A Reconfigurable Embedded Input Device for Kinetically Challenged Persons

Apostolos Dollas

Kyprianos Papademetriou

Nikolaos Aslanides

Dept. of Electronic and Comp. Eng.
Technical University of Crete
Akrotiri Campus
73100 Chania, Crete
Greece

Dollas@mhl.tuc.gr

Tom Kean

Algotronix Ltd.
P.O. Box 23116
Edinburgh EH8 8YB
Scotland
United Kingdom

Tom@Algotronix.com

Abstract. A new input device for kinetically challenged persons has been developed. This device is based on solid-state accelerometers to sense motion in space, a microcontroller to sample the data in real time, and an embedded FPGA to distinguish types of motion from programmable lists of motions. The FPGA computational model for the first version, presented in this paper, is an implementation of finite state machines (FSM) running in parallel, one for each type of motion which is detected by the system. The design is modular, allowing for different lists of motions and/or thresholds on input data to be incorporated with reconfiguration of the FPGA. A personal computer is used to determine the appropriate settings for each motion, which are then converted to FSM. The architecture of the system, types of motions it detects, and its performance characteristics are presented in this work.

1 Introduction - Motivation

Input devices in general are based on the translation of a stimulus to some action. With the pervasiveness of computers today, a large selection of input devices for computer I/O is available for kinetically challenged persons (large keyboards, eye-tracking devices, etc.)[1], [2], [3]. Essentially all research and associated products to date have focused on making computers more accessible to kinetically challenged persons [4], or, in using computers to perform tasks (e.g. turn lights on/off, etc.), whereas physical devices are controlled in traditional, mechanical ways (e.g. wheelchairs are still controlled by joysticks).

G. Brebner and R. Woods (Eds.): FPL 2001, LNCS 2147, pp. 326-335, 2001.

The next logical step is to detach the input devices from the need to be connected to a personal computer, and, by making them embedded, to associate such devices directly with desired tasks, such as wheelchair motion, or direct device control (without the need of a general purpose computer). Furthermore, for kinetically challenged persons, it is desirable to decouple the input device functionality from the actual manipulation of physical entities (e.g. joysticks, trackballs, keyboards, mouse devices) and rather rely on free motion which may be more comfortable. Taking example from advances in virtual reality applications for scientific and entertainment purposes alike, which employ 3D input and manipulation devices, we consider that an input device which is based on free motion of kinetically challenged persons would be useful in real-world applications, if it is reliable in its results, robust in its operation, reasonable in its size and power consumption, and inexpensive. Furthermore, the device should be easily tunable to the specific motion characteristics of different persons. We thus needed to solve the problem in a different design space than that of the corresponding virtual reality input devices.

The work, which is presented in this paper, is a first generation full-scale embedded system. It is complete with:

- motion sampling and off-line preprocessing by a personal computer,
- mapping different motions on finite state machines (FSM) within an embedded FPGA,
- running the algorithms in real time from a standalone system which is based on accelerometers as sensors, a microcontroller for the sampling of sensor data, and a FPGA for real-time processing,
- small and inexpensive construction.

Section 2 of this paper presents related work on sensors and real-time processing using FPGA's. Section 3 has the architecture and applications of the new system, motivating also why the use of FPGA's is needed in the system. Section 4 presents the input subsystem, which also motivates why a microcontroller is needed. It is followed by Section 5 on how a personal computer was used to sample and pre-process input data. Section 6 has the FPGA real time FSM computation model and results including area and speed characteristics. Lastly, there is a section on conclusions and alternative computation models.

2 Related Work

There is no related work which involves motion sensing and reconfigurable logic for persons with kinetic disabilities, so related work on each aspect of this project will be presented separately in this section. Regarding persons with motor disabilities, a real-time system has been designed, such that electromyographic (EMG) biosignals from cranial muscles and electroencephalographic (EEG) biosignals from the cerebrum's occipital lobe, are transformed into controls for two-dimensional (2-D) cursor movement[4]. The above system is based on a Digital Signal Processor (DSP) for its computations, and it is used for computer I/O.

