WiiStick: Enhancing Motion Recognition Capability for Wii Systems

Pei-Ying Chiang, May-chen Kuo, Jessy Lee and C.-C. Jay Kuo
Ming Hsieh Department of Electrical Engineering and Signal and Image Processing Institute
University of Southern California, Los Angeles, CA 90089-2546
E-mails: peiyingc@usc.edu, maychenk@usc.edu, jessylee@usc.edu and cckuo@sipi.usc.edu

***Abstract*— Nintendo's Wii system uses motion acceleration sensors to provide an intuitive user interface with limited motion recognition capabilities. In this work, we investigate these limitations, and propose a new system called WiiStick to enhance the motion recognition performance. One main limitation of Wii is that the direction of movement cannot be recognized accurately if the motion involves both accelerating and rotating. To overcome this difficulty, we propose to add an elastic stick and an IR emitter to the Wiimote. Besides, we develop a motion recognition algorithm based on signal analysis and pattern recognition techniques. As a result, motion can be recognized more effectively. It is shown by experimental results that the motion recognition capability of the Wii system can be greatly improved by the proposed WiiStick system.**

I. Introduction

Researchers in human machine interaction (HMI) devote their efforts to the provision of an interface that allows people to communicate with machines as easy as possible. Traditional user interface (UI) such as the mouse, keyboards and computer languages are machine-oriented interfaces, which need certain amount of learning time for people to get used to it. The human-oriented UIs that understand human gestures and/or speeches (instead of typed words or clicked buttons) are desired. The "multi-touch screen" [1] allows users to manipulate screen objects with fingertips directly, which provides a good example of user-friendly interface. People can learn to control objects on the screen easily and intuitively.

The Nintendo Wii games [2] have become popular in recent years, which have drawn a large number of people to the gaming market due to the novel HMI technique. The Wii system consists of a game console and a novel remote controller called the Wiimote. For example, to play golf games, a player can simply swing the Wiimote as a golf club (instead of using the traditional button-driven keyboard). This intuitive UI lowers the learning curve and creates more excitement due to the full body motion. The Wii-like motion sensing system has a great potential in emerging applications that demand a novel interface between humans and machines.

Motion recognition solutions can be categorized to three types: the vision-based, the motion-based and the hybrid approaches. The vision-based method, *e.g.,* Sony's Eye Toy [3], allows players to interact with a game console by processing image sequences taken by one or multiple cameras. The vision-based method can detect motion more precisely at the cost of higher computational time and system cost. The motion-based method, *e.g.,* Apple's iPhone [4], senses the orientation of the phone and changes the screen display accordingly with a low-cost motion sensor, called the accelerometer. The motion-based method demands less computational time and system cost. Various commercial products adopt the accelerometer for motion recognition because of its low cost and portability features. However, it can only recognize simple motion types. The Wii system adopts a hybrid approach that integrates IR sensing and accelerometer sensing to extend its motion recognition capabilities. There are however limitations of the motion recognition capability of today's Wii system.

Research on the Wii system has been conducted recently. WiiArts [5] and UniGest [6] examined several new applications based on the current Wii technologies. In this work, we investigate the limitation of motion recognition in the Wii system, and propose a new system, called WiiStick, to enhance its motion recognition performance. One main limitation is that the direction of movement cannot be recognized accurately if the motion involves both accelerating and rotating. It is desirable to extend its motion recognition capability.

The virtual fencing game in WiiMedia [7] is closest to our efforts along this research direction. However, WiiMedia can detect three types of fencing motions only. The difficulty of tracking motion trajectory in Wii is demonstrated in their experiments. To overcome this difficulty, we propose to add an IR emitter and an IR sensor to the Wiimote and the Wii console, respectively. Besides, we present a motion recognition algorithm based on signal analysis and pattern recognition. As a result, motion translation can be tracked effectively. It is shown by experimental results that the motion recognition capability of the proposed WiiStick system can be greatly enhanced.

