本文译自：<http://developer.android.com/guide/topics/sensors/sensors_motion.html>

Android平台提供了几个能够监视设备移动的传感器。其中的两个是基于硬件的（加速度传感器和陀螺仪），有三个传感器既可以是基于硬件的，也可以是基于软件的（重力传感器、线性加速度传感器和旋转矢量传感器）。例如，在某些基于软件传感器的设备上会从加速度传感器和磁力仪中提取它们的数据，但是在其他的设备上，它们也可以使用陀螺仪来提取数据。大多数Android设备都有加速度传感器和陀螺仪。基于软件的传感器的有效性是可变的，因为它们要依赖一个或多个硬件传感器来提取对应的数据。

运动传感器用于监视设备的移动，如：倾斜、振动、旋转或摆动。移动通常是直接的用户输入的反映（如，用户在游戏中操作汽车或控制一个球），但也能够是设备所处的物理环境的反映（如，驾驶汽车时的位置移动）。在第一种场景中，你要相对于设备的参照系或应用程序的参照系来监视设备的移动，在第二种场景中，你要相对于全球参照系来监视设备的移动。

通常，移动传感器本身并不用于监视设备的位置，但它们能够跟其他传感器一起来使用，例如跟地磁传感器一起来判断设备相对于全球参照系统的的位置（更多信息，请看位置传感器）。

所有的移动传感器的每个SensorEvent都会返回传感器值的多维数组。例如，一个加速度传感器的传感器事件会返回三个坐标轴方向的加速度力，陀螺仪会返回三个坐标轴方向旋转速率。这些数据是以float数组的形式跟SensorEvent参数一起被返回的。表1概要介绍了Android平台上可用的移动传感器。

表1.Android平台所支持的移动传感器

|  |  |  |  |
| --- | --- | --- | --- |
| **传感器** | **传感器数据** | **描述** | **测量单位** |
| [TYPE\_ACCELEROMETER](http://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ACCELEROMETER) | SensorEvent.values[0] | X轴方向的加速力（包括重力） | m/s2 |
| SensorEvent.values[1] | Y轴方向的加速力（包括重力） |
| SensorEvent.values[2] | Z轴方向的加速力（包括重力） |
| [TYPE\_GRAVITY](http://developer.android.com/reference/android/hardware/Sensor.html#TYPE_GRAVITY) | SensorEvent.values[0] | X轴方向的重力 | m/s2 |
| SensorEvent.values[1] | Y轴方向的重力 |
| SensorEvent.values[2] | Z轴方向的重力 |
| [TYPE\_GYROSCOPE](http://developer.android.com/reference/android/hardware/Sensor.html#TYPE_GYROSCOPE) | SensorEvent.values[0] | 绕着X轴的旋转率 | rad/s |
| SensorEvent.values[1] | 绕着Y轴的旋转率 |
| SensorEvent.values[2] | 绕着Z轴的旋转率 |
| [TYPE\_LINEAR\_ACCELERATION](http://developer.android.com/reference/android/hardware/Sensor.html#TYPE_LINEAR_ACCELERATION) | SensorEvent.values[0] | X轴方向的加速力（不含重力 | m/s2 |
| SensorEvent.values[1] | Y轴方向的加速力（不含重力 |
| SensorEvent.values[2] | Z轴方向的加速力（不含重力 |
| [TYPE\_ROTATION\_VECTOR](http://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ROTATION_VECTOR) | SensorEvent.values[0] | 沿着X轴的旋转矢量(x\* sin(θ/2)). | 无单位 |
| SensorEvent.values[1] | 沿着Y轴的旋转矢量(y\* sin(θ/2)). |
| SensorEvent.values[2] | 沿着Z轴的旋转矢量(z\* sin(θ/2)). |
| SensorEvent.values[3] | 旋转矢量的标量部分((cos(θ/2)).1 |

**1**标量部分是一个可选值。

对于移动检测和监视，旋转矢量传感器和重力传感器是使用频率最高的传感器。旋转矢量传感器尤其灵活，可广泛的应用于移动相关的任务，如，检测手势、监视角度变化、监视相对的方向变化等。例如：如果你正在开发一款使用2维或3维罗盘来增强现实感游戏，或增强相机的稳定性的应用程序，那么旋转矢量传感器是非常理想的选择。在大多场景中，使用这些传感器比使用加速度传感器和地磁传感器或方向传感器要好。

**Android开源项目的传感器**

Android开源项目（AOSP）提供了三个基于软件的移动传感器：重力传感器、线性加速度传感器和旋转矢量传感器。这三个传感器在Android4.0中被更新，并且都使用陀螺仪（其它传感器除外）来改善稳定性和性能。如果你想尝试这些传感器，可以通过使用getVendor()方法和getVersion()方法来识别它们（供应商是：Google Inc；版本号是：3）。通过供应商和版本号来识别这些传感器是必须的，因为Android系统认为这些传感器是第二传感器。例如，如果设备制造商提供了他们自己的重力传感器，那么AOSP的重力传感器会作为第二重力传感器。这三个传感器都依赖陀螺仪：如果设备上没有陀螺仪，这些传感器不会显示，并不可使用。

