An Inertial Measurement Framework for Gesture Recognition and Applications

Ari Y. Benbasat and Joseph A. Paradiso

MIT Media Laboratory
77 Massachusetts Ave., NE18-5FL 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, wireless six-axis inertial measurement unit to fully capture three-dimensional motion. The second, a gesture recognition algorithm, analyzes the data and categorizes it on an axis-by-axis basis as simple motions (straight line, twist, etc.) with magnitude and duration. The third allows an application designer to combine recognized gestures both concurrently and consecutively to create specific composite gestures can then be set to trigger output routines. This framework was created to enable application designers to use inertial sensors with a minimum of knowledge and effort. Sample implementations and future directions are discussed.

1 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 interaction and they can easily be embedded into mobile platforms. However, in most cases, inertial systems are put together in a very *ad hoc* fashion, where a small number of sensors are placed on known fixed axes, and the data analysis relies heavily on *a priori* information or fixed constraints. This requires a large amount of custom hardware and software engineering for each application, with little possibility of reuse. We present a solution in the form of a generalized framework for inertial gesture recognition. The system consists of a compact inertial measurement unit (IMU), a light-weight gesture recognition algorithm, and scripting functionality to allow specified combinations of such gestures to be linked to output routines of a designer's choice. We envision that the IMU could easily be incorporated into almost any device, and a designer would be able to quickly and easily specify the gestures to be detected and their desired response. Sample implementations of the framework are discussed, as well as future possibilities. A more detailed discussion can be found in [5].

I. Wachsmuth and T. Sowa (Eds.): GW 2001, LNAI 2298, pp. 9–20, 2002.

2 Related Works

There are two major threads in the academic literature on the uses of inertial components in gestural interfaces. The first is in the area of musical input. Sawada[18] presented an accelerometer-only system that can recognize ten gestures based on a simple feature set. These gestures are then used to control a MIDI (Musical Instrument Digital Interface) instrument. The Brain Opera, a large-scale interactive musical exhibit produced by the Media Laboratory, included an inertially-instrumented baton as an interface[14], again to a MIDI instrument. In both of these cases, the gesture recognition techniques were quite specific to the application at hand and are therefore difficult to generalize.

The second area is the use of inertial sensors as a stand-alone interface for palmtop computers. The Itsy system from Compaq[4] uses accelerometers both as a static input, with the user tilting the device to scroll images, and a dynamic input, where fanning the device zooms the image in or out. Notable design ideas and concepts for such devices are presented by Small and Ishii[19] and Fitzmaurice[10]. While interesting, the input spaces of devices such as the Itsy tend to be limited as they consider only orientation (either static or dynamic), completely ignoring the kinetic information from the accelerometers.

The work most closely related to ours is that of Hoffman et. al.[13], who used a glove-based accelerometer system for the recognition of a subset of German sign language. The key differences lie in the complexity: our system (as described later) is designed around simpler algorithms (direct data stream analysis versus hidden Markov models) and less densely sampled data (66 Hz versus 500 Hz). Further, our inclusion of gyroscopes increases the range of gestures which can be recognized and the wireless nature of our sensing package allows for its use in more diverse situations.

3 Sensor Hardware

This project's direct lineage can be traced to the Expressive Footwear[15] project, where a printed circuit card instrumented with dynamic sensors (gyroscopes, accelerometers, magnetic compass), among a number of others, was mounted on the side of a dance shoe to allow the capture of multi-modal information describing a dancer's movements. This data was then filtered for a number of specific features, such as toe-taps and spins, which were used to generate music on the fly. While very successful, this system could not be generalized because the chosen set of sensors measured only along specific axes (as suggested by the constraints of the shoe and a dancer's movement) and the circuit card was too large for many applications. Therefore, we decided to create a new system that would contain the sensors necessary for full six degree-of-freedom (DOF) inertial measurement. The system had to be compact and wireless, to allow the greatest range of possibilities. While there are currently a number of 6 DOF systems commercially available, they were all unsuitable for a number of reasons. Crossbow Technologies offers the DMU-6X inertial measurement unit[9] which

Fig. 1. Current IMU hardware and reference frame

has excellent accuracy, but is quite large ($> 30\,\text{in}^3$). The Ascension Technology miniBird 500 [8] magnetic tracker is the smallest available at $10mm \times 5mm \times 5mm$ making it particularly easy to use. However, the closed loop nature of the sensor requires that it be wired, and the base unit is somewhat cumbersome. Also, both of these systems are fairly expensive and neither matches our specification in terms of ease of use (small, wireless).

