# BVH 3D Viewer 项目文档集

## 一、代码核心功能总结

该代码是一个基于 **Pygame+OpenGL** 开发的 BVH（运动捕捉数据格式）3D 可视化与分析工具，核心功能覆盖 “数据解析 - 3D 渲染 - 运动学计算 - 交互操作 - 数据导出” 全流程，具体包括：

1. **BVH 文件解析**：读取 BVH 文件的层级结构（HIERARCHY）和运动数据（MOTION），构建关节树（Joint 类）；
2. **3D 骨骼渲染**：按自定义关节顺序渲染骨骼（关节为黑色球体，连接为黑色线条），支持网格、坐标轴绘制；
3. **运动学数据计算**：
   * 基础数据：关节世界坐标（位置）、速度（帧间位置差 / 帧时间）、加速度（帧间速度差 / 帧时间）；
   * 解剖学角度：肩（外展 / 内收、屈曲 / 伸展）、肘（屈曲）、髋（屈曲 / 伸展、外展 / 内收）、膝（屈曲）角度；
   * 关节活动度（ROM）：统计各解剖学角度的动态范围（最小值 / 最大值）；
4. **交互操作**：
   * 鼠标：左键平移、中键旋转、滚轮缩放、右键恢复视图、拖动时间轴切换帧；
   * 键盘：空格切换播放 / 暂停、左右箭头切换帧；
   * UI 按钮：加载 BVH 文件、导出 CSV 数据、设置关节轨迹（多选关节 + 显示开关）；
5. **轨迹可视化**：播放时显示选中关节的运动轨迹（绿色小点，随播放进度累积绘制）；
6. **数据导出**：将每帧的关节位置、速度、加速度、解剖学角度导出为 CSV 文件（单位：m、m/s、m/s²）。

## 二、需求文档（RD）

### 1. 项目背景

BVH（Biovision Hierarchy）是运动捕捉领域的标准数据格式，包含关节层级结构和帧运动数据。当前缺乏轻量、直观的工具用于 BVH 数据的可视化验证与运动学分析，因此开发本工具以满足 “可视化查看 - 数据计算 - 结果导出” 的核心需求，适用于运动分析人员、动画开发人员等用户。

### 2. 功能需求（FR）

#### 2.1 文件操作需求

|  |  |  |
| --- | --- | --- |
| 需求 ID | 需求描述 | 实现方式 |
| FR-001 | 加载 BVH 文件 | 点击 “Load File” 按钮，通过文件对话框选择 .bvh 文件，解析层级与运动数据 |
| FR-002 | 导出运动学数据 | 点击 “Export Data” 按钮，通过文件对话框选择路径，导出 CSV 格式数据（含帧号、关节位置 / 速度 / 加速度、解剖学角度） |
| FR-003 | 轨迹设置 | 点击 “Trajectory” 按钮，弹出设置窗口：1. 总开关：控制是否显示轨迹；2. 关节多选：按自定义顺序列出关节，支持多选；3. 确认后生成选中关节的轨迹数据 |

#### 2.2 3D 可视化需求

|  |  |  |
| --- | --- | --- |
| 需求 ID | 需求描述 | 实现方式 |
| FR-011 | 骨骼渲染 | 按自定义关节顺序（CUSTOM\_JOINT\_ORDER）渲染：- 关节：黑色球体（半径 2.5*0.4）；- 关节连接：黑色线条（线宽 2.0）；- 末端位点（End Site）：黑色小球体（半径 2.5*0.3）+ 连接线 |
| FR-012 | 辅助元素渲染 | 1. 网格：XZ 平面灰色网格（间隔 50 单位）；2. 坐标轴：X（红）、Y（绿）、Z（蓝）轴 + 标签，长度 16.67 单位；3. 角度标签：肩 / 肘 / 髋 / 膝关节处显示角度弧 + 数值（保留 1 位小数） |
| FR-013 | 轨迹可视化 | 播放时显示选中关节的轨迹：- 样式：绿色小点（点大小 1.0）；- 逻辑：仅绘制当前帧及之前的轨迹点（随播放进度累积） |

#### 2.3 运动控制需求

|  |  |  |
| --- | --- | --- |
| 需求 ID | 需求描述 | 实现方式 |
| FR-021 | 播放 / 暂停 | 1. 点击播放按钮（三角形 / 双竖线图标）；2. 按空格键；3. 播放时按帧时间自动切换帧（目标帧率 = 1 / 帧时间，最高 60FPS） |
| FR-022 | 帧切换 | 1. 拖动时间轴滑块（显示当前帧 / 总帧数）；2. 按左 / 右箭头键（逐帧切换，边界限制：0~ 总帧数 - 1） |
| FR-023 | 视图控制 | 1. 左键拖动：平移视图（X/Y 方向）；2. 中键拖动：旋转视图（俯仰 / 偏航）；3. 滚轮：缩放视图（靠近 / 远离模型）；4. 右键：恢复默认视图（视角：45° 透视，位置：(0,-100,-300)） |

