Orient-2: A Realtime Wireless Posture Tracking System Using Local Orientation Estimation

A.D. Young
ayoung9@inf.ed.ac.uk

M.J. Ling
m.j.ling@ed.ac.uk

D.K. Arvind
dka@inf.ed.ac.uk

Institute of Computer Systems Architecture
School of Informatics, University of Edinburgh
Mayfield Road, Edinburgh EH9 3JZ, United Kingdom

ABSTRACT

A realtime posture tracking system has been developed using a network of compact wireless sensor devices worn by the user. Each device is a complete inertial/magnetic tracking unit which performs *in situ* orientation estimation based on its own sensor readings, using a complementary quaternion-based filter. Compared to existing systems which transmit raw sensor data to a PC for processing, it is shown that this technique reduces bandwidth requirements by 79% for typical usage. In combination with a time division multiple access scheme, this reduction allows for full-body tracking using 15 devices at a 64Hz update rate through a single 250kbps receiver. The data is applied to a rigid body model of the subject to provide a realtime display, and can be exported for use in major animation packages.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems

Keywords

Wireless sensor network, motion tracking, orientation estimation

1. INTRODUCTION

Motion tracking has many applications in fields ranging from animation to physical therapy. However, there are many shortcomings to traditional techniques. Source-based rangefinding and multi-camera computer vision systems have a limited tracking area, and the latter often require much manual post-processing of data. Joint angle sensors impose weight and movement constraints on the user. All these limitations restrict freedom of expression.

Inertial tracking systems are based on measurements made *in situ* on multiple parts of the body by sensor packages consisting of accelerometers, magnetometers and angular rate sensors in three axes. Data from these are used to estimate the orientation of each sensor package. Combined with an articulated rigid body model of the instrumented human, this information can be used to provide a complete posture. Dynamic acceleration data can also be integrated to provide estimates of velocity and position, albeit with drift errors.

There are several commercial inertial tracking products available today. Typical full body tracking systems, such as the Moven [1] motion capture suit, use multiple wired inertial sensors embedded into a Lycra body suit. While these now offer wireless transfer to a PC, the requirement for a specially fitted body suit limits the ease of use of such systems.

Fully wireless orientation sensors are available, such as the Wireless InertiaCube3 [2]. However, these transmit raw sensor data at high sample rates, with radio bandwidth limiting the number of devices that can be used simultaneously. The InertiaCube3 supports a maximum of three devices per receiver, yet full body orientation-based tracking of a human requires a minimum of 15 devices [3].

Existing work within the wireless sensor network community has typically concentrated on smaller devices, such as Eco [4, 5], with lower input dimensionality. Such systems are limited to applications such as gesture recognition and motion detection as they do not provide sufficient sensor data for drift corrected tracking with three degrees of freedom (3DOF). More advanced systems [6, 7] do provide the necessary sensor data to allow orientation based tracking, but again transmit raw sensor data with the associated high bandwidth requirements.

The *Orient-2* system was designed to provide a completely wireless inertial/magnetic 3DOF tracking system, supporting real-time full-body posture tracking through a single receiver, at an update rate suitable for high-framerate animation. A major feature of the design was to perform orientation estimation locally, to minimise bandwidth requirements and eliminate estimation errors due to packet loss.

The hardware platform has been designed to be as compact as possible to minimise restriction of the user, whilst maintaining sufficient battery life for practical use. Typical applications were envisaged to include continuous operation for 1-2 hours, e.g. for a stage performance, or intermittent shorter “takes” over a longer period in an animation studio. It is intended to be flexible enough for use in non-human motion tracking, including onboard storage for data logging in standalone applications.

