The development of motion tracking algorithms for low cost inertial measurement units

- POINTING-AT -

Giuseppe Torre
Interaction Design Centre
University of Limerick
Limerick, Ireland
giuseppe.torre@ul.ie

Javier Torres
Tyndall National Institute
Cork University
Cork, Ireland
javier.torres@tyndall.ie

Mikael Fernstrom
Interaction Design Centre
University of Limerick
Limerick, Ireland
mikael.fernstrom@ul.ie

ABSTRACT

In this paper, we describe an algorithm for the numerical evaluation of the orientation of an object to which a cluster of accelerometers, gyroscopes and magnetometers has been attached. The algorithm is implemented through a set of Max/Msp and *pd* new externals. Through the successful implementation of the algorithm, we introduce *Pointing-at*, a new gesture device for the control of sound in a 3D environment. This work has been at the core of the *Celeritas Project*, an interdisciplinary research project on motion tracking technology and multimedia live performances between the Tyndall Institute of Cork and the Interaction Design Centre of Limerick.

Keywords

Tracking Orientation, Pitch Yaw and Roll, Quaternion, Euler, Orientation Matrix, Max/Msp,*pd*, Wireless Inertial Measurement Unit (WIMU) Sensors, Micro-Electro-Mechanical Systems (MEMS), Gyroscopes, Accelerometers, Magnetometers

1. INTRODUCTION

Motion Tracking technology has interested the multimedia art community for two or more decades. Most of these systems have tried to offer a valid alternative to camera-based system such as VNS[2] and EyesWeb [14]. Between them are: DIEM [1], Troika Ranch[15], Shape Wrap, Pair and Wisear [19], Eco [17], Sensemble [13],The Hands [8]and Celeritas [20, 16] from the authors.

In this paper we describe the algorithm to numerically solve the orientation of each single mote in our Celeritas system. We also aim to give an introduction to the topic to persons that aim to develop their own tracking device (using Arduino for example). Although a full Max/Msp and *pd* library has been developed and made available at [10] , we have listed in the reference of this paper other Max/Msp developers [11, 12, 18] whose work has been freely released though their work focuses only on the conversion between different numerical representation and does not interact with the specific device specified above.

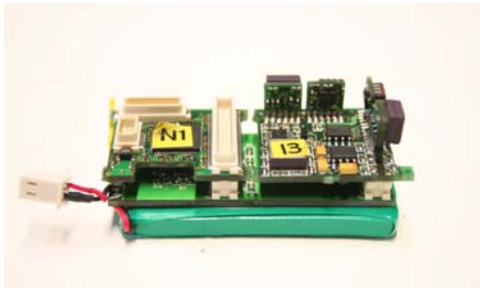

Figure 1: *Mote* and its cluster of sensors with battery pack. Dimensions are 25 x 25 x 50 mm

On the basis of the result achieved, we introduce in the last paragraph *Pointing-at*, a new gesture device for the control of sound in a 3D space or any surround system. The device can be used both in studio and live performances.

Our Celeritas system is built around the Tyndall's 25mm WIMU which is an array of sensors combined with a 12-bit ADC [6, 4, 5, 7]. The sensor array is made up of three single axis gyroscopes, two dual axis accelerometers and two dual-axis magnetometers.

The *accelerometers* measure the acceleration on the three orthogonal axes (U, V and W as shown in Figure 2).

The *gyroscopes* measures the angular rate around the three orthogonal axes.

The *magnetometers* measure the earth magnetic field on the three orthogonal axes.

2. TERMINOLOGY

Before going into the description of the algorithm, we would like to introduce the reader to some of the most common terms in use to make easier the understanding of the following sections. A good explanation of this terms and of the 3D math can be also found at [9].

System of Reference. We will discuss the two systems of reference: the Earth-Fixed one (x, y, z) which has the x axis pointing at the North Pole, y axis at west and z at the Earth' core. The IMU-Fixed frame (u, v, w) with three orthogonal axes parallel to the sensor's sensitive axes.

Quaternions form a 4-dimensional normed division algebra over the real numbers. It is usually written in the form $qw^2+qx^2+qy^2+qz^2 = 1$. Quaternions are used to represent the rotation of an object in a 3D space. They are very common in programming, as they don't suffer from problems with singularities at 90 degrees.

