# Hand Gesture for Taking Self Portrait

Shaowei Chu and Jiro Tanaka

Department of Computer Science, University of Tsukuba,
Tennodai, Tsukuba, 305-8577 Ibaraki, Japan
chushaowei@iplab.cs.tsukuba.ac.jp,
jiro@cs.tsukuba.ac.jp

**Abstract.** We present a new interaction technique enabling user to manipulate digital camera when taking self-portrait pictures. User can control camera's functions such as pan, tilt, and shutter using hand gesture. The preview of camera and GUIs are shown on a large display. We developed two interaction techniques. First one is a hover button that triggers camera's shutter. Second one is cross motion interface that controls pan and tilt. In this paper, we explain algorithms in detailed manner and show the preliminary experiment for evaluating speed and accuracy of our implementation. Finally, we discuss promising applications using proposed technique.

**Keywords:** Hand gesture, self-portrait, human computer interaction, skin color, hand detection, fingertip detection, optical-flow.

## 1 Introduction

Digital camera is a way to take self-portraits allowing us to express ourselves through photography. Nowadays digital cameras have many functions to take self-portrait pictures, such as frontal screen [6], self-timer, face and smile detection [6], motion detection [7]. However, the main drawback of these methods is that it is hard to control the camera from a distance. Moreover the preview is too small to see from a distance. Therefore, when the user sets up the shooting parameters and timer, runs into front to the camera, while it is impossible to change the camera settings or know whether or not they are in the good view to shoot. For such reasons a common scenario is one where user runs back and forward to make sure a good shot is obtained. Besides, when using SLR (single lens reflection) camera, in case of a little bit wrong position, nearer or farther from the camera, the result image will be blured because the lens focal.

In this paper, we explore the hand tracking and hand motion gesture recognition approach to interact with the camera for taking self-portrait pictures. We apply computer vision algorithm on camera live view video to recognize user's hand gestures. The pan and tilt platform guaranties the camera will focus on the user automatically. The large display shows the augmented live view video, which gives a clear preview to user. By using hand gestures, the user can control the camera functions from a distance. Figure 1 shows the system overview.

J.A. Jacko (Ed.): Human-Computer Interaction, Part II, HCII 2011, LNCS 6762, pp. 238–247, 2011.


**Fig. 1.** The system overview (left), setup scene (middle), and augmented live view (right)

## 2 Related Work

Many innovative techniques have been proposed in the literature 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 [3]. Therefore, a number of systems have been proposed.

Chen [4] presents an optical flow with MoSIFT appearance features, to recognize gestures for controlling TV operations. The MoSIFT is computationally expensive, the authors' implementation is based on parallel processing with multi-core processors to improve the recognize latency. However, the result shows that it still requires quiet long processing time. It takes about 2.5 seconds between an activity and a result action.

Lenman [5] shows a study of using gestures interact with pie and marking menus as a remote controller to control electronic appliances in a home environment, such as TV sets and DVD players. The system recognizes 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 then tracking the movement. The problem of this approach is that it is slow and not accurate on complex background. Moreover the user needs to hold the hand pose still and frontally facing the camera for several seconds to activate an operation.

Sixth Sense [6] uses colored marker attached to fingertips and Microsoft Kinect uses depth image sensor to segment people's body, both of their approaches make the hand detection more easier. However we want to pursuit a pure vision based method that is marker-less and can easily embedded into everyday used common architecture digital cameras.

In short, most of the presented interaction techniques of hand gestures interaction have limitations for the purpose of taking self-portraits by using a digital camera. Our proposed interaction technique is application oriented designed especially for self-portrait; a higher speed hand detecting algorithm and a cross motion recognition interface are developed. By using this light-weighted algorithm, it is easier to transfer the algorithm into the camera device.

Our proposal mainly has three contributions. First, we propose a novel technique that enables user to manipulate digital camera conveniently using hand gesture especially when controlling it from a distant. Second, we developed a real-time computer vision algorithm that tracks the hand and fingertip with accuracy and high speed. Third, a cross motion interface used to recognize hand motion direction has been proposed.

## 3 System Overview

In this section we will discuss the hardware we used, and low-level image processing algorithm to detect the hand and hand motion gestures.