The physical design of our wireless inertial measurement unit is a cube 1.25 inches on a side (volume $< 2\,\text{in}^3$) and is shown in figure 1. Two sides of the cube contain the inertial sensors. Rotation is detected with three single axis Murata ENC03J piezoelectric gyroscopes[1]. Acceleration is measured with two two-axis Analog Devices ADXL202 MEMS accelerometers[2]. The sensor data are input to an Analog Devices ADuC812 microcontroller (on the remaining side) using a 12-bit analog-to-digital converter. The raw sensor values are then transmitted wirelessly at an update rate of 66 Hz using a small RF Monolithics transmitter module[3] to a separate basestation, which connects to a data analysis machine via a serial link. The complete system operates at 3 V, draws 26 mA while powered, and runs for about 50 hours on two batteries placed in parallel. These batteries, together with the planar transmit antenna, are also small enough to fit inside of the cube formed by the hardware (see above, at left). The total cost of the system, in prototype quantities, is approximately US $300.

In the two years since the design of our hardware, low-cost inertial packages have become much smaller. Analog Devices currently markets the ADXL202E with dimensions of 5 mm × 5 mm × 2 mm, and is in preproduction of the ADXRS150 gyroscope, which will be available in a similar package. It is now readily possible to make a (almost) flat IMU with a footprint of less than $1\,\text{in}^2$, and this is an opportunity that we are currently pursuing.

Finally, there is no absolute need to use the wireless link to a remote computer. We argue that some of the most interesting devices will be those which in-

[1] Max. angular velocity 300°/ sec. Sensitivity 0.67 mV/°/ sec.
[2] Max. acceleration ±2 g. Pulse width output, sensitivity 12.5%/ g
[3] Frequencies: 315, 916 MHz. Max transmission speed 19.2kbps.

tegrate enough processing power to perform the software functions of the framework (recognition and matching) on board. Such devices would then have a sense of their own motion and could respond to it *in situ*.

4 Problem Statement

Gesture recognition is a wide-ranging research topic. Therefore, it is important at this point to discuss the specifics being addressed in this paper. Our goal was to design generalized algorithms to utilize the hardware described above as fully as possible. Since inertial sensors measure their own motion, the system will either have to be worn by the subjects or embedded in an object manipulated by them. Therefore, we chose to examine (as a first problem) human arm movement, with a gesture defined as any non-trivial motion thereof. It is assumed that these movements will convey information, though their interpretation is considered outside the scope of this paper[4]. Further, to take advantage of the compact, wireless nature of the system, we would like to be able to create stand-alone applications. Therefore, the algorithms should be as efficient as possible, with incremental increases in accuracy being readily traded for increases in speed or algorithmic simplicity.

5 Atomic Gesture Recognition Algorithm

5.1 Atomic Gestures

We define atomic gestures as those that cannot be further decomposed, and which can be combined to create larger composite gestures. The value of this definition is that it should only be necessary to recognize a small set of atoms which span the space of human gestures; thereafter, any gesture of interest can be synthesized from its atoms. The fundamentally important choice of atomic gestures to recognize was made through an examination of both the raw sensor data and the human kinetics literature.

Figure 2 shows the parsed (by the full algorithm) accelerometer traces from a number of simple arm movements and figure 3 shows the curves created by a straight-line and by a there-and-back motion on one axis. Note that this multi-peaked structure is representative of all human arm motion[11]. A straight-line motion will create a two-peaked trace, while the there-and-back motion creates a three-peaked trace[5]. Therefore, we define the atomic gestures simply by the number of contained peaks. While this may not provide a complete basis for the space of human gesture, this decomposition exploits the structure in the data to greatly simplify the gestures recognition with few drawbacks (see section 7.1).

[4] In fact, the point of the framework is that meaning is to be added in the scripting stage by an application designer.

[5] A single-peaked trace would imply a net change in velocity. The arm's limited range makes constant velocity motion impossible for more than short periods.