Dynamic Time Warping (DTW) has been used for real time motion sensing of similar motions in a different time frame, in the context of a mobile robot [6]. This algorithm measures similarity between multivariate time series, and although Hidden Markov Models (HMM) outperform in terms of results the DTW algorithms, the latter are still used due to lower computational requirements. Another motion sensing project, with accelerometers as sensors [7], is using HMM's for (non real-time) signature verification and falls in the general category of "smart pens". A project combining FPGA's, DSP's and accelerometers (and other sensors too) is the "Air Drummer" [8]. The goal of this project is to make a virtual drumset which will enable the generation of drum sounds through the process of "air drumming" or "virtual drumming". Air drumming is the process of mimicking the actions of a drummer through the syncopated motion of one's upper and lower limbs. A part of a project called "3D Eye Tracking Device" [9] combines accelerometers and reconfigurable logic in order to measure successive head and eye movements. An FPGA is used as the interface between sensors and the processor board. An FPGA/DSP combination is used for the online, real-time acquisition and preprocessing of sensor data for medical purposes, and the entire system is used in conjunction with a computer.

On a different type of a real-time reconfigurable application, a system that supports reconnaissance, surveillance and target acquisition operation [10] comprises of microsensors, a DSP module and a FPGA module. The purpose of the FPGA is for signal processing (FFT) and for the control of various subsystems towards low-power consumption. FPGA's have also been used for real time 3D volume visualization [11]. It should be noted that most of these projects use large FPGA's, e.g. in [10] a large Xilinx Virtex FPGA is used, whereas in [11] multiple FPGA's are used. In our system such approaches would not meet size, cost, and power consumption limits for an embedded application.

3 Architecture and Applications of the System

The general characteristics of the new system, defined functionally and not in terms of implementation are:

- low cost solid state (non-mechanical) sensors which are reliable and suitable for motion sensing,
- real-time sampling of the data,
- motion detection for a large number of motions (vocabulary), tunable to different persons needs, and,
- ability to directly control devices through a direct electronic interface which can be easily adapted to mechanical interface control.

The key decision for the system was the usage of solid state accelerometers. Although other sensors were considered (e.g. Hall effect sensors), accelerometers were deemed to be sufficiently small in size, reliable, and low cost. The choice of accelerometers as input devices opened a slew of design considerations, as described in Section 4. From the general characteristics of the architecture we developed a system-level block diagram, shown in Figure 1.

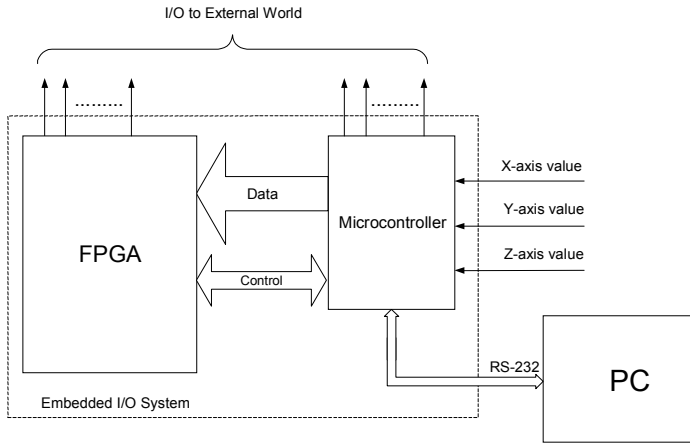

Fig. 1. A concept for the generic diagram of the system

The diagram in Figure 1 is generic, and as the project progressed there were considerations to leave out the FPGA if the microcontroller proved sufficiently powerful, or, to leave out the microcontroller if the FPGA was sufficient. The connectivity with the PC was envisioned as a rapid prototyping tool, so that several algorithms and their computational requirements could be assessed. In the standalone operation of the system the personal computer is not connected. In terms of applications, the original requirement for direct I/O was incorporated into the architecture with I/O lines, which can take any desirable logical values. Through optocouplers, direct interfacing to devices can then be accomplished (e.g. wheelchair Control). Moreover, these I/O lines can be easily converted to infrared (IR) ports, so that remote devices may be accessed (e.g. IR controlled door locks, IR controlled telephone device pick-up). Lastly, the I/O lines can be either decoded, leaving one line per device, or, encoded, allowing for larger numbers of devices to be accessed at the expense of more complex per-device decoding of multiple signals.

