基于IMU的计步算法分析与比较

计步检测系统

2026 年 1 月 2 日

1 概述

本文档详细介绍三种基于惯性测量单元(IMU)加速度数据的计步算法：

1. **峰值检测法** (Peak Detection)
2. **过零检测法** (Zero-Crossing Detection)
3. **自相关函数法** (Autocorrelation)

三种方法共享相同的预处理流程，仅在最终计步算法上有所不同。

2 通用预处理流程

所有三种方法均采用以下预处理步骤：

2.1 步骤1：数据加载

从IMU传感器读取三轴加速度数据 (a_x, a_y, a_z) 及对应时间戳。

2.2 步骤2：计算合成加速度

将三轴加速度合成为标量模值，消除设备方向的影响：

$$a_{mag} = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{1}$$

2.3 步骤3：带通滤波

应用4阶巴特沃斯带通滤波器，保留步态相关频率成分：

- 低截止频率：$f_L = 0.5$ Hz（排除缓慢漂移）
- 高截止频率：$f_H = 5.0$ Hz（排除高频噪声）

- 正常步频范围：$0.5 \sim 3.0$ Hz（约每秒0.5到3步）

滤波器传递函数：

$$H(s) = \frac{\omega_H^n \cdot s^n}{(s^2 + \frac{\omega_L}{Q}s + \omega_L^2)^{n/2} \cdot (s^2 + \frac{\omega_H}{Q}s + \omega_H^2)^{n/2}} \tag{2}$$

使用 `scipy.signal.filtfilt` 进行零相位滤波，避免相位延迟。

3 方法一：峰值检测法 (Peak Detection)

3.1 算法原理

每一步行走会在加速度信号中产生一个明显的峰值。通过检测信号中的局部最大值来计数步数。

3.2 算法步骤

1. **计算自适应阈值：**

$$\text{threshold} = \mu + 0.5\sigma \tag{3}$$

其中 μ 为信号均值，σ 为标准差。

2. **设置最小峰值间距：**

$$d_{min} = \frac{f_s}{f_{max}} = \frac{100}{3} \approx 33 \text{ 样本} \tag{4}$$

其中 f_s 为采样率，f_{max} 为最大步频。

3. **计算峰值突出度：**

$$\text{prominence} = 0.3\sigma \tag{5}$$

4. **峰值检测：**使用 `scipy.signal.find_peaks` 函数，满足：
   - 幅值 $>$ 阈值
   - 与相邻峰值距离 $\geq d_{min}$
   - 突出度 $\geq$ prominence
5. **计数：**峰值数量即为步数

3.3 数学表达

对于信号 $x[n]$，若满足以下条件则 n 为峰值点：

$$x[n] > x[n-1] \wedge x[n] \geq x[n+1] \wedge x[n] > \text{threshold} \tag{6}$$

4 方法二：过零检测法 (Zero-Crossing Detection)

4.1 算法原理

步态信号具有周期性，去除直流分量后，信号会周期性地穿过零点。通过检测从负到正的过零次数来估计步数。

4.2 算法步骤

1. **去除直流分量：**

$$x'[n] = x[n] - \bar{x} \tag{7}$$

其中 $\bar{x} = \frac{1}{N}\sum_{i=0}^{N-1} x[i]$

2. **设置最小过零间隔：**

$$T_{min} = 0.3 \times f_s = 30 \text{ 样本} \tag{8}$$

3. **检测上升沿过零点：**

$$\text{crossing}[n] = \begin{cases} 1 & \text{if } x'[n-1] < 0 \wedge x'[n] \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

4. **应用时间约束：**相邻过零点间隔需 $\geq T_{min}$

5. **计数：**有效过零点数量即为步数

4.3 数学表达

过零检测器的输出：

$$ZC = \sum_{n=1}^{N-1} \mathbf{1}[x'[n-1] \cdot x'[n] < 0 \wedge x'[n] > x'[n-1]] \tag{10}$$

5 方法三：自相关函数法 (Autocorrelation)

5.1 算法原理

步态信号具有周期性，其自相关函数在周期对应的时延处会出现峰值。通过分析自相关函数估计步态周期，进而计算步数。

5.2 算法步骤

1. **去除直流分量：**

$$x'[n] = x[n] - \bar{x} \tag{11}$$

2. **计算自相关函数：**

$$R_{xx}[\tau] = \sum_{n=0}^{N-1-\tau} x'[n] \cdot x'[n+\tau] \tag{12}$$

3. **归一化：**

$$R'_{xx}[\tau] = \frac{R_{xx}[\tau]}{R_{xx}[0]} \tag{13}$$

4. **确定搜索范围：**

$$\tau_{min} = \frac{f_s}{f_{max}} = \frac{100}{3.0} \approx 33 \text{ 样本} \tag{14}$$

$$\tau_{max} = \frac{f_s}{f_{min}} = \frac{100}{0.5} = 200 \text{ 样本} \tag{15}$$

5. **找到第一个主峰：**在 $[\tau_{min}, \tau_{max}]$ 范围内找第一个峰值，对应步态周期 T_s

6. **计算步频：**

$$f_{step} = \frac{f_s}{T_s} \tag{16}$$

7. **估计步数：**

$$N_{steps} = \lfloor D \times f_{step} \rfloor \tag{17}$$

其中 D 为信号总时长（秒）

5.3 数学表达

自相关函数反映信号与其延迟版本的相似度：

$$R_{xx}[\tau] = E[x[n] \cdot x[n+\tau]] = \int_{-\infty}^{\infty} x(t)x(t+\tau)dt \tag{18}$$

6 方法比较

6.1 优缺点对比

6.2 适用场景

- **峰值检测法：**适合室内规律行走、需要精确定位每步时刻的应用
- **过零检测法：**适合对实时性要求高、计算资源受限的嵌入式设备
- **自相关函数法：**适合离线分析、步态周期估计、稳态长时间行走

表 1: 三种计步方法优缺点比较

方法	优点	缺点
峰值检测法	• 直观易理解 • 计算效率高 • 适合规律行走 • 可精确定位每一步	• 对阈值敏感 • 易受噪声干扰 • 不规则步态误检多
过零检测法	• 不依赖幅值阈值 • 对信号幅值变化鲁棒 • 实现简单 • 计算开销小	• 对噪声敏感（产生虚假过零） • 需要良好的滤波 • 难以处理非周期信号
自相关函数法	• 抗噪声能力强 • 可估计步频 • 适合稳态行走分析 • 对局部异常不敏感	• 计算复杂度较高 $O(N^2)$ • 需要较长数据段 • 步频变化时精度下降 • 无法精确定位每一步

6.3 计算复杂度

表 2: 计算复杂度比较

方法	时间复杂度	空间复杂度
峰值检测法	$O(N)$	$O(1)$
过零检测法	$O(N)$	$O(1)$
自相关函数法	$O(N^2)$ 或 $O(N \log N)$*	$O(N)$

*使用FFT加速

7 总结

三种方法各有特点：

1. **峰值检测法**：最常用，平衡了准确性和计算效率
2. **过零检测法**：最简单，适合资源受限场景

3. **自相关函数法**：最鲁棒，适合需要步频分析的场景

实际应用中，可根据硬件资源、实时性要求和精度需求选择合适的方法，或结合多种方法进行融合以提高鲁棒性。