EmNets'07 June 25-26, 2007, Cork, Ireland

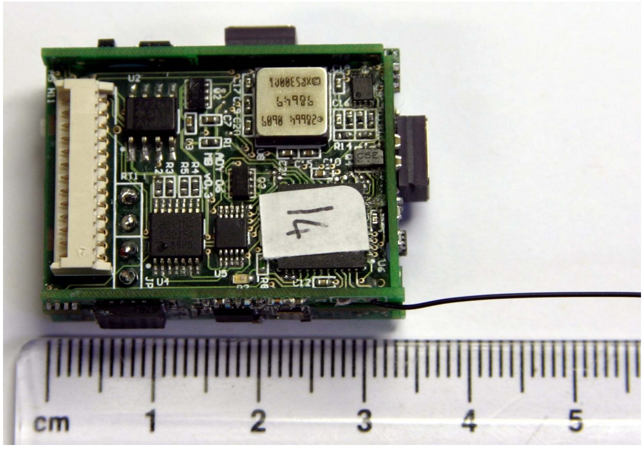


Figure 1: Orient-2 Device

2. SYSTEM DESIGN

The *Orient-2* system comprises the device hardware, onboard firmware, and PC-based software for data visualisation and export. Currently, one complete device is also used to act as a PC-connected basestation, though this will be replaced by a simpler USB device without the sensor components.

2.1 Hardware

The device hardware consists of four interlocking PCBs. These are assembled to form a three-dimensional structure surrounding a 120mAh lithium polymer battery. The complete unpackaged structure, shown in Figure 1, measures $36 \times 28 \times 11$mm including the battery and all components except for the whip antenna for the radio. A 13-way connector provides access for charging, serial communications and debugging.

The processor is a Microchip dsPIC 30F3014 16-bit microcontroller. This device was selected for its digital signal processing features, and ability to perform rapid sampling sweeps of multiple analogue inputs to its onboard 12-bit ADC. The processor is clocked from a 7.37MHz internal RC oscillator, and a 32,768kHz crystal oscillator is used to provide a real time clock. The processor has 2KB of SRAM and 24KB of program memory, together with a 1KB onboard EEPROM. A 4MB external Flash memory chip is included for data logging.

A Chipcon CC1100 transceiver is used for radio communications, which can function in either 433MHz or 868MHz ISM bands. The CC1100 is capable of operating at data rates up to 500kbps; however, due to European regulatory limits on bandwidth only 250kbps may be used [8].

The sensors used are a Freescale MMA7260Q three-axis accelerometer with processor-selectable range from ± 1.5 to $6g$, two Honeywell HMC1052 two-axis magnetometers, and three Analog Devices ADXRS300 MEMS rate gyroscopes. The gyros have a normal maximum measurable rate of $\pm 300^\circ$/s, but a processor-controlled analog switch is used to alter the output feedback loop to produce selectable ranges of $\pm 300, 600, 900$, and 1200°/s [9].

2.2 Software

The firmware on the device processor carries out all aspects of realtime operation, including sampling and normalisation of sensor data and orientation estimation, and transmitting and handling data and control packets. Its behaviour, configuration and calibration settings can be set remotely over the radio, and stored in the onboard EEPROM for reuse on next startup.

On the PC side, a Python module implements the protocol for communicating with a basestation device over the serial port, and exposes a higher-level API for controlling a network of devices. This can be used for scripted data capture, as well as by a Java-based tool developed for rigid body modelling and control of capture sessions.

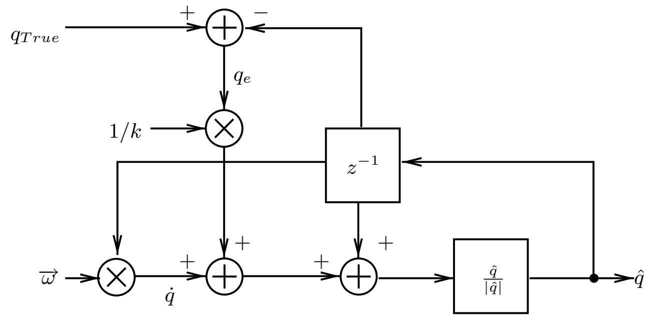


Figure 2: Complementary Quaternion Filter Loop

2.3 Orientation estimation algorithm

The goal of the orientation estimation algorithm is to calculate the relative rotation between the local device coordinate frame and a global reference frame. The global frame is shared by all devices in a network and is defined to have its positive X-axis pointing in the horizontal direction of magnetic north, positive Z down and positive Y east.

