheld remote control on a five-point scale from "very poor" to "excellent". Table 4.5 shows the results. We conducted an ANOVA test, but there was no significant effect (F(2,33) = 0.49, p = 0.62).

Table 4.5: Effectiveness of the interface for taking portraits

	Excellent	*Good*	*Fair*	*Poor*	*Very poor*
Gesture interface	16.7%	50.0%	33.3%	0.0%	0.0%
Remote (button)	41.7%	33.3%	25.0%	0.0%	0.0%
Remote (touch)	41.7%	33.3%	16.7%	8.3%	0.0%

4.6.5 Results of user satisfaction

We asked the participants to rate their level of satisfaction with the adopted function mappings (Table 4.6). All of the participants agreed that mapping of the *nodding* gesture to zoom in and the *shaking* gesture to zoom out were both intuitive and appropriate. Almost 60% of participants were satisfied with the mapping of the *mouth-opening* gesture to the shutter trigger function. About 40% of users reported a lack of satisfaction with the *mouth-opening* gesture; these users complained that this gesture may conflict with mouth motions when they are talking, or may cause unwanted triggering of the shutter countdown while preparing for the picture. Our application included a cancel function in which the user can perform the *head shaking* gesture to cancel the countdown once it has started. The users evaluated this design as useful when they triggered the shutter by accident.

Table 4.6: Evaluation of function mappings

	Excellent	*Good*	*Fair*	*Poor*	*Very poor*
Gesture interface	83.3%	16.7%	0.0%	0.0%	0.0%
Remote (button)	75.0%	25.0%	0.0%	0.0%	0.0%
Remote (touch)	41.7%	16.7%	33.3%	8.3%	0.0%

In addition, the users evaluated their satisfaction with regard to freedom and concentration while preparing for a picture when using the proposed interface compared to the handheld remote control (Table 4.7). About 90% of participants reported a satisfied experience with the gesture interface (ratings of “Excellent” and “Good”), while about 66% participants felt satisfied with the handheld remote control, both the button and touch interfaces. An ANOVA test showed a significant effect, F(2,33) = 3.79, p = 0.03. Moreover, the post hoc tests by Tukey HSD showed that the gesture interface against touch remote control had a significant effect (p = 0.03). However, there were no significant effects of gesture interface vs. button remote control (p = 0.19) and button vs. touch remote control (p = 0.64).

Table 4.7: Freedom and concentration while preparing for a picture

	Excellent	*Good*	*Fair*	*Poor*	*Very poor*
Gesture interface	41.7%	50.0%	8.3%	0.0%	0.0%
Remote (button)	16.7%	50.0%	33.3%	0.0%	0.0%
Remote (touch)	0.0%	66.7%	25.0%	8.3%	0.0%

Finally, participants rated their overall satisfaction with the proposed gesture interface compared to the handheld remote control (Table 4.8). All of the participants reported a satisfied experience with the gesture interface, while about 80% and 60% participants felt satisfied with the button remote control, and touch remote control respectively. An ANOVA test showed a significant effect, F(2,33) = 3.98, p = 0.03. Furthermore, the post hoc tests by Tukey HSD showed that the gesture interface against touch remote control had a significant effect (p = 0.02). But there were no significant effects of gesture interface vs. button remote control (p = 0.17) and button vs. touch remote control (p = 0.63).

Table 4.8: Impression of proposed interface vs. handheld remote control

	Excellent	*Good*	*Fair*	*Poor*	*Very poor*
Gesture interface	41.7%	58.3%	0.0%	0.0%	0.0%
Remote (button)	8.3%	75.0%	16.7%	0.0%	0.0%
Remote (touch)	16.7%	41.7%	33.3%	8.3%	0.0%

4.7 Summary and discussions

4.7.1 Validity of design

Here, we review the appropriateness of our design goals based on the results of the survey and our observations of the participants' interactions with the self-portrait camera. We also examine whether our implementation satisfied these goals based on our observations of how the participants used the system.

Small size of frontal screen - The participants could interact with the camera with head gestures within a distance of 2 m. Although the participants stood 2 m from the camera and could not see the preview in detail on the 3.5-inch screen, all of them could still determine the zooming status and perform gestures to control zooming in/out from close-up to upper body and take the shots.

Considering strong pattern of motion gestures - Our observations in the present study confirm that the *nodding* and *shaking* motions are strong gesture patterns, which were rarely triggered by accident. However, the *mouth-opening* motion was less distinct and was unexpectedly recognized many times, particularly when participants were speaking or preparing for pictures. However, this gesture is still one of the best candidates for an intuitive gesture, as almost 60% of participants expressed a preference for it.

