A Design and Application of Tilt Angle Algorithm Based on the 3-axis Acceleration Sensor

Xianzhen Wu, Zejun Wu*, Boya Zhao and Mingjiang Wang
Key Laboratory of IOT Terminal Pivotal Technology
Harbin Institute of Technology Shenzhen Graduate School
Shenzhen, China
wuxianzhen12@163.com, *wuzejun@126.com, aptxboya@qq.com, mjwang@hit.edu.cn

***Abstract*—This article proposes an algorithm for detecting the direction and the magnitude of the tilt angle based on BMA 150—a 3-axis acceleration sensor, and a page-turning function in Android daily application software is implemented based on this algorithm. This paper firstly introduces the architecture of Android sensor system and the sensor application module. Secondly, it uses the threshold optimization algorithm to identify the five device location states which are reliable, high distinguishability. And then, it makes use of trigonometric function to calculate the tilt angle, thus achieving high precision page-turning function. The experimental data analysis shows that this design gets more accurate tile angle through the algorithm optimization. The angle error is only ±0.39 °. This algorithm effectively removes invalid data. The result proves the page-turning function's feasibility and correctness.**

Keywords: Android, acceleration sensor, tilt angle, page-turning, threshold optimization

I. INTRODUCTION

With the development of science and technology, smart phones have become an integral part of our everyday life. Customers put forward higher requirements according to the operability for application software and software operating comfort. Android is the name of the open source mobile phone operating system based on Linux and it is developed by the Open Handset Alliance, which is set up jointly by Google Company and other 33 companies and enterprises focusing on mobile technology, hardware and software design, mobile phone chips and other fields[1]. The Android platform consists of operating system, middleware, user interface, and application software. It is a truly open platform for mobile application development.

As the increasing demand for Android smart phones, sensors have become a major bright spot of Android mobile phone hardware. Currently, the acceleration sensor is mainly used in the game [2]-[4]. It is rarely used in the daily application software. At present, many papers mainly focus on describing the framework of Android system [5], but there is little study on implementation and optimization for sensors in the Android system. Acceleration sensor has been widely used in the Android phone. However, the data obtained from the acceleration sensor is not optimized. It is original data. So it is important to propose a new algorithm that can effectively optimize and filter the invalid data.

To solve these problems, this paper describes the Android acceleration sensor system hierarchy and presents the page-turning function in the daily application such as calendar and eBook. This function is achieved according to the tilt angle which is calculated by the threshold optimization algorithm and the tilt angle algorithm based on the acceleration sensor.

II. THE STRUCTURE OF SENSOR SYSTEM

A. The Performance of Sensor Hardware

The acceleration sensor in the consumer products is divided into two categories: digital output and analog output. This paper uses a sensor with digital output named BMA 150 3-axis accelerometer. It is directly connected to ARM Cortex-A8 processor through the I^2C interface, so as to realize the function of data measurement and data acquisition. BMA 150 3-axis accelerometer is a highly accuracy G-sensor. Its working current is 200 µA and resolution ratio is 10 bits. This sensor is suitable for the measurement.

B. The Architecture of Android Sensor System

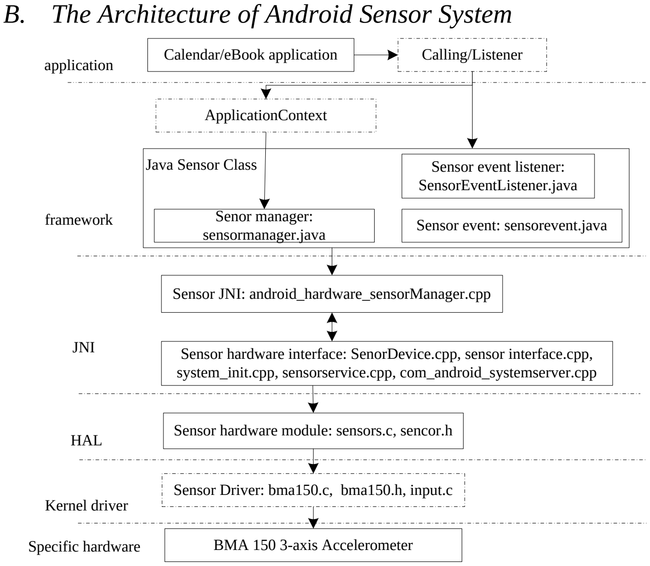

Figure 1. The architecture of Android sensor system

The sensor system of Android consists of the following components: the layers involved from bottom to top are kernel driver layer, hardware abstraction layer, Java native

*Corresponding author

Funded by: The Key Laboratory Projects of Shenzhen (CXB201104220018A)

interface layer, framework layer and the application layer [7]. How application layer makes use of these levels and the relationship between the levels is described in Figure 1.

