Head Nod and Shake Gesture Interface for a Self-portrait Camera

Shaowei Chu
Department of Computer Science
University of Tsukuba
Tsukuba, Ibaraki, Japan
chushaowei@iplab.cs.tsukuba.ac.jp

Jiro Tanaka
Department of Computer Science
University of Tsukuba
Tsukuba, Ibaraki, Japan
jiro@cs.tsukuba.ac.jp

***Abstract* – Interactive interfaces and applications are a flourishing research area. In this paper, we introduce a head gesture interface for a digital camera shooting self-portrait pictures. Natural head nodding and shaking gestures can be recognized in real-time, using optical-flow motion tracking. A double head nod triggers the camera shutter to take shots. Continuous nodding or shaking triggers a zooming interface to zoom the user's face in or out. To make the recognition robust, a safe zone analysis of the head region was conducted to quickly exclude any insignificant head motion, and thresholds of moving direction and length of head motion were selected in a preliminary set-up step. A finite state machine was used to recognize head gestures. Our results show that the proposed head gesture recognition method is a promising interface for a self-portrait camera.**

Keywords-head gesture; self-portrait; human computer interaction; optical-flow; motion tracking.

I. INTRODUCTION

Due to advances in digital cameras, including those in certain smart phones and foldable Liquid-Crystal Display (LCD) screens, taking self-portraits has become much easier. However, a user must physically touch the camera to change the frame or any settings, or in some cases, can use a remote control, but this can result in unnatural poses of the hands in pictures.

Figure 1. With hand gestures, a user can interact with a camera to trigger the shutter and take self-portraits.

We believe that one of the next major trends in the advancement of camera design will be making it more interactive, responsive, and accessible to the user. Applying face, smile, and motion detection [10] functions to a camera is a good step, but does not fully satisfy users because of the modest degree of interaction. In our previous work [1], we proposed hand gestures for self-portrait photos, making the camera more interactive (Fig. 1), but this had limited success. Obviously, the system used in that study (and shown in the figure) is far too large to be incorporated into a portable digital camera; it requires a large display to provide a live view, where the user can see her/his gestures, visual tooltips, and a GUI. In practical applications, the screen of a camera will always be small, making it difficult or impossible to see such details.

Thus, we propose a head gesture interface, which unlike hand gestures, has no strict requirement for visual feedback and thus a small screen is acceptable (as shown in [3]), and it works well even with no feedback device (as shown in [6]). Head gestures can express clear meanings (e.g., a nod means yes and a shake means no), which are difficult to achieve using hand gestures. Moreover, when designing a zooming interface for self-portrait photos, it is difficult to develop a hand gesture-based interface, because the hands may extend outside the camera view when a zooming-in function is executed.

In this paper, we present and implement a head gesture interface, which triggers the camera shutter with a natural double nod and controls the zoom function with continuous nodding and shaking gestures (see Fig. 2).

Figure 2. Using the self-portrait application outdoors.

A Canon 60D digital camera was used, which provides hardware-supported face detection with a 30 FPS at 1056 × 704 resolution video stream. The LCD screen is used as a front-facing screen to provide a live view to the user. For head gesture recognition, a safe zone analysis of the face

 ISBN: 978-1-61208-177-9

region is first conducted to quickly exclude large head motions. The features inside the face region are extracted only when the head motion is minor and is restricted to the safe zone and within a predefined period parameter (currently 500 ms). Then, after the features of the face region are selected, feature tracking is performed in each consecutive frame, and feature motion is recorded to compute the 2D head motion direction and length. Afterwards, user head motion data, in terms of direction and length, are collected through a user test step. Finally, based on the pattern analyzed from the user data, a Finite State Machine (FSM) is used to recognize the head gestures. In our implementation, head nods and shakes can be counted. The proposed interface was positively received in various user experiments.

The rest of this paper is organized as follows. Section 2 introduces related work on head gesture interfaces, Section 3 discusses our implementation, Section 4 introduces the interface design for self-portraits, Section 5 discusses preliminary user experiments and results, and Section 6 provides conclusions, a summary of the proposed interface, and possible future work.