Head gestures only - The original motivation for introducing a head gesture interface was that the head and face are always available in images when developing vision-based gesture interfaces for zooming. Other gestures, such as hand or body gestures, cannot be used if the camera is zoomed close in. In the user study, we found that head gestures can provide an effective interface for controlling a self-portrait camera, particularly for controlling the zooming function.

Real-time processing - Gesture recognition can be achieved rapidly, within 1.2 ms and 0.5 ms for nodding and shaking, and mouth-opening, respectively. Thus, the algorithm used here can be implemented and will show good performance on a modern digital camera.

The observations and survey results obtained here indicate that the proposed gesture interface fulfills almost all of the requirements set at the beginning of the present study, thus suggesting that it is very effective for the stated application. In general, the users reacted very positively; they derived enjoyment from interacting with the self-portrait camera and were satisfied with the concept of the gesture interface. However, there were several slight problems, and these are discussed in the following subsection.

4.7.2 Limitations and future work

Our proposed self-portrait system is a “proof-of-concept” implementation, and so there were some hardware limitations. We used a Canon DSLR camera to take portraits, a PC connected to the camera for image processing and gesture recognition, and an iPhone attached to the camera as a frontal screen. These hardware limitations were frustrating for users when considering a practical self-portrait camera. However, we believe that the development of a smart camera or computational camera [13] equipped with a frontal screen and supporting custom programming will soon be developed. We hope to implement the gesture interface described here in such cameras in the future.

While the proposed gesture-recognition algorithm achieved a good level of accuracy, better results are expected with future improvements. The development of methods for extracting facial features to allow localization of the eyes, nose, mouth, and lips for detecting accurate facial motions is also expected in future.

Another limitation of the proposed interface is the inability to map to many more camera functions, such as aperture, shutter speed, ISO, white balance, and so forth. Further studies are required to develop an improved gesture interface to provide access to such camera functions.

In the future, we will improve the system for intermediate users and consider the result of portrait pictures. For example, we will consider helping users to take professional photographs, like showing example postures on the screen and letting users imitate them.

More long-term plans include discussions with professional photographers and self-portrait experts and exploring practical self-portrait camera features.

We have in mind the following three categories of practical use: 1) Photographers, who are able to take excellent self-portrait photographs of others, but who may find it difficult to take their own photographs. The self-portrait camera can help them explore their self-portraits. 2) Photography learning, people who want to learn portrait photography, but they find it difficult to find a model to take pictures. They can use self-portrait camera to practice on themselves. 3) Common users, who take self-portraits for fun and they want to share them.

4.8 Usage scenarios and potential applications

The proposed head gesture interface is not only can be applied to the self-portrait camera, but also other areas of applications. For example, in the study of surveillance camera that, aims at retrieval of events, people, or faces from video [101]. We can also apply face tracking [102] and gesture recognition to recovery actions of people. Another example is video chat system that, by using head tracking, it can provide an intuitive view changing method to control a PTZ camera [103]. But by applying the gesture recognition technique based on the tracking, it can provide more function controls, such as fast zooming, command triggering, etc. A last example is that, the potential applications on smart phone or tablet PC. The front camera on these devices makes it possible to develop new vision-based interaction techniques, and leveraging face and head gestures to interact with the devices. One successful study was reported in [104].

4.9 Related work

The vision-based head gesture interface has been studied by several researchers [73], [76], [78], [81], [82], [83], [105], [106], [107]. One typical gesture recognition scheme presented by Davis [106] uses feature tracking on face region to estimate head motions, and then utilizes a Finite State Machine (FSM) to recognize *nodding* and *shaking* gestures. The author used the gestures for a simple dialog-box agent and acquired yes/no acknowledgement from the user. However, it needed a hardware equipment, IBM PupilCam, to detect the face, thus limiting its practical application. Li [73] modified the recognition scheme and introduced the block motion vectors to perform motion estimation, which reduced computational requirements and enabled the use of a single camera to run on mobile devices. However, it required the user to place his/her head at predefined positions in order to perform the gestures. Tan [105] introduced the use of coordinates of eyes to judge the direction of face's movement. A hidden Markov model (HMM) was trained to perform *head nodding* and *shaking* recognition. The limitation reported by the author is that, by using eye locating method, it was easy to generate wrong results when the user moved his/her head in one direction continuously. Other proposals such as the use of stereo camera [76], or context knowledge [81] to improve the recognition were also reported.

