Camera-based Flicking Gesture Recognition and Game Applications

Toshiki Sato, Kentaro Fukuchi and Hideki Koike
Graduate School of Information Systems
The University of Electro-Communications
Chofu-shi, Tokyo 182-8585 JAPAN
+81-42-443-5653
{den,fukuchi,koike}@vogue.is.uec.ac.jp

FINGER FLICKING GESTURE

Conventional interfaces of power adjustment for video games employ common controllers (e.g. game pad) or dedicated input devices. For example, in a golf game, a player presses a button of a controller several times to shoot. Typically the pressure power on the button is ignored. However, the feeling from this action is very different from an actual golf swing. Since it is preferred that a video game provides a real and intuitive interaction, this input is not desirable.

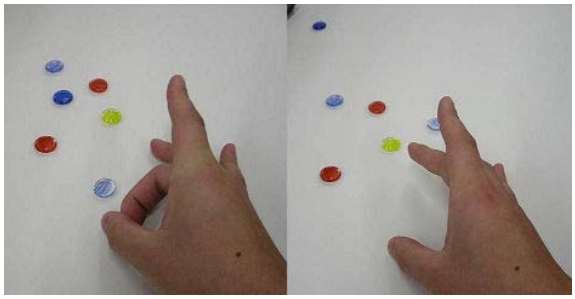

Figure 1: Finger flicking: at first put a forefinger or mid finger on a thumb and add power to the finger (left), then release the finger (right).

Finger flicking, by the way, is a familiar action to us, as anyone who had played marbles or flicked coins. We consider that finger flicking has the same property with golf swing or hitting in respect of action that hits something to a distance. Hence, we propose to use the finger flicking for power adjustment, which can be useful for video games. We named this flicking gesture interaction "*OHAJIKI interface*"[1]. [1]

VISION-BASED FLICKING GESTURE RECOGNITION

It is difficult to put some sensors on a finger to track a quick motion of finger flicking gesture, because the size of the sensor is limited to the size of the finger and it may be different between users. Besides, because it is a very quick motion, the sensor has to be fitted to the finger tightly, and its size and weight should be as small and light as possible. For that reason, we used a vision based recognition technique to track finger flicking motion and measure the position and direction of a hand. We hypothesized that the power of the finger flicking can be recognized by measuring the speed of the finger motion which can be tracked by vision based techniques. However, its accuracy and scan rate are worse than the result by the hardware based recognition typically, because of the low frame rate of the camera. Because of that, it was difficult to track a quick motion by a camera. To solve this problem, we employed a high-speed camera and developed a high throughput image processing system to track a quick finger motion.

IMPLEMENTATION

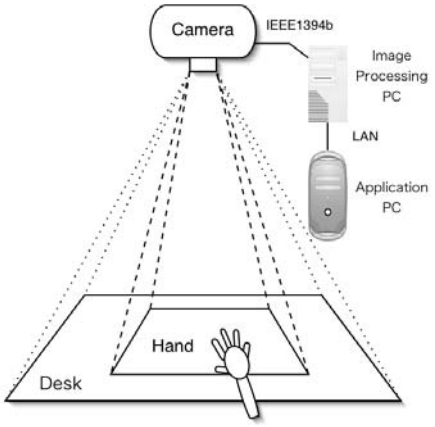

Figure 2: System overview. This system consists of two PCs. An image from the camera is processed by the image processing PC, and the results are sent to the application PC via Ethernet.

Recognition Process

We used a Point Grey Research's high-speed camera, Dragonfly Express, that has several scan modes and it can increase its scan rate by limiting its scan range (see table 1). In order to select an appropriate scan mode, the recognition process is divided to the following three phases.

1. Hand Detection Phase
   At first, the system scans the table to find whether a user is going to flick with his finger or not. To recognize the gesture, the system detects a user's stance of flicking. We

UIST '06, October 15–18, 2006, Montreux, Switzerland

[1]In Japan, playing marbles, or "ohajiki", is a popular game.

defined this stance that it makes a small circle region by a thumb and a flicking finger as seen in figure 3. The system detects this small circle region using a background subtraction and a region extraction process. When the region is detected, the system goes into the next phase.