Euler Angles The Euler angles are usually given in aeronautical term as Pitch, Roll and Yaw as shown in Figure

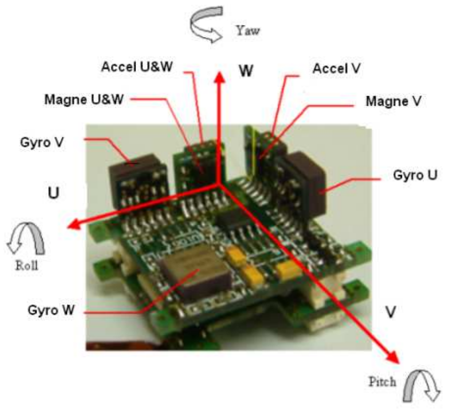

Figure 2: sensor and IMU reference system.

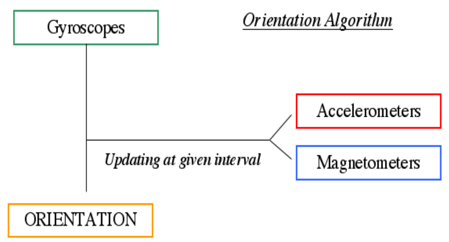

Figure 3: Orientation Algorithm.

2, where: Pitch is the rotation around the lateral (V) axis, Roll around the longitudinal (U) axis and Yaw around the perpendicular (W) one. The calculation involves the usage of non-commutative matrix multiplication.

Orientation Matrix mathematically represents a mathematical bases change in a 3 dimensional space, thus, we can translate the sensors output coordinates, given with respect to the IMU fixed frame, into the reference Earth frame using the Orientation Matrix.

Angle x, y, z describe rotation using a unit vector indicating the direction of the axis and an angle indicating the magnitude of the rotation about the considered axis.

3. ALGORITHM

With our cluster of sensors we calculate the orientation of the sensor with respect of the Earth-fixed frame of reference. The orientation is retrieved using two source of estimation: the output of the gyroscopes and then the combination of accelerometers and magnetometers on the other. The reasons for doing this is that gyroscopes are not self-sufficient for long-term precision because of a drift associated with their reading. Accelerometers and magnetometers, on the other hand, are good for long-term stability but, once again, not good for short-term accuracy due to occasional inaccuracy caused by linear and rotational acceleration. Thus, our algorithm combines the short-term precision of the gyroscopes with the long-term precision of accelerometers and magnetometers.

3.1 Reading the values from the sensor

As the data from the *motes* are sent wirelessly to a base station connected to the host computer via serial port, we designed a C driver to handle this stream. Ultimately, we compiled a new external (mote) to import this stream in Max/Msp or Pd. Values appearing in our host application

Table 1: Unit of measurement

Sensor	from	to
Gyroscopes	ADC	degrees/second
Accelerometers	ADC	m/s^2
Magnetometer	ADC	gauss

environment are ADC in the range of 0 and 4096 (as our microcontroller is 12 -bit resolution). After having calculated the offset by averaging the first thousand incoming values leaving the sensor in steady position, we are able to read the values related to the movement of the sensor by subtracting from the calculated offset. Then we need to convert the ADC values from each sensor into the proper units of measurement as shown in Table 1.

Multiplying the subtracted ADC value by the rate resolution value of each sensor can do this. The rate resolution value can be found in the specification sheet of each sensor or by empirical methods.

3.2 Orientation using Gyroscopes

Sampling at t rate enables us at each Δt to know α, ϕ and θ applying the following formulas:

$$\Delta\theta(k+1) = \Delta t * \Delta\dot{\theta}(k+1);$$

$$\Delta\phi(k+1) = \Delta t * \Delta\dot{\phi}(k+1);$$

$$\Delta\alpha(k+1) = \Delta t * \Delta\dot{\alpha}(k+1);$$

where $\Delta\theta$, $\Delta\phi$, $\Delta\alpha$ represent the incremental angle around W, V, and U respectively. Next, the algorithm constructs the rotation matrix around each particular axis and multiply them together.

$$R(w,\theta,k+1) = \begin{pmatrix} cos\Delta\theta(k+1) & -sin\Delta\theta(k+1) & 0 \\ sin\Delta\theta(k+1) & cos\Delta\theta(k+1) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R(v,\phi,k+1) = \begin{pmatrix} cos\Delta\phi(k+1) & 0 & sin\Delta\phi(k+1) \\ 0 & 1 & 0 \\ -sin\Delta\phi(k+1) & 0 & cos\Delta\phi(k+1) \end{pmatrix}$$

$$R(v,\alpha,k+1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\Delta\alpha(k+1) & -sin\Delta\alpha(k+1) \\ 0 & sin\Delta\alpha(k+1) & cos\Delta\alpha(k+1) \end{pmatrix}$$

which can be generally written as:

$$Rotation(k+1) = R(w,\theta,k+1)*R(v,\phi,k+1)*R(v,\alpha,k+1);$$

Therefore we define our Orientation, in Matrix format, as to be:

$$Orientation(k+1) = Rotation(k+1) * Orientation(k);$$

From these results, the algorithm converts the resulting matrix into quaternion and angle,x,y,z format which facilitate ease of use in graphical oriented programming language such as Max/Msp and *pd*.