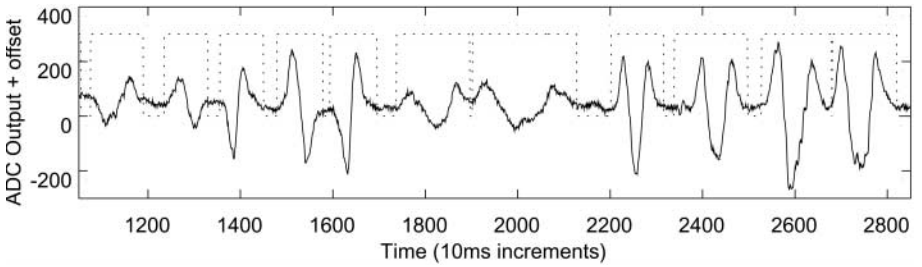

Fig. 2. Parsed acceleration data from simple human arm motion

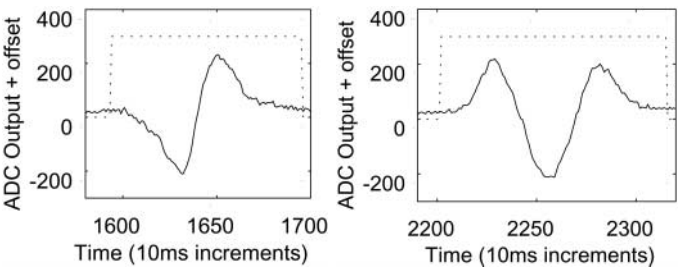

Fig. 3. From figure 2, a straight-line and a there-and-back gesture

Further, we parameterized these gestures in terms of magnitude (length) and duration, which is again fundamental to human motion[2].

We simplify the recognition by examining the data streams on an axis by axis basis (where the axes are those defined by the body frame of the sensor). This allows us to exploit the symmetry in the problem and run the same algorithm repeatedly in a one-dimensional space, which is more efficient than doing so once in a high-dimensional space. This also means that the recognition algorithm can be used with any number of sensors, allowing applications with incomplete, as well as multiple, IMUs.

The literature[6] suggests the definitions given above are not unduly limiting. In the case of atomicity, the study of French Sign Language by Gibet et al.[12] showed that the most common gestures were either straight-line or twist (the simplest gyroscope gesture in our scheme) and further that they most often lay on one of the three main axes (or planes). As for the parameterization, the widely used Hamburg Notation System[16] includes modifiers for both speed and size of gestures.

[6] While the cited references are both from sign language research, it is not our intent to suggest that this system is designed for use in that application, only that their descriptions of arm motion are both interesting and generally applicable.

5.2 Activity Detection

In the activity detection stage, sections of increased energy in the data are flagged on an axis by axis basis for further analysis. To do so, the variance of the data over a fixed window is calculated. Since the variance is proportional to $\Sigma(x^2)-(\Sigma x)^2$, this can be done very efficiently by keeping a running sum and running sum of squares of the data. Ranges where the variance is greater than a set threshold are considered to be periods of activity. The window size and threshold value can be found analytically, based on the sensor noise floor and the minimum attack speed considered to represent a deliberate motion. The threshold should err on the side of false positives, as these will later be rejected by the gesture recognition algorithms.

While the activity detection algorithm is designed primarily to filter the data before it is passed on to the recognition system, other purposes can be served in annotating the data stream. For example, consider a medical system that not only does real time recognition but also caches data for further analysis. While storing detailed data for an entire days' worth of motion is achievable, storing only those data segments where activity was noted would be much more efficient.

5.3 Gesture Recognition

Given a flagged area, the next step is to determine whether a gesture is present therein, and if so which atomic gesture it is, and what its parameters are. Because efficiency is a key goal for the gesture recognition algorithm, very general, powerful, and cycle-hungry systems such as hidden Markov models[17] (HMM) are not applicable. Instead, an algorithm was designed that took advantage of *a priori* knowledge of the structure of human arm muscle motion described above.

Since the velocity of the arm is zero at the ends of the gesture, the integral of the acceleration across it must be zero as well (after subtracting any baseline change due to change in orientation). Therefore, recognition is accomplished simply by tracking across an area of activity, and recording the number of peaks and their integral. A minimum peak size is assumed (to reject noise and dithering) and smaller peaks are subsumed into the previous peak (if possible). If the ratio of the net area under the peaks to the sum of the absolute value of their areas is below a fixed threshold (random walk noise must be accounted for even in the ideal case), a valid gesture is said to be present. Its duration is simply the duration of the area of activity, and its magnitude is proportional to the absolute sum divided by the duration. The parameters are determined after the recognition stage, which allows it to be done in a single pass, rather than via a multi-step search using expectation-maximization (or similar techniques), as is often the case with HMMs[20].