Page-turning function is primarily implemented and described in application layer, but its implementation has to rely on a variety of interfaces provided from low layers, data transmission and the interaction of sensor hardware. The process of realizing the page-turning function is described as follows:

1) The sensor driver is *bma150.c*. I^2C bus driver is loaded first, and then the sensor driver is loaded. The main function is to obtain acceleration values. When it reports data, it will trigger an interrupt, and then submit a task report of the values to the queue in the interrupt handler function and disable interrupts. The sensor driver reads acceleration data in the work queue and reports to the input subsystem, and finally enables interrupts.

2) The sensor HAL layer is used to sensor system achieve Android adaptation layer according to their situation. Its HAL interface definition is in the file *sensors.h*. Poll function pointer is the key point in this file. It must realize blocking until input events are available, and then it will return an array of the sensor data. *Sensors.c* interacts with the underlying layers to obtain data.

3) The sensor JNI layer is considered as supporting role in the entire architecture. In short, it completes a task of transforming from C++ language to Java language. JNI layer provides a range of interfaces for framework layer. These interfaces use callback functions such as sensors_data_poll() to interact with the HAL layer, and these callback functions such as poll() have been defined in the HAL layer. *Com_android_systemserver.cpp* mainly starts a service, and *android_hardware_sensorManager.cpp* is mainly used for the sensor data control in this service.

4) The sensor framework layer provides a variety of classes and objects. It is available for the use of the upper application. The specific communication of *SensorManager.java* is realized by calling *android_hardware_sensorManager.cpp* through JNI. Its subclass SensorThreadRunnable realizes the accelerometer data access.

5) Application layer uses the API provided by the framework layer to receive data for calculation. The page-turning function is implemented according to the calculation result.

C. The Structure of the Sensor Application Module

The working mechanism of sensors in Android is one kind of event mechanism. Event mechanism is used to handle events. It consists of event source, event and listener [6]. The working mechanism of sensors has its own characteristics: first, the sensor must use the SensorManager to register a listener while other components can register listeners by themselves. Second, when the application is not visible, common components naturally become ineffective, but the sensor will continue to work until you manually turn it off. Combining with characteristics of the activity's life cycle, the development idea of the opening and closing timing of sensor is shown in Figure 2.

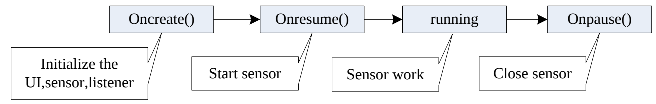

Figure 2. The development idea of sensor

In Android system, the event source of sensor is the class Sensor. The class SensorEvent is corresponding to the event. The class SensorEventListener plays a listener's role. Using the getSystemService() of the activity returns an SensorManager object. The function getDefaultSensor (Sensor.TYPE_ACCELEROMETER) of this object can request to update accelerometer. The registerListener() registers the SensorEventListener and makes the sensor work. The unregisterListener() is used to cancel sensor.

SensorEventListener is used to monitor sensor events. Once the sensor value changes, it will execute the corresponding method onSensorChanged(). It is in this method that this paper filters the data and calculates the direction and the magnitude of the tilt angle.

The page-turning function is implemented according to tilt angle as shown in Table I:

TABLE I. THE PAGE-TURNING FUNCTION ACCORDING TO TILT ANGLE

Description	Calendar /eBook function
Tilted(10～30，30～40，40～50，50°～60，70～80，80～90）to the left	Flip forward by (1/2/3/4/5/6) month/ Flip forward (1/2/3/4/5/6) page
Tilted(10～30，30～40，40～50，50°～60，70～80，80～90）to the right	Flip backward by (1/2/3/4/5/6) month/ Flip backward (1/2/3/4/5/6) page
Tilted(10～30，30～40，40～50，50°～60，70～80，80～90）forwards	Flip forward by (1/2/3/4/5/6) year
Tilted(10～30，30～40，40～50，50°～60，70～80，80～90）backwards	Flip backward by (1/2/3/4/5/6) year