On the other hand, facial gestures, such as mouth-opening and eyebrow-raising, that work together with head gestures, were also studied and proved to be an effective way to trigger commands [82], [83], [107].

Although the head gestures and facial gestures were widely studied, few of the studies addressed the zoom interface by using head gestures. In this work, we present an intuitive function mapping of zoom. Because humans can easily perform the *nodding* and *shaking* gestures continuously, it is appropriate to map them to zooming in/out respectively. This was verified in our user study, as described in Section 5.

We also improved the gesture recognition reported in [73], [106]. We use a safe zone around the face to perform auto-initialization and quickly exclude casual head movements. A simplified motion estimation by using optical-flow for "good features to track" [92] on face is used to achieve good performance and motion tracking accuracy. These improvements offer an effective head gesture interface.

Chapter 5

Conclusion

This thesis described the studies of vision-based gesture interfaces for interacting with a self-portrait camera. It introduced three types of gesture interfaces: 1) *hand gesture interface*, 2) *motion-based hand gesture interface*, and 3) *head gesture interface*. Figure 5.1 shows the study progress curve of the three interfaces. From the starting point, we found some limitations on the conventional approaches regarding taking self-portraits, then we proposed our approach, that is, to use hand gestures to interact with the camera. After we got several results, we proposed the motion-based gestures, and developed a rich gesture interface for controlling a DSLR camera. Finally, in the third study, we considered developing a zooming interface and made it work on a small size of front screen. The three types of interfaces helped users interact with self-portrait camera and take self-portraits with both effectiveness and efficiency. The experiments evaluated the usability and showed that the gesture interface is more useful for interacting with a self-portrait camera than the conventional camera interfaces. We believe the proposed three types of vision-based gesture interfaces are a promising concept for the future development of self-portrait cameras.

5.1 Contributions

Our main contribution in this work is demonstrating a new method, vision-based gesture interface for interacting with a self-portrait camera. We implemented three types of gesture interfaces to explore the possibility of gesture-based interaction.

In *hand gesture interface*, we first investigated the design of a new interaction approach for self-portrait camera. And then we introduced the image processing algorithm of ges-

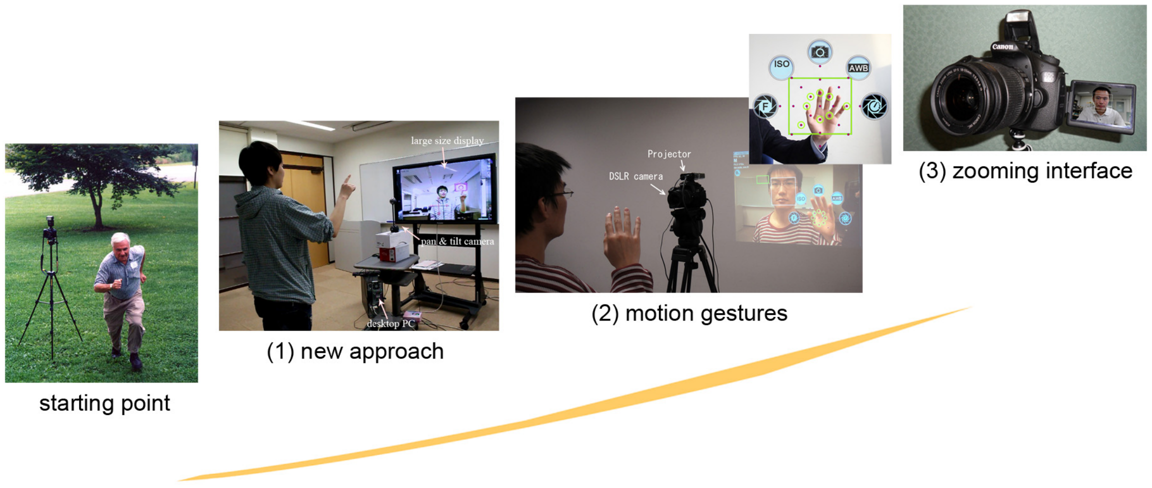

Figure 5.1: The study progress curve.

ture recognition that is used for detecting hand fingertip and hand directional movement gestures. We have implemented a prototype system and showed it to various audiences. Our experiment results showed that the proposed system got good evaluations of user experience, and the accuracy of gesture recognition achieved good performance. The review of conventional interaction approaches, and highlights of the advantages of our approach for taking self-portraits were also reported in this study.

