# SmartScope 测试用例模板和示例

## 1. 测试用例模板

### 1.1 标准测试用例模板

测试用例ID: TC\_[模块]\_[类型]\_[序号]

测试用例名称: [简洁描述测试目标]

测试类型: [单元测试/集成测试/性能测试/专项测试]

优先级: [P0/P1/P2/P3]

测试目标: [详细描述要验证的功能或特性]

前置条件:

* [列出执行测试前必须满足的条件]
* [包括环境、数据、状态等]

测试步骤:

1. [详细的操作步骤]
2. [每个步骤要具体、可执行]
3. [包括输入数据和操作方法]

预期结果:

* [明确的预期输出或行为]
* [可量化的验收标准]

测试数据:

* [输入数据的具体内容]
* [测试文件路径或数据源]

环境要求:

* [硬件配置要求]
* [软件版本要求]
* [网络环境要求]

执行时间: [预估执行时间]

自动化程度: [手动/半自动/全自动]

关联需求: [相关需求ID]

设计者: [测试用例设计人员]

审核者: [测试用例审核人员]

创建日期: [YYYY-MM-DD]

修改日期: [YYYY-MM-DD]

备注:

* [特殊说明或注意事项]
* [已知问题或限制]

### 1.2 性能测试用例模板

性能测试用例ID: PT\_[模块]\_[指标]\_[序号]

测试用例名称: [性能指标描述]

测试类型: 性能测试

优先级: [P0/P1/P2/P3]

性能指标: [响应时间/吞吐量/资源使用率/并发数]

测试目标:

* [具体的性能目标]
* [基准值和目标值]

测试环境:

* 硬件配置: [CPU/内存/存储/网络]
* 软件环境: [操作系统/依赖库版本]
* 负载条件: [并发用户数/数据量/网络条件]

测试场景:

1. [正常负载场景]
2. [峰值负载场景]
3. [压力测试场景]

性能基准:

* 基准值: [当前性能水平]
* 目标值: [期望达到的性能]
* 阈值: [可接受的最低性能]

测试工具:

* [性能测试工具名称]
* [监控工具配置]

测试数据:

* [测试数据集大小]
* [数据特征描述]

执行步骤:

1. [环境准备]
2. [基准测试]
3. [负载测试]
4. [结果收集]

验收标准:

* [具体的性能指标要求]
* [可接受的性能范围]

风险评估:

* [可能的性能瓶颈]
* [测试风险和缓解措施]

## 2. 具体测试用例示例

### 2.1 单元测试用例示例

#### TC\_CAM\_UNIT\_001: 相机设备发现测试

测试用例ID: TC\_CAM\_UNIT\_001

测试用例名称: USB相机设备自动发现功能测试

测试类型: 单元测试

优先级: P0

测试目标: 验证系统能够正确发现和识别连接的USB相机设备

前置条件:

* 系统已安装相机驱动
* 至少连接一个USB相机设备
* 相机设备处于可用状态
* 测试程序已编译完成

测试步骤:

1. 启动相机管理器测试程序
2. 调用相机发现接口 CameraManager::discoverCameras()
3. 检查返回的相机设备列表
4. 验证设备信息的完整性
5. 测试设备连接和断开的动态发现

预期结果:

* 成功发现所有连接的USB相机
* 设备信息包含：设备ID、名称、分辨率、帧率
* 设备列表更新及时（< 2秒）
* 无内存泄漏或异常

测试数据:

* USB相机设备: /dev/video0, /dev/video1
* 预期设备名称: ["cameraL", "cameraR"]

环境要求:

* Ubuntu 20.04 LTS
* USB 3.0 接口
* 双目相机设备

执行时间: 5分钟

自动化程度: 全自动

关联需求: REQ\_CAM\_001

设计者: 张三

审核者: 李四

创建日期: 2024-08-06

修改日期: 2024-08-06

备注:

* 测试不同品牌的相机设备
* 验证热插拔功能

#### TC\_STE\_UNIT\_002: SGBM立体匹配算法测试

测试用例ID: TC\_STE\_UNIT\_002

测试用例名称: SGBM立体匹配算法精度测试

测试类型: 单元测试

优先级: P0

测试目标: 验证SGBM算法的立体匹配精度和性能

前置条件:

* 立体匹配模块已编译
* 标准测试数据集已准备
* 相机标定参数已加载
* 算法参数已配置

测试步骤:

1. 加载标准立体图像对（Middlebury数据集）
2. 设置SGBM算法参数
3. 执行立体匹配计算
4. 生成视差图
5. 与标准答案对比计算精度
6. 记录处理时间

预期结果:

* 视差计算精度 ≥ 95%（错误率 ≤ 5%）
* 处理时间 ≤ 200ms（1280x720分辨率）
* 视差图质量良好，无明显噪声
* 内存使用稳定

测试数据:

* Middlebury数据集: Teddy, Cones, Tsukuba
* 图像分辨率: 1280x720
* 标准视差图: ground\_truth.png

环境要求:

* RK3588开发板
* OpenCV 4.5+
* 8GB内存

执行时间: 10分钟

自动化程度: 全自动

关联需求: REQ\_STE\_002

设计者: 王五

审核者: 赵六

创建日期: 2024-08-06

修改日期: 2024-08-06

备注:

* 测试不同参数组合
* 对比BM算法性能

### 2.2 集成测试用例示例

#### TC\_CAM\_STE\_INT\_001: 相机采集到立体匹配集成测试

测试用例ID: TC\_CAM\_STE\_INT\_001

测试用例名称: 相机采集到立体匹配完整流程测试

测试类型: 集成测试

优先级: P0

测试目标: 验证从相机图像采集到立体匹配的完整数据流

前置条件:

* 双目相机已连接并校准
* 相机管理器和立体视觉模块已初始化
* 系统配置文件已加载
* 测试场景已准备

测试步骤:

1. 启动相机管理器
2. 连接左右相机
3. 开始同步图像采集
4. 将采集的图像传递给立体匹配模块
5. 执行立体匹配计算
6. 生成视差图和深度图
7. 验证数据流的完整性和时序

预期结果:

* 图像采集帧率 ≥ 30fps
* 相机同步精度 ≤ 1ms
* 立体匹配处理 ≤ 200ms
* 端到端延迟 ≤ 300ms
* 数据流无丢失或错误

测试数据:

* 测试场景: 标准测试物体
* 采集时长: 5分钟
* 图像分辨率: 1280x720

环境要求:

* 双目相机系统
* 标准光照条件
* 稳定的测试环境

执行时间: 15分钟

自动化程度: 半自动

关联需求: REQ\_INT\_001

设计者: 钱七

审核者: 孙八

创建日期: 2024-08-06

修改日期: 2024-08-06

备注:

* 监控系统资源使用
* 记录性能指标

### 2.3 性能测试用例示例

#### PT\_STE\_PERF\_001: 立体匹配实时性能测试

性能测试用例ID: PT\_STE\_PERF\_001

测试用例名称: 立体匹配算法实时处理性能测试

测试类型: 性能测试

优先级: P0

性能指标: 处理时间、吞吐量、资源使用率

测试目标:

* 验证立体匹配算法的实时处理能力
* 测量不同分辨率下的处理性能
* 评估系统资源使用效率

测试环境:

* 硬件配置: RK3588, 8GB RAM, NPU
* 软件环境: Ubuntu 20.04, OpenCV 4.5
* 负载条件: 连续图像流处理

测试场景:

1. 正常负载: 30fps图像流
2. 高负载: 60fps图像流
3. 压力测试: 最大吞吐量测试

性能基准:

* 基准值: 当前SGBM算法200ms
* 目标值: 处理时间 ≤ 100ms
* 阈值: 可接受最大200ms

测试工具:

* 性能监控: htop, nvidia-smi
* 时间测量: 高精度计时器
* 资源监控: 自研监控工具

测试数据:

* 图像分辨率: 640x480, 1280x720, 1920x1080
* 测试图像: 标准立体图像对
* 测试时长: 每个场景10分钟

执行步骤:

1. 环境准备和基准测试
2. 640x480分辨率性能测试
3. 1280x720分辨率性能测试
4. 1920x1080分辨率性能测试
5. 资源使用率监控
6. 结果分析和报告生成

验收标准:

* 1280x720@30fps: 处理时间 ≤ 100ms
* CPU使用率 ≤ 80%
* 内存使用 ≤ 2GB
* NPU利用率 ≥ 70%

风险评估:

* 可能的瓶颈: 内存带宽、NPU调度
* 缓解措施: 算法优化、并行处理

### 2.4 专项测试用例示例

#### ST\_ACC\_MEA\_001: 距离测量精度专项测试

测试用例ID: ST\_ACC\_MEA\_001

测试用例名称: 三维距离测量精度验证测试

测试类型: 专项测试（精度测试）

优先级: P0

测试目标: 验证系统三维距离测量的精度和可靠性

前置条件:

* 系统已完成相机标定
* 标准测量件已准备
* 测量环境光照稳定
* 高精度测量设备已校准

测试步骤:

1. 放置标准测量件（已知尺寸）
2. 使用系统进行三维重建
3. 在点云中选择测量点
4. 执行距离测量
5. 与标准值对比计算误差
6. 重复测量10次计算统计值
7. 测试不同距离和角度

预期结果:

* 测量精度: 误差 ≤ 0.5mm
* 重复性: 标准差 ≤ 0.2mm
* 成功率: ≥ 95%
* 测量范围: 50mm - 500mm

测试数据:

* 标准件: 精密加工的几何体
* 标准尺寸: 50mm, 100mm, 200mm, 500mm
* 测量角度: 0°, 30°, 45°, 60°

环境要求:

* 标准光照条件: 500-1000 lux
* 温度稳定: 20±2°C
* 无振动环境
* 标准背景

执行时间: 2小时

自动化程度: 半自动

关联需求: REQ\_MEA\_001

设计者: 周九

审核者: 吴十

创建日期: 2024-08-06

修改日期: 2024-08-06

备注:

* 使用高精度游标卡尺作为参考
* 记录环境条件变化
* 分析误差来源

## 3. 测试用例管理

### 3.1 测试用例分类标准

按测试类型分类:

* UT: 单元测试 (Unit Test)
* IT: 集成测试 (Integration Test)
* PT: 性能测试 (Performance Test)
* ST: 专项测试 (Specialized Test)

按功能模块分类:

* CAM: 相机模块
* STE: 立体视觉模块
* INF: 推理模块
* MEA: 测量模块
* UI: 用户界面模块
* CFG: 配置管理模块

按优先级分类:

* P0: 核心功能，必须通过
* P1: 重要功能，优先测试
* P2: 一般功能，常规测试
* P3: 边缘功能，时间允许时测试

### 3.2 测试用例编号规则

格式: [测试类型]\_[模块]\_[子类型]\_[序号]

示例:

* TC\_CAM\_UNIT\_001: 相机模块单元测试用例001
* IT\_STE\_INT\_001: 立体视觉模块集成测试用例001
* PT\_SYS\_PERF\_001: 系统性能测试用例001
* ST\_ACC\_MEA\_001: 测量精度专项测试用例001

### 3.3 测试用例维护

版本控制:

* 测试用例纳入版本控制系统
* 每次修改记录变更日志
* 定期审查和更新测试用例

质量保证:

* 测试用例设计评审
* 测试用例执行验证
* 测试覆盖率分析
* 测试效果评估

持续改进:

* 根据缺陷分析补充测试用例
* 优化测试步骤和数据
* 提高自动化程度
* 完善测试工具