A Sensor Fusion Method for Smart phone Orientation Estimation

Shahid Ayub
School of Computing and Communication
Lancaster University
Lancaster, UK
Email: s.ayub@lancs.ac.uk

Alireza Bahraminisaab
HW Communications
Lancaster, UK
Email: a.bahraminasab@hwcomms.com

Bahram Honary
School of Computing and Communication
Lancaster University
Lancaster, UK
Email: b.honary@lancs.ac.uk

***Abstract*—Low cost MEMs sensors are integrated in almost every Android device these days. These sensors are very useful in various applications like gaming, navigation, augmented reality etc. Once we know the correct orientation of the device, it becomes easy to develop such applications. Orientation can be determined by sensor fusion of accelerometer and magnetometer but it provides good accuracy as long as device is stationary or not moving linearly and also it suffers from surrounding magnetic interference. Orientation estimation systems often use gyroscope to increase reliability and accuracy. Although gyroscopes provide a quick response to change in angles and do not have problems like interference, they suffer from bias and integration errors which introduce drift in signal. In this paper a DNRF (Drift & Noise Removal Filter) is described that is implemented by sensor fusion of gyroscope, magnetometer and accelerometer which minimizes the drift and noise in output orientation. A numerical error correction approach is also mentioned to minimize the errors caused by gyro signal integration. The orientation results obtained by proposed DNRF method are smooth and less noisy as compared to digital compass.**

I. Introduction

Orientation determination of a moving object using IMU (inertial measuring unit), has been well developed in the field of Inertial Navigation Systems [1],[2],[3],[4]. In [1] the design of multi-sensor attitude determination systems is discussed using quaternion and Euler based algorithms. A Direction Cosine Matrix (DCM) based method for attitude and orientation estimation is discussed in [2]. DCM method has some advantages over the popular methods such as Euler Angle, Quaternion in light of reliability, accuracy and computational efforts. An extended Kalman filter (EKF)-based data fusion algorithm is discussed in [3]. A systematic approach based on wavelet decomposition is utilized to estimate noise covariances used in the Kalman filter formulation.

Gyroscope is a primary source of extracting orientation from these IMUs. Gyros measure the angular rate of the body with respect to body frame coordinates. They can therefore be used to derive orientation by integrating over time if initial orientation is known . All Orientation determination systems that use gyros suffer from bias or drift, which is a low frequency noise component [5]. Accurate estimates could be achieved over longer periods of time by employing high quality gyros and good initial guesses of the attitude. However, high quality gyros are too expensive for applications like smart phones, and also access to such sensors can be limited.

Rest of the papar is organized as follows. In section II, a brief overview of Android platform and available sensors is described. In section III, DNRF method of smartphone orientation is explained in detail including rotation matrix construction (III-A) numerical correction technique (III-B), Gimbal lock (III-C), and complimentary filter (III-D). Finally results are shown and a conclusion is done in section IV.

II. Android Platform

A. Android sensors

These days Android devices come with several built-in sensors like Accelerometer, Gyroscope, Magnetometer, Proximity, Pressure sensors etc. For positioning, tracking [6]& activity recognition applications these sensors play a critical role. In this work an Android based LG Optimus $2X$ smart phone is used. In this section, a brief description of available sensors is mentioned.

Accelerometer : $3-$Axis Accelerometer sensor gives us the acceleration measurements in m/s^2 along each of X, Y, Z axes. It can be used to recognize the motion activities. The most important source of error of an accelerometer is the bias. The bias of an accelerometer is the offset of its output signal from the true value. It is possible to estimate the bias by measuring the long term average of the accelerometers output when it is not undergoing any acceleration.

Fig. 1. SmartPhone Reference Frame

ISBN: 978-1-902560-26-7 © 2012 PGNet

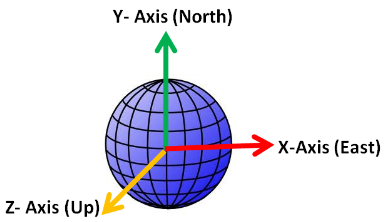

Fig. 2. Earth Reference Frame (ENU)

Magnetometer : Magnetometer sensor measure the magnetic field in micro Tesla in X, Y, Z axis. It can be used in combination with accelerometer to find the direction (yaw) with respect to North when linear acceleration is zero. The main source of measurement errors are magnetic interference in the surrounding environment and in the device.