II. RELATED WORK

Vision-based gesture recognition is believed to be an effective technique for human–computer interaction, as presented in [11]. Some researchers have tried to apply hand gesture with digital cameras for taking self-portraits [1], and have had significant success. The benefits of such an approach are obvious: no additional devices or refitting of the camera are required. However, in particular cases, hand gestures may not be appropriate for self-portraits. For example, when zoomed in, the hands may extend outside the field of view. As a result, we explored a head gesture interface, because the face will always be within the camera view when taking a self-portrait. Furthermore, various studies of head gesture recognition [3][4] and head gesture interfaces [2][6] have shown that a head gesture interface is promising for self-portraits.

To date, there are two main approaches for recognizing head gestures. The Hidden Markov Models (HMM) method uses a pre-training process to collect user data, and then a pattern-recognition algorithm to distinguish between specific gestures [4]. A problem with this approach is that pre-training is required and there is a limited number of head gestures with two delayed digital outputs (nod or shake); in addition, few studies have achieved a recognition rate better than 85%. The other approach uses an FSM-based recognition technique [2][3] to explore temporal information on head motion in each video frame, and switches the states between head moving tendencies. Advantages of this method are that it involves less intense computing than HMM, and an adaptive threshold can be set in user experiments. We believe that a properly constructed FSM is the best option for this task.

In the present work, we provide a precise estimation of head motion direction and motion length in each frame from the video stream with about 50 features. In addition to template-matching of feature points [2], we apply an optical-flow [7] measurement to tracking the motion of the points. Then we divide the tracking region into four sections (Up, Down, Left, and Right) using data collected from experiments. The FSM recognition is constructed based not only on timing and motion direction, but also the motion length. This implementation provides precise results requiring little computation.

We demonstrate that head gestures may be a promising interface for self-portraits. Our proposed method is just a beginning in this area of study.

III. GESTURE RECOGNITION

Our implementation of a head gesture recognition algorithm must meet important requirements for real-time responsive applications: automatic initialization, sufficient sensitivity to recognize natural gestures, and an instantaneous and real-time response with feedback to the user. These requirements guided the development of our head gesture recognition scheme.

A. Camera Device and Face Detection

Gesture recognition of head motion is based on face detection. In addition to implementing a face detection algorithm, we chose the Canon 60D camera, which offers SDK for developers, including embedded hardware support for face detection (Fig. 3).

Figure 3. Canon 60D camera with foldable LCD screen and face detection.

The camera runs at 30 FPS with 1056 × 704 video image sequences for live view and has a foldable LCD screen.

B. Safe Zone to Exclude Large Movements

After obtaining an image with a face region from the video stream, the image is processed to recognize head gestures. The first task is to define a safe zone that excludes large head motions that we assume are not nod or shake gestures. A safe zone, 40% larger than the face region, is initialized based on the face region in the image; then, in the following frames the system needs to check whether the face moved out of that safe zone. If so, then it is considered a large or fast motion, not a nod or shake gesture. The safe zone is then re-initialized and a new check begins. If not, then the safe zone remains as it is and the system waits for the next frame and face region to check again for head motion. When the face does not move out of the safe zone

within a specific period (currently, 500 ms), then we assume the head is still and it may be a good time to recognize head gestures. See Figure 4.

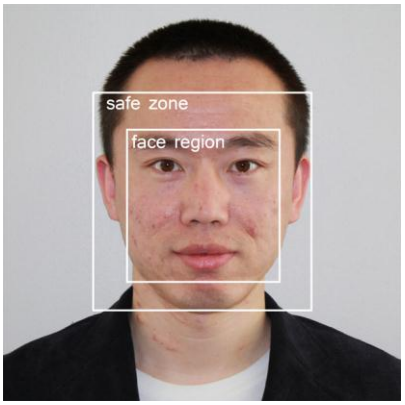

Figure 4. Face region and safe zone.

