Human Motion Tracking With Wireless Wearable Sensor Network: Experience and Lessons

Jianxin Chen[1], Liang Zhou[1], Yun Zhang[1], David Fondo Ferreiro[2]
[1]Key Lab of Broadband Wireless Communication and Sensor Network Technology,
Nanjing University of Posts and Telecommunications, Nanjing, 210003, China
[e-mail: chenjx，liang.zhou, zhangyun@njupt.edu.cn]
[2] Local 14 Campus Universitario de Vigo
Vigo, Pontevedra, 36310, Spain
[e-mail: dfondo@gradiant.org]
*Corresponding author: Jianxin Chen

Received November 23, 2012; revised February 18, 2013; accepted February 18, 2013; published May 31, 2013

Abstract

Wireless wearable sensor networks have emerged as a promising technique for human motion tracking due to the flexibility and scalability. In such system several wireless sensor nodes being attached to human limb construct a wearable sensor network, where each sensor node including MEMS sensors (such as 3-axis accelerometer, 3-axis magnetometer and 3-axis gyroscope) monitors the limb orientation and transmits these information to the base station for reconstruction via low-power wireless communication technique. Due to the energy constraint, the high fidelity requirement for real time rendering of human motion and tiny operating system embedded in each sensor node adds more challenges for the system implementation. In this paper, we discuss such challenges and experiences in detail during the implementation of such system with wireless wearable sensor network which includes COTS wireless sensor nodes (Imote 2) and uses TinyOS 1.x in each sensor node. Since our system uses the COTS sensor nodes and popular tiny operating system, it might be helpful for further exploration in such field.

Keywords: Wireless wearable sensor network, motion tracking, communication protocol, real time

This work is partly supported by the State Key Development Program of Basic Research of China (2011CB302903), by National Natural Science Foundation of China (Grant No. 61201165 and No. 61271240), by Research Grant of Spain Gradiant center，by Priority Academic Program Development of Jiangsu Higher Education Institutions, and by Nanjing University of Posts and Telecommunications Foundation (Grant No. NY211032，NY207021 and NY 211063).

http://dx.doi.org/10.3837/tiis.2013.05.004

1. Introduction

Human motion tracking has been a hot research topic in the computer graphics field due to the wide range of applications. One of the most successful applications is human computer interaction games such as Ninetendo Wii, Microsoft Xbox 360, and Sony PlayStation 3. Besides that, human motion tracking is a promising technique for remote nursing and rehabilitation, in which the patient at home might be nursed or helped by a remote doctor via the Internet [1]. Similar applications can be found in the intelligent media [2], movie making [7] and sports training [6]. Additionally, motion tracking can be used for robot control [9] and new weapon training [17].

Previous techniques for human motion tracking are either cumbersome (mechanical tracking [11], inertial/magnetic tracking systems [15][16][18]), source-dependent (camera tracking [13], optical tracking [14]) or context aware(magnetic tracking [12]), which make them expensive or usage in rather limited scenarios. Recently due to the development of MEMS (Micro Electro-Mechanical Sensor) and wireless communication technique, it is potential to extend the applications of motion tracking by wireless wearable sensor network using MEMS inertial sensors such as accelerometer and gyroscope.

Inertial sensors are mainly used for orientation measurement in 3D (dimension) space. Generally to obtain a 6-DoF (degree of freedom) measurement, there are 3-axis accelerometer and 3-axis gyroscope, which are complemented by 3-axis magnetometer due to the inherent shortcomings of each other. For example, gyroscope suffers from drifting problem for long duration; Accelerometer might measure attitude in a static condition; Magnetometer might measure azimuth in horizontal plane. Therefore, these three kinds of sensors are complemented for motion tracking. With inertial sensors for motion tracking, human limb is divided into more than 15 independent segments according to the human biomechanics [20]. Each segment is attached with a 6-DoF measurement unit which includes a 3-axis accelerometer, 3-axis magnetometer and 3-axis gyroscope. Therein, for a whole body motion tracking there are required more than 15 6-DoF measurement units.

Currently, in the motion tracking system with inertial measurement units, all these measurement units are wired to a central controller (CC) [15][16][18]. This CC is in charge of reading sensor data from these measurement units and transmitting to the PC via WiFi [29] for motion rendering. As WiFi is not energy-saving, it requires high power for work prolonging. In addition, a high sampling rate for each measurement unit is required, e.g. 50 Hz - 200 Hz to guarantee the tracking accuracy and reliability. As nine channels (3-axis accelerometer, 3-axis magnetometer and 3-axis gyroscope) data in each measurement unit and WiFi communication, it is required a powerful storage battery for CC so as to provide enough energy for CC and all measurement units. Herein, the storage battery is heavy. On the other hand, all measurement units around limb are wired to CC, such wire line more or less adds burden to human motion. Even particular cloth is made to hold all these measurement units, CC and wire lines. Finally the weight of this cloth is nearly about 11kg [18]. Such extra load affects human motion. Recently, with the development of low power wireless communication technique, it is possible to use the wireless wearable sensor network for human motion tracking which combats the above problems since each sensor node is, light small and independent, which makes it more flexible for human wearing. Furthermore, the low power wireless communmication technique such as Zigbee/IEEE 802.15.4 consumes much less energy compared to WiFi. Therein it is possible to avoid the weight storage battery.

