# SmartScope 工业双目内窥镜系统 - 测试计划文档

## 1. 引言

### 1.1 文档目的

本测试计划文档旨在为 SmartScope 工业双目内窥镜系统的测试活动提供全面的指导，确保系统质量满足需求规格说明书中的各项要求。本文档面向测试团队、开发团队、项目管理人员和质量保证人员。

### 1.2 项目背景

SmartScope 是一个基于双目立体视觉技术的工业内窥镜检测系统，集成了图像采集、立体视觉处理、深度学习推理和三维测量功能。系统需要在工业环境中稳定可靠地运行，对测试的全面性和严格性要求很高。

### 1.3 测试目标

* 验证系统功能完整性和正确性
* 确保系统性能满足需求指标
* 验证系统在各种环境下的稳定性
* 确保用户体验符合预期
* 验证系统安全性和可靠性

### 1.4 测试范围

* 功能测试：所有功能模块的测试
* 性能测试：响应时间、吞吐量、资源使用
* 兼容性测试：硬件和软件兼容性
* 安全性测试：数据安全和系统安全
* 可用性测试：用户界面和操作体验
* 稳定性测试：长时间运行和压力测试

### 1.5 测试策略

采用分层测试策略：

* 单元测试：开发阶段，开发人员负责
* 集成测试：模块集成阶段，测试团队负责
* 系统测试：系统完成阶段，测试团队负责
* 验收测试：交付前，用户和测试团队共同负责

## 2. 测试环境

### 2.1 硬件环境

* 主机配置：
* 处理器：RK3588 ARM64 8核
* 内存：8GB LPDDR4
* 存储：64GB eMMC + 256GB SSD
* 显示：1920x1080 IPS 显示器
* 相机：双目 USB 3.0 相机（1280x720@30fps）
* 测试设备：
* 标准测试样品（已知尺寸的物体）
* 标定板（棋盘格标定板）
* 测量工具（游标卡尺、千分尺）
* 网络设备（路由器、交换机）

### 2.2 软件环境

* 操作系统：Ubuntu 20.04 LTS ARM64
* 开发环境：
* Qt 5.15.2
* OpenCV 4.5.4
* PCL 1.10.1
* ONNX Runtime 1.12.0
* RKNN Runtime 1.4.0
* 测试工具：
* Google Test (gtest)
* Qt Test Framework
* Valgrind（内存检测）
* Perf（性能分析）
* Stress-ng（压力测试）

### 2.3 网络环境

* 局域网环境：1000Mbps 以太网
* 无网络环境：离线模式测试
* 不稳定网络：模拟网络中断和延迟

## 3. 测试组织

### 3.1 测试团队

* 测试经理：负责测试计划制定和执行监督
* 功能测试工程师：负责功能测试用例设计和执行
* 性能测试工程师：负责性能测试和优化建议
* 自动化测试工程师：负责自动化测试框架和脚本
* 用户体验测试员：负责可用性测试和用户反馈

### 3.2 职责分工

* 开发团队：单元测试、缺陷修复、测试环境支持
* 测试团队：测试用例设计、测试执行、缺陷报告
* 项目管理：测试进度监控、资源协调、风险管理
* 质量保证：测试质量审查、标准制定、流程改进