4 The Input Subsystem

Several generations of input subsystems were implemented and experiments were conducted in real-time input sampling. All versions were based on some kind of solid state accelerometer, a board with the ATMEL AVR 90S8515 microcontroller (but with different interfaces and software), and a personal computer to hold files of sampled data for post processing. The two most significant versions were

- the three-axes analog accelerometer system, with three Analog Devices ADXL05 accelerometers, and with a microcontroller system to sample and average input data, and,
- the two-axes digital (PWM) accelerometer based system, with a single two-axes Analog Devices ADXL210 integrated circuit.

We found that analog accelerometers have extensive noise, leading to the need for signal conditioning. Following extensive experimentation with the analog accelerometer system we determined that the inclusion of a third axis was not necessary for our purposes, as it complicated unnecessarily the computational requirements. Indeed, for a system which is attached during operation to a wrist, the degree of rotations is bounded in first place, and, limited partial rotations are permissible as long as the final system can detect the desired motion among a fixed list of motions.

The calculations to convert accelerometer output to fixed point digital data are performed by the microcontroller. Subsequently, in the development process the data is sent to the PC via RS232, whereas in the final embedded system the data is sent to the FPGA via an 8-bit port. Every X,Y acceleration value is interpreted as one 16-bit number. The first 8-bits transmitted to the FPGA represent the integer part of the acceleration value and the remaining 8-bits represent the fractional part. Thus, each sample comprises of four bytes of X,Y acceleration information, plus some control bits. This application takes roughly 2Kbytes, or 25% of the microcontroller on-chip program memory. It uses little of the CPU power, but most of the timers and counters. A system calibration process is done by the microcontroller and entails measurement of accelerometer output for zero acceleration. The calibration leads to a slightly different frequency of operation for each implementation; however, the standard frequency of sampled data (determined by worst case analysis) for the FPGA is 2.5 msec per set of samples, or, 400 samples/sec, where each sample is comprised of two 16-bit numbers. We note that a 50Hz sampling could detect all motion by a human hand, so the 400 samples afford an 8X oversampling with the associated robustness of data. The resolution of the system is 1.25 mg for the nominal, 2.5 msec sampling rate.

5 The Personal Computer Data Preprocessing System and Motion Detection Experiments

After sampling the input data, several strategies were considered, based on the processing of accelerometer data, or, results after one or two integrations (velocity and position, respectively). After considerable experimentation we concluded independently that previous results suggesting that these approaches led to excessive error [5], [12] were correct, and we abandoned such approaches in favor of processing the original data. The main reason for the excessive error is that since an integration is performed by multiplying the output of an accelerometer by t (velocity) and t^2 (position), any errors in the output of the accelerometer are also multiplied by these factors. Due to the inherent accumulation of absolute positional errors, inertial sensors (e.g. accelerometers) are much better suited for relative motion sensing or tracking.

Many experiments have been performed, and thirteen motions have been successfully sampled and processed. This “vocabulary” can be dynamic, and depending on the freedom of motion of a person it can be adjusted both in terms of sensitivity and in terms of range of motions. Eight of the thirteen motions detected by our present system are shown in Figure 2, whereas Figure 3 shows X-Y accelerometer data (in g)

over time for two motions. Detection of each motion is independent of the detection of the remaining motions, and in terms of resources it is subject only to the availability of logic blocks (CLB's) on a FPGA, leading to a broad range of potential implementations with varying cost and performance.