Orient 2 was the first system built with wireless wearable sensor network [19], which includes 15 wireless wearable sensor nodes. In such system each sensor node performs the on-situ data fusion for three kinds of sensors which reduces the data transmission via wireless communication, herein conserving energy. Besides that, other systems such as [8][6][25] using wireless sensor nodes just track parts of human body. In [8], the authors used 6 sensor nodes to track arbitrary human upper body motion including a trunk and two limbs. In [6], a 6 DoF wireless sensor node is designed to precisely measure the dynamics of a golf club, which supports golf swing training, custom club fitting, and club design. In [25], a compact wireless, wearable system was designed to measure signals indicative of forces, torques and other descriptive and evaluative features that the human body undergoes during bursts of extreme physical activity. In [26], the authors present a high rate wireless sensor network usage for interactive dance (1Mbit/s RF rate), in which a choreographer translates the movements of multiple dancers into real-time audio or video content to accompany the performance. The system supports processing of up to 25 nodes with 100 Hz full state updates. [30] [31] focused on the inertial/magnetic unit implementation such as Wagromag, *ZIMUED* which might be used for part of limb's rehabilitation. Other system using one or several inertial/magnetic units for Parkinson's disease [32], upper limb rehabilitation [33], gait analysis [34], and etc..

However, all these wireless sensor nodes for motion tracking are particular designed, which requires specific technique to accomplish. Therein it is not easy for fast deployment. In addition, current wearable sensor node has no operating system. As we have known, in each sensor node, there are multiple tasks in each sensor node, e.g. radio receiving, transmitting, sampling, data fusion, time synchronization, and media access control, etc.. If there is no operating system embedded, it is not easy to schedule all these tasks. In this paper, we build a motion tracking system using the COTS (commercial-off-the-shelf) wireless wearable sensor nodes and sensor boards, in addition with a popular tiny operating system for low power environments. It not only fastens the system building procedure, but also extends the scalability and flexibility of the system. Our system is not the first wireless wearable sensor network for motion tracking, but it is the first such system built with COTS products and using the low power operating system-Tinyos-1.x.We show that with the operating system it is more flexible to schedule multitasks and cooperation among multiple wireless sensor nodes which construct a wearable wireless sensor network (WWSN). Additionally, although TinyOS-1.x is not a real time operating system, our test results show that it has the potential to use for some real time applications depending on the platform.

The remainder of this paper is organized as follows: Section II introduces the system architecture including sensor nodes, sensor boards, the embedded operating system and wireless communication protocol. The detail of a TDMA implementation in the tiny operating system is described in Section III. Section IV discusses performance aspects such as packet loss, energy efficiency and the accuracy of orientation tracking. Lessons learned from the building of a prototype system are described in Section V. Section VI draws the conclusion and lists the future work.

2. System Architecture

2.1 System Introduction

Our wearable motion-tracking prototype system consists of 9 wearable sensor nodes and a base station, which constructs a wireless wearable sensor network (WWSN) with a star topology. The nine sensor nodes are attached the main segments of limb such as: left

upper/forearm, right upper/forearm, left upper/lower leg, right upper/lower leg and abdomen (**Fig.1**). Each wearable sensor node includes three kinds of MEMS sensors: 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer to complementarily measure the position of independent segment. In each sensor node, the on-situ data fusion is accomplished by extended Kalman filtering algorithm to obtain the real time orientation of each segment so as to reduce the data transmitting amount, which herein conserves the energy. All sensor nodes transmit the measured orientation information to the base station for real time motion reconstruction and rendering. Since in each sensor nodes there are multiple tasks such as data sampling, fusion, radio transmitting and listening, we use the popular tiny operating system—TinyOS, which also reduces the implementation complexity due to plenty of supports.

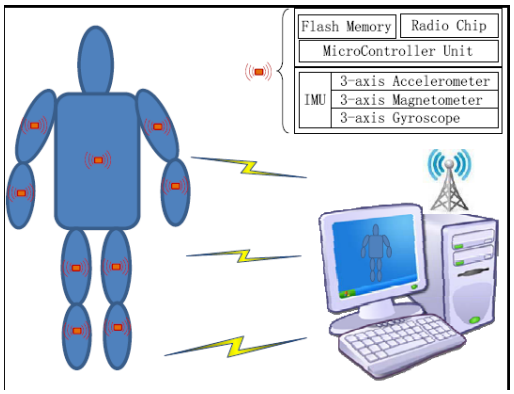


Fig. 1. Human motion tracking system with wireless wearable sensor network

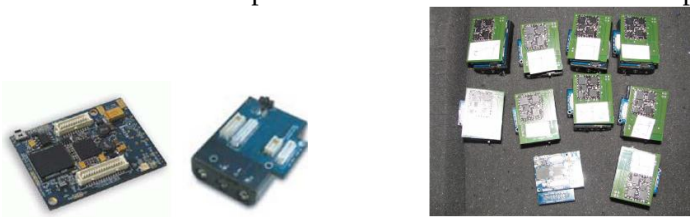
The base station acts as a gateway between the WWSN and the PC. It not only receives the data from the WWSN, but also might send data to the wearable sensor nodes such as control (configuration) or synchronization information. The control information depends on the application requirements, e.g. some special segments are of interest to the users for rehabilitation diagnosis. The time synchronization is required for TDMA implementation so as to coordinate all wearable sensor nodes for reliable communication. All data from the sensor nodes are forwarded by the base station to the PC for motion reconstruction and rending in a 3D virtual space.

2.2 Wireless Sensor Node

Our system uses mostly off-the-shelf hardware. Specifically, the wearable sensor nodes are Imote2 [3], which has the 32-bit microcontroller (MCU)–Intel Xscale PXA 271 [4]. This processor can work with dynamic frequency ranging from 13 to 416 MHz, making it possible to use in an energy-saving scenario. Although energy saving was not a goal of our project, moving in that direction is now easier since we have started with this platform. Furthermore, three 1.5V batteries in each Imote2 provide enough power for the inertial sensor board which usually requires 3.3V.

