Angulo

Daniel Kraft

September 28, 2011

1 Introduction

Angulo is a tool which utilizes the internal sensors (for acceleration and magnetic field) of a mobile device to allow measuring angles with it. We assume that both acceleration (the Earth's gravitational field) and the surrounding magnetic field are constant and fixed in space; thus if we measure both at two times and compare the measured vectors (their directions in particular), we can calculate how the device was rotated between the measurements.

Thus you can set a "reference point" with the device pointing in one direction, and rotate it into another one, and have the angle between both directions calculated. With this tool, one can for instance find out the inclination of a slope or actually simply measure an angle between any two given directions.

The code is free (released under the GPL), check it out at `http://www.sourceforge.net/projects/angulo/`. **Angulo** runs on Google's Android operating system (and obviously needs devices which actually have the sensors used). This document is not intended as an usage introduction, for that see the built-in help text or the webpage at `http://www.domob.eu/projects/angulo.php`. Instead, I want to describe the internal logic here (in particular, how the shown results are calculated).

Angulo displays measured angles mainly in degrees; for small angles (and in particular slopes) sometimes also percent are used — where the percent value corresponds to the tangens of the actual angle. If the angle is small (below 45 degrees), **Angulo** as a utility feature displays the same angle again in percent.

2 With a Single Sensor

Assume for a moment that we only consider data from a single sensor, and let it without loss of generality be acceleration. If we measure twice, we get two vectors $a, b \in \mathbb{R}^3$. a corresponds to the acceleration at the first measurement (which **Angulo** calls the "reference point"), b to that at the second time (which is the "current" value updated continuously during runtime).

Now, if ϕ denotes the angle in-between the vectors a and b, elementary vector analysis implies that

$$\cos\phi = \frac{a \cdot b}{|a|\,|b|}. \quad (1)$$

Hence, using the arccosine we can easily calculate the angle between both directions. However, since the inner product is symmetric, the same angle will be reported if we rotate in the opposite direction — which is somewhat counterintuitive. On the other hand, the cross-product $a \times b$ is *anti*symmetric. Thus we can set the sign of ϕ depending on the direction of $n = a \times b$; since there's no "absolute reference" however, **Angulo** simply uses the sign of the sum of all component of n as the assumed sign of $\sin\phi$ and thus ϕ. That way, if a and b are flipped (which corresponds to rotating in the opposite direction), also n is flipped and the sign will be opposite. This is of course an arbitrary choice (and a lot of other options would be possible), but it makes **Angulo** display a useful sign in most circumstances. Just note to be careful in particular with the sign; but usually, the interesting quantity is anyways the magnitude of the angle and not its direction (since that is most of the time apparent).

In addition to Equation 1, another relation is

$$|\sin\phi| = \frac{|a \times b|}{|a|\,|b|}. \quad (2)$$

If we combine Equation 1 and Equation 2, we get (except for the sign of $\sin\phi$, but the same solution as above applies for the angle's sign) an even more (numerically) accurate result for the angle.

The problem with this approach is that it only measures the angle between the two actually measured vectors — which may not always correspond to the angle the device was actually rotated! For instance, if you rotate it exactly around the direction of acceleration (which usually means horizontally), the measured value *does not change at all* and consequently, **Angulo** will display $\phi = 0$ despite there obviously being a nonzero rotation. Since **Angulo** actually displays the two angles calculated independently from the direction of acceleration and magnetic field, one can sometimes mitigiate this problem by choosing one of them over the other, depending on the axis of rotation; but it may well be that *none* is correct on its own. See Section 3 for a way to combine both values into a single measurement that does not have that problem.

3 Combined Angle

As mentioned above, one gets a better estimate on the actual angle if the measurements of both sensors are combined into a single value — this is what **Angulo** displays in large, green writing on the bottom of the screen and the value is supposed to be the most accurate estimate available in most cases. This works well as long as the directions of both quantities used are not linearly dependent — but since the acceleration is usually pointing straight through the floor and the magnetic field corresponds to Earth's magnetic field, those vectors being linearly dependent means that the magnetic field is also pointing straight up or down, which in turn only happens at the magnetic poles. Thus this situation is quite unlikely, and usually the the assumption of linear independence is fulfilled.

To make it more precise, let a and b be the values of acceleration and magnetic field at the "reference measurement", respectively. Let further a and b be linearly independent. After the device is rotated, we assume that the new directions are Ra and Rb, where R is the three-dimensional rotation matrix corresponding to the movement of the device. We assume that the "current" vectors can be represented in this way and clearly $R \in SO(3)$.

3.1 Rotation Angle

Assume now that R is known. If we choose a suitable orthonormal coordinate system (with the axis of rotation being the third basis vector), the matrix representation of R will be

$$R' = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Since the trace of a matrix is invariant under basis changes, also

$$\operatorname{tr} R = 1 + 2\cos\phi. \tag{3}$$

It can further be shown (see for instance the thread at `http://www.mathworks.com/matlabcentral/newsreader/view_thread/160945`) that additionally

$$\left| \begin{pmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{pmatrix} \right| = 2\,|\sin\phi| \tag{4}$$

holds. Taking Equation 3 and Equation 4 together, we can again nicely find ϕ itself (up to the sign, but again this is not clearly defined). For the sign, the same rule of thumb is used as in Section 2 — namely all (in total six) components of $a \times Ra$ and $b \times Rb$ are summed up, and the sign of this sum is used to adapt the sign of the resulting angle.

3.2 Finding the Matrix

It remains now to actually *find* the matrix R from the four vectors a, b, Ra, Rb that are known. Note that since R is orthogonal and in particular regular, Ra and Rb are also linearly independent. As a first step, we can orthonormalize the vectors a and b and extend them to a right-handed orthonormal basis (a', b', c') of $\mathbb{R}^3$:

$$a' = \frac{a}{|a|}, \;\; \tilde{b} = b - (a' \cdot b) \cdot a' \neq 0, \;\; b' = \frac{\tilde{b}}{\left|\tilde{b}\right|}, \;\; c' = a' \times b' \tag{5}$$

Let B be the matrix formed by those basis vectors as columns; then $B \in SO(3)$ and applying $B^{-1} = B^{\top}$ onto a vector represents it in the basis (a', b', c').

Now, the columns of the representation of R in this basis are simply the images under R of the basis vectors. But since R is linear and also orthogonal, *it preserves norms and inner products.* Thus, we get Ra', Rb' and Rc' by simply applying the recipe of Equation 5 onto Ra and Rb in the places of a and b. Let B' be the matrix whose columns are the vectors Ra', Rb' and Rc', which are known this way. Then the full matrix is simply:

$$R = B'B^{\top}$$

Hence, we know all that is necessary for calculating the angle.