The rest of the paper is organized as follows. The motion recognition module of the Wii system is reviewed and its limitations are discussed in Sec. II. The WiiStick system is described in Sec. III. The experimental results are shown in Sec. IV. Finally, concluding remarks and future work are discussed in Sec. V.

II. Motion Recognition with Wii

A. Wiimote and Acceleration Recognition

As shown in Fig. 1, the Wiimote contains a built-in motion acceleration sensor that can detect the acceleration in three dimensions (*i.e.* along the x-, y-, z-axes). Besides, it contains infrared (IR) sensors used to track the translational motion. When the Wiimote lies still horizontally, the acceleration value is measured according to Earth's gravity, which is a vertical force along the z-axis. Thus, the accelerometer outputs an acceleration of (0, 0, 1g), where 1g indicates one gravity. While rolling (*i.e. rotating along the y-axis*) the Wiimote slowly, the gravity force applied to the z-axis sensor is decreasing but to that the x-axis is increasing. When the Wiimote lying on its left side, the accelerometer outputs an acceleration of (1g, 0, 0). Similarly, the acceleration value becomes (0, 1g, 0) when the Wiimote lies on its top.

The rotational motion patterns along the x-, y- and z-axes are called roll, pitch and yaw, respectively. Also, we use A_x, A_y and A_z to denote the force (or the acceleration) along the corresponding axis. Based on the acceleration value measured by the accelerometer, if the Wiimote is rotated slowly, the rotational angle of roll and pitch can be estimated by

$$\theta = \sin^{-1}(A_x) \text{ and } \zeta = \sin^{-1}(A_y),$$

where θ is the roll angle and ζ is the pitch angle [8]. Thus, the rotational angle of roll and pitch can be easily estimated. However,

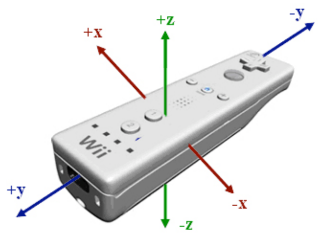

Fig. 1. Illustration of the accelerometer and its three axes.

.

the yaw motion cannot be detected since the gravity has the same magnitude along the three axes and none of A_x, A_y, A_z would change by this movement. To solve this problem, the Wiimote has built-in IR sensors (receivers) that detect the yaw motion by sensing the relative translation from the IR emitters attached to the Wii console (*i.e.*, the so-called IR sensor bar often placed on the top of the monitor). However, to sense radiation from emitters, the built-in IR sensors have to be constantly facing the emitters. Besides, the IR detecting range is about 36 degrees [7], which somehow limits the sensing range.

The following rule is developed to detect the occurrence of motion. When the vector sum of gravity force components, A_x, A_y and A_z, is equal to 1g (*i.e.* Earth's gravity), the Wiimote is not accelerating. If the Wiimote is thrust rapidly, the magnitude of the acceleration would increase depending on the speed of motion. Thus, to detect motion, we can use the following criterion:

$$\sqrt{A_x^2 + A_y^2 + A_z^2} > \Theta,$$

where Θ is a threshold close to but larger than 1g. Its value determines the sensitivity of the sensing function. For example, a larger threshold implies lower sensitivity.

B. Limitations of Wii

The low cost accelerometer has several limitations, which makes it difficult to enrich the application variety. In the experiment of playing the tennis game in Wii Sports, we observed that users can hit a ball back without considering the direction. The ball's direction is simply affected by "the timing of the motion performed" as well as "the acceleration magnitude". For example, the balls can go to the left even the player thrusts the Wiimote to the right. In fact, a player can successfully hit the ball back when he/she thrusts the Wiimote to any direction. In other words, the direction information is not used in the decision process because the motion sensor cannot determine the motion trajectory accurately when the underlying motion involves both acceleration and rotation. Only simple detection such as "Is the player accelerating the device?" or "What orientation (or rotation) is the device being held?" has been been made in today's Wii system.