The left of **Fig. 2** depicts the sensor node of Imote2 and the battery board. The right of **Fig. 2** depicts the nine wearable sensor nodes in addition to one interface board for our system. Each sensor node is attached with a sensor board. The interface board combined with Imote2 IPR 2400 act as a base station connected to the PC via the USB cable. We choose this sensor node for the consideration of its powerful computation capability which might perform the on-situ data fusion algorithm. In addition, the MCU with high performance might guarantee the low delay for multiple tasks processing and scheduling under tiny operating system even in

rather low power constraint context. Besides that, Imote2 is a convenient low power platform which already had a few software components that we could re-use for our purpose.



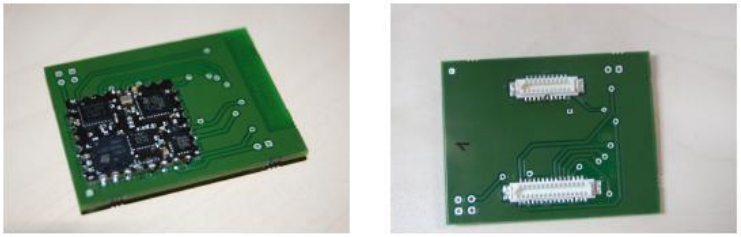
Imote2 Nine Sensor Nodes With Sensor Board

Fig. 2. Imote2 Sensor Nodes

2.3 Sensor Board

Existing sensor boards for wireless sensor network cannot monitor rigid body orientation in 3D space. Although there are few off-the-shelf IMUs with 6 DoF, they are not specially designed for the Imote2 platform. If to use them, we need to design an adaption board to connect to the Imote2 platform. Currently there are two kinds of IMUs with 6 DoF: with in-situ data fusion or external data fusion. Both of them include 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer. The IMU with in-situ data fusion algorithm outputs the orientation representation such as quaternion or Euler angle, while the IMU without in-situ data fusion algorithm outputs the original sensor data via nine channels from 3-axis accelerometer, magnetometer and gyroscope. All this data are dealt with in the PC connected with the base station. Compare to the former IMU, this one has more data output for each sample due to nine channels. For example, if data from each channel is represented with 2 bytes, there are 18 bytes of each sample. While for the former IMU, only 8 byes are enough if orientation is represented with quaternion and each quaternion element is represented with 2 bytes. This could save much energy since wireless communication costs most of power in energy- constrained context. Herein, we prefer to the one with in-situ data fusion algorithm from the consideration of energy-saving since our platform is powered by battery.

Currently, VN 100 satisfies our requirements [5]. It has a 32-bit microcontroller inside, which performs an extended Kalman filter algorithm for data fusion. In VN 100, both orientation representations are provided, i.e. Euler angle or quaternion. They can be updated with a maximum frequency of 200 Hz. Besides that, the output might include sensor raw information from 3-axis accelerometer, 3-axis gyroscope or 3-axis magnetometer.



Top Bottom

Fig. 3. Adapter board with IMU for Imote2

To combine IMU VN 100 with sensor node Imote2 platform, we designed an adapter board for them as in **Fig. 3**. On the top of this board, it is VN 100. On the other side, it includes DF-20 and DF-9 interfaces which connects to Imote2 platform. DF-9 interface is used to provide the power and ground to the sensor board, while DF-20 is used for data transmission, e.g. SPI interface.

2.4 Operating System

In this platform, there are multiple tasks such as the orientation information sampling, packet transmitting, packet receiving, media access control of , etc.. Thus a tiny operating system will be flexible to tackle with these tasks. For the chosen platform, currently three kinds of operating systems are supported, e.g. Linux, SOS, and TinyOS. In our system, we use Tinyos-1.x for the consideration of its popularity in the energy efficient scenarios. In addition, we want to see its potential capability for real time applications since Tinyos-1.x is not real time operating system.

2.5 Media Access Control

In WWSN, there are multiple sensor nodes which use the same wireless bandwidth resource. Due to the real time requirements for motion tracking, TDMA (time division multiple access) is the optimal scheme for those sensor nodes sharing the radio channel. Different from other energy-constrained applications where the concerned events happen occasionally or sporadically, motion tracking requires each sensor node to sample and transmit periodically with a frequency more than 20Hz. Although such frequency is not very high for unique sensor node, for more than one sensor node with the same frequency, it adds more challenge to system implementation due to the usage of low power wireless communication technique, which has limited bandwidth resource such as 250kbits/s (Zigbee/IEEE 802.5). Therein the more sensor nodes, the higher sampling rate, the less bandwidth frequency each sensor node will have. Moreover, each sensor node uses TinyOS-1.x(a tiny operating system) to schedule multiple tasks. As it is not real time, the time skew and jitter make it difficult to obtain the accurate timeline. Therein it is difficult for time synchronization during TDMA implementation. We will discuss it in detail in the following section.

3. TDMA Implementation

For TDMA implementation, the main challenge comes from time synchronization, which ensures all wireless sensor nodes to have a similar timeline. Otherwise it will appear that more than one sensor nodes transmit at the same time, which results in collision. In addition, the duration of each time slot (*TS*) should also be defined carefully. The longer the duration, the more bandwidth is uniquely allocated for one sensor node, while the slower the data updating is for motion tracking; the shorter the duration is, the faster the data updating is for motion tracking, while the more severe timeline should be acquired by synchronization. To make tradeoff, several items should be considered: data amount for each sample, radio transmission rate and system delay. In the following we will discuss in detail.