III. TILT ANGLE ALGORITHM IMPLEMENTATION

A. Setting Threshold of The Initial Value and Device States Detection Module

BMA 150 3-axis acceleration sensor measures the acceleration in m/s^2 in three axis directions, including the acceleration of gravity and motion. The relationship between the acceleration coordinate system and the device screen coordinate system is shown in Figure 3.(a). Sensor takes the upper left corner as the origin. The x-axis direction is parallel to the left edge or the right edge of the screen. The forward direction is positive. The y-axis direction is parallel to the bottom edge or the top edge of the screen. The direction to the right is the positive direction. The z-axis direction is perpendicular to the screen. The upward direction is the positive direction. The Figure 3.(b) shows the coordinate system when the device tilts. The θ is the tilt angle.

Therefore, we can use the sensor to detect motion of the device. Whether the device is in a static state or in a fast moving state, the signal itself has some errors that are caused by transient non-stationary characteristics and jitter. Application needs to remove these errors for preventing their impact on the future results. When the device is placed horizontally and the screen is up, which is considered as the initial state, Android system collects data from sensor in the x-axis, y-axis, and z-axis. The values are shown in Fig. 4.

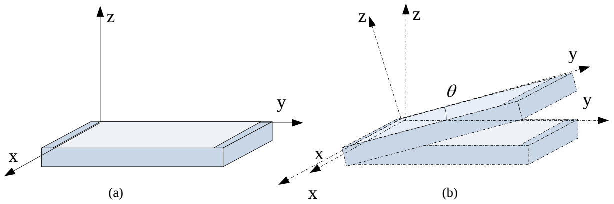

Figure 3. The accelerometer coordinate system: (a) when the device is placed horizontally (b) when the device tilts

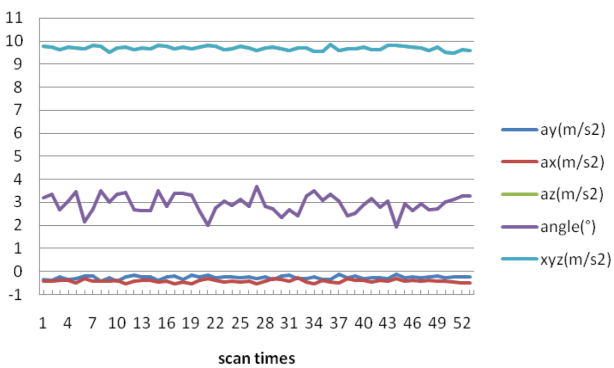

Figure 4. Acceleration in each direction and angle when horizontally placed (screen up)

We can see that: the acquired data is in the -0.42~-0.03m/s^2 range (average of -0.2) in y-axis and -0.3~-0.7m/s^2 (average of -0.42) in x-axis while the theoretical initial value of the acceleration should be 0. In the z-axis, although the device has no trend of movement, it is affected by gravity, so the sensor value is the acceleration caused by the support force from the desktop. Data range is 9.4~10.0 m/s^2. It needs to be ensured that the data of x-y plane in stationary horizontal state has no effect on the accumulation or processing of the data obtained while moving. Here set the zero point thresholds on the x-axis and y-axis. The y-axis threshold is φ_y =0.22, the x-axis threshold is φ_x=0.3.

The device state is divided into five states according to the threshold value: tilted to the left state, tilted to the right state (rotated on the y-axis), tilted forwards state, tilted backwards state (rotated on the x-axis) and the initial state (placed horizontally). The conversions between these five states are shown in Figure 5.

The Figure 5 shows the definition of the five states. If the absolute value of the difference between acceleration in y-axis and -0.2 is less than φ_y and the absolute value of the difference between acceleration in x-axis and -0.42 is less than φ_x, the state is the initial state. When the acceleration change in the y-axis is positive and the change is greater than φ_y, and the acceleration change magnitude in y-axis is greater than that in x-axis, and the value in y-axis isn't in the initial state, it is the tilted to the left state. The difference between the tilted to the right state and the tilted to the left state is that the value change of the tilted to the right state in the y-axis is negative and less than -φ_y. When the acceleration change in the x-axis is positive and this change is greater than φ_x, and the acceleration change magnitude in the x-axis is greater than that in the y-axis, and the value in x-axis isn't in the initial state, it is the tilted forwards state. The difference between the tilted backwards state and the tilted forwards state is that the value change of the tilted backwards state in the x-axis is negative and less than -φ_x. The obtained experimental data shows that these thresholds effectively remove small jitter data and the threshold algorithm can improve the classification accuracy and robustness.