The safe zone is a key mechanism in the automatic initialization principle mentioned at the beginning of Section 3. Tracking, described in the next section, is reinitialized when the face moves out of the safe zone.

C. Feature Points Tracking and Motion Calculation

The features and their motion are calculated in consecutive images from the camera at 30 FPS, and the safe zone of the face is scaled to a 140 × 140 resolution size to perform feature extraction and tracking. There are two reasons for this. First, it identifies the face region size according to person and circumstantial context, such as user distance to the camera, helping to achieve a domain-independent model theory [4]. Second, it allows the calculation to be steady and fast for real-time application.

Within the face safe zone, we extract image features for tracking head motion. Several feature-extracting algorithms are known [8][9], but we chose a fast extracting method derived from the Hessian matrix, and selected the top 50 feature points as good features to track, as defined by Shi and Tomasi (S-T) features.

The Hessian-defined features rely on a matrix of the second-order derivatives (∂2x, ∂2y, ∂x, ∂y) of image intensities. These are used because they are not sensitive to light. For each pixel point (x, y) in a second derivative image, the autocorrelation matrix over a small window around it is calculated, as follows:

$$M(x,y)=\begin{bmatrix} \sum_{-K\le i,j\le K} w_{i,j} I_x^2(x+i,y+j) & \sum_{-K\le i,j\le K} w_{i,j} I_x(x+i,y+j) I_y(x+i,y+j) \\ \sum_{-K\le i,j\le K} w_{i,j} I_x(x+i,y+j) I_y(x+i,y+j) & \sum_{-K\le i,j\le K} w_{i,j} I_y^2(x+i,y+j) \end{bmatrix}, \quad (1)$$

where $w_{i,j}$ is a weighting term that can be defined as uniform or used to create a circular window around a pixel, and I is the intensity of a pixel. Good S-T features are found and placed in the image where the autocorrelation matrix of the second derivatives has two large eigenvalues of the matrix.

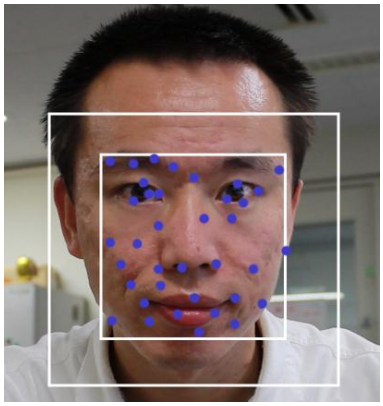

Figure 5. Numerous feature points are extracted.

After features are extracted (Fig. 5), a Lucas-Kanade [7] optical-flow measurement is taken to track each feature point's motion. The Lucas-Kanade method assumes that the displacement of the image contents between two nearby instant frames is small and approximately constant within the neighborhood of the point P under consideration. Thus, the optical flow equation can be assumed to hold for all pixels within a window centered at P. Moreover, by tracking the features with a pyramid layer of images, precise measurement of the velocity of feature motion can be achieved. This method functions well for tracking S-T features in the image. The optical-flow measurement results are feature points in the current frame's displacement from the previous frame. By calculating each feature's displacement, the motion length and direction of each feature point becomes clear.

Because of the error rate in optical-flow tracking, certain points will report a wrong result or be lost to tracking during head motion. Increasing the number of feature points is helpful to obtain reliable motion information. We calculate a set of feature points within the face region and mean values of length and direction as the main parameters in each frame.

It takes approximately 2.6 ms to extract the top 50 feature points and about 3.4 ms to track them using optical-flow measurements, which will suffice for real-time applications.

D. User Experiments

A user test step is applied to obtain head motion direction and length in frames while the user nods and shakes his/her head. The collected data are used to design the threshold of gesture recognition. The task is straightforward; users are asked to keep nodding or shaking during a period, and the moving direction degree and motion length in each frame with time is recorded.