#### 2.4 数据计算需求

|  |  |  |
| --- | --- | --- |
| 需求 ID | 需求描述 | 实现方式 |
| FR-031 | 基础运动学数据 | 1. 位置：通过关节矩阵（matrix [:3,3]）获取世界坐标；2. 速度：帧间位置差 / 帧时间（前 2 帧速度设为 0）；3. 加速度：帧间速度差 / 帧时间（前 2 帧加速度设为 0） |
| FR-032 | 解剖学角度 | 1. 肩：外展 / 内收（YZ 平面与 Y 轴夹角）、屈曲 / 伸展（XY 平面与 Y 轴夹角）；2. 肘 / 膝：三点夹角（如肘：上臂 - 前臂 - 手）；3. 髋：屈曲 / 伸展（YZ 平面与 Y 轴夹角）、外展 / 内收（XY 平面与 Y 轴夹角） |
| FR-033 | 关节活动度（ROM） | 统计所有帧中各解剖学角度的最小值和最大值 |

#### 2.5 数据展示需求

|  |  |  |
| --- | --- | --- |
| 需求 ID | 需求描述 | 实现方式 |
| FR-041 | 位置面板 | 左侧显示所有关节的实时位置（单位：m，保留 4 位小数），按自定义关节顺序排列，超出屏幕范围截断 |
| FR-042 | 速度面板 | 右侧显示所有关节的实时速度（单位：m/s，保留 4 位小数），按自定义关节顺序排列，超出屏幕范围截断 |
| FR-043 | 状态信息 | 底部显示：1. BVH 数据信息（帧率、总帧数）；2. 软件帧率（FPS）；3. 当前帧号 |

### 3. 非功能性需求（NFR）

|  |  |  |
| --- | --- | --- |
| 需求 ID | 需求类别 | 需求描述 |
| NFR-001 | 性能 | 1. 渲染帧率：稳定达到目标帧率（最高 60FPS）；2. 数据计算：加载 1000 帧 BVH 文件时，计算时间≤5 秒 |
| NFR-002 | 兼容性 | 1. 支持标准 BVH 格式（含 HIERARCHY、MOTION 模块，支持位置 / 旋转通道）；2. 兼容 Windows/macOS 系统（依赖 Pygame/OpenGL 跨平台特性） |
| NFR-003 | 易用性 | 1. UI 布局清晰：按钮（加载 / 导出 / 轨迹）位于顶部，播放按钮 / 时间轴位于底部，面板位于左右侧；2. 操作反馈：加载失败 / 无文件时弹出提示框，导出成功 / 失败打印日志 |
| NFR-004 | 容错性 | 1. 文件读取失败时（如路径错误、格式非法），打印错误信息且不崩溃；2. 轨迹设置时无加载文件，弹出 “请先加载 BVH 文件” 提示 |

### 4. 用户场景（US）

|  |  |  |
| --- | --- | --- |
| 场景 ID | 用户角色 | 场景描述 |
| US-001 | 运动分析人员 | 1. 加载运动员跑步的 BVH 文件；2. 播放并查看髋关节屈曲角度的动态变化；3. 选中 “RightFoot” 关节，显示其运动轨迹；4. 导出所有帧的关节位置和角度数据，用于后续 biomechanics 分析 |
| US-002 | 动画开发人员 | 1. 加载角色走路的 BVH 文件；2. 拖动时间轴逐帧检查骨骼姿态是否正确（如手指末端位点是否对齐）；3. 旋转视图从侧面观察膝关节屈曲是否自然；4. 确认无误后导出数据用于动画引擎适配 |

### 5. 输入输出（IO）

* **输入**：标准 BVH 文件（.bvh），需包含：
  1. HIERARCHY 模块：ROOT/JOINT 定义、偏移量（OFFSET）、通道（CHANNELS，支持位置 / X/Y/Z 旋转）、末端位点（End Site）；
  2. MOTION 模块：帧数（Frames）、帧时间（Frame Time）、帧运动数据（每行对应一帧的所有通道值）；
* **输出**：
  1. 可视化输出：3D 骨骼、轨迹、辅助元素、UI 面板；
  2. 文件输出：CSV 格式数据文件（含帧号、关节位置 / 速度 / 加速度、解剖学角度）。