### 3.1 Hardware

The camera we used is Logitech Orbit AF Camera, capturing 640 x 480 resolution video with 30 FPS, 1024 x 768 for 15 FPS and 1600 x 1200 for 5 FPS, with 189-degree field of pan and 102-degree field of tilt.

The large display we used has a size of 30 inches which will give a clear augmented live view from a distance. We use down sized 320 x 240 resolution live view video applies computer vision algorithm, 640 x 480 resolution for live view show and 1600 x 1200 resolution for taking shots.

### 3.2 Hand Contour Segmentation and Fingertip Detection

The hand tracking applies low-level image processing operations on each frame of video in order to detect the locations of the fingers. While a model-based [2] approach that uses temporal information could provide more robustness to situations such as complex backgrounds, has been implemented. Our algorithm of hand detection is described in the following:

Given a captured image, every pixel is categorized to be either a skin-color pixel or a non-skin-color pixel. We use skin-color detection algorithm, described in [1], to segment the skin-color pixels. After segmenting, an amount of noise pixels in the image is inevitable, we apply median filtering [8] with 3 x 3 kernel to remove extraneous noise. Then we apply single connected component contour finding algorithm (implemented in OpenCV [9]) to locate hand contours. We abandon the small area contours, which are less than 600 pixels, and leave only larger ones as hand candidates.

After we obtain the contours, the next job is to detect the fingertips. The fingertips are detected based on the contour information by using a curvature-based algorithm similar to the one described in [2]. The algorithm can detect the fingertips and finger orientation base on 2D hand silhouette. Figure 2 shows the processing of hand fingertips detection from the original image to hand fingertips detected result image. During this process, because the color of the face is also skin-color, we have detected the face as well; this will be useful for face detection and recognition in our future work.

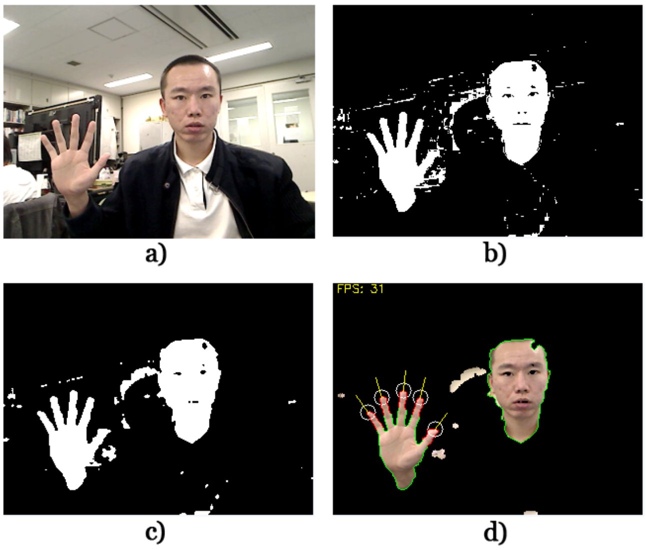


**Fig. 2.** Hand fingertips detection procedure. a) original image. b) skin-color segment. c) median filtered. d) detected fingertips with colored marker.

### 3.3 Motion Gesture Recognition

In this section we will describe a motion recognition interface designed for recognizing 4 motion direction gestures, wave UP, DOWN, LEFT and RIGHT, by using the optical-flow measurements. The motivation of developing such interface is that when the user moves his/her hand with a certain speed, the hand image will blur and it is very difficult to detect the fingertips. The optical-flow [9] denotes the movement of an image patch between two frames of a video sequence that can measure the motion gesture even with high speed movement which is suitable for our scenario. There are various techniques for estimating the optical-flow [11] [12], and it is proved efficient [10] for gesture recognition.

Calculating the optical-flow in real-time for the whole image at 320 x 240 resolution might require a lot of computing power. We restrict the optical-flow measurement with limited feature points (29 points) and within a small region. The feature points movement will be extracted in each frame and calculating the mean value of both orientation and speed. The noise of small and large movement of feature points will be cut off, leaving only reliable movement among them. In order to recognize the gestures we analyse the pattern of feature points movement in video frame sequence, from no movement to movement, and to no movement, then distinguish a specific motion gesture.

