

# Compact, Configurable Inertial Gesture Recognition

Ari Y. Benbasat and Joseph A. Paradiso

Responsive Environments Group

MIT Media Laboratory

20 Ames St., Cambridge, MA 02139 USA

{ayb,joep}@media.mit.edu

## ABSTRACT

We describe an inertial gesture recognition framework composed of three parts. The first is a compact, six-axis inertial measurement unit to fully measure human motion. The data from this unit is then analyzed by a gesture recognition algorithm which considers data on an axis-by-axis basis and categorizes them as simple motions (line, twist, etc.) with magnitude and duration. An application designer can then combine these atoms together both concurrently and consecutively to create composite gestures which can be tied to output routines. The framework was implemented on a Palm III to demonstrate its light-weight nature and to evoke devices which possess both a sense of their own motion and the ability to respond to it.

## Keywords

inertial measurement, gesture recognition, palm pilot

## INTRODUCTION

Inertial measurement components, which sense either acceleration or angular rate, are being embedded into common user interface devices more frequently as their cost continues to drop. These devices hold a number of advantages over other sensing technologies: they directly measure important parameters for human interfaces and can easily be embedded into mobile platforms. However, in most cases, these inertial systems require a large amount of custom hardware and software engineering to be done for each application, with little reuse possible.

This paper presents a solution in the form of a framework for inertial gesture recognition. The system consists of a compact inertial measurement unit, a light-weight atomic gesture recognition algorithm, and scripting functionality to allow combinations of atoms to be linked to output routines. A sample implementation and future possibilities are discussed. A more detailed discussion can be found in [2].

## SENSOR HARDWARE

The first step in the framework is the inertial measurement unit (IMU), which will provide the full six degrees of motion sensing, three axes each of rotation and acceleration. While there are many 6-axis IMUs on the market, most of them fail to meet our requirements. The Crossbow Technologies DMU-6X and the Intersense IS600 (shown at SIGGRAPH98) are representative in that they are far too large for our purposes and offer very little access to the underlying hardware (a detriment to our research).

The final design of our IMU is a cube 1.25" on a side and is shown in Figure 1a. Two sides of the cube contain the

inertial sensors. Rotation is detected with three single axis Murata ENC03J piezoelectric gyroscopes<sup>1</sup>. Acceleration is measured with two two-axis Analog Devices ADXL202 MEMS accelerometers<sup>2</sup>. The sensor data are input to an Analog Devices ADuC812 microcontroller with 12-bit analog-to-digital converter (ADC) and an 8051 microprocessor core. Gyroscope data is collected using the ADC, while the accelerometer data is collected via timing measurements. The raw sensor values are sent using a small RF Monolithics transmitter module<sup>3</sup> to a separate base station, which connects to the data analysis machine via a serial link. The device update rate is approximately 66 Hz. The complete system runs at 3V and draws only 26mA when powered.

There is no necessity to use the wireless link. In fact, the authors argue that the most interesting devices are those which incorporate enough processing power to perform the software functions of the framework (recognition and matching) on board. This would allow them to have a sense of their own motion and to respond to it *in situ*.

## GESTURE RECOGNITION

For stand-alone implementations to be possible, it is necessary to have as simple a recognition scheme as possible. Therefore, very general, powerful, and cycle-hungry systems such as hidden Markov models, are not acceptable. Instead an algorithm was designed that took advantage of a *priori* knowledge of the structure of human muscle motion.

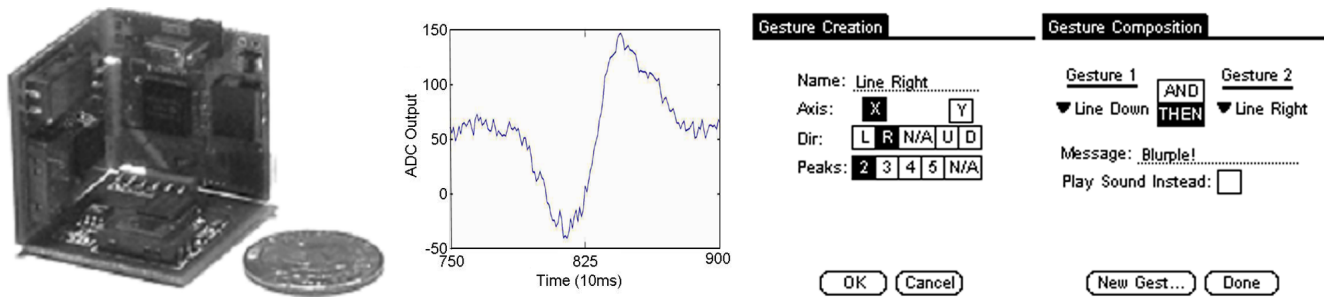
The algorithm designed works in two stages. In the activity detection stage, sections of increased energy are flagged for further analysis on an axis by axis basis. They are identified by applying a threshold to the variance of the signal within a time interval window. The threshold set is using knowledge of the maximum attack speed, and the window size is set to guarantee separation between the expected in-motion variance and the sensor noise floor. Note that this algorithm recognizes only fairly deliberate motions with a sharp start and stop.