3.1 Time Synchronization

To implement TDMA, the same timeline should be used for all sensor nodes. Therein all wearable sensor nodes for motion tracking should be synchronized. Same as other energy-constrained applications, the low power sensor node suffers from the problems of clock skew and jitter，which is difficult to define the accurate timeline. Additionally the system delay is unpredicted too since Tinyos 1.x is not real time operating system, which makes it difficult to reserve the gap between two continuous time slot. Moreover, different from other energy-constrained applications where the monitored event appears occasionally, for motion tracking all sensor nodes are working periodically with rather high frequency. Those sensor nodes transmit their data to the base station one by one. Therein, the severe

timeline should be acquired otherwise it will result in collision.

In our system, to combat the above problems, we avoid the accurate timeline by passive listening scheme. Specifically, we let the first sensor node works periodically using its own clock, i.e. a timer is defined in sensor node 1. When the timer is time out, sensor node 1starts to sample and transmit the data. Other sensor nodes although work periodically, they do not use their own clock. They transmit data following the first sensor node. This period is also called frame, which includes several *TS*s. All *TS*s are allocated to other sensor nodes in turn, and during each *TS* there is only one sensor node to transmit data.

For example, if the second node finds that current transmission from sensor node 1, it continues to listening. When it finds that sensor node 1 finishes transmitting, it is starting to transmit. Except for sensor node 2, other sensor nodes such as 3, 4, 5, …, 9, find current transmission from the first sensor node, then it stops listening. They are waiting and starting to listening after (*i*-5/2)*TS*, here *i* denotes the sensor ID (In our system 1-9 are used for sensor IDs). For example, if sensor node 3 finds the current transmission from sensor node 1, it stops listening, and waits for 1/2*TS*. After 1/2*TS* it starts to listen and find current transmission from sensor node 2. It still listens and starts to transmit until sensor node 2 finishes. Here we add extra 1/2*TS* due to that the sensor ID is in the header field of a frame, and 1/2*TS* is enough for transmitting the frame heads.

Using such scheme, there is no need to keep gap between two adjacent *TS*s, which could use the bandwidth completely. But it might happen that the last sensor node transmission collides with the first sensor node transmission in the WWSN due to the accumulated delay. Specifically, if the accumulated delay is larger than the retransmission period of the first sensor node, a collision will appear. To avoid such collision, in each frame we add another *TS* at the end of the frame, which could arise partial bandwidth waste.

3.2 Duration of Time Slot and Frame

3.2.1 Time Slot Duration

According to above analysis, now we need to define the duration of *TS* and frame. As motion tracking is real time application, the duration of *TS* should be defined carefully. As we know, TS duration depends on the data transmission rate of wireless communication, USB serial rate between the base station and PC, and the sample length from each sensor node. In our system, Imote2 is a Zigbee/IEEE 802.15.4 compatible device, which has a radio transmission rate of 250 Kbits/s. The sample from each sensor node consists of orientation information represented with quaternion including 16 bytes (Each quaternion element is a floating point number denoted by 4 bytes). Besides that, the sensor ID and a packet sequence of 4 bytes should be added in addition to the IEEE 802.15.4 protocol stack overhead of 15 bytes (in Tinyos-1.x). Finally, there are total of 35 bytes for each sample data transmitted over radio channel. Then the transmission delay is

$$d_t = 35 \times 8/250\text{k} = 1.12 \text{ ms}. \quad (1)$$

For the serial port forwarding, in our system there is a USB cable connecting the base station and the computer which simulates the serial port. For Tinyos-1.x, the java tool *SerialForwarder* in the PC reads data from the serial port and forwards to the above java application for rendering. The default transmission rate of *SerialForwarder* is set to 115.2kbit/s. Different from the message transmitted over the radio channel, the data forwarded by the *SerialForwarder* has a length of 30 bytes, which discards the preamble

sequence (4 bytes) and starting label of a frame (1 byte). Then the forwarding delay from serial port d_f is

$$d_f = 30 \times 8/115.2\text{k} = 2.083 \text{ ms}. \quad (2)$$

To lessen this delay, a feasible solution is to increase the serial port rate. But according to our experiments, when the transmission rate is greater than 120kbits/s, *SerialForwarder* reports transmission error. Therefore we have to configure the serial port rate as 115.2 Kbits/s. Finally we define TS duration of 5 ms, which is verified by lots of experiments under TinyOS 1.x on Imote2. In our implementation we found that the serial port cannot forward the packets if the timer duration of TS is less than 5ms.

3.2.2 Frame Length

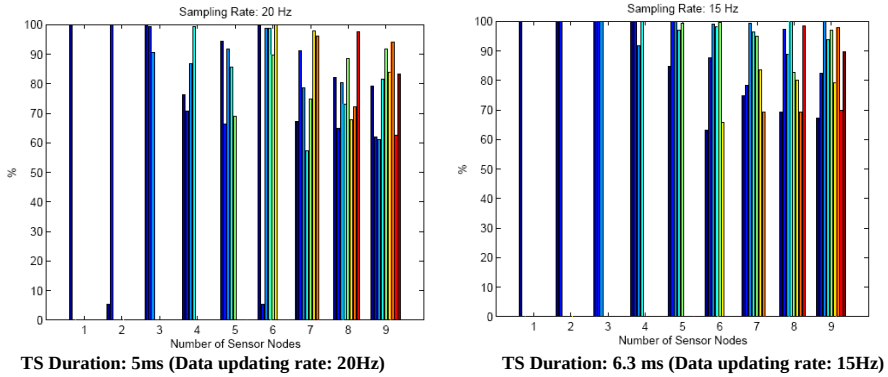
As we have 9 sensor nodes around the human body, and each sensor node takes one *TS* to send data, there are at least 9 *TS*s in one frame. According to above analysis, using our time synchronization scheme, the transmission from last sensor node (sensor node 9) might collide with the first sensor node due to the accumulated delay for all sensor nodes. To solve this problem, we add an extra *TS* at the end of the frame to act as a gap. Therein, there are 10 *TS*s in one frame, with each *TS* of 5ms.