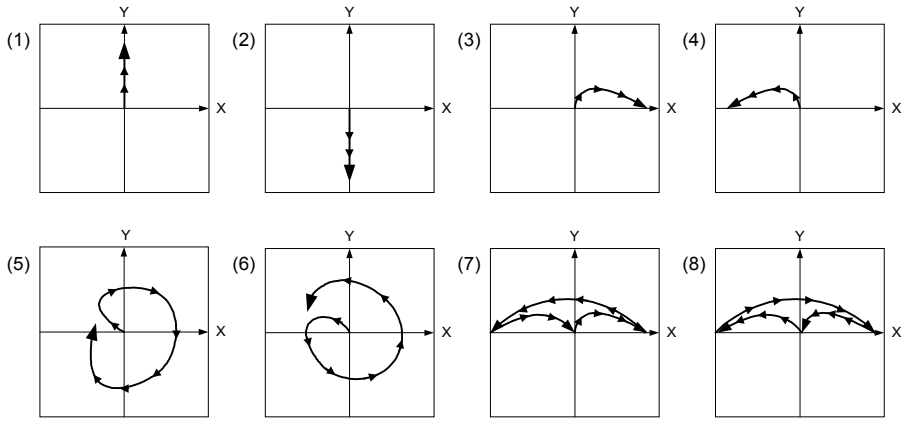

Fig. 2. Partial vocabulary of motions for the first generation of the system

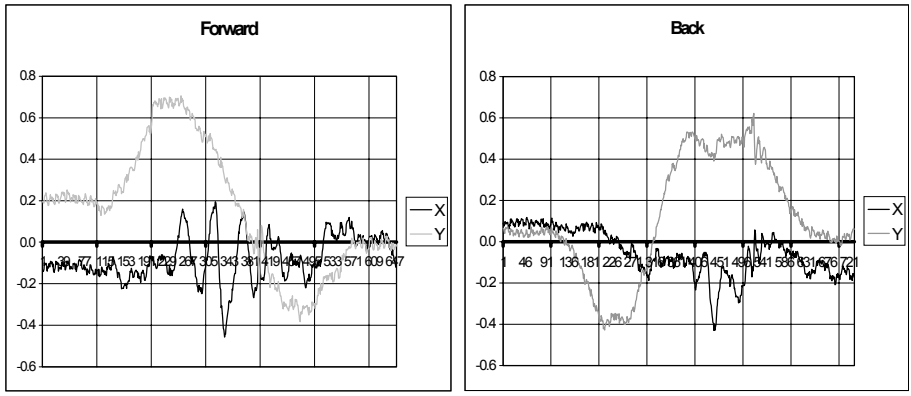

Fig. 3. Acceleration (in g) chart for motions (1) and (2) of Figure 2 (forward and back). The average duration of each motion is about 1.3 sec

Although a recent generation PC can perform a large range of computations, our experiments were computationally limited to reasonable models for final embedded implementation by microcontroller or FPGA. After consideration with several alternatives, we determined that a parallel implementation of finite state machines (FSM) on an FPGA gave a linear increase in the range of motions that could be detected with no performance loss. On the other hand, for a small number of motions

(2-4 motions) a microcontroller would be acceptable computationally. Especially in the case of a system that needs to be tuned to each individual's operational parameters and motion preferences with different FSM's, a microcontroller-based solution would be severely limited.

6 The Embedded FPGA Real Time Processing System - Results

We briefly motivated in the previous section why the FPGA turned out to be necessary for our application. We will now state why the microcontroller is also necessary. In the final version of the system, a standardized embedded system with broad usage will need to be "tuned" to each individual's motion preferences and abilities. This entails a process which is most conveniently performed by a personal computer. Whereas an FPGA could certainly perform the input sampling, it is tasks such as the calibration for each accelerometer, to which we referred in Section 4, and accurate timing of operations, in which a microcontroler is vastly superior due to timers, interrupts, RS 232 port, and other built-in resources. These make the cost-performance of a combined FPGA/microcontroller system superior to any FPGA-only or microcontroller-only approach. Whereas we could use any kind of FPGA or microcontroller, we opted for a Xilinx XC4010 low cost FPGA, and the ATMEL AVR 90S8515 microcontroller. Together with the accelerometers, such a system can be constructed for less than $50, which is consistent with our low-cost goal. The diagram of the final system is shown in Figure 4, and bears substantial resemblance to that of Figure 1, but comes after study to justify the role of each subsystem.