Given a flagged area, the next step is to determine whether a gesture is present therein, which atomic gesture it is, and what its parameters are. Parameterized atomic gestures cannot be decomposed, and are combined to create composite gestures. An examples is a straight-line are parameterized in terms of magnitude (length) and duration. Figure 1b show the acceleration curve created by a straight-line on one axis. This multi-peaked structure is

<sup>1</sup>Max. angular velocity 300°/sec. Sensitivity 0.67mV/°/sec.

<sup>2</sup>Max. acceleration ±2g. Pulse width output, sensitivity 12.5%/g

<sup>3</sup>Frequencies: 315, 916 MHz. Max transmission speed 19.2kbps.



**Figure 1: a) The IMU; b) Acceleration data for a line gesture; c) & d) Gesture creation and composition interfaces**

representative of all human arm motion[3]. Adding another continuous line segment (a there and back motion) simply adds another peak. Therefore, our atoms were defined simply by the number of peaks.

We exploit this structure for the recognition. Since the velocity of the arm is zero at the ends of the gesture, the integral across it must be zero as well (after any orientation baseline change is subtracted). Therefore, recognition is accomplished simply by tracking across an area of interest, and recording the number of peaks and their mass (assuming certain minimum peak sizes to ignore noise and dithering). If the sum over the area is small compared to the sum of the unsigned area, a gesture is present. The duration is found trivially, and the magnitude is proportional to the absolute sum. Note that the parameters are found after the recognition stage, which allows it to be done in a single pass.

For gyroscope gestures, the integration criteria no longer holds. Still, a simple threshold on peak sum proved robust in testing.

### SCRIPTING

Up to this stage in the framework, an application designer need not have concerned themselves with the details. Their only role is in the final scripting phase. In this phase, it is possible to combine gestures both consecutively and concurrently to create composite gestures. AND, OR and THEN logical combinations are allowed between atoms, and matches on individual atoms can be restricted to those with certain parameters. The user can add new matching functions if desired.

Given a gesture they wish to recognize, a designer only has to perform it a few times, note the atomic gestures it produces and their order, and then write a simple script to recognize that combination. It can then be tied to any output routine they desire. A sample application described in [2] did not suffer from the granularity imposed by the recognition or scripting stage.

### SAMPLE IMPLEMENTATION

To demonstrate the ease of implementation of these algorithms, it was decided to build a gesture recognition system into a Palm III personal digital assistant (PDA). The authors admit that a PDA is not the best place for a gesture-based interface, since it is difficult to track the screen as it is moved. Tilting and shaking interfaces tend to be more appropriate, such as in [1]. Regardless, this platform was chosen because of its ubiquity and ease of use.

Our implementation used a reduced sensor set (only 2 accelerometers) because of space restrictions with the Palm III case. It was otherwise complete, providing not only the

recognition algorithms, but also simple scripting capabilities, allowing atoms to be defined (Figure 1c) and combined (Figure 1d). A designer can simply perform a gesture, see the atoms it creates in the output screen, and then create a composite gesture and specify the output (currently only a text message or beep sound are available).

The Palm III uses a Motorola DragonBall processor (on par with the original Macintosh), which runs at 16 MHz and has no floating point unit. Further, both program and data memory are limited (and shared with other applications). Nonetheless, we were able to implement our framework, running at 50 Hz. Several features made this possible: the algorithms use only integer operations, few calculation are done at each time step, and the same analysis is run on each axis.

### FUTURE DIRECTIONS

Given the ability to create simple portable computers (wearables, PDAs, etc.), the question is how this portability can be exploited. One way to do so is to give these devices a sense of their own movement and the ability to respond thereto. A possible example is shoe-based physical therapy system which cues the wearer to awkward movements. More intriguing is the concept of a generalized inertial gesture system, which can switch from application to application simply by attaching it to a new object and downloading (with little bandwidth) a new set of gestures.

### CONCLUSIONS

We built an inertial gesture recognition framework comprising a compact IMU, an atomic gesture recognition algorithm and a scripting system. The system was light-weight enough to be implemented on Palm III. This work points to a future where interface designers can use easily configured inertial sensors in a wide variety of settings.

### ACKNOWLEDGMENTS

We appreciate the support of the Things That Think Consortium and other sponsors of the MIT Media Laboratory. Mr. Benbasat also acknowledges the support of the Natural Sciences and Engineering Research Council of Canada and the Toshiba Corporation.

### REFERENCES

1. Bartlett, J.F. Rock n' Scroll is Here to Stay. *IEEE Computer Graphics and Applications*. 20(3):40-50, May/June 2000.
2. Benbasat, A. An Inertial Measurement Unit for User Interfaces. SM Thesis, MIT Media Laboratory, Sept. 2000.
3. Flash, T. and Hogan, N. The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model. *J. Neuroscience*, 5(9):2318-2330, 1985.