Gyroscope : Gyro sensor provides us the angular velocities in rad/sec along each of 3 axis. It can be used to get the correct orientation of the device while in motion. If there is a magnetic interference in the surrounding environment, the heading calculated from the magnetometer is not accurate. Moreover, roll and pitch calculated from accelerometer sensor are only accurate when mobile is stationary or its acceleration is zero. In this situation, gyroscope can be used to get the correct orientation. But the problem with gyroscope is that there are bias and numerical errors.

The bias of a gyro is the average output from the gyroscope when it is not undergoing any rotation (i.e: the offset of the output from the true value). The gyro bias, shows itself after integration as an angular drift, increasing linearly over time. Another error arising in gyros is the 'calibration error', which refers to errors in the scale factors, alignments, and linearities of the gyros. Such errors are only observed whilst the device is turning. Such errors lead to the accumulation of additional drift in the integrated signal, the magnitude of which is proportional to the rate and duration of the motions.

B. Data Acquisition

Raw sensor data is obtained by using Android API function. It reports a new sample value whenever a change in sensor signal is detected. We have the option to select the different sampling rate e,g., *NORMAL*, *UI(User Interface)*, *GAME*, and *FASTEST*. In this work sampling rate is set to *FASTEST*. After accessing the sensor data it is passed through a low pass filter to remove unwanted noise from signal. A moving average filter of window size w is designed as an equivalent to low pass filter.

III. DNRF FOR SMART PHONE ORIENTATION

DNRF method shown in figure (3) consists of the following stages. At first Rotation matrix is constructed and then numerical correction is applied to correct the orthogonality of rotation matrix after that a combination of low and high pass

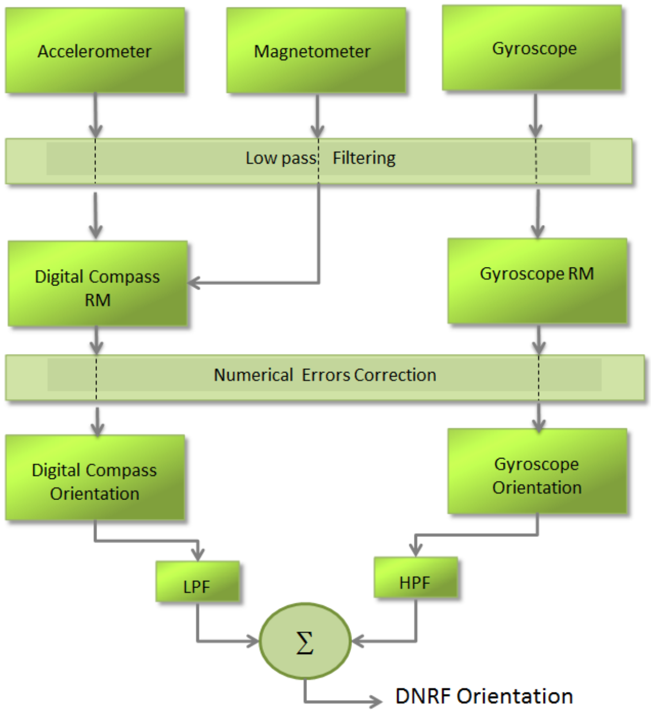

Fig. 3. DNRF Algorithm Flow Diagram

filter is applied to get the smooth and drift free orientation. Finally a correction is done for Gimbal lock condition.

Smart phone orientation means to determine the rotation angles relative to some fixed reference frame. It can be explained by two co-ordinate systems. One coordinate system is fixed to smart phone also known as Device reference frame (DRF) and other is fixed to Earth and also known as Earth reference frame (ERF) or navigation frame.

In Android API the device co-ordinate frame is defined relative to the screen of the phone. The *X-axis* is horizontal and points to the right, the *Y-axis* is vertical and points towards the top of the screen and the *Z-axis* points outside the front face of the screen as shown in figure 1.