For gyroscope gestures, the integration criterion no longer holds (as the data is one less derivative away from absolute motion). Still, in testing, a simple threshold on peak sum proved effective in distinguishing gesture from noise. Duration found as above and magnitude is now proportional to the absolute sum itself.

To improve recognition rates, gestures are further analyzed to account for two special cases. Broken gestures – those made up of two or more areas of activity – are tested for by combining areas of activity (less than a fixed time apart) with no valid gesture and reevaluating the combination. Composite gestures – where two or more gestures are within a single area of activity – are found by splitting and reexamining areas of activity at the point where two adjoining peaks whose masses have the same polarity. In figure 2, the data around 18000 ms was originally parsed as a broken gesture, and the data between 25000 ms and 28000 ms was originally parsed as a composite gesture.

It is interesting to note that even the current modest data sampling rate and accuracy may be more than is needed. In the case of a recorded sample stream of atomic gestures, the same recognition rate was achieved with both the full data stream and one downsampled to 30 Hz and 8 bits[5]. This suggests that the redundancy in human gesture could be even further exploited than we have here, and more lightweight algorithms, and therefore devices, could be created.

6 Scripting

Until this stage in the framework, an application designer needs not have concerned themselves with the details. Their only (required) role in the framework is in the final scripting phase. In this phase, the designer can combine gestures both consecutively and concurrently to create composite gestures of interest. Matches on individual atoms can be restricted to those with certain parameters, and OR and AND logical combinations are allowed between atoms (though the user can add new matching functions if desired). Such logical combinations can then be placed in temporal order to create full composite gestures. These gestures are then connected to output routines.

Given a composite gesture they wish to recognize, a designer only has to perform it a few times, note the atomic gestures recognized and their order, and then write a simple script to recognize that combination. It can then be tied to any output functionality they desire (auditory, graphical, etc.).

It is also possible at this stage to analyze additional sensor data of interest (e.g. bend sensors in a data glove), though the designer would need to provide recognition algorithms for that data. The output of those algorithms could then be added to the atomic gestures found by the inertial gesture recognition and matched and composed in the same fashion.

Note that there is no necessity to use the scripting system provided. The atomic gestures can be used as inputs to an HMM or other recognition algorithm, thereby fulfilling a role similar to that of phonemes.

7 Applications and Implementations

While the purpose of this paper is to present a framework for easily creating inertial sensing based applications, that potential cannot be evaluated without considering applications created using it. This section will discuss the limitations

that must be considered when creating applications, and then describes both a large-scale application and a stand-alone implementation of the system.

7.1 System Limitations

While we attempted to create as general a wireless system as possible, various features of both inertial sensing generally and this framework specifically impose certain restrictions on the gestural applications that can be implemented. The most important constraint to consider is the lack of an absolute reference frame. Given the class of sensors chosen (based on their price and size), it is not possible to track orientation relative to a fixed frame for longer than approximately five seconds[5]. Therefore, it is necessary for the body reference frame itself to have some meaning associated to it, such that the application designer will be able to construct output functions that are appropriate regardless of the instrumented object's orientation in the world frame.

The second constraint, imposed by the set of atomic gestures chosen, is that the system cannot track multi-dimensional gestures, except for those which are separable in space and time. An arbitrary straight line can always be decomposed, and therefore can be recognized, while a rotating object tracing the same linear path would not be recognizable because the local axis of acceleration changes with time, making the decomposition impossible. However, the physical constraints of the gestural system will often prevent such movements, especially in the case of human movement.

The final set of constraints are those imposed by the algorithms used for analysis and gesture recognition. Gestures must come to a complete stop to be found by the activity detection algorithm, making fluid motions and transitions hard to pick up. Note that as long as movement on one axis comes to a stop, the recognition of at least part of the gesture can be attempted. For example, while drawing a square shape can be done repeatedly with a single motion, each line segment will contain a start and a stop on the appropriate axis.