The algorithm chosen was a complementary quaternion-based filter, based on the work of Bachmann [10]. This filter uses two orientation estimates, with differing noise characteristics, to produce a single orientation estimate combining the advantages of each. It is important to note that due to the complementary nature of the filter there is no group delay in the filter response. The choice of filter algorithm was influenced by the relative simplicity of the implementation, compared to more complex Kalman filters [11], allowing the filter to be implemented locally on each device.

The first estimate is a relative estimate produced by integrating the outputs of the rate gyroscopes. This estimate produces smooth movement with very low latency, but suffers from low frequency drift due to gyroscope null offset accumulation and numerical integration errors.

The second estimate is an absolute estimate calculated from the measured acceleration and magnetic field vectors. These two vectors are used to directly calculate the rotation matrix relative to the Earth-fixed frame. The acceleration vector provides one basis vector, the magnetic field projected into the X-Y plane provides the second, and the cross product the third. This estimate has the advantage that it produces an absolute estimate but suffers from movement-induced error due to dynamic accelerations.

The absolute and relative estimates are used as inputs to the complementary filter shown in Figure 2. The filter operates by taking the angular rate input from the gyroscopes, $\overrightarrow{\omega}$, performing a quaternion multiplication with $\hat{q}$, the orientation estimate, to produce a rate quaternion $\dot{q}$. An error factor, q_e, is calculated by subtracting $\hat{q}$ from the absolute estimate, q_{True}. This error factor is scaled by the filter coefficient, k, and added to $\dot{q}$. The resulting quaternion is accumulated to produce the new orientation estimate.

The absolute estimate is only calculated if the magnitude of the acceleration vector is close to $1g$. Movement induced error is thus reduced by discarding corrections obviously affected by dynamic accelerations.

2.4 Communications

A common packet format is used for both serial and radio communications, with a variable length header format used to minimise the communication overhead for all packets. On the serial line, packets are transmitted immediately; on the radio, two MAC algorithms are implemented.

A simple Carrier Sense Multiple Access (CSMA) scheme is used for remote configuration of devices when idle. This can also be used to transmit data packets, but is unsuitable for realtime capture using multiple devices; in the presence of unsynchronised devices sending data with isochronous transmission timing, starvation of individual devices is frequent. Additionally, the use of the radio receiver for carrier sensing and reception of unscheduled control packets consumes a significant amount of power.

To overcome these problems a Time Division Multiple Access (TDMA) scheme is used during capture. Transmissions are grouped into frames with one transmission slot per device. At the start of each frame, all devices synchronise their local clocks to a synchronisation packet transmitted by the basestation. Once synchronised, a device can safely shut down its radio to save power until it is due to transmit data, and again after it has transmitted. At the end of the frame all devices turn on their receivers for a short window to receive the next synchronisation packet. Missed synchronisation packets due to channel conditions can be tolerated for a programmable time, which is set to prevent the relative drift between device clocks from causing collisions. If a synchronisation outage exceeds this time, devices will stop transmitting and switch on their receivers continuously to resynchronise or receive alternative commands. Data capture can be stopped explicitly by setting a flag on the synchronisation packets.

2.5 Body modelling

By mapping the received orientation data from multiple devices onto a rigid body model, the posture of a subject can be tracked in real time. A Java tool, MotionViewer, has been written to perform this function.

A rigid body model is constructed from a hierarchy of joints. The model is constructed from multiple joint objects, stored in a tree structure, each specifying an offset from their parent joint in the parent's local co-ordinate frame. This internal model was chosen as it simplifies export to existing animation software.

Sensor devices are mapped to individual joints, and capture initiated from the software. At the end of each data frame the new orientations from the devices are applied to the body model and the relative positions of all joints updated by traversing the tree.

Additional tasks can be performed at frame updates by writing plugins. Currently three plugins have been implemented: an OpenGL visualisation allowing live monitoring of subject posture, a BioVision Hierarchy (BVH) [12] export plugin allowing export to commercial animation software, and an experimental absolute positioning plugin.