The earth reference frame used in this work is East-North-Up (ENU), where positive *X-axis* is along East. *Y-axis* is along North and *Z-axis* points out of the center of the Earth as shown in figure 2. Our task is to get the correct orientation of smart phone relative to the ERF.

A. Rotation Matrices

The transformation from the device frame to the Earth frame requires the rotation about three axes. These angular rotations can be represented as the Euler angles. A convenient way of describing these rotations is using a Rotation matrix. If one defines these angles as Yaw (Rotation around *Z-axis*, figure 4), Pitch (Rotation around *X-axis*, figure 5), and Roll (Rotation around *Y-axis*,figure 6), one can transform a vector from the device to the earth frame by simply multiplying it with rotation matrix. The rows of rotation matrix are actually the projections of ERF axes on DRF axes, and columns are the projection of DRF axes on ERF axes.

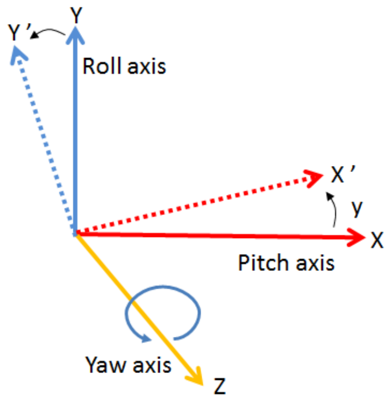

Fig. 4. Yaw-Rotation around Z-Axis

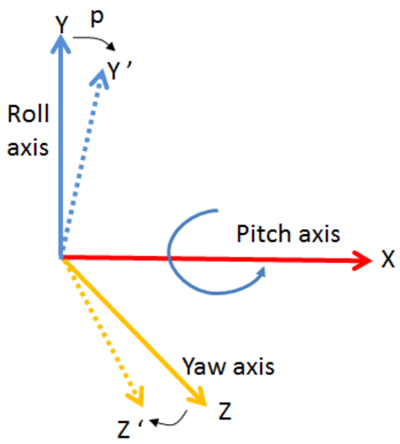

Fig. 5. Pitch-Rotation around X-Axis

One of the key properties of the rotation matrix is its orthogonality. The orthogonality of the rotation matrix in mathematical terms means that any pair of columns (or rows) of the matrix are perpendicular, and that the sum of the squares of the elements in each column (or row) is equal to one. The rotation matrices along each of the three axis are mentioned below

$$R_z = \begin{bmatrix} C_y & S_y & 0 \\ -S_y & C_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_p & S_p \\ 0 & -S_p & C_p \end{bmatrix} \tag{2}$$

$$R_y = \begin{bmatrix} C_r & 0 & -S_r \\ 0 & 1 & 0 \\ S_r & 0 & C_r \end{bmatrix} \tag{3}$$

where $C = cos$, $S = sin$, $y = Yaw$, $p = Pitch$ and $r = Roll$. Final rotation matrix R is obtained by multiplying the above

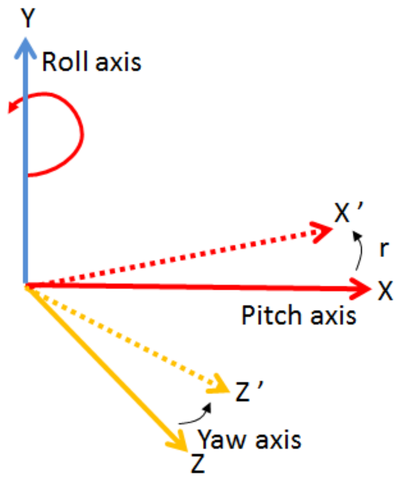

Fig. 6. Roll-Rotation around Y-Axis

three rotation matrices.

$$R(t) = R_z * R_x * R_y \tag{4}$$

$$R(t) = \begin{bmatrix} C_rC_y + S_rS_pS_y & C_pS_y & -S_rC_y + C_rS_pS_y \\ -C_rS_y + S_rS_pC_y & C_pC_y & S_rS_y + C_rS_pC_y \\ S_rC_p & -S_p & C_rC_p \end{bmatrix} \tag{5}$$

At first the rotation matrix is initialized from the compass values that is constructed using accelerometer and magnetometer sensors. After that in each cycle it is updated by using gyro signal. Gyro signal is integrated over time to get the change in angles. The following equations shows the update of rotation matrix using gyros as described in [7].

$$\Delta R = I + \Omega(t) \times dt \tag{6}$$

where I is 3×3 identity matrix and

$$\Omega(t) = \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix} \tag{7}$$

The updated rotation matrix is obtained by multiplying it with ΔR

$$R(t + dt) = R(t) \times \Delta R \tag{8}$$

B. Numerical Correction

Numerical Integration of gyro signal introduces errors, which are of the following two types.