Optical-flow estimates are often very noisy; we add 29 feature points, and calculate the main motion measurements using mean orientation value, the result shows it is increase to indicate the expected gesture by adding more points. The layout of feature points affects the optical-flow measurement also; we apply a circle-like layout which proves effective for recognizing four motion direction. Figure 3 shows a frame sequence in video time line, that to recognizing a RIGHT hand motion.

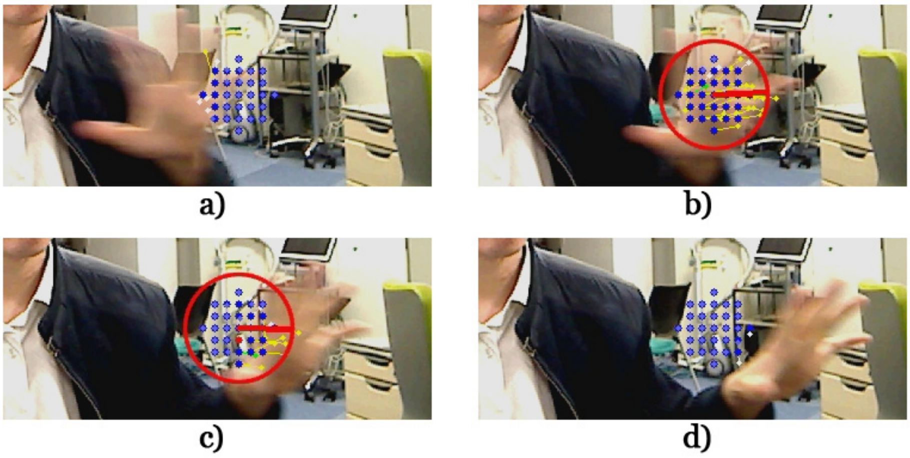

**Fig. 3.** The procedure of motion gesture frame time line. a) frame t, no motion detected. b) frame t + 1, motion orientation detected, the clock and arrow indicate the motion orientation is right. c) frame t + 2, motion orientation is right. d) frame t + 3, no motion detected.

## 4 Interaction Techniques

We use the large display to give augmented live view, including the highlighted color marker of hand, fingertips and graphical interface, as in Figure 4. Two interfaces have been implemented.

### 4.1 Hover Button Interface

The hover button is a fingertip air touch interface to activate the camera shutter. On the large display the detected hand will be marked with green contour pixels, the fingertips will be highlighted with red pixels. User can put out any fingertip into the region of shutter button to activate the button. Because of the noise of detection and cluster background, (s)he needs to put the fingertip into the region of button steady for a specific period, one second, to activate the shutter button. After the shutter is activated, a big colored number on large display will be shown counting down from 3 to 1, which indicates the user should prepare for a pose, and taking a shot ( see Figure 1 b) ) .

### 4.2 Cross Motion Interface

The second interface is a motion based interface as described in section 3.3, which is used to adjust the camera's pan and tilt. We designed a GUI for cross motion interface (see Figure 4, Cross motion interface). The four direction arrows indicate it can recognize four motion directions, pan left, right and tilt up, down. When user uses a hand to make a cross motion within a specific short period cross the interface, then the hand motion direction, UP, DOWN, LEFT and RIGHT, can be recognized by using optical-flow measurements described in section 3.3. The camera pan & tilt operations will function according to the hand motion direction. To make the interface more robust, which prevents the camera from moving in a disorderly manner, the gestures of slow motion and move continually within the interface region will be excluded. For 20 milliseconds after performing a motion, the user needs to stand still (no motion on the interface). After that, a pattern of motion will be recognized and the pan & tilt operations will be functional immediately. Figure 3 shows one of RIGHT motion gesture. Figure 4 shows the augmented live view image including two GUIs.

