天线数据采集程序

代码说明书

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制： | 张琛 | 日期： | 2014-12-25 |
| 审核： | 张琛 | 日期： | 2014-12-25 |
| 审核： | name | 日期： | yyyy-mm-dd |
| 批准： | name | 日期： | yyyy-mm-dd |

修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 版本号 | CR号 | 修改描述 | 作者 |
| 2014-12-25 | 1.0 |  | 初稿 | 张琛 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

1，功能说明

2，输入输出和内部配参

3，技术方案描述

3.1 XXXXX模块(或文件或函数)

3.1.1 XXXXXX模块(或文件或函数)

3.2 XXXXXXXX

XXXXXXXX

XXXXXXXX

关键词：

Android，惯性传感器，相机，智能手机

摘要：

在《天线工参计算代码说明书格式》文档中描述的天线工参计算算法中，需要采集智能手机的惯性传感器数据和彩色相机数据。在本文档中，描述了一个基于Android平台的天线数据采集程序的实现。

缩略语清单：

|  |  |  |
| --- | --- | --- |
| 缩略语 | 原语言全称 | 中文解释 |
| IMU | Inertial Measurement Unit | 惯性测量单元 |
|  |  |  |
|  |  |  |
|  |  |  |

# 功能说明

本应用为Android平台的天线数据采集程序。采集的数据包括：

1. 照片/视频
2. IMU运动传感器数据

采集数据之前，用户需要设定好：

① 工程名

② 照片分辨率；若是视频，还需设定码率、帧率；

③ 焦距，以及对焦模式；

④ 设备朝向，以及下倾角；

⑤ IMU数据采集模式设为单独或者 合并采集

# 输入输出和内部配参

（整个算法的输入输出和内部配参）

输入数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 变量代码名 | 中文名称 | 描述 | 默认值和取值范围（单位） |
| \_picNames（MainActivity.java） | 照片文件名 | {照片名, 时间戳} 字符串对保存在 collection-proj.xml文件 |  |
| \_picTimestamps | 拍照时间戳 | Null |  |
| \_newSessionNode | IMU数据 | 传感器数据保存在sensor\_ID.xml文件 |  |

输出数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 变量代码名 | 中文名称 | 描述 | 默认值和取值范围（单位） |
| \_confFile | 工程描述文件 | 保存当前工程组织信息 |  |
| dataXmlFile | IMU数据文件 | 保存一次采集过程中的IMU数据 |  |
|  |  |  |  |

内部配置参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 变量代码名 | 中文名称 | 描述 | 默认值和取值范围（单位） |
| cntDataXml | IMU数据文件计数器 | 统计当前工程中已有IMU文件个数 |  |
| dataXmlTask | 数据文件存储任务 | 异步存储任务，负责保存IMU文件 |  |
| confXmlTask | 工程描述文件存储任务 | 异步存储任务，负责保存工程描述配置文件 |  |

# 技术方案描述



图1，算法总体模板与流程

## 主模块 OpenCameraZC

1，功能说明，对应文件

本模块为程序入口主模块，负责程序的界面显示、逻辑实现。主要文件为：

MainActivity.java,

Preview.java,

CollectionProjXml.java

2，输入输出和内部配参的列表（表格模板同上）

输入数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 变量代码名 | 中文名称 | 描述 | 默认值和取值范围（单位） |
| class MainActivity | 主类 | 程序启动入口 |  |
| class Preview | 相机预览窗口类 | 相机预览界面显示，以及点击拍照时的逻辑部分 |  |
| class CollectionProjXml | 工程描述类 |  |  |

输出数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 变量代码名 | 中文名称 | 描述 | 默认值和取值范围（单位） |
| collection-proj.xml | 工程描述文件 |  |  |
| pic\_ID.jpg | 照片文件 | ID为文件序号 |  |
| sensor\_0.xml | IMU数据文件 | ID为文件序号 |  |

3，此模块的流程图



图2，主模块 OpenCameraZC流程

4，流程步骤的说明，按照流程图来写，对应的代码文件

MainActivity.java:

* + - * 1. 主程序入口启动 onCreate()； 继续 onResume()；
        2. 界面布局 layoutUI()
        3. 注册IMU监听单元 \_listener.registerWithSensorManager()
        4. 相机预览界面启动 preview.onResume()
        5. 主程序暂停 MainActivity.onPause()
        6. 主程序销毁 MainActivity.onDestroy()

Preview.java:

* + - * 1. 相机预览界面启动 preview.onResume()
        2. 打开相机 openCamera()
        3. 设定拍照参数 setupCameraParameters()

负责处理主要拍照参数设定逻辑，例如对焦模式、焦距、曝光、白平衡、ISO、etc.

* + - * 1. 对焦完成后拍照 takePictureWhenFocused()

负责处理拍照时（e.g., 自动对焦完成后、手动点击拍照按钮）数据采集逻辑。

* + - * 1. 预览界面暂停（e.g., 程序切出时调用）onPause()

### ~~模块1的支线模块1~~

~~（模板同模块1）~~

### ~~模块1的支线模块2~~

~~（模板同模块1）~~

## 模块2： MySensorListenerTest

1，功能说明，对应文件

本模块为主模块OpenCameraZC的依赖lib之一。是IMU单元监听接口，用于获得并导出IMU数据。对应文件:

MySensorListener.java

Consts.java

2，输入输出和内部配参的列表（表格模板同上）

输出数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 变量代码名 | 中文名称 | 描述 | 默认值和取值范围（单位） |
| sensorData | IMU传感器数据 | MySensorListener.MySensorData类型的数据，实际为各种IMU数据集合的包装类。通过\_listener.getSensorData获取 |  |

3，此模块的流程图



图3，MySensorListenerTest模块流程图

4，流程步骤的说明，按照流程图来写，对应的代码文件

MySensorListener.java:

1) registerWithSensorManager() 向主模块中获得的 SensorManager 注册监听器；

2) onSensorChanged() 回调函数为数据获取接口，每一帧新数据在此被填充到各个缓存中；

3) unregisterWithSensorManager() 结束时释放监听。

## 模块3： SensorDataXml

1，功能说明，对应文件

本模块为主模块OpenCameraZC的依赖lib之二。主要功能为将传入的MySensorListener.MySensorData数据填充到xml数据结构，以便序列化到文件。主要功能位于NewSessionNode.java，与主模块中的CollectionProjXml.java功能类似，由于历史原因，作为一个单独的lib，被其他模块调用。

2，输入输出和内部配参的列表（表格模板同上）

输入数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 变量代码名 | 中文名称 | 描述 | 默认值和取值范围（单位） |
| sensorData | IMU传感器数据 | MySensorListener.MySensorData类型的数据 |  |

3，此模块的流程图



图4，SensorDataXml模块流程图

4，流程步骤的说明，按照流程图来写，对应的代码文件

见主模块中 Preview.java 文件：

1） 实例化对象：

|  |
| --- |
| \_newSessionNode = new NewSessionNode(); |

|  |
| --- |
| nsNode.setBeginTime(beginTime); |

2） 传入IMU传感器数据：

|  |
| --- |
| nsNode.setEndTime(System.*currentTimeMillis*() \* Consts.*MS2S*);  nsNode.addNode(\_mainActivity.\_listener.getSensorData()); |