**使用加速度传感器**

加速度传感器测量作用于设备的加速度，包括重力作用。下列代码显示了如何获得默认的加速度传感器的一个实例：

private SensorManager mSensorManager;

private Sensor mSensor;

...

mSensorManager = (SensorManager) getSystemService(Context.SENSOR\_SERVICE);

mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER);

从概念上，加速度传感器通过测试作用于传感器本身的作用力（Fs），并使用下列公式来判断作用于设备的（Ad）加速度：

Ad = - ∑Fs / mass

但是，重力作用始终会影响下列公式的测量结果：

Ad = -g - ∑F / mass

由于这个原因，当设备位于桌面（并没有加速度）时，加速度传感器的读取结果时g = 9.81 m/s2。同样，当设备自由落体运动时，因为快速向下的加速度是：9.81 m/s2，所以加速度传感器读取的结果是：g = 0 m/s2。因此要测量设备的实际加速度，重力作用的影响必须要从加速度传感器数据中删除。这种问题可以通过高通滤波来实现。相反，低通滤波能够被用来隔离重力作用。下面的示例演示了如何做这件事情：

public void onSensorChanged(SensorEvent event){

// In this example, alpha is calculated as t / (t + dT),

// where t is the low-pass filter's time-constant and

// dT is the event delivery rate.

final float alpha = 0.8;

// Isolate the force of gravity with the low-pass filter.

gravity[0] = alpha \* gravity[0] + (1 - alpha) \* event.values[0];

gravity[1] = alpha \* gravity[1] + (1 - alpha) \* event.values[1];

gravity[2] = alpha \* gravity[2] + (1 - alpha) \* event.values[2];

// Remove the gravity contribution with the high-pass filter.

linear\_acceleration[0] = event.values[0] - gravity[0];

linear\_acceleration[1] = event.values[1] - gravity[1];

linear\_acceleration[2] = event.values[2] - gravity[2];

}

**注意：**你能够使用不同的技术来过滤传感器数据。上述的代码示例使用了一个简单的过滤器常量（alpha）来创建一个低通滤波器。该滤波器常量源自时间常量（t），它粗略的表述了过滤器添加给传感器事件的延迟和传感器事件发送频率(dt)。为了演示，示例代码使用一个0.8的alpha值。如果你要使用这种滤波方法，你需要根据实际情况选择不同的alpha值。

加速度传感器使用标准的传感器器坐标系统。在实践中，当设备以它自然的方向被平方在桌面时，下列条件是适用的：

1. 如果在左边推动设备（让它向右移动），那么X轴的加速度值就是正值；
2. 如果在底部推动设备（让它向上移动），那么Y轴的加速度值就是正值；
3. 如果向着天空，用A m/s2 的加速度来推动设备，那么Z轴的加速度值就是A+9.81，它对应了设备的加速度（+A m/s2）减去重力作用（-9.81 m/s2）。
4. 静止的设备的有一个+9.81的加速度值，它对应设备的加速度（0 m/s2减去重力作用，也就是-9.81 m/s2）。

通常，加速度传感器是一种用于监视设备移动的比较好的传感器。几乎每种Android手持设备和平板电脑都有加速度传感器，而且它的能耗要比其他的移动传感器要低10倍。唯一的缺点是，你要实现低通和高通滤波来消除重力作用和减少噪音的影响。

Android SDK提供了一个如何使用加速度传感器的示例程序（Accelerometer Play）

**使用重力传感器**

重力传感器提供了三个维度的矢量，用来指示重力的方向和重量。下列代码显示了如何获取一个默认的重力传感器的实例：

private SensorManager mSensorManager;

private Sensor mSensor;

...

mSensorManager = (SensorManager) getSystemService(Context.SENSOR\_SERVICE);

mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE\_GRAVITY);

单位与加速度传感器所使用的单位（m/s2）相同，并且坐标系统也与加速度传感器所使用的坐标系相同。

**注意：**当设备处于静止状态时，重力传感器的输出应该与加速度传感器的输出相同。

**使用陀螺仪**

陀螺仪以rad/s（弧度/每秒）为单位围绕设备的X、Y、Z轴来测量速率或旋转角度。下列代码显示了如何获取一个默认的陀螺仪的实例：

private SensorManager mSensorManager;

private Sensor mSensor;

...

mSensorManager = (SensorManager) getSystemService(Context.SENSOR\_SERVICE);

mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE\_GYROSCOPE);

该传感器的坐标系统与加速度传感器所使用的坐标系统是相同的。逆时针方向旋转是正值，也就是说，如果设备是逆时针旋转，那么观察者就会看到一些有关以设备原点为中心的正向的X、Y、Z轴的位置。这是标准的正向旋转的数学定义，并且与方向传感器所使用的用于滚动的定义不同。

通常，陀螺仪的输出会被集成到时间上，以便计算在一定时间不长之上旋转角度的变化。例如：

// Create a constant to convert nanoseconds to seconds.