### 3.3 沟通机制

* 每日站会：测试进度同步
* 周报告：测试状态和问题汇报
* 缺陷评审会：缺陷优先级和修复计划
* 里程碑评审：阶段性测试结果评估

## 4. 测试类型和方法

### 4.1 功能测试

#### 4.1.1 相机功能测试

测试目标：验证相机管理和控制功能

测试内容：

* 相机设备发现和连接
* 相机参数设置和获取
* 双目相机同步采集
* 相机异常处理和恢复
* 热插拔功能

测试方法：

* 黑盒测试：验证功能正确性
* 边界值测试：参数极值测试
* 异常测试：设备断开、参数错误
* 兼容性测试：不同型号相机

验收标准：

* 相机连接成功率：≥ 95%
* 参数设置响应时间：≤ 100ms
* 同步精度：≤ 1ms
* 异常恢复时间：≤ 5s

#### 4.1.2 立体视觉功能测试

测试目标：验证立体匹配和深度计算功能

测试内容：

* 图像预处理效果
* 立体匹配算法准确性
* 深度图生成质量
* 算法参数调节
* 处理性能

测试方法：

* 标准数据集测试
* 实际场景测试
* 算法对比测试
* 参数敏感性测试

验收标准：

* 深度计算精度：≤ 2% 误差
* 处理帧率：≥ 10 FPS
* 算法切换时间：≤ 1s

#### 4.1.3 三维重建功能测试

测试目标：验证点云生成和处理功能

测试内容：

* 点云生成准确性
* 点云滤波效果
* 三维可视化功能
* 点云文件保存和加载

测试方法：

* 几何精度测试
* 视觉效果评估
* 文件格式兼容性测试

验收标准：

* 点云精度：≤ 1mm 误差
* 渲染帧率：≥ 30 FPS
* 文件保存成功率：100%

#### 4.1.4 测量功能测试

测试目标：验证三维测量工具的准确性

测试内容：

* 距离测量精度
* 面积测量精度
* 角度测量精度
* 测量工具易用性

测试方法：

* 标准件测量对比
* 重复性测试
* 不同角度测试
* 用户操作测试

验收标准：

* 距离测量精度：≤ 0.5mm 误差
* 面积测量精度：≤ 2% 误差
* 角度测量精度：≤ 1° 误差
* 操作成功率：≥ 95%

### 4.2 性能测试

#### 4.2.1 响应时间测试

测试目标：验证系统响应时间满足需求

测试内容：

* 界面操作响应时间
* 图像处理响应时间
* 文件操作响应时间
* 网络操作响应时间

测试方法：

* 自动化性能测试脚本
* 多次测量取平均值
* 不同负载条件测试

验收标准：

* 界面响应：≤ 100ms
* 图像处理：≤ 200ms
* 文件操作：≤ 500ms

#### 4.2.2 吞吐量测试

测试目标：验证系统处理能力

测试内容：

* 图像采集吞吐量
* 数据处理吞吐量
* 并发操作能力
* 批量处理能力

测试方法：

* 持续负载测试
* 峰值负载测试
* 并发用户模拟

验收标准：

* 图像采集：≥ 30 FPS
* 立体匹配：≥ 10 FPS
* 点云生成：≥ 5 FPS

#### 4.2.3 资源使用测试

测试目标：验证系统资源使用合理性

测试内容：

* CPU 使用率监控
* 内存使用监控
* 存储空间使用
* 网络带宽使用

测试方法：

* 系统监控工具
* 长时间运行监控
* 不同工作负载测试

验收标准：

* CPU 使用率：≤ 80%
* 内存使用：≤ 4GB
* 存储增长：≤ 100MB/小时

### 4.3 兼容性测试

#### 4.3.1 硬件兼容性测试

测试目标：验证系统在不同硬件上的兼容性

测试内容：

* 不同型号相机兼容性
* 不同分辨率显示器
* 不同存储设备
* 不同网络设备

测试方法：

* 硬件替换测试
* 配置变更测试
* 驱动兼容性测试

#### 4.3.2 软件兼容性测试

测试目标：验证系统在不同软件环境的兼容性

测试内容：

* 不同 Linux 发行版
* 不同库版本
* 不同配置文件格式
* 文件格式兼容性

测试方法：

* 环境迁移测试
* 版本升级测试
* 配置导入导出测试

### 4.4 安全性测试

#### 4.4.1 数据安全测试

测试目标：验证数据安全保护机制

测试内容：

* 敏感数据加密
* 访问权限控制
* 数据备份恢复
* 操作审计日志

测试方法：

* 安全扫描工具
* 权限绕过测试
* 数据泄露测试

#### 4.4.2 系统安全测试

测试目标：验证系统安全防护能力

测试内容：

* 输入验证机制
* 缓冲区溢出防护
* 恶意文件检测
* 网络安全防护

测试方法：

* 渗透测试
* 模糊测试
* 恶意输入测试

### 4.5 可用性测试

#### 4.5.1 用户界面测试

测试目标：验证用户界面的易用性

测试内容：

* 界面布局合理性
* 操作流程直观性