In *motion-based hand gesture interface*, we introduced a completely real-time motion-based gesture recognition technique, which provides the robustness against lighting and color conditions, and offered to develop a rich gesture interface for controlling not only digital, but also analog parameters. The motion estimation was implemented by arranging a set of tracking points at a specific region on an image frame, and the optical flow at these points is calculated repeatedly in sequences of image frames to recognize the gestures. This method can greatly reduce the computing complexity and does not require any pre-training. The three types of motion gestures, *waving*, *eight-direction selection*, and *circling*, were recognized. By combining these gestures we developed a rich gesture interface for controlling a professional DSLR camera with various brightness-sensitive functions, such as aperture, shutter speed, ISO, white balance, and shutter trigger. Our experiments confirmed that the proposed motion-based gesture recognition method can achieve good performance and get good accuracy. And the gesture interface performed much better than those using conventional techniques such as contact-based interface and self-timers.

In *head gesture interface*, we presented a head gesture recognition algorithm for developing a zooming interface. We also aimed to develop a gesture interface that works well with a small sized front facing screen. The intuitive *nodding* and *head-shaking* gestures were recognized and control the camera zoom in/out on the face, and a *mouth-opening* gesture triggers the camera to take a picture. In the experiments, we confirmed that *n-odding* and *shaking* gestures are very suitable for mapping to zooming function and work well with small size front facing screen. We also found that the gesture interface has a high degree of usability (effectiveness, efficiency, and satisfaction) and performed much better than handheld remote control for interacting with a self-portrait camera.

Summary of our conclusions, 1) gesture interface is an effect technique to interact with self-portrait camera. It has a high degree of usability; it is superior to conventional techniques, such as contact-based self-timer and remote controller. 2) the motion-based gestures are appropriate when developing a vision-based gesture interface for self-portrait cameras.

5.2 Future work

Our study demonstrated the gesture interface can be a promising technique for interacting with a self-portrait camera. In the future, first we would like to consider the result of portrait photographs, such as professional studio photograph to improve not only the interaction approach but also the quality of photography. Second, we want to explore the development of self-portrait system with gesture interface that works with little feedback or without visual feedback, such as imaginary interface [108], eye-free interface [109]. This will allow the self-portrait camera to work outdoors without clear preview on the front screen. Finally, we want to pursue the potential of proposed gesture interfaces to other applications, such as HMD, AR, robot interaction, etc.

Bibliography

[1] "History of the camera," http://en.wikipedia.org/wiki/History_of_the_camera.

[2] "Digital photography," http://en.wikipedia.org/wiki/Digital_photography.

[3] "Digital versus film photography," http://en.wikipedia.org/wiki/Digital_versus_film_photography.

[4] L. Huang, T. Xia, J. Wan, Y. Zhang, and S. Lin, "Personalized portraits ranking," *Proceedings of the 19th ACM international conference on Multimedia*, pp. 1277–1280, 2011.

[5] D. Okabe, M. Ito, J. Chipchase, and A. Shimizu, "The social uses of purikura: photographing, modding, archiving, and sharing," *Pervasive Image Capture and Sharing Workshop, Ubiquitous Computing Conference*, pp. 2–5, 2006.

[6] "5 reasons why you should take a self portrait," http://www.iheartfaces.com/2012/04/self-portraits-tutorial/.

[7] S. Counts and E. Fellheimer, "Supporting social presence through lightweight photo sharing on and off the desktop," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 599–606, 2004.

[8] "100 seriously cool self-portraits (and tips to shoot your own!)," http://photozz.com/fizz/14216553.aspx.

[9] "4 tips for taking gorgeous self-portrait and outfit photos," http://www.shrimpsaladcircus.com/2012/01/self-portrait-outfit-photography-guide.html.

[10] "Taking a great self portrait with your camera,"
http://www.squidoo.com/self-portrait-tips.

[11] S. Chu and J. Tanaka, "Hand gesture for taking self portrait," *Proceedings of the 14th international conference on Human-computer interaction: interaction techniques and environments - Volume Part II*, pp. 238–247, 2011.

[12] "Nikon coolpix s1200pj camera,"
http://www.nikonusa.com/en/Learn-And-Explore/Article/g022fmeg/Built-in-Projector.html.