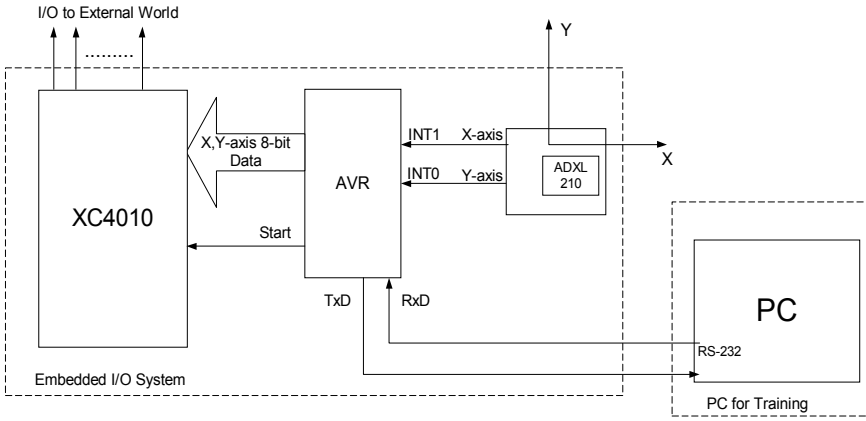

Fig. 4. Final diagram of the overall system

The computational model is that of parallel FSM's, each of which is comprised of stages for detection of values/ranges of X-Y data, followed by similar stages, or stages to wait for a predefined period of time. This way each motion is represented in terms of thresholds which need to be exceeded for the state to be active, followed by periods of "not examining the input", which are useful in avoiding local minima (from irregular motion or noise). These FSM's, one per motion that is desirable to be detected, are reset by a master FSM and operate in parallel. Once an FSM has detected a motion, its ID denotes which motion was detected, and the master FSM outputs the type of motion and resets the FSM's. The overall organization of the parallel detection of motions is shown in Figure 5.

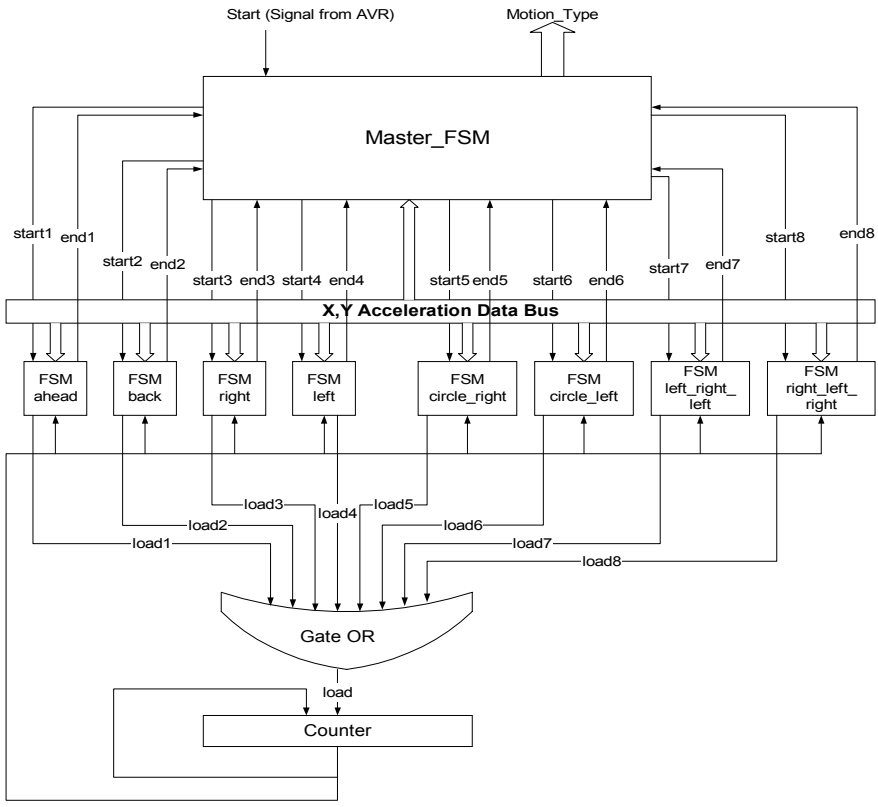

Fig. 5. Implementation of FSM's in hardware

In order to present quantitative results for the speed and the complexity of this scheme, Table 1 shows the usage of CLB's for the eight motions of Figure 2. Although usage of CLB's is not uniform per FSM, a good approximation is that a Xilinx XC4010 can hold the interface to the microcontroller, the master FSM, and all eight of these motion FSM's. Transferring the design to a larger (but still low cost) FPGA such as the Xilinx XC4044 allows for dozens of motions to be detected.

Motion	CLB's	Frequency
"Forward" (1)	30	59MHz
"Back" (2)	30	61MHz
"Right" (3)	34	53MHz
"Left" (4)	34	52MHz
"CircleRight" (5)	54	34MHz
"CircleLeft" (6)	54	34MHz
"Right-left-right" (7)	49	39MHz
"Left-right-left (8)	50	38Mhz

Table 1. CLB usage and speed operation while implementing 2 FSM's in XC4010XL.

The regular structure of the FSM's makes customization of the system through reconfiguration quite easy. Without a new synthesis process each time, once training has determined thresholds and delays for each FSM, a simple script can alter the configuration file, meaning that an application can be developed which customizes the system through reconfiguration, but without the usual synthesis process.

