趣感科技“微马”智能护膝系统介绍

**系统架构**

系统结构上采用三级结构：

1. 节点上报测量的加速度计、陀螺仪、磁力计的原始数据
2. 微处理器收到惯性传感器上报的原始数据后，应用融合算法计算各节点的姿态数据并上报姿态数据给上位机（PC，安卓手机或苹果手机）。
3. 上位机收到融合后的姿态数据后进行膝关节姿态识别计算，并将计算得到的姿态信息结合原始数据对跑步过程中的步频、步幅等指标进行解算。

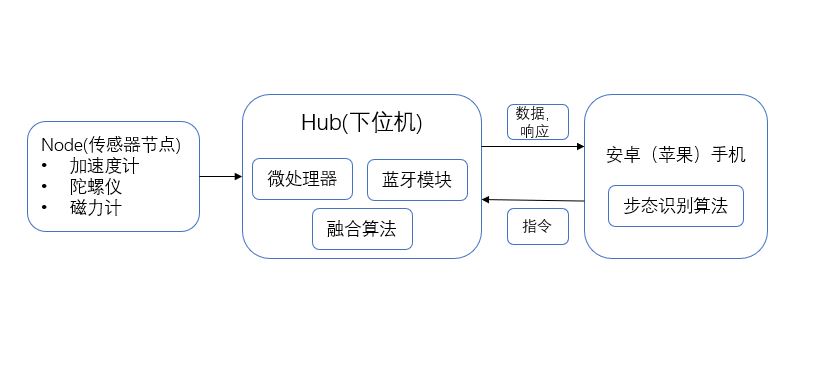
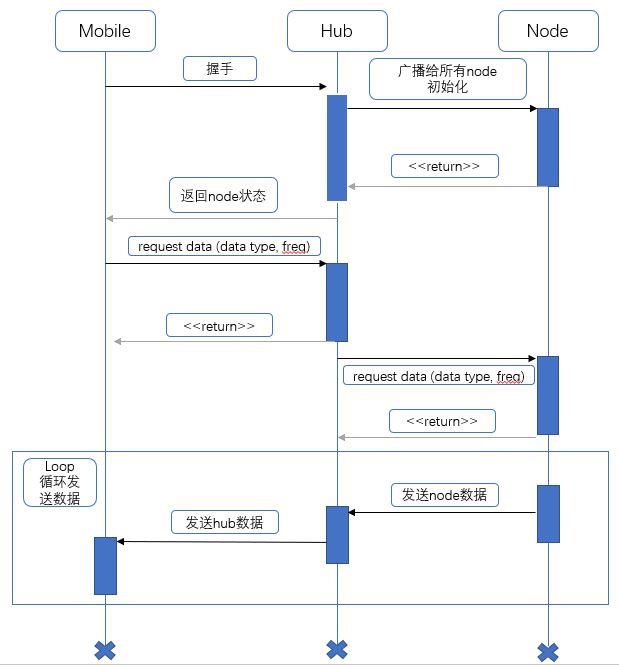


图1. 系统架构



# 程序架构

手机端

* **MiniMarathon**
  + **(C) BTPort**
    - string **getBtPort**() : 初始化蓝牙，返回蓝牙名称。
    - bool **Connect**(string device): 连接设备
    - bool **SendData**(IntPtr dataPtr, uint offset, uint count): 发送数据。
    - void **rawDataReceived**(uint nodeid, float accX, float accY, float accZ,float gyroX, float gyroY, float gyroZ, float magX, float magY, float magZ): 接收原始数据
    - void **SetInitQ**(): 设置初始化位置。
  + **(C) MotionCaptureWrapper**
    - void **init**(): 初始化算法库
    - void **mc\_onDataReceived**(IntPtr pMC, [In] byte[] pbuf, int lbu): 获取数据
    - void **mc\_reset**(IntPtr pMC): 复位
    - void **mc\_GetCurrentAngle**(IntPtr pMC, [In] double[] outbuf, int buflen);获取当前膝关节角度
    - void **mc\_UpdateJoints**(IntPtr pMC);更新角度
    - void **mc\_GetStrideLength** (IntPtr pMC, int nodeid, float length): 步幅
    - void **mc\_GetKneeVibration** (IntPtr pMC, int nodeid, float vibra); 膝关节落地抖动。
    - void **mc\_GetStrideVelocity** (IntPtr pMC, int nodeid, float velocity); 步速
    - void **mc\_getCadence** (IntPtr pMC, int nodeid, float cadence); 步频
    - void **mc\_getStrideTime** (IntPtr pMC, int nodeid, float time); 步幅周期时间
    - void **mc\_getSwingTime** (IntPtr pMC, int nodeid, float time); 摆动时长
    - void **mc\_getStanceTime** (IntPtr pMC, int nodeid, float time); 落地时长
    - void **mc\_getKneeForce** (IntPtr pMC, int nodeid, float force); 膝关节受力
    - void **mc\_getFootPart** (IntPtr pMC, int nodeid, int part); 脚底最先触地部位
    - void **mc\_getEvaluation**(IntPtr pMC, string eva); 评估跑步情况

**步态识别算法概览**

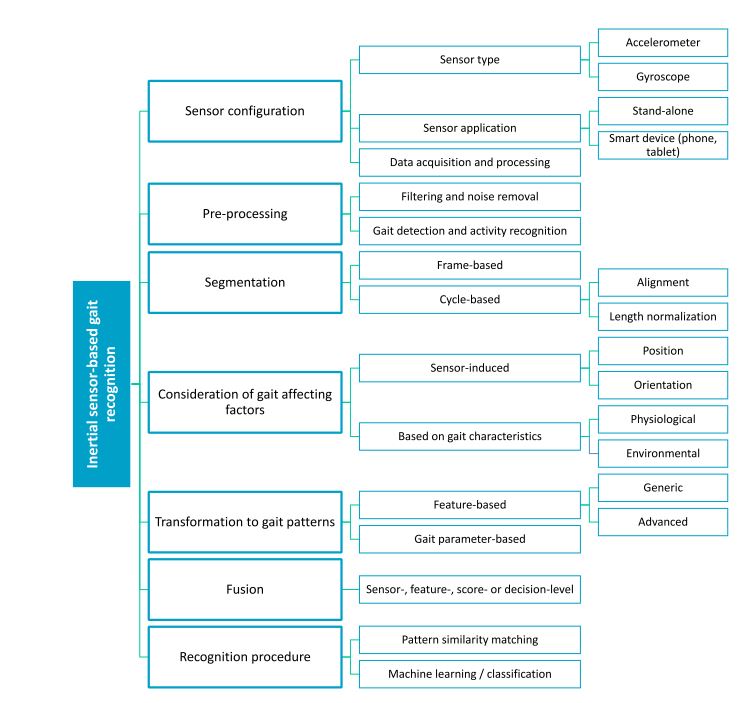
基于惯性传感器的步态识别方法的基本概述如图所示。算法的处理按照以下原则运行：

（a）基于姿态传感器设置，在用户步态中获取惯性数据;

（b）在预处理和分割步骤之后，获取的惯性数据被转换为步态模式;

（c）通过适当的识别程序将进入的步态模式与已标记步态进行比较。

在以上步骤中，一些方法可能需要利用Kalman等融合算法和模式识别。

****

实验

