

Spike Camera

课程项目文件结构

- 代码包
- `result.avi`结果视频文件
- `Imaging with Spike Camera.pdf`项目报告
- `Source Code of Project Report.tex`报告的Latex源代码

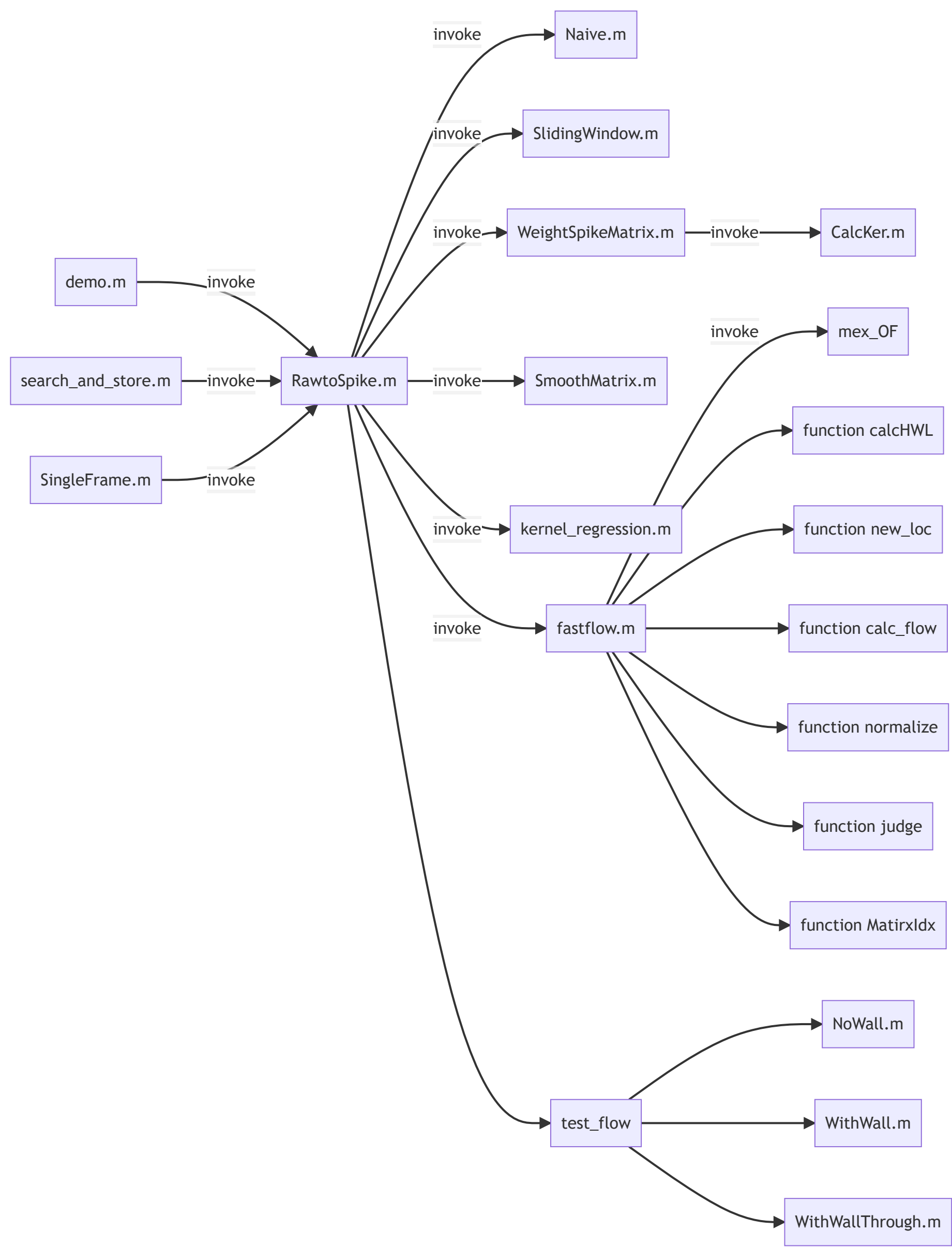
代码包结构

- `result`
 - 生成出来的 `.avi` 视频文件
- `demo.m`
 - 调用 `RawtoSpike` 生成视频，播放并保存
- `search_store.m`
 - 对全部数据，调用 `RawtoSpike` 使用不同方法批量生成视频并保存
- `RawtoSpike.m`
 - 调用某种方法（可选），将raw数据处理成 `uint8` 类型的矩阵序列
- `SingleFrame.m`
 - 只保存单帧
- `Naive.m`
 - 使用Naive方法
- `SlidingWindow.m`
 - 使用滑动窗口方法
- `WeightSpikeMatrix.m`
 - 使用带时间加权的滑动窗口方法
- `kernel_regression.m`
 - 使用3-D kernel regression方法
- `fastflow.m`
 - 使用flow方法估计运动速度
 - `function calcHWL`
 - `function new_loc`
 - `function calc_flow`
 - `function normalize`
 - `function judge`
 - `function MatrixIdx`
- `mex_OF.m`
 - 用于计算flow的函数。供 `fastflow.m` 调用
- `CalcKer.m`
 - 根据半窗长 `half_window_length`，计算长为 `2*half_window_length+1` 的 `kernel` 向量（高斯核）。供 `WeightSpikeMatrix.m` 调用
- `SmoothMatrix.m`
 - 对每一单帧作降噪处理
- `testflow`

为了解决 `optical flow` 算法中遇到的遮挡问题，我们创建了一些简单的数据来观察 `optical flow` 在给定场景下的行为

- `NoWall.m`
一个黑色背景下缓慢移动的白色方块
- `WithWall.m`
黑色背景下缓慢移动，并被灰墙遮挡的白色方块
- `WithWallThrough.m`
黑色背景下缓慢移动，并被灰墙遮挡的贯通白色方块

各函数的相互调用



说明

1. 为减少舍入带来的误差，计算过程中统一采用了 `double` 数据类型，只在处理输入和输出时使用 `uint8`。
2. 默认 `demo.m` 会播放产生的结果，`search_and_store.m` 不会。
3. 默认每次运行 `demo.m`、`search_and_store.m` 都会保存结果。如果没有修改保存路径，会导致之前的文件被覆盖。
4. 切换生成图像的方法时，主要应修改 `RawtoSpike.m` 文件。若使用 `search_and_store` 保存文件，还应在 `search_store` 中修改字符串 `approach`。

小组分工

成员：傅增、王立楹、邢雨菡、詹松昆

代码：傅增（`fastflow`算法部分、`kernel_regression`算法部分）、王立楹（其余部分）

代码优化：王立楹

中期汇报：王立楹、傅增

结题汇报：邢雨菡、詹松昆

代码调试及生成结果：傅增、王立楹、邢雨菡、詹松昆（每人各负责一个视频）

报告大纲：詹松昆

最终报告写作：邢雨菡

整理汇总：詹松昆