In terms of speed of the FPGA, the structure alone of the solution readily shows that it is a few orders of magnitude faster than what is needed. For example, typical speeds for the FPGA operation are in excess of 35 MHz, whereas we have a sample every 400Hz. Even if we transfer the 4-byte data from the AVR to the XC4010 at its nominal frequency of 8MHz bursts, the FPGA can still run at less than 10MHz and have most of its cycles unused (two cycles for each of the X-Y data set -four cycles total- are sufficient to determine the next state of each FSM). In practice the minimum allowable clock frequency is used to lower power requirements.

7 Conclusions and Alternative Computational Models

In conclusion, a low cost system has been developed to detect motions from an initial vocabulary range of thirteen motions (eight are shown in this paper). The system was implemented with an ATMEL AVR 90S8515 microcontroller, a Xilinx XC4010 FPGA, and Analog Devices ADXL 210 accelerometers. The prototype of the system comprises of several printed circuit boards and breadboards, connected with ribbon cable.In terms of computational models, the general architecture allows for much more sophisticated computations, including fuzzy logic algorithms, discrete Hidden Markov Models, and neural networks. Although we do have preliminary results with such models, we have found that these are much more intensive in resources, and to the extent that the simplistic FSM model that was presented here works adequately, it should be preferred for reasons of cost performance.

Subsequent steps of this work include clinical testing (which is underway), the development of training tools for the system to automatically determine thresholds for the FSM's and better integration of the subsystems in one or two printed circuit boards.

Acknowledgements

We wish to acknowledge funding for this work by the Greek Secretatiat of Research and Technology (GSRT) and the British Council, under the Britain Greece Joint Research Program, and, the EPET II European Union program under the Second Framework of Support to Greece. We also acknowledge generous donations to our educational and research activities by Xilinx Corporation, and electronics parts support by Analog Devices. We also thank University of Edinburgh Professors Gordon Brebner and John Gray for hosting at the Division of Informatics the Greek partners of the program during a week-long work visit to Edinburgh.

References

1. Lau C., O' Leary S. "Comparison of computer input devices for persons with severe physical disabilities", Am.J.Occup. Ther. Nov. 1993, 47:11.
2. Hawley M.S. et al. "Wheelchair mounted integrated control systems for multiply handicapped people", J. Biomed Eng. 1992 May, 14:3.
3. Luttgens K. et al Kinesiology-Scientific Basis of Human Motion" 8th Ed.,Brown and Benchmark Publishers, 1992, ISBN 0-697-11632-8.
4. Armando B. Barreto, Scott D. Scargle, Malek Adjouadi, "A practical EMG-based human-computer interface for users with motor disabilities", Journal of Rehabilitation Research and Development Vol. 37 No. 1, January/February 2000
5. C. Verplaetse "Inertial proprioceptive devices: Self-motion-sensing toys and tools", IBM Systems Journal, Vol 35, NOS 3&4, 1996.
6. T. Oates; M. D. Schmill and P. R. Cohen, "Identifying Qualitatively Different Outcomes of Actions: Gaining Autonomy Through Learning". In Proceedings of the Fourth International Conference on Autonomous Agents, pages 110-111, 2000.
7. Jim Broyles, Nicolas Karlsson, Rob Swanson, and Eduardo Velarde, "The Signature Verification System", January 1997, http://www.analog.com/publications/magazines/accel_news/issue6/2.html.
8. Jean-Louis, Racine Jeremy Risner, "Air Drummer", http://www.me.berkeley.edu/ME235/groups/psi
9. Kayser-Threde GMBH, 3D Eye Tracking Device, Technical Description, November 2000, Munchen.
10. S. Scalera, M. Falco, and B. Nelson, "A Reconfigurable Computing Architecture for Microsensors", In Proceedings of the FCCM April 2000, pages 59-67
11. A. Kaufman, F. Dachille, B. Chen, I. Bitter, K. Kreeger, N. Zhang, and Q. Tang, "Real-Time Volume Rendering", Special Issue on 3D Imaging of the International of Imaging Systems and Technology, 2000, pp.
12. C. Verplaetse, "Can A Pen Remember What It Has Written Using Inertial Navigation ?: AnEvaluation Of Current Accelerometer Technology", part of a class project for "Physics and Media" class, http://xenia.media.mit.edu/~verp/projects/smartpen/ruff_drapht.html.