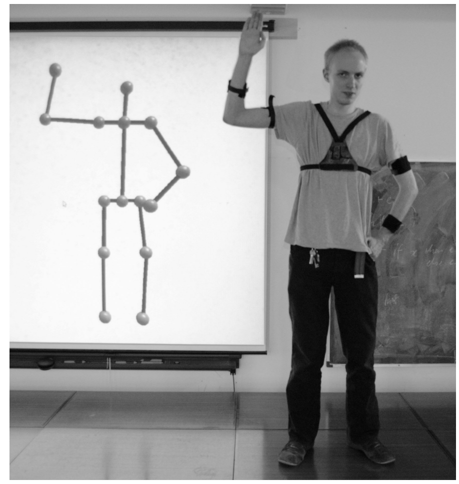


Figure 3: Example output from the MotionViewer software

3. OPERATION

Compared to many other motion/posture capture systems, the *Orient-2* system is very quick to set up and unobtrusive to the user. The sensors are compact and can be worn on comfortable straps, underneath or over clothing. The only equipment that needs to be deployed in the tracking area is the basestation device with serial connection to a PC. Since little processing power is required on the PC, the basestation could easily be connected instead to a PDA carried by the user. By providing a gateway to a wireless LAN or cellular network, this would allow an almost unlimited tracking area.

Currently the system is limited to tracking posture only, but full spatial motion capture could be achieved by combining the *Orient-2* system with a simple vision or source-based position tracker. Only a single reference point needs to be tracked to anchor the entire body model. There are also a number of techniques that can be used to derive motion data from the inertial sensor data alone. Future work will be directed towards implementing and combining these approaches.

3.1 Typical procedure

A typical capture session involves the following steps:

1. A model of the subject being tracked is prepared. This specifies the connectivity of joints, their offsets, and a default calibration stance. The model can be imported from an existing BVH file, e.g. from a previous optical capture session.

2. Sensor devices are placed onto the body segments to be tracked. The assignment of device IDs to body segments is set in the MotionViewer tool.

3. Devices are started remotely and begin transmitting orientation updates. A realtime display is visible immediately.

4. The alignment between sensors and body segments is fixed by having the user stand in the calibration stance.

5. Capture proceeds and posture information is logged to a file in BVH format.

3.2 Calibration

Each *Orient-2* device must be calibrated to compensate for the scaling and offset of its individual sensor inputs, before the sensor data can be used for accurate orientation estimation. This is largely a one-off operation for each device, although the calibration is sensitive to differences in temperature as well as gradual drift in component values, and may need to be repeated occasionally.

Satisfactory calibration can be performed by manually putting the device through a series of orientations and rotations, prompted by a script on the PC that determines the appropriate calibration values from the resulting data and transmits these back to the device. In the future more accurate automated calibration will be performed to increase accuracy, using a PC controlled rotation platform.

3.3 Data export and usage

The BVH output has been used with AutoDesk MotionBuilder. This tool allows mapping motion capture data from a simple rigid body model onto a detailed character with its own kinematic constraints. Many other commercial animation packages also support the import of BVH data files.

The output data requires no post-processing to be performed. Any errors in the initial body model dimensions and sensor alignment can be corrected after capture.

4. PERFORMANCE

4.1 Bandwidth and latency

In existing comparable systems, where data is relayed to a central point for processing, each device must transmit sampled data from all sensors at a high update rate in order to reduce the effect of cumulative integration errors in the processing of rotation rate data. In such a system a typical packet payload would consist of nine 12-bit values. At an update rate of 180Hz[1] this corresponds to a data rate of 19,440bps per device. Any packet losses will introduce integration errors in the final orientation estimate, though interpolation techniques can be used to mitigate these to some degree.