4. Discussion

In this section, we will discuss some of the main characteristics of our system such as packet lost, energy efficiency and tracking accuracy.

4.1 Packet Loss

In energy-constrained wireless sensor network, the transmitted packet might be lost due to following reasons: limited communication range, transmission collision, network congestion, and channel interference, etc.. In our system, the main reason comes from the transmission collision, which arises from the accumulated delay of all sensor nodes. Just as we discussed above, the system processing delay and clock skew are difficult to estimate. Therein the last several sensor nodes might collide with the first sensor node because the first sensor node works periodically. When the accumulated delay is longer than this period, the collision will happen. In our prototype system, we compared two scenarios with different TS durations: 5 ms and 6.3 ms, which correspond to data updating rates of 20 Hz and15 Hz respectively.



TS Duration: 5ms (Data updating rate: 20Hz) **TS Duration: 6.3 ms (Data updating rate: 15Hz)**

Fig. 4. Network throughputs

Here we use network throughputs to show the packet loss under the defined scenarios. **Fig.4** depicts throughputs with different number of sensor nodes in terms of *TS* durations. In the experiments, the sensor node is added with one by one. In both scenarios, there is no collision and the throughput is maximal, i.e. 100% when there is only one sensor node. But when more sensor node is added, the network throughput is not always 100%. Especially for the high data updating rate such as 20Hz, the network throughput varies randomly. Even for some cases, it decreases to 10% in the left of **Fig. 4** when there are 2 or 6 sensor nodes. This comes from the high updating rate, which results in the received buffer overflow in the base station.

Furthermore, when more sensor nodes are added to the WWSN, the maximum throughput tends to decrease in both scenarios. For example, when there are more than 4 sensor nodes in the WWSN, the network throughput decreases significantly. In the scenario of 20Hz date updating rate, the maximum throughput is 93% for 9 sensor nodes, and the minimum throughput is 63%. From the figure, we cannot find any relation between it and the number of sensor nodes in WWSN. But from the figure, we can deduce that the average throughput decreases with the number of sensor nodes.

Now we decrease the date updating rate to 15Hz. As that in the right of **Fig. 4,** the network throughput is rather better, not only the average throughput, but also the single node's throughput. For example, where there are few sensor nodes such as less than 4 in WWSN, the throughput for every sensor node is close to 100%. When the sensor nodes are added more, the throughput although decreases for each sensor node, it still higher compared to the scenario of date updating rate of 20Hz. On the other hand, totally the average throughput decreases with the number of sensor nodes. Besides the buffer overflow problem, collision is also one of the main reasons which results in packet loss, especially for the high date updating rate. In our tests we find that the collision might occur in the case of a high quisition rate, e.g. 20 Hz, since for the case of 15Hz, this effect is not so significant. That is because in the case of low data updating rate, the maximum throughput is possible to be close to 100%, which shows that first sensor node transmits successfully since all other sensor nodes work following it. In that case, the packet loss mainly comes from the buffer overflow in the gateway.

4.2 Energy Efficiency

Although the energy efficiency is not our focus in this prototype system, it is still interesting since the system is powered with battery. As each sensor node works periodically with high frequency, the system lifetime depends on the power of three 1.5V AAA batteries.

In this system, the main energy consumption comes from the data sampling and transmission. As the data fusion algorithm is done in the VectorNAV sensor board, we only concern the sampling procedure. According to VectorNAV 100[5], the current consumption is about 65mA during working. But when it is added to Imote2 platform, the system current is about 130 mA. This increased amount mainly comes from the radio listening and transmission. If we remove the sensor board, the current for the Imote2 platform is about 110 mA under similar data transmission rate. Therefore, energy consumption is mainly coming from the data transmission.

1) Transmission: In our system, all sensor nodes transmit in a round robin fashion. When the first sensor node is transmitting, all other sensor nodes start listening until they find that this transmission is not belong to their previous one (The detailed procedure could refer to TDMA implementation.). Here all sensor nodes have listened at least one complete packet. Previous research results showed that the energy consumption of radio module while listening or receiving messages is very similar to when it is working [24], which means all sensor nodes

cost extra energy for those listening that does not destine to them.

To combat this energy waste, we could make the sensor nodes sleep after they complete transmission. But another problem might arise. Ideally, each sensor node transmit periodically. However since there are multiple sensor nodes transmission in this period, it requires that all sensor nodes are precisely synchronized. As we discussed above, due to the unpredictable system processing delay, it is not easy to complete this task. Therein, a feasible solution is to wake up one sensor node one *TS* earlier than a whole frame. For example, a sensor node sleeps after it finishes transmitting, and it does not wakes up after one frame, but after one frame minus 3/2*TS*. Here we reserve 1/2*TS* due to the unexpected system processing delay.

2) Sampling: In our system, the sensor board VN 100 provides the orientation with quaternion representation. It might provide the orientation data with a maximum frequency of 200Hz. According to the above analysis, the feasible transmission period is at most 20Hz, therein when the orientation data updating frequency is above 20Hz, it is always ready for reading.