A graduate student took part in this experiment to collect data. Measurements of the degree of motion begin at the top-right corner of the user's face as it appears in the display (Fig. 6). Figure 7 shows selected sequence data of nodding

 ISBN: 978-1-61208-177-9

and shaking gestures with motion degree (right vertical coordinate), length (left vertical coordinate), and timing (horizontal coordinate).

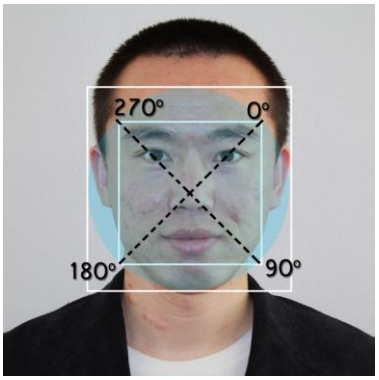

Figure 6. Motion direction angle coordinate.

Two important pieces of statistical information can be obtained from the data. First, when nodding (Fig. 7a), the motion directions are between 90 ° and 180 ° when tilting the head down and 270 ° and 360 ° when tilting the head up. In contrast, the motion directions in the shaking gesture (Fig. 7b) are smooth and steady along the 45 ° and 225 ° line. Second, the motion length data change periodically during the head gesture, but below a peak value of 6.

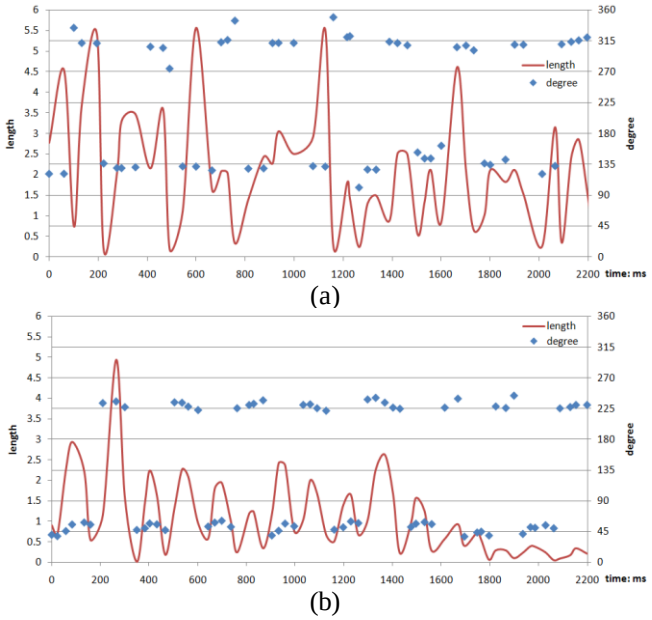

Figure 7. Nodding (a) and shaking (b) gesture data.

The time interval of up-down and left-right movement was 122.2 ms and 137.5 ms, respectively.

E. Recognition Design

Based on the data analyzed above, we can conclude that head shaking is a more steady motion than nodding. Thus, we separated the motion regions for moving direction recognition into four regions (Fig. 8):

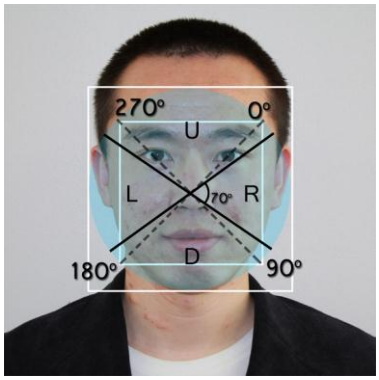

Figure 8. Motion direction region (U: Up, R: Right, D: Down, L: Left).

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 motion length value must be larger than 0.5 and less than 6.0 in the recognition procedure.

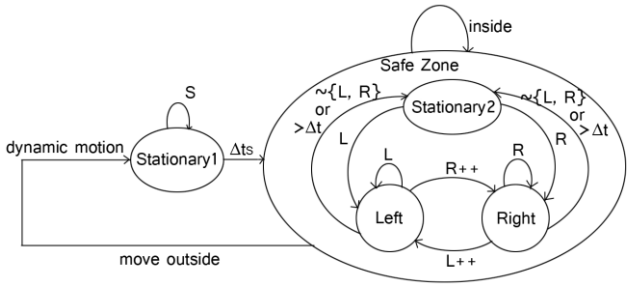

Figure 9. Finite State Machine for recognizing head shakes.