2. Search Window Setting and Angle Calculation Phase
   Next, the system sets a search window to track a flicked finger, and calculates the flicking direction based upon the positions of the arm and the circle region. When the user keeps his hand still for a while (0.5 second), the system goes to the next phase. This dwell time is needed to change the scan range of the camera.

Figure 3: Left: Circle region and search window position. The window is set to the position that moving finger passes through its inside. Right: Flicking angle is based on the position of circle and hand region, or a virtual marble and circle position.

3. Fingertip Detection and Result Calculation Phase
   The system changes its scan range to the search window and tracks the finger motion inside the window. This process uses background subtraction and template matching that uses a small disc template to detect a fingertip. When the fingertip runs out from the window, the system stops tracking and calculates the velocity of the finger from the result.

   The velocity of the finger is calculated by the following steps: let T be a length of time (msec) from the first detection of a finger to the last, and L_n represents the distance in pixels between positions of fingers of n and $n+1$, and N is the number of detected fingertips. The velocity of the finger V (pixels/msec) is calculated as $V = \frac{\sum_{i=1}^{N-1} L_i}{T}$

Table 1: Frame rate for each phase of recognition

Phase	Image size	Frame rate	Scan rate
stance detection	640×480	200 fps	160 fps
window setting	320×240	320 fps	130 fps
fingertip detection	32×32	630 fps	630 fps

Applications

We developed two applications using OHAJIKI interface.

3D Golf Swing Game

We developed a simple 3D golf game that a player can shoot a ball by a finger flicking gesture. A player controls the direction of the shot by moving his hand on the input surface, and then controls the power of the shot by finger flicking (the left of figure 4).

Figure 4: Left: Playing 3D Golf Swing Game. Right: Playing Tabletop Marble Game.

Tabletop Marble Game

While the golf game application had a separated game display from the input surface, the second application, a tabletop virtual marble game system, allows a user to flick a virtual marble projected on the table directly with his finger. Figure 2 shows the hardware structure of the system.

In this implementation, the position of the search window is set over the virtual marble on the table, and the direction of the flicking gesture is estimated from the positions of the virtual marble and the center of hand circle region.

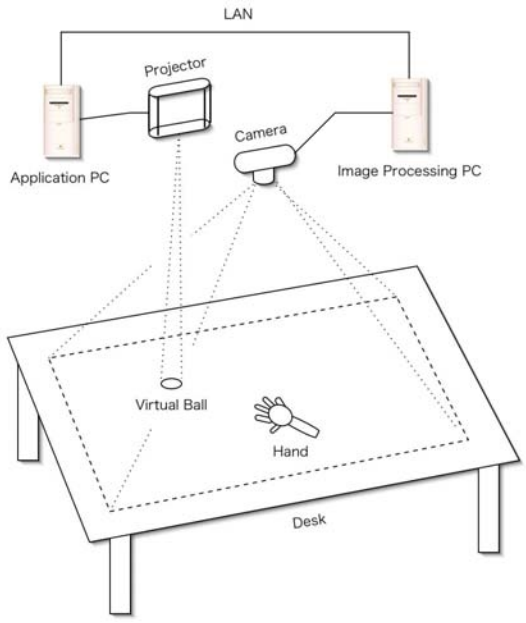

Figure 5: System overview: A projector and a camera are fixed on the ceiling and look down at a table surface. Unlike the 3D golf game, the camera captures whole of the table surface (640 $\times$ 480 pixels). The virtual marble and other game informations (e.g. scores) are displayed on the table surface.

REFERENCES

1. Sato, T., Fukuchi, K. and Koike, H. OHAJIKI Interface: Flicking Gesture Recognition with a High-Speed Camera. To be appeared in *Proceedings of ICEC 2006*.