Integration Error : Numerical integration uses a finite time step and data that is sampled at a finite sampling rate. We assume that rotation rate is constant over the time step; this introduces an integration error that is proportional to the angular velocity.

Quantization Error : We know that digital representation is finite, so all the bits of a value are not stored at analogue to digital converter and hence there will be quantization error.

These numerical errors can violate the orthogonal property of the rotation matrix as mentioned earlier in section (III-A). Since the rows and columns are supposed to represent unit vectors; their magnitude should be equal to one, but numerical errors could cause them to get smaller or larger than one. The rows and columns are supposed to be perpendicular to each other; numerical errors could cause them to lean into each other. It is necessary to correct these errors to preserve the orthogonality of rotation matrix so that we may extract accurate angles from it. The rotation matrix in equation (5) can be written in terms of matrix elements as shown in equation (9) . To maintain the orthogonality condition, we compute the dot product of the R_1 and R_2 rows of the matrix which should be zero, so the result is a measure of how much the R_1 and R_2 rows are rotating towards each other.

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \tag{9}$$

$$R_1 = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \tag{10}$$

$$R_2 = \begin{bmatrix} r_4 \\ r_5 \\ r_6 \end{bmatrix} \tag{11}$$

$$Error = R_1^T.R_2 = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} . \begin{bmatrix} r_3 \\ r_4 \\ r_5 \end{bmatrix} \tag{12}$$

We distribute this error equally into R_1 and R_2, which brings back the rows in the opposite direction.

$$R_{1orth} = R_1 - (\frac{Error}{2})R_2 \tag{13}$$

$$R_{2orth} = R_2 - (\frac{Error}{2})R_1 \tag{14}$$

The next step is to adjust the R_3 row of the matrix to be orthogonal to the R_1 and R_2. This can be accomplished by taking the cross product of the R_1 and R_2.

$$R_{3orth} = R_{1orth} \times R_{2orth} \tag{15}$$

In the end we want to make sure that each row has a magnitude equal to one. One way of doing this is to use Taylor's expansion, the resulting equations for row vectors are

$$R_{1norm} = \frac{1}{2}(3 - R_{1orth} \cdot R_{1orth})R_{1orth} \tag{16}$$

$$R_{2norm} = \frac{1}{2}(3 - R_{2orth} \cdot R_{2orth})R_{2orth} \tag{17}$$

$$R_{3norm} = \frac{1}{2}(3 - R_{3orth} \cdot R_{3orth})R_{3orth} \tag{18}$$

C. Gimbal Lock (Singularity) Removal

After preserving the orthogonality of rotation matrix, Euler angles are obtained from corrected rotation matrix elements by using the following equations.

$$Yaw = atan2\,(r_2, r_5) \tag{19}$$

$$Pitch = -asin2\,(r_8) \tag{20}$$

$$Roll = atan2\,(r_7, r_9) \tag{21}$$

When the pitch value is $90\,°$, i.e., $Cos(pitch) = 0$, which means we are either looking straight up or straight down then we cannot use the above equations since all the matrix elements involved in calculating yaw and roll are zero [8]. This is the Gimbal lock (Singularity) situation, where yaw and roll effectively rotate about the same vertical axis. In this case, we will arbitrarily assign all rotation about the vertical axis to yaw and set roll equal to zero. Now we know the value of pitch and roll, and all we have left is to solve for yaw. So with the assumptions that when $Cos(pitch) = 0$, setting the value of $Roll = 0$,so $Sin(roll) = 0$, $Cos(roll) = 1$ & $Sin(pitch) = 1$. Putting these assumptions back in the rotation matrix in (5), we get the simplified version.