According to the data sheet of VN 100, the serial port or SPI output rate can be set depending on the application requirement. But this operation does not affect the registration updating of orientation. That means it does not change the sensor sampling rate. So that even we read the orientation data with a low frequency such as 20Hz, the 3-axis accelerometer, 3-axis magnetometer and 3-axis gyroscope are still sampled with high frequency which is enough for data fusion and outputs the orientation data with a rather higher frequency of 200 Hz. As VN 100 sensor board does not provide the sampling rate configuration in the firmware, it is not possible to decrease the internal sampling rate.

4.3 Orientation Tracking

In our prototype system, the sensor board VN 100 provides the orientation data represented either by Euler angle (roll, pitch and yaw) or quaternion. This orientation data is calculated from the data fusion algorithm for 3-axis accelerometer, a 3-axis magnetometer and a 3-axis gyroscope, which uses an extended Kalman filter.

As each MEMS sensor suffers from its shortcoming, the orientation output after data fusion algorithm is still easily affected by the outside environment such as temperature changing, metal objects in the surrounding, etc.. In addition, since the wearable modules are powered with battery, it generates soft iron effect on the 3-axis magnetometers. Considering these components, we showed the accuracy of the orientation tracking. To see it intuitively, we used the Euler angle representation, e.g. roll, pitch and yaw in stationary. Note that all the experiments were done after iron/hard iron effect cancellation, otherwise the results would be very inaccurate.

In the stationary condition, we take yaw as an example for the orientation tracking as it has a maximum variability of 2° (rms) (According to VN 100 data sheet, the measurement variances of pitch and yaw are 0.5° (rms)). We studied the stationary clockwise rotation in the horizontal plane by moving the wearable sensor node following the sequences: 0°→90°, 90°→180°, 180°→270°, and 270°→360°. **Table 1** lists the measurement results about the yaw in our system. In addition, since it is not easy to find an absolute horizontal plane, we list the related roll and pitch in the table too.

Table 1. Yaw variation in the stationary condition

status	yaw	pitch	roll
0°	102.93°	-2.73°	-11.51 °
90°,	17.33 °	-13.09 °	-0.88 °

180°	-73.80 °	-3.84 °	13.56 °
270°	-165.88 °	13.45 °	-2.41 °
360°	103.90 °	3.02 °	-11.44 °

Table 2 lists the errors of yaw for stationary rotation. The maximal measurement error is 4.4°, and the minimum measurement error is 0.22°. The measurement variance (rms) is 2.5006 °, which is closed to that of 2° (rms) according to VN 100's data sheet. Note that, although the sensor node is put on the horizontal plane in our test, it is not so absolute, which could be deduced from the variations of roll and pitch.

Table 2. Rotation error of Yaw

Rotation	Error
0°→90°	-4.4°
90°→180°	-1.13°
180°→270°	-2.08 °
270°→360°	-0.22 °

4.4 Real Time Rendering

In this section, we focus on the real time rendering. As we wish the tracked object be rendered in real time, two metrics will be concerned on: delay and orientation accuracy. **Fig. 5** depicts a sequence snapshots from a video of motion tracking. The continuous capturing is accomplished by KMplayer tool[28], and every three seconds a snapshot happens.

From these snapshots, we can find the delay between the human motion and the rendered 3D picture, which was analyzed in the above section. Although this delay is not significant, it can be perceived if the human motion and the 3D rendering in the same snapshot as in **Fig. 5**. However, for most applications such as habilitation, such delay is acceptable.

Fig. 5. Delay analysis

In addition, the real time rendering is based on the motion reconstruction of all independent segments. In our system, each segment attached with a sensor node is taken as an independent rigid body with 6 DoF. Actually, for human body, two adjacent segments are constrained by a joint. Therefore one segment has only 3 DoF in terms of the other segment. For example, the forearm might rotate from 0° to 180° around the elbow relating to the upper arm. Herein, for 3D rendering, those constraints could be added for optimizing the homogeneous rendering. In addition, as the inertial sensor might suffer from the outside interference, the combination of

two adjacent segments might result in a broken joint. Thus for the rendering, some optimized algorithms are required to combat these problems.

5. Lessons Learnt

5.1 Processing Bottleneck in Gateway

During our tests, we found that with higher sampling rate, some packets can not be processed even if they are received correctly by the radio chip in the gateway. However, as the operating system is not real time, a task is posted in the task queue after a frame is received from the radio channel. This received frame will be processed only when the task is scheduled according to the FIFO rule in TinyOS. Therein, when further packets are received by the radio chip before the task is processed, the previous packets will be overwritten by them resulting in packet loss.

In our system, the tool *SerialForwarder* is used to transport the data from the base station to the PC. The default rate of *SerialForwarder* is 115.2 kbits/s. As we know, the radio rate is up to 250 Kbit/s. Then the serial port can not forward all received packets if the received packet rate is greater than the port capacity. This is the reason why there are a lot of packet loss when the sampling rate of each sensor node is higher than 20 Hz. We tried to improve the serial port rate, but the *SerialForwarder* tool does not work correctly over 120 kbits/s. Herein, how to improve the serial port rate is crucial for accurate tracking.

5.2 Sensor Calibration

Calibration is a procedure to align the measurement to the actual value. It is done by comparing the output of a sensor with a known measurement. Usually for each sensor two parameters are calibrated: offset and scale factor. Since these two parameters determine the accuracy of the measurements, the calibration should be accomplished before the sensor usage.

For VN 100, these calibrations are not performed independently for each sensor. As the sensor board is an encapsulated component and no interface is open for the calibration of offset and scale factor, we cannot cancel such errors independently. Nevertheless, for this sensor board, hard/soft iron effect calibration can be performed for the magnetometer, which is crucial for the correct output of azimuth. For example in the stationary test section, the tracking variance is only about 2° after hard/soft iron effect cancellation. But before it, the variation is not so good as shown in **Table 3**. This table lists the rotation variation of yaw before calibration, where the Z axis is perpendicular to the ground plane.