3.3 Orientation using Accelerometers and Magnetometers

So far we considered the 3 x 3 Orientation Matrix as the matrix describing the orientation of the IMU-fixed frame in relation to the Earth-fixed frame. Conversely, the Inverse Rotation Matrix describes the orientation of the Earth-fixed

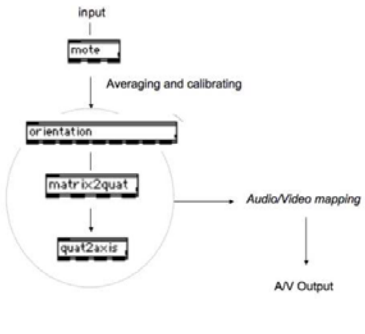

Figure 4: Pseudo Max patch.

Figure 5: *Pointing-at* out of the shell.

frame in relation to the IMU-fixed frame and can be written as:

$$Orientation^{-}1 = \begin{pmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{pmatrix}$$

The two dual-axis magnetometers enable the reading of the earth's magnetic field on the three orthogonal axes. These values are used to calculate the first column of the Inverse Orientation Matrix (a11, a21,a31) using the following set of formulas:

$$\begin{pmatrix} a11 \\ a21 \\ a31 \end{pmatrix} = 1/H * \begin{pmatrix} Hu \\ Hv \\ Hw \end{pmatrix}$$

where H is the earth magnetic field magnitude and Hu, Hv and Hw are the magnetic filed vector measured by the sensor along U, V and W respectively.

To calculate the third column (a13, a23, a33) of the Inverse Orientation Matrix, we use the values read from the two dual-axis accelerometers. The formula used is:

$$\begin{pmatrix} a13 \\ a23 \\ a33 \end{pmatrix} = 1/g * \begin{pmatrix} Gu \\ Gv \\ Gw \end{pmatrix}$$

where g describe the earth gravity acceleration magnitude and Gu, Gv and Gw the acceleration vector measured by the IMU along U, V and W

Finally the third column (a12, a22, a23) is calculated from the cross product between the first and the third column as described below:

$$\begin{pmatrix} a12 \\ a22 \\ a32 \end{pmatrix} = \begin{pmatrix} (a21 * a33) - (a31 * a23) \\ (a31 * a13) - (a11 * a23) \\ (a11 * a23) - (a21 * a13) \end{pmatrix}$$

4. ORIENTATION LIBRARY AND MAPPING

For the purpose of the numerical evaluation of the sensor's orientation an ad hoc set of Max/Msp and *pd* external were developed. The most important are listed below..

Orientation Calculates the Orientation of the IMU. The inlet receives a list made up of the following elements: pitch, yaw, roll, n_pack, sampling time, Alpha_rate_resolution, Phirateresolution, Theta_rate_resolution. Each of the 9 outlets are element of the 3 x 3 Matrix format representing the IMU's orientation.

matrix2quat Converts the 3 x 3 Matrix to Quaternion format.

quat2axis Converts the quaternion format to the angle, x, y, z format.

azi_ele Converts the input to azimuth and elevation numbers making the format readable by Vector Base Amplitude Panning (VBAP) or other multi-channel libraries.

A schematic of the max patch is shown in Figure 4.

5. APPLICATION DEVELOPMENT

On the basis of the above algorithm, we developed several applications for multidisciplinary live performances like Vitruvian for live dance performance and DjMote. In this paper we introduce *Pointing-at*, a new gesture device for the control of sounds in a 3-D surround environment.

5.1 *Pointing-at*

Pointing-at is a new wearable wireless glove that uses the results of our Celeritas project and the reliability of the Tyndall's 25mm WIMU to control sounds in a 3-D or any set of surround system. Its design is focused on the analysis of the methodologies concerning the gestural mapping of sounds in a fully three-dimensional environment. As the most natural movement related to directionality is the simple pointing of the hand in a given direction, we decided to use the orientation of the hand/arm as indicator of this direction. Thus, we fitted the WIMU in a glove, which has a protective pocket on the top of the hand's dorsum as shown in Figure 5.

5.2 Gesture Mapping

The orientation data retrieved from the reading of the WIMU are translated into azimuth and elevation coordinates making the data compatible to libraries such as VBAP[3]. Azimuth and elevation are calculated taking into account the z-axis the main axis of reference. The third variable that is a characteristic of surround sound editor systems is source distance. The gestural mapping of this parameter has been solved in the following way: a 90 degree roll movement enable the azimuth to be read as distance value in the range between 0 and 90 where 0 indicates the farthest distance and 90 the closest (Fig 6).