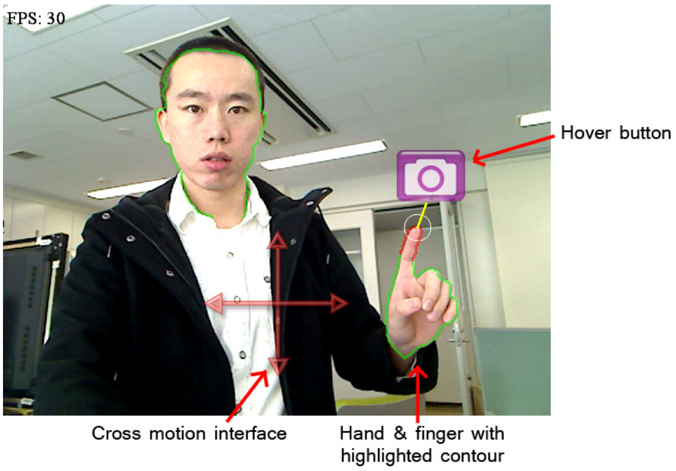


**Fig. 4.** Augmented live view, mixes with colored hand marker and GUIs

## 5 Performance Evaluation

### 5.1 Performance of Hand Fingertip Detection and Optical-Flow

We evaluated two algorithms with regard to speed and accuracy. The experiments were performed on a computer with a 2.5GHz CPU, using USB 2.0 Pan & Tilt camera.

Table 1 shows the process time of hand fingertip detection. The result shows that the hand fingertip detection algorithm can run in real-time with about 180 FPS. Optical-flow measurement of cross motion interface is less than 3 milliseconds.

**Table 1.** The process time of hand fingertip detection with video resolution at 320 x 240

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

### 5.2 Accuracy of Hand Tracking and Cross Motion Recognition

The goal of this evaluation is to test the distance where the hand tracking and cross motion interface work well.

For testing the hand tracking, we asked the users to put the hand out in front of the camera to see whether or not the hand can be detected. By various people test the algorithm works well from 0.5 meter to 2 meters away from camera. For testing the accuracy of cross motion interface, we asked users to perform gestures and to see whether they are correctly recognized. Ten volunteers participated in the experiment. The results are shown in tables 2, 3 and 4. Left column of the table means user performed gestures, corresponding row is result of recognized gestures, on average percentage.

**Table 2.** Cross motion interface accuracy (distance 0.5 meter)

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

**Table 3.** Cross motion interface accuracy (distance 1.2 meters)

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

**Table 4.** Cross motion interface accuracy (distance 2 meters)

| Gesture | LEFT | RIGHT | UP | DOWN | NONE |
|---|---|---|---|---|---|
| LEFT | 52% | | | | 48% |
| RIGHT | | 64% | | | 36% |
| UP | | | 85% | 1% | 14% |
| DOWN | | | 3% | 88% | 9% |

## 6 Discussion

### 6.1 Limitations

We have shown the accuracy and effectiveness of our proposed gesture interfaces. However a few limitations do exist.

First, lighting sensitivity is a main weakness for skin-color based hand detection. Outdoors or in highly reflective sunshine conditions, the human skin-color will change to white color, what will cause difficulty to segment. Although many researchers have used uniform background segment [2], color histogram [15], flocks of features [13] to improve the hand tracking, these methods are either slow or less accurate. Therefore we choose another method to adjust camera optics parameters, Exposure, Gain, Brightness, Contrast, Color Intensity, etc., which maintains the video image with uniform balanced lighting condition. Moreover, fortunately, modern camera have self-adaptive functions to adjust them automatically or manually.

Second, the distance from the user to the camera is another issue; the user needs to stand within 2 meters from the camera, otherwise it will be difficult to detect the hand. The cross motion interface also has this problem.

Third, the optical-flow measurements need a cluster background; if the image region on the optical-flow cross motion interface is a completely uniform color, the optical-flow measurements will inaccurate. Such situations occur when the user worn a uniformly colored clothes.

### 6.2 Possible Applications

Our proposed two gesture based interface can be applied to various applications.

The hand and finger detection algorithm can be widely used in vision based human computer interaction. For example, for HMD and Augmented Reality, the hand gesture recognition is very practical to use for interacting with virtual objects and interface. The finger orientation can do distance pointing as described in [16]. Two hand gestures interaction can also be developed (See Figure 5).