Table 3. Rotation deviation of Yaw

Angle Variation	Rotation Angle	Deviation
Initial→90	-69.4 °→-6.1 °	20.6 °
90°→180°	-6.1 °→68.2 °	15.7 °
180°→270°	68.2 °→180 ° -180 °→-136.1 °	-65.7 °
270°→360°	-136.1 °→-69.3 °	-23.2 °

Observe the last column in **Table 3**, the measurement deviation about 360° is little, i.e. 0.1°, but the measurement deviation about 90° is significant. Especially for the rotation from 180°→270°, the deviation is -65.7°. While for the smallest deviation, it is also 15.7°. Thus in

these scenarios without the calibration for the hard/soft iron effect, the measurement is far away from the actual rotation.

5.3 Battery Type

In our prototype, each wearable sensor node is powered with 3 AAA batteries(1.5V), which generate the hard and soft iron effect for the 3-axis magnetometer. As we discussed in the above section, the hard/soft iron effect distorts the orientation outputs directly.

In our wearable sensor nodes, the batteries are closed to the sensor board, which produces the hard and soft iron effect on the magnetic field. Generally, the hard iron effect exhibits an additive and constant field, therein compensating the hard-iron effect is relatively simple. But the soft-iron distortion is not additive due to the result of material that influences a magnetic field.

Since each kind of battery has different characteristics, to obtain a similar soft iron effect, we choose all the batteries from the same company. Additionally, as we use 3 batteries (each of 1.5V), the battery capacity also determines soft iron effect during operation. Therein, when the battery power depletes quickly, the hard/soft iron effect should be compensated dynamically, otherwise the measurement accuracy will decrease. However, VN 100 sensor board does not provide online calibration algorithm, thus the improvement should be added in a further step to accomplish a more accurate tracking for longer duration.

6. Conclusion

Although wireless wearable sensor network emerged as a key technique for human motion tracking, we have argued that designers and researchers should equally consider the ineffectiveness of such systems. By implementing the prototype system with COTS wireless sensor nodes and sensor board accompanying with the popular tiny operating system (Tinyos-1.x), we discussed the possible challenges from the multiple sensor nodes cooperation, packet loss, energy efficiency, orientation tracking and real time rendering under real time high fidelity communication environments. We also described the lessons learnt during system implementation from the considerations of processing bottleneck in the gateway, sensor calibration, and soft/hard iron effect coming from the batteries. We have outlined several critical gaps in wireless wearable sensor network for human motion tracking research and suggested specific areas that may serve as a basis for future work related to it. In the future, packet loss, energy efficiency and real time rendering will be further improved under the real time high fidelity communication environments.

Acknowledgement

We are grateful to Amili for making the video and pictures, Brezhnyev Kirill's (a student of Ottovon-Guiricke Universitaet Magdeburg) kind help for the 3-D graphic rendering implementation using D-H forward kinematic.

References

[1] Huiyu Zhou and Huosheng Hu, "A Survey - Human Movement Tracking and Stroke Rehabilitation," *TECHNICAL REPORT: CSM-420 ISSN 1744-8050*, University of Essex, Colchester United Kingdom, Dec, 2004. Article (CrossRef Link).

[2] Heemin Park, Jeff Burke and Mani B. Srivastava, "Intelligent lighting control using wireless

sensor networks for media production," *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS,* Vol. 3, NO. 5, pp. 423-443, October 2009. Article (CrossRef Link).
[3] Imote 2 data sheet, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf.
[4] Intel; PXA27x Processor family developer?s manual.
[5] VectorNav 100 user Manual, 2009, http://www.vectornav.com /Downloads/Support/UM001.pdf
[6] R. Urtasun, D. J. Fleet and P. Fua, "Monocular 3-D Tracking of the Golf Swing," in *Proc. of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* Vol. 2, pp.1199, June, 2005. Article (CrossRef Link).
[7] Daehwan Kim, Daijin Kim, "Self-occlusion handling for human body motion tracking from 3D ToF image sequence," in *Proc. of 3DVP '10 Proceedings of the 1st international workshop on 3D video processing, pp.*57-62. Article (CrossRef Link).
[8] Y. Jung and D. Kang and J. Kim, "Upper body motion tracking with inertial sensors," *IEEE International Conference on Robotics and Biomimetics(ROBIO)*, pp. 1746-1751, IEEE, December, 2010. Article (CrossRef Link).
[9] Christian Smith and Henrik I Christensen, "Wiimote Robot Control Using Human Motion Models, 2009," www.csc.kth.se/ccs/Publications/iros09b.pdf.
[10] Interactive Control of Avatars Animated with Human Motion Data.
[11] Gypsy 7 Motion Capture System, http://www.metamotion.com/gypsy/gypsy-motion-capture-system.htm.
[12] Polhemus tracking systems, http://www.polhemus.com/?page=Research AndTechnology.
[13] http://www.vicon.com/support/solution view.php?id=1471.
[14] Joseph Bray, "Markerless Based Human Motion Capture: A Survey", http://www.visicast.co.uk /members/move/Partners/Papers/MarkerlessSurvey.pdf.
[15] Animazoo IGS-900-M, http://www.animazoo.com/index.php/igs-190-m.
[16] Innalabs 3Dsuit, http://www.3dsuit.com/cn/product/.
[17] Richard W. Pew, Anne S. Mavor, "National Research, Modeling human and organizational behavior: application to military simulations". Article (CrossRef Link).
[18] D.Roetenberg, H.Luinge, and P.Slycke, "Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors". *Xsens Technologies-B.V.*, April 2009. Article (CrossRef Link).
[19] A. Young, M. Ling, and D. Arvind. "Orient-2: A wireless real-time posture tracking system using local orientation estimation," in *Proc. of the 4th Workshop on Embedded Networked Sensors*, pages 53–57, June 2007. Article (CrossRef Link).
[20] L.Vincent and F.Pascal, "Monocular Model-based 3d Tracking of Rigid Objects, Foundations and Trends," *Computer Graphics and Vision,* vol.1, Issue 1, pp. 1-9, 2005. Article (CrossRef Link).
[21] C.Orwat, A.Graefe, T.Faulwasser, "Towards pervasive computing in health care-A literature review," *BMC Medical Informatics and Decision Making*, pp.1-19, June 2008, 8(26). Article (CrossRef Link).
[22] T.B.Moeslunda, A.Hiltonb and V.Krgerc, "A survey of advances in vision based human motion capture and analysis," *Computer Vision*. pp. 90-126, November 2006. Article (CrossRef Link).
[23] Forward kinematics: the Denavit-Hartenberg Convention, http://www4.cs.umanitoba.ca/~jacky/ Robotics/Papers/spong kinematics.pdf.
[24] Hongchao Zhou and Xiaohong Guan. "Idle-listening Reduction for Data Aggregation in Distributed Sensor Networks," *IEEE Transaction on Parallel and Distributed Systems*, 2009. Article (CrossRef Link)
[25] M. Lapinski and E. Berkson and T. Gill, M. Reinold and J. A. Paradiso, "A Distributed Wearable, Wireless Sensor System for Evaluating Professional Baseball Pitchers and Batters," in *Proc. of International Symposium on Wearable Computers*, pp.131-138, iswc, *September* 2009. Article (CrossRef Link)
[26] R. Aylward and J. A. Paradiso, "A compact, high-speed, wearable sensor network for biomotion capture and interactive media," in *Proc. of the 6th international conference on Information processing in sensor networks,* ACM, pp. 380-389 , April, 2007. Article (CrossRef Link).
[27] G.X. Lee and K.S. Taher, "Unrestrained Measurement of Arm Motion Based on a Wearable

