# SmartScope 简化测试方案

## 🎯 测试目标

将原来的100个测试项目简化为20个核心测试项目，每个测试都有具体的操作步骤和验收标准。

## 📋 简化测试项目清单

### 第一阶段：基础功能测试 (8项，预计2周)

#### T001: 相机连接测试

测试目标: 验证双目相机能够正常连接和工作

具体操作步骤:

1. 连接左右两个USB相机到开发板
2. 运行相机测试程序：

cd /home/eddysun/App/Qt/SmartScope

python3 camera\_test\_fixed.py

1. 观察程序输出，确认检测到两个相机设备
2. 检查实时图像显示是否正常
3. 测试相机参数调节功能

验收标准:

* 成功检测到左右相机 ✓
* 图像显示清晰无花屏 ✓
* 帧率达到15fps以上 ✓
* 参数调节响应正常 ✓

预计时间: 30分钟

#### T002: 相机同步测试

测试目标: 验证双目相机时间戳同步精度

具体操作步骤:

1. 启动相机同步测试程序：

./build/test\_camera\_sync --duration 60

1. 程序运行1分钟，记录左右相机时间戳
2. 分析时间戳差异统计数据
3. 检查同步失败次数

验收标准:

* 时间戳差异 ≤ 5ms ✓
* 同步成功率 ≥ 95% ✓
* 无明显丢帧现象 ✓

预计时间: 20分钟

#### T003: 配置文件测试

测试目标: 验证系统配置加载和保存功能

具体操作步骤:

1. 修改config.toml文件中的相机参数
2. 重启应用程序
3. 检查参数是否正确加载
4. 在界面中修改参数并保存
5. 重启后验证参数持久化

验收标准:

* 配置文件正确加载 ✓
* 参数修改能够保存 ✓
* 错误配置有提示信息 ✓

预计时间: 15分钟

#### T004: 立体匹配基础测试

测试目标: 验证立体匹配算法基本功能

具体操作步骤:

1. 准备标准测试图像对（左右图像）
2. 运行立体匹配测试：

cd reference\_code/lightstereo\_inference

./run\_example.sh

1. 检查生成的视差图质量
2. 记录处理时间

验收标准:

* 成功生成视差图 ✓
* 视差图无明显噪声 ✓
* 处理时间 ≤ 500ms ✓

预计时间: 25分钟

#### T005: 深度计算测试

测试目标: 验证从视差图到深度图的转换

具体操作步骤:

1. 使用T004生成的视差图
2. 运行深度计算程序：

python3 test\_depth\_calculation.py

1. 检查深度图的合理性
2. 验证深度值范围

验收标准:

* 深度图生成成功 ✓
* 深度值在合理范围内(50-2000mm) ✓
* 深度图与场景匹配 ✓

预计时间: 20分钟

#### T006: 用户界面基础测试

测试目标: 验证主界面基本功能

具体操作步骤:

1. 启动主程序：

./SmartScopeQt

1. 测试菜单栏各项功能
2. 测试工具栏按钮响应
3. 测试页面切换功能
4. 测试参数面板操作

验收标准:

* 界面正常显示 ✓
* 按钮响应及时 ✓
* 页面切换流畅 ✓
* 无界面卡死现象 ✓

预计时间: 30分钟

#### T007: 文件操作测试

测试目标: 验证图像和数据文件的保存加载

具体操作步骤:

1. 在主界面中采集图像
2. 保存左右图像为JPG格式
3. 保存视差图为PNG格式
4. 重新加载保存的文件
5. 验证文件完整性

验收标准:

* 文件保存成功 ✓
* 文件格式正确 ✓
* 重新加载显示正常 ✓

预计时间: 20分钟

#### T008: 日志系统测试

测试目标: 验证日志记录功能

具体操作步骤:

1. 启动程序并执行各种操作
2. 检查logs/app.log文件内容
3. 故意触发错误操作
4. 验证错误日志记录

验收标准:

* 日志文件正常生成 ✓
* 日志内容完整准确 ✓
* 错误信息记录清晰 ✓

预计时间: 15分钟

### 第二阶段：集成功能测试 (6项，预计1.5周)

#### T009: 端到端图像处理流程测试

测试目标: 验证从相机采集到深度图生成的完整流程

具体操作步骤:

1. 启动完整的图像处理流程
2. 放置测试物体在相机前
3. 开始实时处理：

# 在主程序中点击"开始处理"按钮

# 或运行端到端测试程序

./build/test\_end\_to\_end

1. 观察实时视差图和深度图生成
2. 记录处理帧率和延迟

验收标准:

* 实时处理帧率 ≥ 5fps ✓
* 端到端延迟 ≤ 1秒 ✓
* 深度图质量良好 ✓
* 无明显处理错误 ✓

预计时间: 45分钟

#### T010: 测量功能测试

测试目标: 验证距离测量功能的准确性

具体操作步骤:

1. 准备已知尺寸的标准件（如100mm的标准块）
2. 将标准件放置在相机视野内
3. 生成点云数据
4. 使用测量工具测量距离：

* 在界面中选择测量工具
* 点击标准件的两端
* 记录测量结果

1. 重复测量5次，计算平均值和误差

验收标准:

* 测量误差 ≤ 2mm ✓
* 重复性误差 ≤ 1mm ✓
* 测量操作流畅 ✓

预计时间: 40分钟

#### T011: 多场景适应性测试

测试目标: 验证系统在不同场景下的表现

具体操作步骤:

1. 明亮场景测试:

* 在充足光照下测试
* 记录图像质量和处理效果

1. 昏暗场景测试:

* 降低环境光照
* 调节相机曝光参数
* 测试处理效果

1. 复杂纹理测试:

* 使用有丰富纹理的物体
* 测试立体匹配效果

1. 简单纹理测试:

* 使用纹理较少的物体
* 观察算法表现

验收标准:

* 明亮场景处理正常 ✓
* 昏暗场景可调节适应 ✓
* 复杂纹理匹配良好 ✓
* 简单纹理有基本效果 ✓

预计时间: 60分钟

#### T012: 异常处理测试

测试目标: 验证系统异常情况下的处理能力

具体操作步骤:

1. 相机断开测试:

* 运行过程中拔掉一个相机
* 观察系统反应和恢复能力

1. 内存不足测试:

* 运行内存消耗程序模拟内存不足
* 观察系统降级处理

1. 文件权限测试:

* 修改配置文件为只读
* 测试系统错误处理

1. 网络中断测试:

* 断开网络连接
* 测试离线功能

验收标准:

* 异常情况有明确提示 ✓
* 系统不会崩溃退出 ✓
* 能够自动恢复或降级 ✓

预计时间: 50分钟

#### T013: 性能压力测试

测试目标: 验证系统在高负载下的表现

具体操作步骤:

1. 长时间运行测试:

# 运行2小时连续处理

timeout 7200 ./SmartScopeQt --auto-mode

1. 高频操作测试:

* 快速切换不同功能
* 频繁保存和加载文件
* 连续调节参数

1. 资源监控:

# 监控系统资源使用

htop &

watch -n 1 'free -h && df -h'

验收标准:

* 2小时运行无崩溃 ✓
* CPU使用率 ≤ 85% ✓
* 内存使用稳定无泄漏 ✓
* 响应时间无明显增加 ✓

预计时间: 3小时（可后台运行）

#### T014: 数据一致性测试

测试目标: 验证数据处理的一致性和可重复性

具体操作步骤:

1. 重复处理测试:

* 使用相同的输入图像
* 重复处理10次
* 对比输出结果的一致性

1. 参数影响测试:

* 改变算法参数
* 观察输出变化的合理性

1. 数据完整性测试:

* 保存处理结果
* 重新加载并验证数据完整性

验收标准:

* 相同输入产生一致输出 ✓
* 参数变化影响可预期 ✓
* 数据保存加载无损失 ✓

预计时间: 35分钟

### 第三阶段：验收测试 (6项，预计1周)

#### T015: 精度验证测试

测试目标: 使用标准件验证系统测量精度

具体操作步骤:

1. 准备标准测量件:

* 50mm标准块
* 100mm标准块
* 200mm标准块

1. 精度测量流程:

# 每个标准件测量10次

for i in {1..10}; do

echo "第${i}次测量"

# 在界面中进行测量操作

# 记录测量结果

done

1. 数据分析:

* 计算平均值、标准差
* 分析误差分布
* 生成精度报告

验收标准:

* 50mm测量误差 ≤ 1mm ✓
* 100mm测量误差 ≤ 1.5mm ✓
* 200mm测量误差 ≤ 2mm ✓
* 重复性标准差 ≤ 0.5mm ✓

预计时间: 90分钟

#### T016: 用户操作流程测试

测试目标: 验证完整的用户操作流程

具体操作步骤:

1. 新用户上手测试:

* 模拟新用户首次使用
* 按照用户手册操作
* 记录操作难点和耗时

1. 典型工作流程:

* 系统启动 → 相机连接 → 参数调节
* 图像采集 → 立体处理 → 测量操作
* 结果保存 → 报告生成 → 系统关闭

1. 效率测试:

* 记录完整流程耗时
* 识别操作瓶颈

验收标准:

* 新用户30分钟内掌握基本操作 ✓
* 完整流程15分钟内完成 ✓
* 操作界面直观易懂 ✓

预计时间: 2小时

#### T017: 兼容性测试

测试目标: 验证系统在不同环境下的兼容性

具体操作步骤:

1. 不同相机测试:

* 测试不同品牌USB相机
* 测试不同分辨率设置

1. 系统环境测试:

* 不同Ubuntu版本（如果可用）
* 不同硬件配置

1. 文件格式测试:

* 导入不同格式图像
* 导出不同格式结果

验收标准:

* 主流USB相机兼容 ✓
* 系统环境适应性好 ✓
* 文件格式支持完整 ✓

预计时间: 2小时

#### T018: 安全性测试

测试目标: 验证系统安全性和稳定性

具体操作步骤:

1. 输入验证测试:

* 输入非法参数值
* 加载损坏的文件
* 测试边界条件

1. 权限测试:

* 测试文件访问权限
* 测试设备访问权限

1. 内存安全测试:

# 使用valgrind检测内存问题

valgrind --tool=memcheck --leak-check=full ./SmartScopeQt

验收标准:

* 非法输入有适当处理 ✓
* 权限控制正确 ✓
* 无内存泄漏或越界 ✓

预计时间: 90分钟

#### T019: 性能基准测试

测试目标: 建立系统性能基准数据

具体操作步骤:

1. 处理速度基准:

# 测试不同分辨率处理速度

echo "640x480分辨率测试"

time ./test\_stereo\_640x480

echo "1280x720分辨率测试"

time ./test\_stereo\_1280x720

1. 资源使用基准:

* 记录CPU使用率峰值和平均值
* 记录内存使用峰值和平均值
* 记录存储空间使用

1. 建立基准数据库:

* 记录所有性能指标
* 建立性能回归检测基准

验收标准:

* 640x480处理 ≤ 100ms ✓
* 1280x720处理 ≤ 300ms ✓
* CPU平均使用率 ≤ 60% ✓
* 内存使用 ≤ 2GB ✓

预计时间: 60分钟

#### T020: 最终验收测试

测试目标: 综合验证系统是否满足交付要求

具体操作步骤:

1. 功能完整性检查:

* 逐一验证所有功能模块
* 确认无关键功能缺失

1. 质量指标验证:

* 汇总前19项测试结果
* 验证是否达到验收标准

1. 用户验收:

* 邀请最终用户试用
* 收集用户反馈意见

1. 交付准备:

* 整理测试报告
* 准备用户文档
* 制作安装包

验收标准:

* 所有P0测试项目100%通过 ✓
* 所有P1测试项目≥95%通过 ✓
* 用户满意度≥85% ✓
* 文档完整准确 ✓

预计时间: 4小时

## 📊 测试执行计划

### 时间安排

* 第一阶段: 2周（基础功能测试）
* 第二阶段: 1.5周（集成功能测试）
* 第三阶段: 1周（验收测试）
* 总计: 4.5周

### 人员安排

* 测试工程师: 2人
* 开发支持: 1人（兼职）
* 用户代表: 1人（验收阶段）

### 测试环境

* 硬件: RK3588开发板 + 双目相机
* 软件: Ubuntu 20.04 + 项目环境
* 工具: 标准测量件、测试脚本

## 🎯 成功标准

### 通过标准

* 关键测试: T001-T008, T015, T020 必须100%通过
* 重要测试: T009-T014, T016-T019 必须≥95%通过
* 整体通过率: ≥95%

### 质量指标

* 功能完整性: 100%
* 性能达标率: ≥90%
* 稳定性: 无崩溃，内存稳定
* 用户满意度: ≥85%