[13] A. Adams, E.-V. Talvala, S. H. Park, D. E. Jacobs, B. Ajdin, N. Gelfand, J. Dolson, D. Vaquero, J. Baek, M. Tico, H. P. A. Lensch, W. Matusik, K. Pulli, M. Horowitz, and M. Levoy, "The frankencamera: an experimental platform for computational photography," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 29:1–29:12, 2010.

[14] "Canon hack development kit," http://chdk.wikia.com/wiki/CHDK.

[15] "Magic lantern firmware,"
http://magiclantern.wikia.com/wiki/Magic_Lantern_Firmware_Wiki.

[16] "Breeze system," http://www.breezesys.com/.

[17] S. R. Gomez, "Interacting with live preview frames: in-picture cues for a digital camera interface," *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*, pp. 419–420, 2010.

[18] "Canon digital camera software developers kit (canon sdk),"
http://usa.canon.com/cusa/consumer/standard_display/sdk_homepage.

[19] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[20] A. Atia and J. Tanaka, "Interaction with tilting gestures in ubiquitous environments," *International Journal Of UbiComp*, vol. 1, no. 3, pp. 1–13, 2010.

[21] P. Mistry and P. Maes, "Sixthsense: a wearable gestural interface," *ACM SIGGRAPH ASIA 2009 Sketches*, pp. 11:1–11:1, 2009.

[22] G. Cohn, D. Morris, S. Patel, and D. Tan, “Humantenna: using the body as an antenna for real-time whole-body interaction,” *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pp. 1901–1910, 2012.

[23] M. Chen, L. Mummert, P. Pillai, A. Hauptmann, and R. Sukthankar, “Controlling your tv with gestures,” *MIR 2010: 11th ACM SIGMM International Conference on Multimedia Information Retrieval*, pp. 405–408, 2010.

[24] C. Harrison, D. Tan, and D. Morris, “Skinput: appropriating the body as an input surface,” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 453–462, 2010.

[25] J. Kim, S. Mastnik, and E. André, “Emg-based hand gesture recognition for realtime biosignal interfacing,” *Proceedings of the 13th international conference on Intelligent user interfaces*, pp. 30–39, 2008.

[26] T. S. Saponas, D. S. Tan, D. Morris, J. Turner, and J. A. Landay, “Making muscle-computer interfaces more practical,” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 851–854, 2010.

[27] T. Schlömer, B. Poppinga, N. Henze, and S. Boll, “Gesture recognition with a wii controller,” *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pp. 11–14, 2008.

[28] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, “Gesture recognition with a 3-d accelerometer,” *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing*, pp. 25–38, 2009.

[29] R. Francese, I. Passero, and G. Tortora, “Wiimote and kinect: gestural user interfaces add a natural third dimension to hci,” *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 116–123, 2012.

[30] J. Eisenstein and W. E. Mackay, “Interacting with communication appliances: an evaluation of two computer vision-based selection techniques,” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1111–1114, 2006.

[31] S. Malik and J. Laszlo, "Visual touchpad: a two-handed gestural input device," *Proceedings of the 6th international conference on Multimodal interfaces*, pp. 289–296, 2004.

[32] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," *International conference on computer graphics and vision*, pp. 85–92, 2003.

[33] P. Yogarajah, J. Condell, K. Curran, A. Cheddad, and P. M. Kevitt, "A dynamic threshold approach for skin segmentation in color images," *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 2225–2228, 2010.

[34] M. S. Iraji and A. Yavari, "Skin color segmentation in fuzzy ycbcr color space with the mamdani inference," *American Journal of Scientific Research*, pp. 131–137, 2011.

[35] J. Kovac, P. Peer, and F. Solina, "Human skin colour clustering for face detection," *IEEE EUROCON 2003: International Conference on Computer as a Tool*, pp. 144–148, 2003.

[36] M. Jankowski, "Morphological image processing," *The Mathematica Journal*, vol. 8, no. 1, pp. 147–159, 2001.

[37] S. Perreault and P. Hebert, "Median filtering in constant time," *Image Processing, IEEE Transactions on*, vol. 16, no. 9, pp. 2389–2394, 2007.

[38] R. D. Yapa and H. Koichi, "A connected component labeling algorithm for grayscale images and application of the algorithm on mammograms," *Proceedings of the 2007 ACM symposium on Applied computing*, pp. 146–152, 2007.

[39] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[40] T. Fukasawa, K. Fukuchi, and H. Koike, "A vision-based non-contact interactive advertisement with a display wall," *Proceedings of the 5th international conference on Entertainment Computing*, pp. 394–397, 2006.