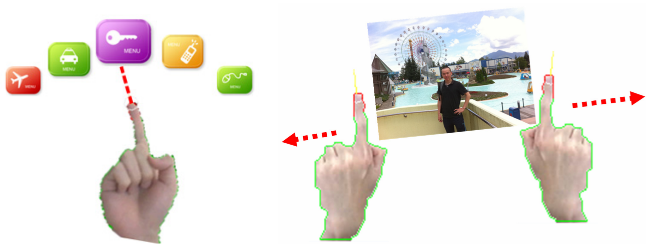

**Fig. 5.** Finger orientation interface and two hand manipulation of picture

The motion based gesture recognition is also a novel approach to recognize the human hand gestures. All of the gestures represent a sequence of motion in video. By carefully analyzing the motion frame sequence in video time line, specified gesture can be recognized. We developed a motion based interface and introduced it into our application for controlling the pan and tilt, and we obtained good results. However, this is just a primary attempt, numerous interface can be implemented, such as the slider bar, clock move interface, button. Moreover, if we set the interface on the region of the face, then the head shake and nod gesture can be recognized as well.

## 7 Conclusions and Future Work

In this paper we presented the hand gestures for taking self-portraits. The camera's pan, tilt and shutter can be controlled by hand gestures. We use skin-color and k-curvature algorithm to detect the fingertip, optical-flow measurement to recognize the hand motion gesture, both of them running with accuracy and high speed. By using our system, the user can control the self-portrait process from a distance. Our result showed that hand gesture is promising interface as a means of manipulating digital camera remotely.

In the future, we have plan to support more functionalities with a SLR camera and a more accurate Pan and Tilt platform for our experiment. We also want to apply face tracking and recognition techniques to self-portrait system which will be benefit for user tracking and multi-users scenario. Finally, new interaction techniques that can work outdoors, with small or screen-less live view cameras, have also come into consideration.

## References

1. Kovač, J., Peer, P., Solina, F.: Human Skin Colour Clustering for Face Detection. In: IEEE EUROCON 2003: International Conference on Computer as a Tool, vol. 2, pp. 144–148 (2003)
2. Malik, S., Laszlo, J.: Visual Touchpad: A Two-handed Gestural Input Device. In: The ACM International Conference on Multimodal Interfaces, pp. 289–296 (2004)
3. Mistry, P., Maes, P.: SixthSense: a wearable gestural interface. In: ACM SIGGRAPH ASIA 2009 Sketches, pp. 11:1–11:1 (2009)
4. Chen, M.Y., Mummert, L., Pillai, P., Hauptmann, A., Sukthankar, R.: Controlling Your TV with Gestures. In: MIR 2010: 11th ACM SIGMM International Conference on Multimedia Information Retrieval, pp. 405–408 (2010)
5. Lenman, S., Bretzner, L., Thuresson, B.: Using Marking Menus to Develop Command Sets for Computer Vision Based Hand Gesture Interfaces. In: Proceedings of the Second Nordic Conference on Human-computer Interaction, pp. 239–242 (2002)
6. Sony Party Shot (2009), http://www.sony.jp/cyber-shot/party-shot/
7. Samsung Self-Portrait Camera: TL225 (2009), http://www.samsung.com/us/photography/digital-cameras/
8. Pereault, S., Hebert, P.: Median Filtering in Constant Time. IEEE Transactions on Image Processing 16, 2389–2394 (2007)
9. OpenCV: Open Source Computer Vision, http://opencv.willowgarage.com/wiki/
10. Zivkovic, Z.: Optical-flow-driven Gadgets for Gaming User Interface. In: International Conference on Entertainment Computing, pp. 90–100 (2004)
11. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: 7th International Joint Conference on Artificial Intelligence, vol. 2, pp. 674–679 (1981)
12. Beauchemin, S.S., Barron, J.L.: The Computation of Optical Flow. ACM Computing Surveys, 433–467 (1995)
13. Kölsch, M., Turk, M.: Fast 2D hand tracking with flocks of features and multi-cue integration. In: Conference on Computer Vision and Pattern Recognition Workshop, vol. 10, p. 158 (2004)

14. Letessier, J., Bérard, F.: Visual tracking of bare fingers for interactive surfaces. In: Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST 2004), pp. 119–122 (2004)
15. Lee, T., Höllerer, T.: Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking. In: International Symposium on Wearable Computers (IEEE ISWC), pp. 11–13 (2007)
16. Wang, F., Cao, X., Ren, X., Irani, P.: Detecting and Leveraging Finger Orientation for Interaction with Direct-Touch Surfaces. In: Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST 2009), pp. 23–32 (2009)