The second limitation of the accelerometer is that, when a motion involves both rotating and accelerating, the accelerometer cannot tell the difference among many different types of motion. For example, the accelerometer outputs $A_x = 1g$ when the Wiimote is rolled clock-wise 90 degree along the y-axis. However, it also outputs $A_x = 1g$ when the Wiimote is thrust to the left with force of the same magnitude (*i.e.* 1g). Thus, the accelerometer is rarely used to estimate motion trajectory but detect simple motion such as sudden motion with acceleration or the orientation of the Wiimote.

To extend existing functionalities, there is a need to improve the motion recognition capability of the Wii system. In the next section, we propose a new system called WiiStick that includes an

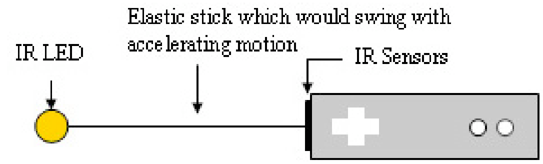

Fig. 2. Illustration of a Wiimote with an added stick connected to an IR LED.

IR-enhanced detection device and an advanced motion recognition algorithm. The moving direction of a certain motion type can be well recognized although the orientation of the device is changed during the motion.

III. Enhanced Motion Recognition with WiiStick

A. System Overview

To overcome the limitations of Wii as described in Sec. II, we propose a Wii-like system called *WiiStick* that has an additional device added to the Wiimote to facilitate motion recognition. As shown in Fig. 2, the add-on device consists of an elastic stick, which will swing with an accerlation motion, an IR LED attached to the other end of this stick, which serves as an IR emitter.

The main advantage of this add-on device is that the elastic stick not only follows but also exaggerates hand's movement. Then, the IR sensor on the top of the Wiimote can estimate the motion by tracking the motion of the IR LED. This idea is motivated by a fencing sword. The combination of the elastic stick and the Wiimote works just like the elastic blade and the hilt. The other advantage is that the attached IR LED always moves with the Wiimote so that the IR sensor inside the Wiimote can always face to this IR emitter. A player does not have to worry about whether his/her Wiimote is facing to the fixed IR emitter. With the add-on device, we can capture the IR translation information for more accurate motion recognition. In this framework, both signals from the IR sensor and the motion sensor in the Wiimote are used for motion recognition. Examples of X-Y translational motion measured by the IR sensor for two motion types (*i.e.*, circle drawing and right-downward thrusting) are shown in Fig. 3.

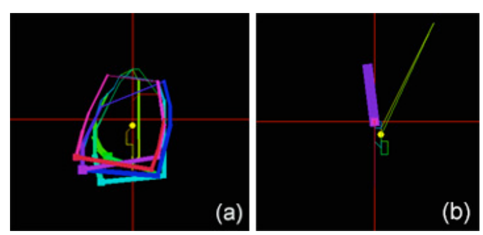

Fig. 3. The X-Y translation of the attached IR LED during the motion of (a) drawing circles (b) thrusting right downward.

We propose a motion recognition algorithm to process the captured IR signals and the acceleration information in the WiiStick system. This algorithm consists of the following three steps.

1) Motion Segmentation
   Separate pre-motion and actual-motion periods for accurate motion recognition.
2) IR Signal Post-Processing
   A post-processing step is used to fix errors of the IR signal.
3) Pattern Recognition
   This is used to categorize each motion type.

They are detailed below.

B. Motion Segmentation

To extract useful motion signals, the quantity

$$S_i = \sqrt{A_{x_i}^2 + A_{y_i}^2 + A_{z_i}^2},$$

which represents the acceleration magnitude, is calculated at each time step to check whether motion occurs or not (see Sec. II for detail). If there is a time period of duration Ψ, where all S_i values in Ψ are larger than threshold t, we claim the occurrence of a motion. Besides, we set up a time interval (*e.g*. 30 frames. Also, note that the Wiimote reports up to 100 times per sec.) so that we can extract signals for motion recognition for every interval. Then, we segment Ψ into two parts: pre-motion and actual-motion.