[41] T. Lee and T. Hollerer, "Handy ar: Markerless inspection of augmented reality objects using fingertip tracking," *Wearable Computers, 2007 11th IEEE International Symposium on*, pp. 1–8, 2007.

[42] A. D. Cheok, S. W. Fong, K. H. Goh, X. Yang, W. Liu, and F. Farzbiz, "Human pacman: A mobile entertainment system with ubiquitous computing and tangible interaction over a wide outdoor area," *Fifth International Symposium on Human Computer Interaction with Mobile Devices and Services*, pp. 209–223, 2003.

[43] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 1–10, 2007.

[44] A. Sherstyuk, A. Dey, C. Sandor, and A. State, "Dynamic eye convergence for head-mounted displays improves user performance in virtual environments," *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 23–30, 2012.

[45] A. Henrysson, J. Marshall, and M. Billinghurst, "Experiments in 3d interaction for mobile phone ar," *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pp. 187–194, 2007.

[46] A. Henrysson, M. Billinghurst, and M. Ollila, "Face to face collaborative ar on mobile phones," *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 80–89, 2005.

[47] F. Wang, X. Cao, X. Ren, and P. Irani, "Detecting and leveraging finger orientation for interaction with direct-touch surfaces," *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pp. 23–32, 2009.

[48] "Sony party shot," http://www.sony.jp/cyber-shot/party-shot/.

[49] "Casio tryx camera," http://www.casio-intl.com/asia-mea/en/dc/ex_tr150/.

[50] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Journal of Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[51] S. Lenman, L. Bretzner, and B. Thuresson, "Using marking menus to develop command sets for computer vision based hand gesture interfaces," *Proceedings of the second Nordic conference on Human-computer interaction*, pp. 239–242, 2002.

[52] A. D. Wilson, “Robust computer vision-based detection of pinching for one and two-handed gesture input,” *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pp. 255–258, 2006.

[53] R. Wang, S. Paris, and J. Popović, “6d hands: markerless hand-tracking for computer aided design,” *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 549–558, 2011.

[54] R. Y. Wang and J. Popović, “Real-time hand-tracking with a color glove,” *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 63:1–63:8, 2012.

[55] B. Stenger, T. Woodley, and R. Cipolla, “A vision-based remote control,” *Studies in Computational Intelligence*, vol. 285, pp. 233–262, 2010.

[56] A. Bruhn, J. Weickert, and C. Schnörr, “Lucas/kanade meets horn/schunck: combining local and global optic flow methods,” *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[57] M. Bayazit, A. Couture-beil, and G. Mori, “Real-time motion-based gesture recognition using the gpu,” *IAPR Conference on Machine Vision Applications (MVA)*, pp. 9–12, 2009.

[58] M. Takahashi, M. Fujii, M. Naemura, and S. Satoh, “Human gesture recognition using 3.5-dimensional trajectory features for hands-free user interface,” *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, pp. 3–8, 2010.

[59] A. Fathi and G. Mori, “Action recognition by learning mid-level motion features,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–8, 2008.

[60] J. Hardy, E. Rukzio, and N. Davies, “Real world responses to interactive gesture based public displays,” *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, pp. 33–39, 2011.

[61] R. Salvador, T. Romão, and P. Centieiro, “A gesture interface game for energy consumption awareness,” *Proceedings of the 9th international conference on Advances in Computer Entertainment*, pp. 352–367, 2012.

[62] "Open source computer vision library (opencv)," http://opencv.org/.

[63] "Windows direct2d," http://msdn.microsoft.com/en-us/library/windows/desktop/dd370990.

[64] J. Fiss, A. Agarwala, and B. Curless, "Candid portrait selection from video," *ACM Transactions on Graphics*, vol. 30, no. 6, pp. 128:1–128:8, 2011.

[65] T. Murase, A. Moteki, N. Ozawa, N. Hara, T. Nakai, and K. Fujimoto, "Gesture keyboard requiring only one camera," *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, pp. 9–10, 2011.

[66] T. Murase, A. Moteki, G. Suzuki, T. Nakai, N. Hara, and T. Matsuda, "Gesture keyboard with a machine learning requiring only one camera," *Proceedings of the 3rd Augmented Human International Conference*, pp. 29:1–29:2, 2012.

[67] J. Hannuksela, M. Barnard, P. Sangi, and J. Heikkilä, "Adaptive motion-based gesture recognition interface for mobile phones," *Proceedings of the 6th international conference on Computer vision systems*, pp. 271–280, 2008.