The timing-based FSM described in [2] was used to recognize the gestures. Figure 9 shows a transition chart for head shaking. It contains two main states: Stationary1, a motionless state, and Safe Zone, when the face is within the safe zone discussed above for a certain period of time. Inside the Safe Zone state, there are Stationary2, Left motion, and Right motion. The transition from Left to Right or reverse transition will add one factor to the shaking count.

F. Performance and Implementation

To increase the speed and efficiency of image processing, we used OpenCV [5] to implement the algorithm. The program was written in C++ with multi-threading. Image drawing was under Direct2D API support. The processing performance is shown in Table 1, based on an Intel Core 2 Quad CPU 2.5 GHz PC.

 ISBN: 978-1-61208-177-9

TABLE 1. IMAGE PROCESSING PERFORMANCE

Image processing	Process time (ms)
Resize face region	0.2
Extract features	2.6
Feature tracking	3.4
Gesture recognition	2.6

Note that feature extraction is performed on when initializing tracking; it is not executed in each frame. In addition, as shown in Table 1, in each frame with a 140 x 140 resolution safe zone, the process of tracking features requires the most computation time. Thus, one way to reduce the calculation time would be to further scale down the size of the safe zone. Finally, the gesture-recognition task, which calculates the motion of every feature point, depends on the number of points: tracking fewer points requires less time.

The Canon 60D camera is connected to the computer through a USB cable. Currently, the Canon 60D camera does not support programming to configure its foldable frontal screen. Thus, we developed an iPhone application to show the live view video and GUI. The iPhone application runs at about 13 FPS with a 360 × 240 resolution motion jpeg image through a WiFi connection.

IV. INTERFACE DESIGN FOR SELF-PORTRAITS

The original goal of this study was to design a natural head gesture interface for self-portrait photos. From the recognition procedure and the results described above, we designed an application based on counted nods and shakes. On the frontal screen, the user's face region and safe zone are circled with lines. Graphical tooltips for the number of nods and shakes are drawn in the top area above the face region instantaneously when the user performs a gesture.

The double nod gesture (performing the nod gesture twice) triggers the camera shutter immediately. After the shutter is triggered, the self-timer is activated and a countdown (set at 5 s, because it takes 2–4 s to drive the mechanical lens and run the auto focus on the user's face) is shown on the front-facing screen. At the same time, the camera adjusts the lens, focuses, and opens a flash when needed. The user can prepare her/his pose during this period and get ready for the self-portrait.

Continuous nodding or head shaking (when the gesture is performed three or more times) triggers the zoom function, where nodding zooms in and shaking zooms out. This function is performed with smooth gesture transitions.

Figure 10. Canon 60D camera with an iPhone as a front-facing screen.

The camera can be set on any steady object (e.g., by using a gorillapod [12]) or on a tripod indoors or outdoors. An iPhone was attached to the camera as a front-facing screen (Fig. 10).

V. DISCUSSION

Compared to our previous work on hand gestures, the head gesture interface is more suitable for providing a zoom interface. As mentioned in Section 1, it is not practical to use hand gestures for zooming. In addition, head gesture recognition is independent of user distance to the camera, which simplifies gesture recognition. Moreover, the implementation of the recognition algorithm does not vary depending on lighting conditions; it works well under any light conditions. Furthermore, using a frontal screen, the system is portable and can be taken outdoors.

While detailed user experiments have not yet been performed, informal user feedback from students at our university has been very positive. After a very brief introduction to the system, users were free to explore the system on their own. Feedback from students not specializing in computer science was more positive; they found the gesture-based manipulations to be intuitive and understandable, with descriptions such as "accessible" to describe the overall system and the general idea. The nod/shake counts shown above the face region of the display made it simple for users to understand, and they became familiar with the routine of recognition based on the visual tooltips.