By pre-motion, we refer to the motion preparation stage. For example, a play may slightly raise the hand before thrusting downward. Then, hand raising is the pre-motion while thrusting downward is the actual motion. To separate these two parts, our algorithm searches for each local maximum as a key point since, when a player performs a motion, the acceleration value would first increase and then decrease. The local maximum of the acceleration value can be used as a clue for data segmentation. In the algorithmic implementation, we choose another threshold t'. (Both t and t' are statistical values of previous experience. The systme is more sensitve if a smaller threshold value is chosen.) If the local maximum is larger than t', it is the actual-motion; otherwise, the pre-motion. The ranges of these two parts are obtained by extending from the local maximum to the time instances where S_i stop decreasing. In Fig. 4, we show how to segment the pre-motion and the actual motion parts from captured signal S_i.

Since pre-motion occurs before actual-motion as described above, estimating pre-motion can help predict actual-motion by assigning a statistical weight to each candidate motion. For example, for a player who always raises the hand before thrusting downward, if the pre-motion is recognized as "raising the hand", then we may assign the highest weight to "thrusting downward" to the coming motion.

C. IR signal post-processing

The IR trajectory of some motion measured from the Wiimote is shown in Fig. 5(a), where we observe quite a few gaps in the path of the captured IR signal. Since the IR reception is not fast enough to catch the whole path and the detection range of the IR sensor is limited, IR sequences can break into many small pieces when the IR LED is out of the detection range. Post-processing can be used to connect these broken pieces.

Generally speaking, to fill the gap of IR translational curves, we calculate the velocities of IR motion before and after the gap, and use them for interpolation. The detail is described below. Assume

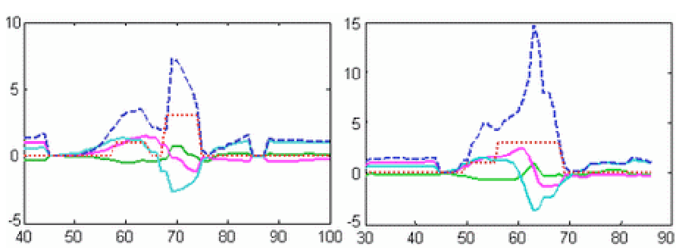

Fig. 4. Two samples of acceleration sequences. The horizontal axis shows the time whereas the vertical one represents the acceleration value. The red dot line represents the segmentation result. The first lower peak (square) of the red dot line is the pre-motion period and the higher peak next to it is the valid motion period. (The blue dash line is S_i at each time step, other lines are three acceleration values.)

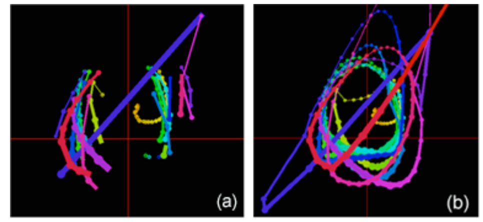

Fig. 5. IR LED translation sequences: (a) data directly captured from the IR sensor and (b) results after post-processing.

that the gap starts at t_s and ends at t_e. We calculate velocities of IR translation at time $t_s - 1$ and $t_e + 1$ and denote them with V_{pre} and V_{post}, respectively. Then, we use the following equation to interpolate missing IR points between the gaps:

$$IR[t_s - 1 + i] = IR[t_s - 1] + (V_{pre} \times i) + (\delta_V \times i^2) + (\text{offset} \times i),$$

where $i = 1 \cdots g_{size}$, $g_{size} = (t_e - t_s + 1)$, $IR[t] = (x(t), y(t))$ at time t, $\delta_V = (V_{post} - V_{pre})/g_{size}$ and

$$\text{offset} = \frac{IR[t_e+1] - (IR[t_s-1] + V_{pre} \times g_{size} + \delta_V \times g_{size}^2)}{g_{size}}.$$

If the data keep missing until the motion stops, we cannot get V_{post} in this case. Then, we adopt the following equation

$$IR[t_s - 1 + i] = Ir[t_s - 1] + V_{pre} \times i, \quad i = 1 \cdots g_{size},$$

The post-processing result is shown in Fig.5(b).