[68] M. Lee and M. Billinghurst, "A wizard of oz study for an ar multimodal interface," *Proceedings of the 10th international conference on Multimodal interfaces*, pp. 249–256, 2008.

[69] M. Storring, T. B. Moeslund, Y. Liu, and E. Granum, "Computer vision-based gesture recognition for an augmented reality interface," *4th IASTED International Conference on Visualization, Imaging, and Image Processing*, pp. 766–771, 2004.

[70] G. R. Bradski and J. W. Davis, "Motion segmentation and pose recognition with motion history gradients," *Machine Vision and Applications*, vol. 13, no. 3, pp. 174–184, 2002.

[71] Z. Zivkovic, "Optical-flow-driven gadgets for gaming user interface," *International Conference on Entertainment Computing*, pp. 90–100, 2004.

[72] 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*, pp. 112–117, 2012.

[73] R. Li, C. Taskiran, and M. Danielsen, "Head pose tracking and gesture detection using block motion vectors on mobile devices," *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, pp. 572–575, 2007.

[74] A. Crossan, M. McGill, S. Brewster, and R. Murray-Smith, "Head tilting for interaction in mobile contexts," *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 6:1–6:10, 2009.

[75] L.-P. Morency, C. Sidner, C. Lee, and T. Darrell, "Contextual recognition of head gestures," *Proceedings of the 7th international conference on Multimodal interfaces*, pp. 18–24, 2005.

[76] L.-P. Morency and T. Darrell, "From conversational tooltips to grounded discourse: head posetracking in interactive dialog systems," *Proceedings of the 6th international conference on Multimodal interfaces*, pp. 32–37, 2004.

[77] F. Althoff, R. Lindl, and L. Walchshausl, "Robust multimodal hand- and head gesture recognition for controlling automotive infotainment systems," *VDI conference: the driver in the 21st century*, pp. 187–205, 2005.

[78] I. Yoda, K. Sakaue, and T. Inoue, "Development of head gesture interface for electric wheelchair," *Proceedings of the 1st international convention on Rehabilitation engineering and assistive technology: in conjunction with 1st Tan Tock Seng Hospital Neurorehabilitation Meeting*, pp. 77–80, 2007.

[79] C. L. Sidner, C. Lee, L.-P. Morency, and C. Forlines, "The effect of head-nod recognition in human-robot conversation," *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pp. 290–296, 2006.

[80] C. Liu, C. T. Ishi, H. Ishiguro, and N. Hagita, "Generation of nodding, head tilting and eye gazing for human-robot dialogue interaction," *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pp. 285–292, 2012.

[81] L.-P. Morency, C. Sidner, C. Lee, and T. Darrell, "Head gestures for perceptual interfaces: The role of context in improving recognition," *Artificial Intelligence*, vol. 171, no. 8-9, pp. 568–585, 2007.

[82] T. Palleja, E. Rubion, M. Teixido, M. Tresanchez, A. F. d. Viso, C. Rebate, and J. Palacin, "Using the optical flow to implement a relative virtual mouse controlled by head movements," *Journal of Universal Computer Science*, vol. 14, no. 19, pp. 3127–3141, 2008.

[83] Y. Gizatdinova, O. Špakov, and V. Surakka, "Comparison of video-based pointing and selection techniques for hands-free text entry," *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 132–139, 2012.

[84] P. Dalka and A. Czyzewski, "Lip movement and gesture recognition for a multimodal human-computer interface," *Proceedings of the International Multiconference on Computer Science and Information Technology*, pp. 451–455, 2009.

[85] P. Dalká and A. Czyzewski, "Human-computer interface based on visual lip movement and gesture recognition," *International Journal of Computer Science and Applications*, vol. 7, no. 3, pp. 124–139, 2010.

[86] O. Tuisku, V. Surakka, T. Vanhala, V. Rantanen, and J. Lekkala, "Wireless face interface: using voluntary gaze direction and facial muscle activations for human-computer interaction," *Interacting with Computers*, vol. 24, no. 1, pp. 1–9, 2012.

[87] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1:511–1:518, 2001.

[88] C. Garcia and M. Delakis, "Convolutional face finder: a neural architecture for fast and robust face detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 11, pp. 1408–1423, 2004.

[89] C.-R. Chen, W.-S. Wong, and C.-T. Chiu, "A 0.64 mm real-time cascade face detection design based on reduced two-field extraction," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 19, no. 11, pp. 1937–1948, 2011.