6. CONCLUSION AND FUTURE WORK

In this paper we described an algorithm that is used to retrieve the orientation of an object that has attached to it a cluster of sensor made up of accelerometers, gyroscopes and magnetometers. We also introduced a new wearable wireless glove, *Pointing-at*, for the gestural control of sounds in a 3-D surround space. The device was tested in our lab and proved to be a reliable tool for live performances. At the moment our team is working on the implementation of a bending sensor (see red strip in Figure 5) to enable the

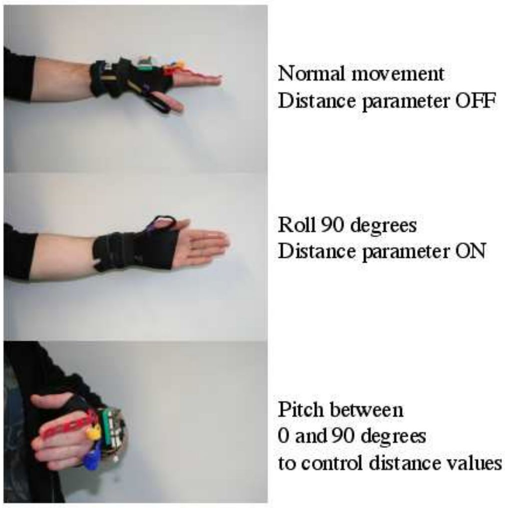

Figure 6: Gesture to control distance parameter.

grabbing and release feature of sound on the fly. In future works we aim to reduce the size of the WIMU to 10mm to improve wearability.

7. ACKNOWLEDGMENTS

We would like to thank the Tyndall National Institute for allowing us access under the Science Foundation Ireland sponsored National Access Programme. Many thanks also to the staff at the Interaction Design Centre of the University of Limerick for their input in to the project and realization of Pointing-at. Finally thanks to all the researchers and students at Tyndall and UL who gave an important input to this project.

8. ADDITIONAL AUTHORS

Additional Author: Brendan OFlynn, Tyndall National Institute - Cork University, email: brendan.oflynn@tyndall.ie

9. REFERENCES

[1] http://hjem.get2net.dk/diem/products.html.
[2] http://homepage.mac.com/davidrokeby/vns.html.
[3] http://www.acoustics.hut.fi/ ville/.
[4] http://www.analog.com/en/prod/0,2877,adxl202,00.html.
[5] http://www.analog.com/en/prod/0,,764_801_adxrs150,00.html.
[6] http://www.analog.com/en/prod/0,,ad7490,00.html.
[7] http://www.atmel.com/dyn/products/product_card.asp?part_id=2018.
[8] http://www.crackle.org/the%20hands%201984.htm.
[9] http://www.euclideanspace.com/.
[10] http://www.idc.ul.ie/idcwiki/index.php/celeritas.
[11] http://www.jasch.ch/code_release.html.
[12] http://www.maxobjects.com/?v=libraries&id_library=111.
[13] R. Aylward, S. D. Lovell, and J. Paradiso. "sensemble: A wireless, compact, multi-user sensor system for interactive dance". *The 2006 International Conference on New Interfaces for Musical Expression (NIME'06)*, (134 - 139), June 2006.
[14] A. Camurri, S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, and G. Volpe. Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal*, 24(1):57– 69, April.
[15] M.Coniglio. http://www.troikaranch.org/mididancer.html.
[16] B. O'Flynn, G. Torre, M. Fernstrom, T. Winkler, A. Lynch, J. Barton, P. Angove, and C. O'Mathuna. Celeritas - a wearable wireless sensor system for interactive digital dance theater. *Body Sensor Network, 4th Internation Workshop on Wearable and Implantable Body Sensor Network*, 2007.
[17] C. Park and P. H. Chou. Eco, ultra-wearable and expandable wireless sensor platform. *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pages 162 – 165, April 2006.
[18] D. Sachs. A forearm controller and tactile display, 2005.
[19] D. Topper and P. Swendensen. Wireless dance control: Pair and wisear. *Proc. Of the 2005 International Conference on New Interfaces for Musical Expression (NIME'05),*, pages 76 –79, May 2006 Canada.
[20] G. Torre, M. Fernstrom, B. O'Flynn, and P. Angove. Celeritas wearable wireless system. *Proc. Of the 2005 International Conference on New Interfaces for Musical Expression (NIME'07*, pages 205 – 208, 2007 New York.