Implementing the orientation estimation filter locally on each device removes the sensitivity of the estimate to packet loss, and reduces update size as only the output of the filter needs to be transmitted. With quaternions of 16-bit values, the data rate per device for 180Hz is reduced by 41% to 11,520bps. Additionally, as sampling of the filter output has only to meet the Nyquist criterion with regard to the movements of the subject, the update rate can be substantially reduced. An update rate of 64Hz is more than sufficient for animation purposes and requires only 4,096bps, 21% of the raw datarate. At this update rate 15 devices can be tracked simultaneously through a single basestation, allowing for full body tracking.

The sensor sampling and filter update rate used on the device is independent of the filter output sampling rate, though it should be a multiple for constant latency. Currently a rate of 256Hz is used, though this could easily be set higher for less integration error at some cost in power; sampling of the inputs takes 320 µs and the filter update requires only 280 µs.

[1]Typical rate used in commercial tracking devices [2].

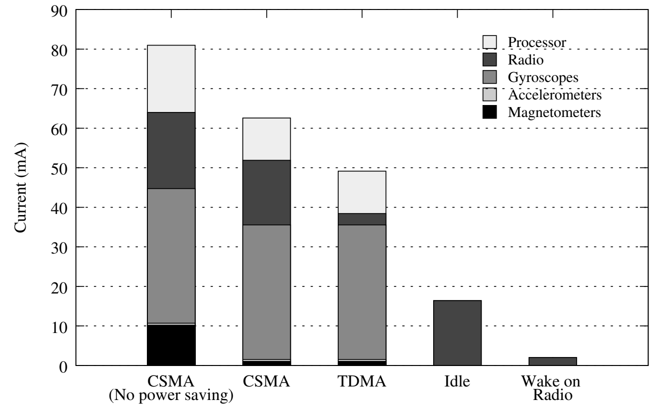


Figure 4: Current Consumption

In order to accurately sample body posture with multiple devices, sampling operations are synchronised to the TDMA schedule during capture. The delay between the synchronised sampling at the start of a TDMA frame and the final transmission at its end introduces a lower bound on the system latency of one frame period. The frame period is given by the number of devices being tracked, multiplied by the transmission slot time. At 250kbps, a 1ms slot time is used, which provides sufficient time to transmit a quaternion update packet with some safety margin. The system latency thus has a lower bound of 15ms for full-body tracking.

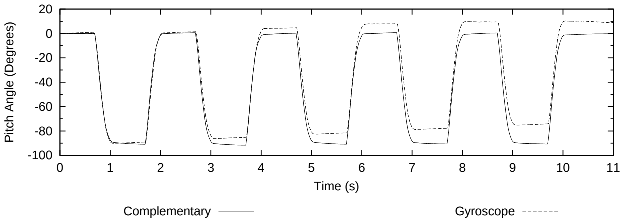
4.2 Power consumption

The *Orient-2* hardware and firmware have been designed to conserve energy wherever possible. All subsystems support power control options and every effort has been made to reduce power consumption through duty cycling. In addition the devices can be placed in a low power sleep mode when not in use, using the built in receiver polling function of the CC1100 radio to wake them remotely.

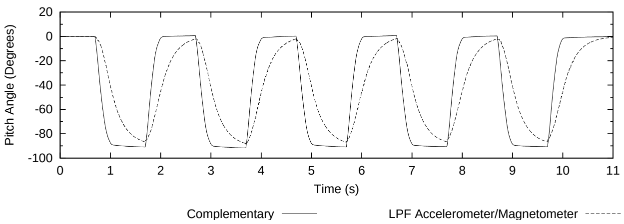
Average power reduction from sensor duty cycling and the use of TDMA can be seen in Figure 4, which plots total measured current draw from the battery in different modes, with a usage breakdown based on a combination of measurements and estimates from device datasheets. In tests using the TDMA mode the devices have been run for 140 minutes with the 120mAh battery.

The largest current draw is from the rate gyroscopes, which require 6mA each at 5V. The 5V supply is generated from the 3.7V nominal battery using a charge pump regulator, which was found to be running at an efficiency of only 50-60%. This inefficiency was compounded by the inability to duty cycle the gyroscopes due to their long (35ms) startup time. Future design iterations will seek to improve the conversion efficiency.