[90] L. Trujillo and G. Olague, "Automated design of image operators that detect interest points," *Journal of Evolutionary Computation*, vol. 16, no. 4, pp. 483–507, 2008.

[91] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[92] J. Shi and C. Tomasi, "Good features to track," *Cornell University Technical Report*, 1993.

[93] M. Y. Ivory and M. A. Hearst, "The state of the art in automating usability evaluation of user interfaces," *ACM Computing Surveys*, vol. 33, no. 4, pp. 470–516, 2001.

[94] A. Folstad, E. Law, and K. Hornbaek, "Analysis in practical usability evaluation: a survey study," *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pp. 2127–2136, 2012.

[95] C. Manresa-Yee, P. Ponsa, J. Varona, and F. J. Perales, "User experience to improve the usability of a vision-based interface," *Interacting with Computers*, vol. 22, no. 6, pp. 594–605, 2010.

[96] E. Friess, "Discourse variations between usability tests and usability reports," *Journal of Usability Studies*, vol. 6, no. 3, pp. 102–116, 2011.

[97] C. K. Coursaris and D. J. Kim, "A meta-analytical review of empirical mobile usability studies," *Journal of Usability Studies*, vol. 6, no. 3, pp. 117–171, 2011.

[98] "Iso 9241-11," *Ergonomic requirements for office work with visual display terminals (VDTs), Part 11, guidance on usability*, 1998.

[99] H. Wang, S. Z. Li, and Y. Wang, "Face recognition under varying lighting conditions using self quotient image," *Proceedings of the Sixth IEEE international conference on Automatic face and gesture recognition*, pp. 819–824, 2004.

[100] J. Molnár, D. Chetverikov, and S. Fazekas, "Illumination-robust variational optical flow using cross-correlation," *Computer Vision and Image Understanding*, vol. 114, no. 10, pp. 1104–1114, 2010.

[101] S. Calderara, R. Cucchiara, and A. Prati, "Multimedia surveillance: content-based retrieval with multicamera people tracking," *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pp. 95–100, 2006.

[102] A. Mian, "Realtime face detection and tracking using a single pan, tilt, zoom camera," *23rd International Conference on Image and Vision Computing New Zealand*, pp. 1–6, 2008.

[103] K. Yamaguchi, T. Komuro, and M. Ishikawa, "Ptz control with head tracking for video chat," *CHI 2009 Extended Abstracts on Human Factors in Computing Systems*, pp. 3919–3924, 2009.

[104] J. Francone and L. Nigay, "Using the user's point of view for interaction on mobile devices," *23rd French Speaking Conference on Human-Computer Interaction*, pp. 4:1–4:8, 2011.

[105] W. Tan and G. Rong, "A real-time head nod and shake detector using hmms," *Expert Systems with Applications*, vol. 25, no. 3, pp. 461–466, 2003.

[106] J. W. Davis and S. Vaks, "A perceptual user interface for recognizing head gesture acknowledgements," *Proceedings of the 2001 workshop on Perceptive user interfaces*, pp. 1–7, 2001.

[107] K. Grauman, M. Betke, J. Lombardi, J. Gips, and G. R. Bradski, "Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces," *Universal Access in the Information Society*, vol. 2, no. 5, pp. 359–373, 2003.

[108] S. Gustafson, D. Bierwirth, and P. Baudisch, "Imaginary interfaces: spatial interaction with empty hands and without visual feedback," *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pp. 3–12, 2010.

[109] I. Oakley and J.-S. Park, "Designing eyes-free interaction," *Proceedings of the 2nd international conference on haptic and audio interaction design*, pp. 121–132, 2007.

List of Publications

Journals

Shaowei Chu and Jiro Tanaka. "Development of a Head Gesture Interface for a Self-portrait Camera," *The Transactions of Human Interface Society*, Vol.15, No.3, 2013.

Conference proceedings

Shaowei Chu and Jiro Tanaka. "Head Nod and Shake Gesture Interface for a Self-portrait Camera," *The Fifth International Conference on Advances in Computer-Human Interactions (ACHI 2012)*, pp.112-117, Valencia, Spain, January 30 to February 4, 2012.

Shaowei Chu and Jiro Tanaka. "Hand Gesture for Taking Self Portrait," *Proceedings of 14th International Conference on Human-Computer Interaction (HCI International 2011), Human-Computer Interaction, Part II, LNCS 6762*, pp.238-247, Orlando, Florida, USA, July 9-14, 2011.