The distance to the camera was an issue raised by one person; although our implementation is completely distance-independent, when a user is a significant distance from the camera, it obviously becomes difficult to see the preview on the frontal screen clearly. However, when indoors, the system can be plugged into a larger display, as we did in [1].

The automatic face tracking and zooming-in to the face region may not satisfy a user who wishes to have full-body pictures. In such a case, a pan and tilt platform could be used, and the zooming-in function could zoom in on the center of the image rather than on the face region.

The issue of taking profile pictures was also raised. This can be done by facing the camera and performing a double nod to trigger the shutter, then turning into the profile position and waiting for the picture to be taken. Another possibility would be to modify the system so that it can estimate the head pose based on eye location [3], and the adjust the thresholds of motion direction for profile nod and shake gestures.

For pictures of multiple users, a face recognition technique could be applied to a main user only, and functions could be triggered based only on that person's gestures.

One participant suggested that our interface could be useful for handicapped persons, who could take hands-free self-portraits.

VI. CONCLUSIONS AND FUTURE WORK

We presented a head gesture interface for self-portraits. The shutter is activated by double nods, and zoom functions

 ISBN: 978-1-61208-177-9

are performed by continuous nods and shakes. In the gesture recognition procedure, a safe zone is used to exclude large, irrelevant motions. Numerous feature points are extracted and tracked based on optical-flow measurements, and FSM is used to recognize head gestures. Our system runs in real-time, counting nods and shakes. We performed trial runs and discussed the primary results, which showed that this method is a promising interface for self-portraits.

In the near future, flexible and context-based interfaces will be integrated into the system to support more configuration functions of the camera. In addition, more user experiments will be conducted to refine the system.

REFERENCES

[1] S. Chu and J. Tanaka, "Hand gesture for taking self portrait," HCI International 2011, Human-Computer Interaction, Part II, LNCS 6762, pp. 238-247, Orlando, FL, USA, July 9-14, 2011.

[2] J. 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, Orlando, FL, USA, 2001, doi: http://doi.acm.org/10.1145/971478.971504.

[3] 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, Singapore, September 10-12, 2007, doi: http://doi.acm.org/10.1145/1378063.1378157.

[4] J. Lee and S. Marsella, "Learning a model of speaker head nods using gesture corpora," Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 289-296, Budapest, Hungary, July 9-14, 2009.

[5] Open Source Computer Vision Library (OpenCV): Last visited on November 22nd, 2011. http://opencv.willowgarage.com/wiki/

[6] 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, Singapore, 2007, doi: http://doi.acm.org/10.1145/1328491.1328511.

[7] J. Bodily, B. Nelson, Z. Wei, D, Lee, and J. Chase, "A Comparison Study on Implementing Optical Flow and Digital Communications on FPGAs and GPUs" ACM Transactions on Reconfigurable Technology and Systems, vol. 3, Issue 2, pp. 6:1-6:22, 2010, doi: http://doi.acm.org/10.1145/1754386.1754387.

[8] L. Trujillo and G. Olague, "Automated design of image operators that detect interest points" Journal of Evolutionary Computation, vol. 16, Issue 4, pp. 483-507, 2008, doi: http://dx.doi.org/10.1162/evco.2008.16.4.483.

[9] H. Bay, T. Tuytelaars, and L. Gool, "SURF: speeded up robust features," Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359, 2008.

[10] Casio TRYX camera: Last visited on November 22nd, 2011. http://exilim.casio.com/digital_cameras/TRYX/TRYX.

[11] M. Pranav and M. Pattie, "SixthSense: a wearable gestural interface," ACM SIGGRAPH ASIA 2009 Sketches, pp. 11:1-11:1, 2009, doi: http://doi.acm.org/10.1145/1667146.1667160.

[12] Joby Gorillapod: Last visited on November 22nd, 2011. http://joby.com/gorillapod.

 ISBN: 978-1-61208-177-9