$$R(t) = \begin{bmatrix} C_y & 0 & S_p S_y \\ -S_y+ & 0 & S_p C_y \\ 0 & -S_p & 0 \end{bmatrix} \tag{22}$$

Now we can compute Yaw from r_4 and r_1 elements of the rotation matrix, which contain the sine and cosine of the yaw, respectively.

$$Yaw = atan2\,(-r_4, r_1) \tag{23}$$

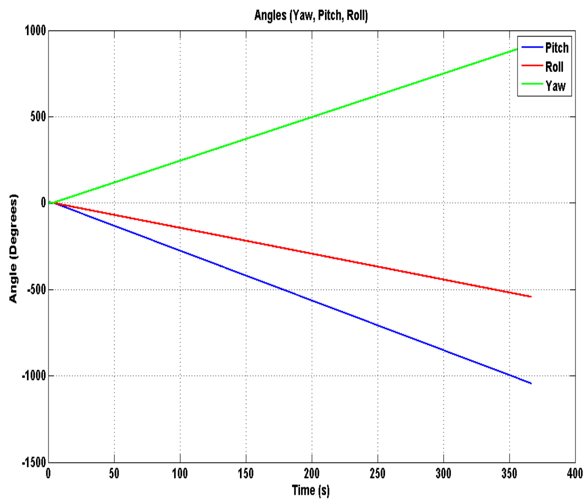

Fig. 7. Integrated Gyro Signal, Drifted Yaw, Pitch, Roll angles: Rest

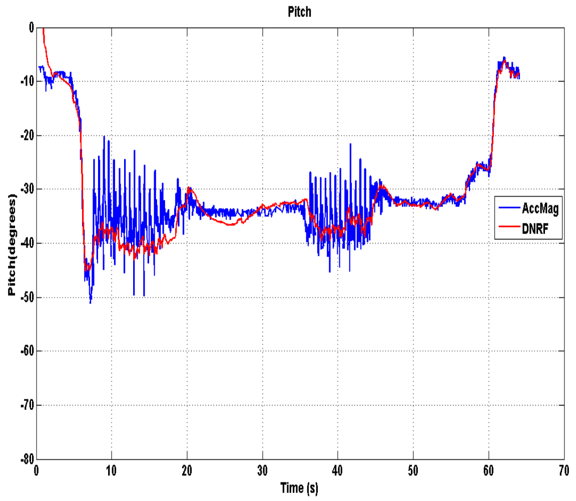

Fig. 8. Pitch Angle

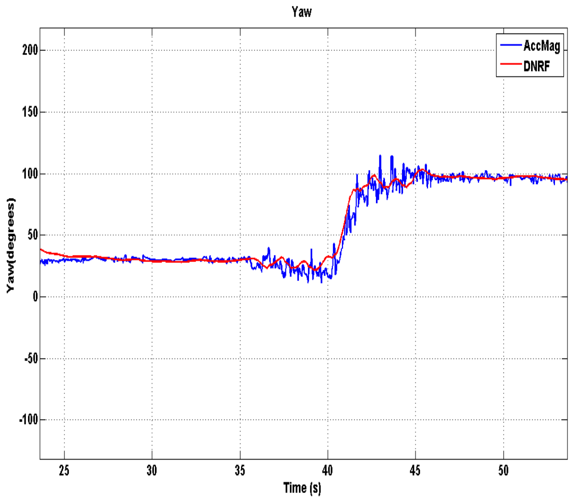

Fig. 9. Yaw Angle

D. Complimentary Filter

This is main part of *DNRF* method. To avoid the gyro drift and noise, gyroscope output is applied only for orientation changes in short time intervals, while the magnetometer/accelerometer orientation is used as support information over a long periods of time. This is equivalent to low-pass filtering of the accelerometer and magnetic field sensor signals and high pass filtering of the gyroscope signals. In other words low-pass filter is eliminating short-term fluctuations from the signal.