Wireless Sensor Network," *IEEE Trans. Instru. Meas.*,59,5,1309 - 1317, May 2010. Article (CrossRef Link)
[28] KMplayer, http://kmplayer.kde.org/.s
[29] IEEE 802.11 Working Group (1997-11-18). IEEE 802.11-1997: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ISBN 1-55937-935-9.
[30] Olivares, A.; Olivares, G.; Mula, F.; G´orriz, J.M.; Ram´ırez, J., "Wagyromag: Wireless sensor network for monitoring and processing human body movement in healthcare applications," *J. Syst. Archit*. pp.905–915, 2011, 57. Article (CrossRef Link).
[31] Cifuentes, C. Braidot, A. ; Rodriguez, L. ; Frisoli, M. ; Santiago, A. ; Frizera, A. , "Development of a wearable ZigBee sensor system for upper limb rehabilitation robotics," in *Proc. of Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference*, pp.1989- 1994, June 2012. Article (CrossRef Link)
[32] Geoffrey Lo, Ashwin Ram Suresh, Leo Stocco, Sergio González-Valenzuela, and Victor C. M. Leung, "A Wireless Sensor System for Motion Analysis of Parkinson's Disease Patients," in *Proc. of Work in Progress workshop at PerCom 2011*, pp. 372-375. March 2011, Article (CrossRef Link).
[33] Z. Zhang, Q. Fang, F. Ferry, "Upper limb motion capturing and classification for unsupervised stroke rehabilitation," in *Proc. of IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pp. 3832-3836, November, 2011. Article (CrossRef Link).
[34] S.S.;Pochappan, D.K.Arvind, J.Walsh, A.M.Richardson, J. Herman, "Mobile Clinical Gait Analysis Using Orient Specks," *Wearable and Implantable Body Sensor Networks (BSN)*, 2012 Ninth International Conference on, pp.172 – 177, May, 2012. Article (CrossRef Link).

Jianxin Chen received his Ph.D degree in Electronics Engineering from Shanghai Jiaotong University in 2007. He worked as a postdoctor in IPP Hurray Research Group, Portugal from May 2008 to July of 2009, and Gradiant ETSI Telecommunication, Spain from 2009 to 2010. Now he works in Nanjing University of Posts and Telecommunications. His research interests include computer network and body sensor network.

Liang Zhou received his Ph.D. degree major at Electronic Engineering both from Ecole Normale Superieure (E.N.S.), Cachan, France and Shanghai Jiao Tong University, Shanghai, China in 2009. Now, he is a professor in Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education, China. His research interests are in the area of multimedia communications and wireless networks.

Yun Zhang received his Barchelor degree from Beijing University of Post and Teleommunication in 1985, master degree from Nanjing University of Posts and Telecommunications in 1991 with major at Electronic system, Ph.D degree major at Computer Application from Suzhou University , Jiangsu, China in 2009. Now, he is a professor in Computer School, Nanjing University of Posts & Telecommunications. His research interests are in the area of computer networking and embedded system.

David Fondo was born in Spain, he obtained the Telecommunication Engineering degree by the University of Vigo (Spain) in 2002 and the Advanced Studies Diploma (pre-PhD level) by the same university in 2005. David worked as a project engineer from 2002 to 2005 for a university department and as a hardware engineer for Pro-Bel Ltd at United Kingdom from 2006 to 2007. He is currently a project manager for Gradiant. His fields of interest are signal processing using reconfigurable digital electronics (FPGAs) and digital communications.