D. Wavelet-based Pattern Recognition

To verify the effectiveness of our WiiStick, the Wavelt-based pattern recognition is chosen because it is easy to implement. The IR LED translations tracked by the IR sensor is used as the motion sequence for recognition. Before categorizing a motion, a normalization step is needed to process the motion sequence. Then, we represent a normalized planar curve using the wavelet descriptor [9] and analyze its associated motion type based on wavelet coefficients. Specifically, we compute the differences of wavelet coefficients between the unknown motion and known motion types and use the Euclidean norm as the error measure. We classify the unknown motion pattern to a given one that has the smallest error. In our experiment, two representative sequences of each motion type are selected as the template, which have the highest recognition rate among all samples.

IV. EXPERIMENTAL RESULTS

To visualize the motion analysis result in real time, we connected the Wiimote to the PC with Bluetooth and built a 3D model of the Wiimote so that we can render the analysis results with OpenGL as shown in Fig. 6. In the experiment, we designed a set of motion patterns commonly used in a fencing game. Some samples of the fencing motion are illustrated in Fig. 7. We consider the following eleven motion types: drawing circles clockwise (T1), drawing circles counter-clockwise (T2), slashing downward (T3), upward (T4), leftward (T5), rightward (T6), forward (T7), left-downward (T8), right-downward (T9), left-upward (T10) and right upward (T11). Some examples are illustrated in Fig. 7.

We capture both acceleration values and IR signals from the Wiimote for motion recognition. For each motion type, we collected 20 to 30 samples, and applied the motion recognition algorithm as described in Sec. III. The eleven sample templates are shown in Fig. 8. Some templates look similar in the figure, *e.g*. (9) and (10). However, one path is forward while the other is backward in the

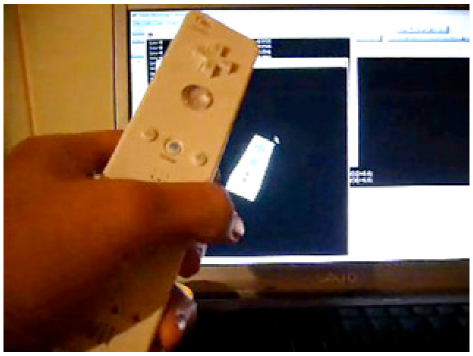

Fig. 6. The 3D model of the Wiimote is rendered at the PC screen so that analytical results are shown in real time.

curve tracing, which cannot be shown in the plot. Since they have different dynamics, their wavelet curve descriptors are different.

Experimental results in terms of correct recognition rates for eleven motion types are shown in Tables I and II. Results in Table I were obtained by analyzing the captured acceleration values only, which corresponds to today's Wii system. Results in Table II were obtained by analyzing the captured acceleration values and IR signals, which corresponds to the proposed WiiStick system.

If only the acceleration values are used in the Wii system, the averaged recognition rate is about 67%. By including both acceleration values and IR signals in the proposed WiiStick system, the averaged recognition rate can go up to 88%. The improvement is very substantial. In the WiiStick system, motions of drawing circles including their directions (*i.e.*, T1 and T2) can be perfectly recognized, and most motion types (*e.g.*, T3, T4, T5, T6, T9 and T10) can be recognized with a correct rate over 80%. The above results show that the motion recognition capability of Wii has been greatly enhanced by WiiStick.

Although the proposed motion recognition algorithm works reasonably well with distinctive motions. However, it demands further

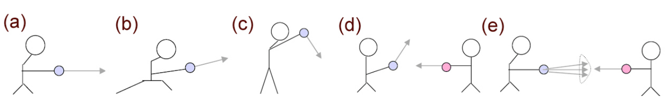

Fig. 7. (Illustration of three attack motions: (a) forward, (b) upward and (c) downward, and two defense motions: (d) hold sword vertically and (e) draw circles.