## 三、目标文档（PD）

### 1. 核心目标

1. 提供 **直观的 BVH 数据可视化能力**：支持 3D 骨骼渲染、视图交互，帮助用户快速验证 BVH 数据的正确性；
2. 提供 **完整的运动学分析能力**：自动计算位置、速度、加速度、解剖学角度、关节活动度，满足运动分析的基础需求；
3. 提供 **便捷的数据导出能力**：将分析结果导出为 CSV 格式，支持后续第三方工具（如 Excel、MATLAB）进一步处理。

### 2. 次要目标

1. 优化用户交互体验：设计清晰的 UI 布局、丰富的操作方式（鼠标 / 键盘 / 按钮），降低使用门槛；
2. 提升可视化效果：支持轨迹显示、角度标签，帮助用户聚焦关键运动细节（如关节轨迹、角度变化）；
3. 保证工具稳定性：处理文件读取、数据计算中的异常场景，避免崩溃，提供明确的错误提示。

### 3. 技术目标

1. **渲染性能**：在主流配置（CPU i5 + 集成显卡）下，加载 1000 帧、50 个关节的 BVH 文件时，渲染帧率稳定≥30FPS；
2. **数据准确性**：
   * 位置计算误差≤0.001m（与 BVH 原始数据对比）；
   * 角度计算误差≤0.1°（与标准运动学工具对比）；
3. **兼容性**：支持 90% 以上的标准 BVH 文件（如 Vicon、OptiTrack 设备输出的 BVH 文件）；
4. **可维护性**：代码模块化设计（如 Joint 类封装关节属性、独立函数封装核心逻辑），便于后续功能扩展（如添加力计算、骨骼皮肤渲染）。

## 四、代码结构解析

### 1. 核心类

|  |  |  |  |
| --- | --- | --- | --- |
| 类名 | 核心属性 | 核心方法 | 功能描述 |
| Joint | name（关节名）、parent/children（父子关节）、offset（偏移量）、channels（通道）、matrix（4x4 变换矩阵）、position/velocity/acceleration（运动学数据）、rom（关节活动度） | add\_child（添加子关节）、set\_offset（设置偏移）、set\_channels（设置通道）、set\_end\_site（设置末端位点） | 封装 BVH 关节的层级关系、几何属性、运动学数据，是骨骼树的基础单元 |

### 2. 核心函数分类

#### 2.1 文件解析类

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 输入 | 输出 | 功能 |
| parse\_bvh | file\_path（BVH 文件路径） | root\_joint（根关节）、joints（关节字典）、motion\_data（运动数据数组）、frames（帧数）、frame\_time（帧时间） | 读取 BVH 文件，解析层级结构（构建 Joint 树）和运动数据（转换为 numpy 数组） |

#### 2.2 矩阵与运动学计算类

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 输入 | 输出 | 功能 |
| update\_joint\_matrices | joint（当前关节）、frame\_data（单帧运动数据）、all\_joints（关节字典） | 无（更新关节 matrix） | 递归计算关节的变换矩阵：根关节含位置偏移，子关节基于父关节矩阵 + 自身偏移 + 旋转通道 |
| get\_world\_position | joint（关节） | 3D 数组（世界坐标） | 从关节 matrix 中提取世界位置（matrix [:3,3]） |
| calculate\_anatomical\_angles | joints（关节字典） | 字典（角度名：角度值） | 计算肩、肘、髋、膝的解剖学角度（基于三点向量夹角、轴向量投影） |
| calculate\_kinematics | joints（关节字典）、all\_frames\_data（所有帧运动数据）、frame\_time（帧时间） | positions/velocities/accelerations/anatomical\_angles（每帧数据列表） | 批量计算所有帧的关节位置、速度、加速度、解剖学角度 |
| calculate\_rom | all\_anatomical\_angles（所有帧角度数据） | 字典（角度名: [最小值，最大值]） | 统计各解剖学角度的动态范围（关节活动度） |

#### 2.3 渲染类

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 输入 | 输出 | 功能 |
| draw\_skeleton\_custom\_order | joints（关节字典） | 无（OpenGL 渲染） | 按自定义顺序渲染骨骼（关节球体、连接线条、末端位点） |
| draw\_grid | 无 | 无（OpenGL 渲染） | 渲染 XZ 平面灰色网格（间隔 50 单位） |
| draw\_axes\_and\_labels | 无 | 无（OpenGL 渲染） | 渲染 X/Y/Z 坐标轴 + 标签（红 / 绿 / 蓝） |
| draw\_joint\_trajectories | show\_trajectories（轨迹开关）、selected\_joints（选中关节）、joint\_trajectories（轨迹数据）、current\_frame（当前帧） | 无（OpenGL 渲染） | 播放时渲染选中关节的轨迹（绿色小点，累积到当前帧） |
| draw\_position\_panel/draw\_velocity\_panel | display（窗口尺寸）、current\_positions/current\_velocities（当前帧数据）、joints（关节字典） | 无（OpenGL 2D 渲染） | 渲染左右侧的位置 / 速度面板（单位转换：cm→m，保留 4 位小数） |
| draw\_2d\_ui | display、current\_frame、frames、is\_playing、fps 等 | 无（OpenGL 2D 渲染） | 渲染 UI 元素：按钮（加载 / 导出 / 轨迹）、播放按钮、时间轴、状态信息 |