7.2 Sample Application

This system was first used in `(void*)`[6], an interactive exhibit created by the Synthetic Characters group at the MIT Media Lab. In this exhibit, a user could control one of three semi-autonomous virtual characters, causing them to dance. Drawing our inspiration from Charlie Chaplin's famous 'buns and forks' scene in The Gold Rush[7], we created an input device whose outer casing was two bread rolls, each with a fork stuck into the top, thereby mimicking a pair of legs(see figure 4). An IMU was placed inside each of the buns, with the data sent to a separate computer for recognition. A variety of gestures (kicks, twirls, etc) with both one- and two-handed versions were recognized and used as commands to the virtual characters.

An HMM-based gesture recognition system was used in this project with reasonable success both at SIGGRAPH '99 and in demonstrations at the Media Laboratory. Its main source of error was overconstraint of the HMM parameters,

Fig. 4. Buns and Forks Interface to `(void*)`

leading to a lack of consistency not only among users, but also among slight variations in the IMU position over time. Adjusting the parameters had limited success in improving this situation.

The gesture recognition for `(void*)`was recently redone using the framework described above, and achieved a similar level of accuracy as the original implementation[5]. The superiority of the framework lies, by design, in its speed and ease of use. It occupies few processor cycles, freeing them up for more complex output tasks. Also, the script took less than two hours to write, with most gestures being trivially defined (e.g. a kick is composed of concurrent twist and straight-line atoms). In contrast, the HMM-based system was so complex that it was not able to process data at even half the sensor update rate and each training cycle for each gesture took numerous hours.

7.3 Stand-Alone Implementation

To demonstrate the ease of implementation of these algorithms, we built a gesture recognition system into a Palm III personal digital assistant (PDA). While a PDA is not the best platform for a gesture-based interface, since it is difficult for the user to visually track the screen as it is moved, 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 within the Palm III case. It was otherwise complete, providing not only the recognition algorithms, but also simple scripting capabilities, allowing atoms to be interactively defined and combined (figure 5). Individual recognized atoms and their parameters were displayed in a separate output mode. Therefore, a designer can simply perform a gesture, see the atoms it creates on the output screen, and then use the GUI shown above to encode the composite gesture it represents and specify the output when detected (currently only a text message or beep sound are available).

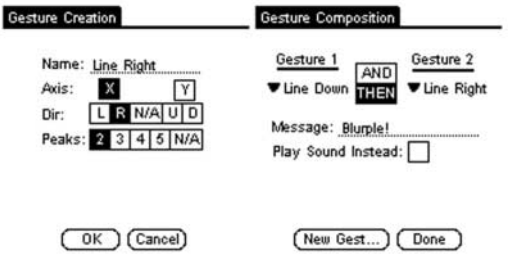

Fig. 5. Gesture creation and composition

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, in approximately 200 lines of C code. Several features made this possible: the algorithms use only integer operations, few calculations are done at each time step, and the same analysis is run on each axis. This experience suggests that even faster implementations could be achieved by optimizing the algorithms for this platform.

8 Future Directions

8.1 Short Term

The next step for this project, as alluded to in Section 3, is to examine the creation of much smaller (particularly thinner) versions of the inertial hardware and the application possibilities which they would open. These sensor packages would be designed around a stacked configurable circuit board architecture, with a high-bandwidth transceiver on the bottom, a general microprocessor and ADC in the middle, and sensor board on top. This work, being done in collaboration with the National Microelectronics Research Centre in Cork, Ireland, will exploit not only the smaller inertial sensors now on the market, but also advanced fabrication techniques such as bare die bonding and stacking, to make as compact a system as possible.

One goal in building such systems (of the many they would enable) is to extend the work done in the Expressive Footwear project. Instead of measuring only the feet of a single dancer, the vision is of whole ensembles of dancers with both wrist and ankles instrumented. This would present challenges not only in terms of infrastructure, but also in the need for quick and efficient gesture recognition such that the large volume of data from the performers can be analyzed and responded to in real time.

8.2 Long Term

Given the ability to create simple portable computers (wearables, PDAs, etc.), the question is how this portability can be exploited. The most 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 (using little bandwidth) the scripts for a new set of gestures. These objects would then have a sense of their own movement and the ability to respond thereto.