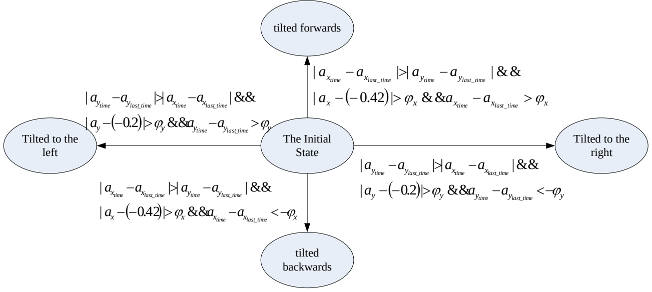

Figure 5. The five states transition diagram

B. *Calculating Angle of Inclination*

During the process of swing, the hand moves the device firstly, and then it starts to apply a force of the reverse direction to let the device still down. Finally the device goes back to the initial position or being tilted. Although during the process of swing it is not at constant speed, the device is tilted to a maximum angle and then the gravitational acceleration is the only element in working at that time. Based on this principle, this paper realizes the page-turning function by calculating the tilt angle.

Since acceleration sensors only obtain gravity in a stationary state, there is a gravitational acceleration of 1 g. The tilt angle in the vertical plane is calculated by using this property. When device is inclined in space, 3-axis accelerometer measures the acceleration component in x-axis, y-axis and z-axis. We use inverse trigonometric function algorithm. Firstly, the algorithm makes the magnitude of the acceleration vector projected onto X-Y plane, and then computes the vertical angle and lastly converts it to degrees.

The definition of the formulas is as follows:

$$\theta = \arctan \frac{\sqrt{a_x^2 + a_y^2}}{a_z} \tag{1}$$

$$\sqrt{a_x^2 + a_y^2 + a_z^2} = 1g\ . \tag{2}$$

Equation (1) is the basic principles of measuring the tilt angle of the object. If the device has an acceleration of movement at the same time, this formula is no longer accurate. So it is necessary to add a restriction to this formula: the resultant acceleration is about 1 g. In Figure 6, the a_{xyz} is the resultant acceleration. The *angle1* is the calculated tilt angle without any restriction. The *angle2* is the calculated tilt angle with restriction of equation (2).

Figure 6 shows the trend of two calculated tilt angles when the device was inclined. Two curves are similar when the device is statically placed, but the peaks of them are very different when the device is tilted in the fast moving. The peak of the *angle1* without restriction is higher than the *angle2* wave while it does not match with the actual measured angle. The actual angles are 22° and 40°. The *angle2* is more accurate than the *angle1* in the figure. This paper selected three angles for repeated measurement and calculation in 30 times, and the result is shown in Table II. The three angles are 22°, 40° and 60°. After experimental analysis of these data, the angle error is only ±0.39°. The angle range is 0~180°according the tilt angle algorithm. This paper achieves the page-turning function in 0~90°.

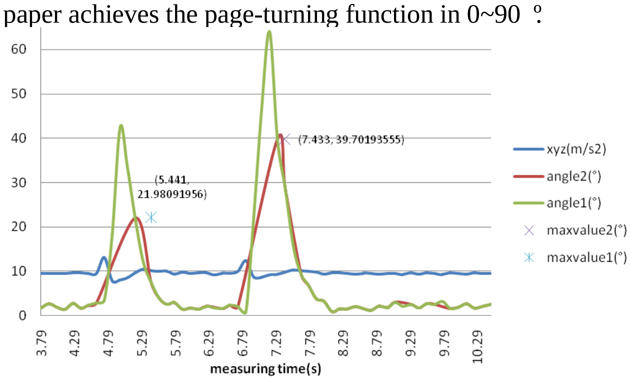

Figure 6. Resultant acceleration and tilt angle under two conditions

TABLE II. THE TILE ANGLE UNDER DIFFERENT CIRCUMSTANCES AND ACCURACY

The title angle of actual measurement	The title angle after calculation (mean)	Deviation
22°	21.96858704°	-0.0314°
40°	40.05685438°	-0.0568°
60°	59.61886069°	-0.38113°