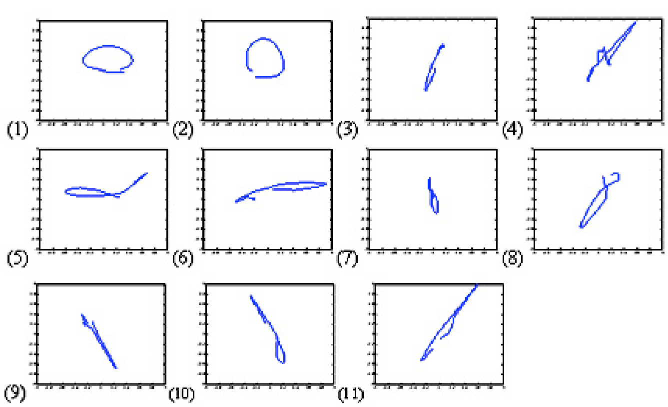

Fig. 8. Sample templates of motion types 1-11.

improvement in some cases. For T7, T8 and T11, the recognition rates are lower due to some similarity of their motion templates. For example, the template of T11 (right-upward) is similar to T4 (upward) so that the motion of T11 is sometimes recognized as T4.

TABLE I
CORRECT MOTION RECOGNITION RATES WITH ACCELERATION SIGNALS ONLY.

Motion Type	T1	T2	T3	T4	T5	T6
Recognition rate	0.86	0.54	0.81	0.81	0.81	0.54
Motion Type	T7	T8	T9	T10	T11	Avg
Recognition rate	0.68	0.40	0.27	0.81	0.90	0.67

TABLE II
CORRECT MOTION RECOGNITION RATES WITH BOTH ACCELERATION AND IR SIGNALS.

Motion Type	T1	T2	T3	T4	T5	T6
Recognition rate	1.00	1.00	0.81	0.95	0.95	0.92
Motion Type	T7	T8	T9	T10	T11	Avg
Recognition rate	0.68	0.72	1.00	0.95	0.68	0.88

V. CONCLUSION AND FUTURE WORK

In this work, we investigated the limitations of the Wii system in motion recogntion, and proposed a new system called WiiStick to enhance the motion recognition performance. Specifically, we added an IR emitter and an elastic stick to the Wiimote. Besides, we developed a motion recognition algorithm based on signal analysis and pattern recognition. It was shown by experimental results that the motion recognition capability of the Wii system is greatly improved by the proposed WiiStick system. In the future, we would like to improve the recognition rate furthermore so that the system is robust enough for real-world game applications. In addition, we would like to develop other applications for WiiStick. One application scenario is to trace and analyze the trajectory of a golf ball by examining the movement of the IR LED attached to the elastic stick. This information provides the system some useful feedback on how a player swings his/her Wiimote or WiiStick.

REFERENCES

[1] J. Y. Han, "Low-cost multi-touch sensing through frustrated total internal reflection," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, 2005, pp. 115–118.

[2] "Wii," Nintendo Co. Ltd, 2006. [Online]. Available: http://wii.com/

[3] "EyeToy," Sony Computer Entertainment, 2002. [Online]. Available: http://www.eyetoy.com

[4] "iPhone," Apple Inc, 2007. [Online]. Available: http://www.apple.com/iphone/

[5] G. G. Hyun-Jean Lee, Hyungsin Kim and A. Mazalek, "WiiArts: creating collaborative art experience with wiiremote interaction," in *Proceedings of the 2nd international conference on Tangible and embedded interaction*. ACM, 2008, pp. 33–36.

[6] S. J. Castellucci and I. S. MacKenzie, "UniGest: text entry using three degrees of motion," in *CHI '08 Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 3549–3554.

[7] A. Skihiko, E. Geslin, and S. Richir, "WiiMedia: motion analysis methods and applications using a consumer video game controller," in *2007 ACM SIGGRAPH symposium on Video games*. ACM, 2007, pp. 133–140.

[8] "A Gnu/Linux port for the Nintendo Wii," WiiLi.org. [Online]. Available: http://www.wiili.org/

[9] C.-H. Chuang and C.-C. J. Kuo, "A wavelet descriptor of planar curves: theory and applications," in *IEEE Trans. on Image Processing*. IEEE, 1996, pp. 56–70.