#### 2.4 UI 交互类

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 输入 | 输出 | 功能 |
| open\_trajectory\_settings | joints（关节字典）、all\_joint\_positions（所有帧位置）、轨迹相关变量 | 更新后的轨迹变量（show\_trajectories/selected\_joints 等） | 弹出 Tkinter 窗口，支持轨迹总开关、关节多选，生成选中关节的轨迹数据 |
| export\_data\_dialog | 所有运动学数据（positions/velocities 等） | 无（生成 CSV 文件） | 弹出文件对话框，按自定义关节顺序导出 CSV 数据（单位转换：cm→m） |
| unproject | winX/winY/winZ（窗口坐标） | obj\_point（3D 世界坐标） | 将窗口 2D 坐标反投影为 3D 世界坐标（用于交互计算） |

### 3. 主函数（main）流程

1. **初始化**：
   * 初始化 Pygame/GLUT/OpenGL（窗口尺寸 1280x960，双缓冲 + 深度测试）；
   * 定义 UI 元素（按钮矩形、轨迹相关变量、运动学数据列表）；
   * 调用 reset\_view () 设置默认视图；
2. **事件循环**：
   * 处理窗口关闭、鼠标事件（点击 / 拖动 / 滚轮）、键盘事件（空格 / 箭头键）；
   * 点击按钮触发对应逻辑（加载文件、导出数据、轨迹设置、播放暂停）；
3. **渲染循环**：
   * 清空颜色 / 深度缓冲区；
   * 渲染辅助元素（网格、坐标轴）；
   * 若加载 BVH 文件，更新关节矩阵，渲染骨骼、角度标签、轨迹（播放时）；
   * 渲染 2D UI（按钮、时间轴、状态信息）和位置 / 速度面板；
   * 翻转缓冲区，按目标帧率（1 / 帧时间或 60FPS）控制循环速度。

## 五、使用说明

### 1. 环境准备

1. 安装依赖库：

|  |
| --- |
| pip install pygame pyopengl pyopengl-accelerate numpy tkinter |

1. 运行代码：直接执行脚本（Python 3.7+ 环境）。

### 2. 操作步骤

#### 2.1 加载 BVH 文件

1. 点击顶部左侧 “Load File” 按钮；
2. 在文件对话框中选择目标 .bvh 文件；
3. 加载成功后，窗口底部会显示 BVH 数据信息（帧率、总帧数），同时 3D 视图中渲染骨骼。

#### 2.2 控制视图与运动

|  |  |
| --- | --- |
| 操作 | 效果 |
| 左键拖动 | 平移视图（左右 / 上下） |
| 中键拖动 | 旋转视图（绕 X/Y 轴） |
| 滚轮上滚 | 放大视图（靠近骨骼） |
| 滚轮下滚 | 缩小视图（远离骨骼） |
| 右键点击 | 恢复默认视图 |
| 空格键 | 切换播放 / 暂停 |
| 左箭头键 | 切换到上一帧 |
| 右箭头键 | 切换到下一帧 |
| 拖动时间轴滑块 | 快速切换到指定帧 |

#### 2.3 设置关节轨迹

1. 点击顶部 “Trajectory” 按钮；
2. 在弹出窗口中：
   * 勾选 “显示关节轨迹”（总开关）；
   * 在列表中按住 Ctrl 键多选目标关节（按自定义顺序排列）；
3. 点击 “确认”，播放时选中关节会显示绿色轨迹点（随播放进度累积）。

#### 2.4 导出运动学数据

1. 确保已加载 BVH 文件；
2. 点击顶部 “Export Data” 按钮；
3. 在文件对话框中选择保存路径和文件名（默认后缀 .csv）；
4. 导出成功后，日志会打印 “数据成功导出到 [路径]”，CSV 文件包含每帧的关节位置、速度、加速度、解剖学角度。

### 3. 注意事项

1. 若加载 BVH 文件失败，检查文件是否为标准格式（含 HIERARCHY 和 MOTION 模块）；
2. 轨迹仅在 “播放状态” 下显示，暂停时不更新轨迹；
3. 位置 / 速度面板仅显示当前帧数据，超出屏幕范围的关节会被截断；
4. 导出的 CSV 数据已将单位从 cm 转换为 m（符合国际单位制）。

（注：文档部分内容可能由 AI 生成）