High-Pass Filter actually allows short-duration signals to pass through while filtering out signals that are steady over time. This can be used to cancel out drift. It is actually done by replacing the filtered high frequency components of magnetometer/accelerometer orientation with the corresponding gyroscope values. The combination of this low pass and high pass filter can be expressed mathematically in equation (24). The first part on right hand side in equation (24) is high pass filter while second part is low pass filter.

$$\theta = (\alpha) * (\theta_g) + (1 - \alpha) * (\theta_{am}) \tag{24}$$

Here θ is the filtered orientation, θ_g is gyrscope orientation and θ_{am} is the orientation calculated from accelerometer and magnetometer. α is the coefficient of digital filter. If we know the sample time and time constant then we can find the filter co-efficient α using equation 25.

$$\alpha = \frac{\tau}{\tau + T_s} \tag{25}$$

Here T_s is the sample period and time constant τ of a filter is the relative duration of signal it will act on. For a low-pass filter, signals much longer than the time constant pass through unaltered while signals shorter than the time constant are filtered out. Similarly for a high pass filter signals shorter than time constant pass through unaltered while signal longer than time constant are filtered out. The figure 7 shows the drift in angles after integration when mobile is at rest position. The final results of *DNRF* method are shown in the figure 9, 8 and 10. It also compares the output with orientation obtained from accelerometer and magnetometer. The output put of DNRF method is much smoother and fast response as compared to orientation calculated from accelerometer and magnetometer.

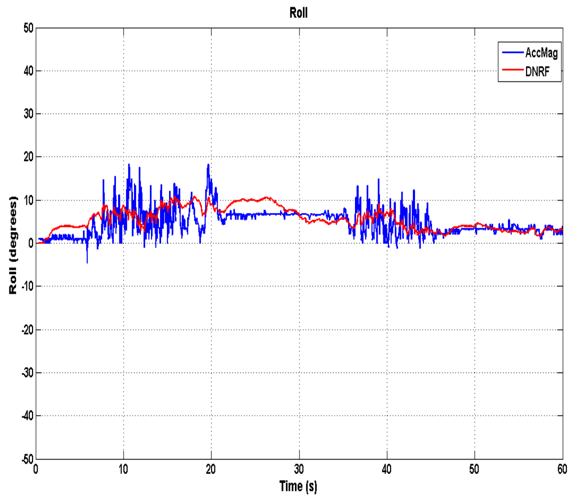

Fig. 10. Roll Angle

IV. Conclusion

In this work an affective and less computational method of smart phone orientation estimation is described which reduces the numerical as well as bias errors in gyro signal. It has been observed that the proposed *DNRF* method provides a smooth, and less noisy orientation as compared to conventional digital compass. Moreover, by adding Gimbal lock removal block it becomes possible to get orientation in any arbitrary position.

ACKNOWLEDGMENT

Author is thankful to COMSATS and HW Communications for providing support in this research work.

REFERENCES

[1] D. E. Gebre-Egziabher, J. D. Powell, and R. C. Hayward, "Design of multi-sensor attitude determination systems," *IEEE Transaction on Aerospace and Electronics Systems*, vol. 40, no. 2, pp. 627–649, APRIL 2004.

[2] Y.-S. S. Nguyen Ho Quoc Phuong, Hee-Jun Kang and Y.-S. Ro, "A dcm based orientation estimation algorithm with an inertial measurement unit and a magnetic compass," *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 859–876, Feb 2009.

[3] P. Setoodeh, A. Khayatian, and E. Farjah, "Attitude estimation by separate-bias kalman filter-based data fusion," *The Journal of Navigation, The Royal Institute of Navigation*, vol. 57, p. 261273, 2004.

[4] T. H. . J.-M. P. Robert Mahony, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transaction on Automatic Control*, vol. 53, pp. 1203–1218, 2008.

[5] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House Boston—London, 2008.

[6] D. Smittle and V. Georgiev, "Indoor localization on mobile phone platforms using adaptive dead reckoning," in *MU Summer Undergraduate Research and Creative Achievements Forum, Univ. of Missouri Columbia, MO.*, July 29 2010.

[7] W. Premerlani and P. Bizard, "Direction cosine matrix imu: Theory," Tech. Rep., May 17 2009.

[8] F. Dunn and I. Parberry, *3D Math Primer for Graphics and Game Development*. Wordware Publishing, Inc., 2002.