* 错误提示清晰性
* 帮助文档完整性

测试方法：

* 用户体验评估
* 任务完成度测试
* 学习曲线分析

#### 4.5.2 可访问性测试

测试目标：验证系统的可访问性

测试内容：

* 键盘导航支持
* 屏幕阅读器兼容
* 色彩对比度
* 字体大小调节

测试方法：

* 无障碍测试工具
* 特殊用户群体测试

### 4.6 稳定性测试

#### 4.6.1 长时间运行测试

测试目标：验证系统长期运行稳定性

测试内容：

* 24小时连续运行
* 内存泄露检测
* 资源释放验证
* 性能衰减监控

测试方法：

* 自动化稳定性测试
* 内存分析工具
* 性能监控工具

#### 4.6.2 压力测试

测试目标：验证系统在高负载下的表现

测试内容：

* 最大负载测试
* 并发操作测试
* 资源耗尽测试
* 恢复能力测试

测试方法：

* 压力测试工具
* 负载生成器
* 故障注入测试

## 5. 测试用例设计

### 5.1 测试用例分类

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

### 5.2 测试用例模板

测试用例ID：TC\_XXX\_001

测试用例名称：相机连接功能测试

优先级：P0

前置条件：系统已启动，相机已连接

测试步骤：

1. 启动应用程序
2. 点击"连接相机"按钮
3. 选择左相机设备
4. 选择右相机设备
5. 点击"确认连接"

预期结果：

1. 相机连接成功
2. 状态栏显示"相机已连接"
3. 实时图像正常显示

测试数据：标准USB相机设备

备注：测试不同型号相机

### 5.3 核心测试用例

#### 5.3.1 相机功能测试用例

* TC\_CAM\_001：相机设备发现测试
* TC\_CAM\_002：相机连接断开测试
* TC\_CAM\_003：相机参数设置测试
* TC\_CAM\_004：双目同步测试
* TC\_CAM\_005：相机异常恢复测试

#### 5.3.2 立体视觉测试用例

* TC\_STE\_001：图像预处理测试
* TC\_STE\_002：BM算法立体匹配测试
* TC\_STE\_003：SGBM算法立体匹配测试
* TC\_STE\_004：深度图生成测试
* TC\_STE\_005：算法性能对比测试

#### 5.3.3 测量功能测试用例

* TC\_MEA\_001：距离测量精度测试
* TC\_MEA\_002：面积测量精度测试
* TC\_MEA\_003：角度测量精度测试
* TC\_MEA\_004：测量工具交互测试
* TC\_MEA\_005：测量结果保存测试

## 6. 测试执行计划

### 6.1 测试阶段划分

#### 6.1.1 第一阶段：单元测试（2周）

* 时间：开发完成后立即开始
* 负责人：开发团队
* 内容：
* 核心算法单元测试
* 工具类函数测试
* 数据结构测试
* 异常处理测试
* 目标：单元测试覆盖率 ≥ 90%

#### 6.1.2 第二阶段：集成测试（3周）

* 时间：模块集成完成后
* 负责人：测试团队
* 内容：
* 模块间接口测试
* 数据流测试
* 配置管理测试
* 错误传播测试
* 目标：集成测试通过率 ≥ 95%

#### 6.1.3 第三阶段：系统测试（4周）

* 时间：系统集成完成后
* 负责人：测试团队
* 内容：
* 功能完整性测试
* 性能基准测试
* 兼容性测试
* 安全性测试
* 目标：系统测试通过率 ≥ 95%

#### 6.1.4 第四阶段：验收测试（2周）

* 时间：系统测试完成后
* 负责人：用户和测试团队
* 内容：
* 用户场景测试
* 可用性测试
* 最终性能验证
* 文档验证
* 目标：用户验收通过

### 6.2 测试进度安排

| 阶段 | 开始时间 | 结束时间 | 持续时间 | 里程碑 |

|------|----------|----------|----------|--------|

| 单元测试 | Week 1 | Week 2 | 2周 | 单元测试完成 |

| 集成测试 | Week 3 | Week 5 | 3周 | 集成测试完成 |

| 系统测试 | Week 6 | Week 9 | 4周 | 系统测试完成 |

| 验收测试 | Week 10 | Week 11 | 2周 | 验收测试通过 |

| 缺陷修复 | Week 12 | Week 13 | 2周 | 产品发布 |

### 6.3 测试资源分配

#### 6.3.1 人力资源

* 测试经理：1人，全程参与
* 功能测试工程师：2人，主要负责功能测试
* 性能测试工程师：1人，负责性能和稳定性测试
* 自动化测试工程师：1人，负责自动化测试框架