A significant amount of power is wasted due to a silicon bug [13] in the current revision of the dsPIC preventing the use of the sleep mode while maintaining the RTC. This causes the processor to draw an additional 9mA on average by limiting it to idle mode.



(a) Gyroscopic Inertial Estimate vs Complementary Filter



(b) Accelerometer/Magnetometer Estimate vs Complementary Filter

Figure 5: Orientation estimates during repeated 90° **rotations**

4.3 Orientation estimation

The correct behaviour of the orientation algorithm has been verified through visual tests and through the use of a servo motor platform. Figure 5 shows results from a test using three devices, one using only inertial estimation, one using absolute estimation and one using the complementary filter, mounted on the platform and subjected to repeated 90° pitch rotations. All devices were calibrated by hand to find offset and scaling factors. The effect of gyroscope null bias and correction is clearly visible in Figure 5(a) with the pure inertial estimate having drifted approximately 10° over an eleven second run. Figure 5(b) illustrates the loss of high frequency information and significant delay introduced by the low pass filtering of the absolute estimate required to reduce movement errors.

The effect of altering the filter coefficient, k, has been explored through mathematical analysis and empirical testing. Essentially k represents the blending factor between the two orientation estimates. Increasing the value of k increases output resolution, by decreasing noise from the accelerometers, but increases initial settling time and static output error, however if k is increased too far then gyroscope bias can no longer be corrected. Measurements of filter output with the device in a static orientation and k=128 have shown output resolution to be limited by noise in the inertial gyroscope estimate.

5. CONCLUSION

A fully wireless posture tracking system has been developed. The use of *in situ* orientation estimation has avoided errors due to communication problems, and reduced bandwidth requirements. The reduction is sufficient to allow real-time full-body tracking over a single 250kbps radio channel The resulting ability to use inexpensive, low-power radio hardware helps the devices maintain a useful battery life whilst being compact enough to present minimal obstruction to the user.

Videos of the complete system in operation are available at: `http://www.tardis.ed.ac.uk/~ayoung/`

6. ACKNOWLEDGEMENTS

This research was supported by grants from the Scottish Funding Council (R37329) and the UK Science and Engineering Research Council (C523881) for the Research Consortium in Speckled Computing.

7. REFERENCES

[1] *Moven*, Xsens Motion Technologies. [Online]. Available: http://www.xsens.com/download/MovenLeaflet.pdf

[2] *Wireless InertiaCube3 Datasheet*, Intersense. [Online]. Available: http://www.intersense.com/uploadedFiles/Products/InertiaCube3.pdf

[3] R. Zhu and Z. Zhou, "A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 2, June 2004, pp. 295–302.

[4] C. Park, J. Liu, and P. H. Chou, "Eco: an ultra-compact low-power wireless sensor node for real-time motion monitoring," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, 2005, pp. 398–403.

[5] C. Park, P. H. Chou, and Y. Sun, "A wearable wireless sensor platform for interactive dance performances," in *Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on*, 2006, p. 6 pp.

[6] R. Aylward, S. D. Lovell, and J. A. Paradiso, "A compact, wireless, wearable sensor network for interactive dance ensembles," in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, 2006, p. 4 pp.

[7] A. Lynch, B. Majeed, B. O'Flynn, J. Barton, F. Murphy, K. Delaney, and S. C. O'Mathuna, "A wireless inertial measurement system (wims) for an interactive dance environment," *Journal of Physics: Conference Series*, vol. 15, pp. 95–100, 2005.

[8] *Application Note AN039 - Using CC1100/CC1150 in European 433/868 MHz bands*, Chipcon.

[9] *ADXRS300 Application Note AN-625*, Analog Devices.

[10] E. R. Bachmann, "Inertial and magnetic tracking of limb segment orientation for inserting humans into synthetic environments," Ph.D. dissertation, Naval Postgraduate School, Monterey, California, 2000.

[11] E. Foxlin, "Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter," in *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, 1996, pp. 185–194, 267.

[12] "Bvh format." [Online]. Available: http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html

[13] *dsPIC 30F3014/4013 Rev. A1 Silicon Errata*, Microchip.