Based on these formulas, the max-angle is calculated by determining the effective wave crest. But the factor affecting the peak value to be considered is that the sensitive sensors are possible to produce value changes for jitter. This kind of value changes is invalid. The applications use a fixed threshold optimization algorithm to remove these invalid values. Through the observation of experimental data in Fig. 4, the average value of horizontal angle is 2.71°±1.2°. So the threshold is φ_{angle}=1.2°. If the data meet the following condition: $a_{angletime}$-$a_{angletime-1}$>φ_{angle}, put the data to compare to obtain the max-angle. The value of φ_{angle} is smaller, the reaction is more sensitive. Tests show that φ_{angle} meet angle requirements. The maxvalue1 and maxvalue2 is the calculated max-angle. They are consistent with actual measurement of angles. The algorithm effectively blocks jitters and invalid data.

C. The Result of the Experiments

The Calendar application is shown in Figure 7. When the device is placed horizontally, the date in the left figure is May, 2013. When the device is tilted to the left to 12.45°, the calendar automatically jumps to April, 2013. The eBook application is shown in Figure 8. When the device is tilted to the right to 13.74°, eBook flips backward 1 page. The change of angle has not been happened in the subsequent period of time when device does not move. They meet the design requirements.

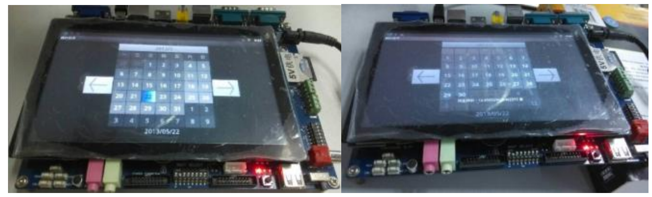

Figure 7. Horizontal and tilted to the left to 12.45 °in calendar

Figure 8. Horizontal and tilted to the right to 13.74 °in eBook

IV. CONCLUSIONS

This paper introduces the structure of Android sensor system and presents an algorithm for detecting the direction and magnitude of the tilt angle. Page-turning function in daily application has been tested on BMA 150 3-axis accelerometer based on this algorithm. The feasibility and the correctness of the threshold optimization algorithm and tile angle algorithm are proved in this paper. The threshold optimization algorithm effectively eliminates invalid data and dithering. It improves the classification accuracy and robustness in application. Through comparative analysis, tilt angle algorithm under the condition of 1g obtains more accurate angle. The average angle error is only ±0.39°. Page-turning function's main features are: (1) Realize the device position state detection. This detection is more reliable, higher distinguishability. (2) Accurately achieve page-turning function according to angle. The page-turning function basically meets the design requirements. This algorithm is of great significance and good expandability in practical daily application. For example, this algorithm can be used to select songs or reject phone calls. It depends on what we actually need.

REFERENCES

[1] A. H. Yang, Y. M. Pu, "the Algorithm of Gravity Master Sensor Game Based on the Android Platform," National Conference on the Information Technology and Computer Science, 2012, pp.678-671.

[2] Y. Chen, K. J. Qiu, H. Xu, F. C. Zhu, "The Design and Implementation of the Blind Mobile Phone based on The Gravity Sensor," Electronic Test, 2013(6), pp.110-111.

[3] Y. Z. Li, C. X. Li, H. Wang, "Light sensor realization and data optimization algorithm based on Android," Journal of Xi' an University of Posts and Telecommunications, vol.17, no.4 Jul. 2012, pp.76-78.

[4] S. H. Guo, The Key of Grand Creation: The Interactive Application Development of Android for Cellphone, China Railway Publishing House, 2011, pp.150-153.

[5] S. Komatineni, D. MacLean, S. Hashimi, Pro Android 3, Posts & Telecom Press, 2011, pp.715-749

[6] D. Han, Android creative examples, Beijing University Of Post & Telecomunication, 2012, pp.347-351

[7] http://blog.chinaunix.net/uid-26997997-id-3485501.html. Feb. 2013.