#### 6.3.2 硬件资源

* 测试主机：3台（不同配置）
* 测试相机：6套（不同型号）
* 测试样品：标准件、复杂件各10套
* 网络设备：路由器、交换机各1台

#### 6.3.3 软件资源

* 测试管理工具：TestLink
* 缺陷管理工具：Jira
* 自动化测试框架：自研框架
* 性能测试工具：自研工具

## 7. 缺陷管理

### 7.1 缺陷分级

#### 7.1.1 严重级别

* 致命（Critical）：系统崩溃、数据丢失、安全漏洞
* 严重（Major）：主要功能无法使用、性能严重下降
* 一般（Minor）：次要功能异常、界面问题
* 轻微（Trivial）：文档错误、建议改进

#### 7.1.2 优先级别

* 紧急（Urgent）：立即修复
* 高（High）：当前版本必须修复
* 中（Medium）：下个版本修复
* 低（Low）：有时间再修复

### 7.2 缺陷处理流程

1. 缺陷发现：测试人员发现并记录缺陷
2. 缺陷分析：测试经理分析缺陷级别和优先级
3. 缺陷分配：分配给相应开发人员
4. 缺陷修复：开发人员修复缺陷
5. 缺陷验证：测试人员验证修复效果
6. 缺陷关闭：确认修复后关闭缺陷

### 7.3 缺陷报告模板

缺陷ID：BUG\_XXX\_001

缺陷标题：相机连接失败

发现人：张三

发现时间：2024-08-04 10:30

严重级别：严重

优先级别：高

所属模块：相机管理

测试环境：Ubuntu 20.04 + RK3588

重现步骤：

1. 启动应用程序
2. 连接USB相机
3. 点击"连接相机"按钮

实际结果：连接失败，提示"设备不存在"

预期结果：连接成功，显示实时图像

附件：错误截图、日志文件

## 8. 测试报告

### 8.1 测试报告类型

* 日报告：每日测试进度和问题
* 周报告：周测试总结和计划
* 阶段报告：测试阶段完成报告
* 最终报告：项目测试总结报告

### 8.2 测试指标统计

* 测试用例执行率：已执行用例数 / 计划用例数
* 测试用例通过率：通过用例数 / 已执行用例数
* 缺陷发现率：发现缺陷数 / 测试用例数
* 缺陷修复率：已修复缺陷数 / 发现缺陷数
* 代码覆盖率：测试覆盖代码行数 / 总代码行数

### 8.3 质量评估标准

* 功能质量：功能测试通过率 ≥ 95%
* 性能质量：性能指标达标率 ≥ 90%
* 稳定性质量：稳定性测试通过率 ≥ 95%
* 安全性质量：安全测试通过率 = 100%
* 可用性质量：用户满意度 ≥ 85%

## 9. 风险管理

### 9.1 测试风险识别

* 技术风险：新技术学习成本、技术难点
* 资源风险：人力不足、设备不够
* 进度风险：开发延期、测试时间压缩
* 质量风险：需求变更、质量标准提高

### 9.2 风险应对策略

* 技术风险：提前技术调研、专家咨询
* 资源风险：资源预留、外部支持
* 进度风险：并行测试、优先级调整
* 质量风险：需求冻结、标准明确

### 9.3 应急预案

* 关键缺陷：紧急修复流程、回归测试
* 环境故障：备用环境、快速恢复
* 人员变动：知识传递、快速补充
* 进度延误：加班加点、范围调整

## 10. 总结

### 10.1 测试计划要点

* 全面覆盖功能、性能、兼容性、安全性等各个方面
* 分阶段执行，确保测试质量和进度
* 明确责任分工，建立有效沟通机制
* 建立完善的缺陷管理和风险控制体系

### 10.2 成功标准

* 所有P0级别测试用例100%通过
* 所有P1级别测试用例≥95%通过
* 系统性能指标全部达标
* 用户验收测试通过
* 产品质量满足发布要求

### 10.3 持续改进

* 测试过程中收集经验教训
* 优化测试方法和工具
* 完善测试流程和规范
* 提升团队测试能力