The value of such a system can be seen in the example of a shoe-based physical therapy system which cues the wearer to damaging or awkward movements. Often, a person who is walking incorrectly will take notice only because their feet (ankles, shins, etc.) begin to hurt. However, they will likely be incapable of correcting the situation, because the feedback received has only a weak temporal coupling to the action, which makes learning more difficult[1]. Further, this feedback is based on the outcome of an action rather than its quality, which would be far more valuable[3]. If the shoe-based system mentioned above emitted different tones (e.g.) for different types of incorrect motion, it would be possible to greatly shorten the feedback latency, and therefore correct problems far more effectively.

9 Conclusions

We built an inertial gesture recognition framework comprising three parts. A compact inertial measurement unit is used to collect data from the object of interest and wirelessly transmit it to a personal computer. This data is then analyzed with a windowed variance algorithm to find periods of activity, and a generalized gesture recognition algorithm is applied to those periods. Gestures are recognized in an atomic form on an axis-by-axis basis using a number of physically based constraints, and those atoms can be combined into more complicated gestures using an output scripting system. This system was designed for use by application designers and allows output functions to be linked to specific gesture inputs. The overall framework was light-weight enough to be implemented on Palm III and this work points to a future where interface designers can use easily configured inertial sensors in a wide variety of settings.

Acknowledgements

We would like to express our thanks to our collaborators in the Synthetic Characters Group, who provided both the impetus and wonderful output functionality for the original version of the IMU. We appreciate the support of the Things That Think Consortium and the 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. J. Anderson. Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review*, 94(2):192–210, 1987. 19
2. C. Atkeson and J. Hollerbach. Kinematic features of unrestrained vertical arm movements. *Journal of Neuroscience*, 5(9):2318–2330, 1985. 13
3. W. Balzer, M. Doherty, and R. O'Connor, Jr. Effects of cognitive feedback on performance. *Psychological Bulletin*, 106(3):410–433, 1989. 19
4. J. F. Bartlett. Rock 'n' scroll is here to stay. *IEEE Computer Graphics and Applications*, 20(3):40–45, May/June 2000. 10
5. A. Y. Benbasat. An inertial measurement unit for user interfaces. Master's thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology, September 2000. 9, 15, 16, 17
6. B. Blumberg et al. `(void*)`: A cast of characters. In *Conference Abstracts and Applications, SIGGRAPH '99*, pages 169–170. ACM Press, 1999. 16
7. C. Chaplin, director. The Gold Rush, 1925. 100 mins. 16
8. Ascension Technology Corp. http://www.ascension-tech.com/. 11
9. Crossbow Technology, Inc. http://www.xbow.com/html/gyros/dmu6x.htm. 10
10. G. W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36(7):38–49, July 1993. 10
11. T. Flash and N. Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience*, 5:1688–1703, 1985. 12
12. S. Gibet et al. Corpus of 3D natural movements and sign language primitives of movement. In *Proceedings of Gesture Workshop '97*, pages 111–121. Springer Verlag, 1997. 13
13. F. Hoffman, P. Heyer, and G. Hommel. Velocity profile based recognition of dynamic gestures with discrete hidden Markov models. In *Proceedings of Gesture Workshop '97*, page unknown. Springer Verlag, 1997. 10
14. T. Marrin and J. Paradiso. The digital baton: a versatile performance instrument. In *Proceedings of the International Computer Music Conference*, pages 313–316. Computer Music Association, 1997. 10
15. J. A. Paradiso, K. Hsiao, A. Y. Benbasat, and Z. Teegarden. Design and implementation of expressive footwear. *IBM Systems Journal*, 39(3&4):511–529, 2000. 10
16. S. Prillwitz et al. Hamnosys. Version 2.0; Hamburg Notation System for sign languages. An introductory guide. In *International Studies on Sign Language and Communication of the Deaf Vol. 5*, page unknown, 1989. 13
17. L. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–86, 1989. 14
18. H. Sawada and S. Hashimoto. Gesture recognition using an accelerometer sensor and its application to musical performance control. *Electronics and Communications in Japan, Part 3*, 80(5):9–17, 1997. 10
19. D. Small and H. Ishii. Design of spatially aware graspable displays. In *Proceedings of CHI '97*, pages 367–368. ACM Press, 1997. 10
20. A. Wilson. *Adaptive Models for the Recognition of Human Gesture*. PhD thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology, September 2000. 14