private static final float NS2S = 1.0f / 1000000000.0f;

private final float[] deltaRotationVector = new float[4]();

private float timestamp;

public void onSensorChanged(SensorEvent event) {

// This timestep's delta rotation to be multiplied by the current rotation

// after computing it from the gyro sample data.

if (timestamp != 0) {

final float dT = (event.timestamp - timestamp) \* NS2S;

// Axis of the rotation sample, not normalized yet.

float axisX = event.values[0];

float axisY = event.values[1];

float axisZ = event.values[2];

// Calculate the angular speed of the sample

float omegaMagnitude = sqrt(axisX\*axisX + axisY\*axisY + axisZ\*axisZ);

// Normalize the rotation vector if it's big enough to get the axis

// (that is, EPSILON should represent your maximum allowable margin of error)

if (omegaMagnitude > EPSILON) {

axisX /= omegaMagnitude;

axisY /= omegaMagnitude;

axisZ /= omegaMagnitude;

}

// Integrate around this axis with the angular speed by the timestep

// in order to get a delta rotation from this sample over the timestep

// We will convert this axis-angle representation of the delta rotation

// into a quaternion before turning it into the rotation matrix.

float thetaOverTwo = omegaMagnitude \* dT / 2.0f;

float sinThetaOverTwo = sin(thetaOverTwo);

float cosThetaOverTwo = cos(thetaOverTwo);

deltaRotationVector[0] = sinThetaOverTwo \* axisX;

deltaRotationVector[1] = sinThetaOverTwo \* axisY;

deltaRotationVector[2] = sinThetaOverTwo \* axisZ;

deltaRotationVector[3] = cosThetaOverTwo;

}

timestamp = event.timestamp;

float[] deltaRotationMatrix = new float[9];

SensorManager.getRotationMatrixFromVector(deltaRotationMatrix, deltaRotationVector);

// User code should concatenate the delta rotation we computed with the current rotation

// in order to get the updated rotation.

// rotationCurrent = rotationCurrent \* deltaRotationMatrix;

}

}

标准的陀螺仪提供了原始的旋转数据，并不带有任何过滤或噪音和漂移（偏心）的校正。在实践中，陀螺仪的噪音和漂移会引入错误，因此需要对此进行抵消处理。通常通过监视其他传感器，如重力传感器或加速度传感器来判断漂移（偏心）和噪音。

**使用线性加速度传感器**

线性加速度传感器用一个三维向量，提供设备坐标系中每个坐标轴的加速度，但不包括重力。下列代码显示了如何获取默认的线性加速度传感器的实例：

private SensorManager mSensorManager;

private Sensor mSensor;

...

mSensorManager = (SensorManager) getSystemService(Context.SENSOR\_SERVICE);

mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE\_LINEAR\_ACCELERATION);

概念上，这种传感器根据下列关系来 提供加速度数据：

线性加速度 = 加速度 – 重力加速度

通常，在想要获取没有重力影响的加速度数据时，使用这种传感器。例如，你能够使用这种传感器来看你的汽车运行速度。线性加速度传感器始终有一个偏移量，你需要删除这个偏移量。做这件事的最简单的方法就是在你的应用程序中建立一个校准处理。在校准期间，你能够要求用户把设备放到桌面上，读取所有三轴的偏移量，然后从加速度传感器中读取的数据中直接减去对应的偏移量，来获取线性加速度。

该传感器的坐标系统与加速度传感器的坐标系相同，它的测量单位是（m/s2）。

**使用旋转矢量传感器**

旋转矢量用角和轴的组合来代表设备的取向，即设备绕着轴（X、Y或Z）旋转角度θ。下列代码显示了如何获取默认的旋转矢量传感器的实例：

private SensorManager mSensorManager;

private Sensor mSensor;

...

mSensorManager = (SensorManager) getSystemService(Context.SENSOR\_SERVICE);

mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE\_ROTATION\_VECTOR);

这三个旋转矢量元素分别被表述以下形式：

x\*sin(θ/2)

y\*sin(θ/2)

z\*sin(θ/2)

其中sin(θ/2)是旋转的数量，旋转轴的方向是旋转矢量的方向。

这三个旋转矢量元素等于一个四元数（cos(θ/2),X\*sin(θ/2),Y\*sin(θ/2),Z\*sin(θ/2)的最后三部分。旋转矢量元素是无单位的。X、Y和Z轴的定义与加速度传感器相同。该参照坐标系统是用直接正交基来定义的（如图1）

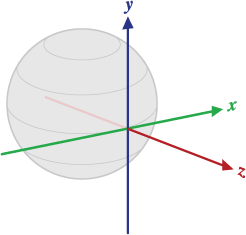


图1.旋转矢量传感器使用的坐标系统

这个坐标系统有以下特点：

1. X轴是用向量积YxZ来定义的，它在设备的当前位置与地面相切，并指向东方；
2. Y轴在设备当前位置与地面相切，并指向地磁场的北极。
3. Z指向天空，并垂直与地面。

Android SDK提供了一个演示如何使用旋转矢量传感器的示例应用程序。这个实例应用程序位于API Demos代码中（OS-RotationVectorDemo）