**Android P ITS测试流程**

ITS，全称为Image Test Suite，是CTSVerifier测试中的一项。主要用于验证图像内容的正确性。此项测试从CTS 7.0\_r8测试包开始加入。进行ITS测试需要单独工具和灯箱等辅助设备，所以单独列出来说一下。

Google要求从Android P开始，在非Low RAM的新设备上必须支持CameraHAL3.2或者更高的版本，如果是Low RAM设备可以不支持HAL3(官网强烈建议支持)。

设备如果支持HAL3版本在CTSVerifier测试中就需要按照本文档介绍跑Camera ITS Test项，但如果设置还是HAL1版本，则Camera ITS Test测项可以直接自动pass。

可以使用如下命令检查HAL版本：

adb shell "dumpsys media.camera | grep 'Camera HAL'"

检查Camera HAL device device@后面的数值，1.X即为HAL1版本，3.X即为HAL3版本

# 测试工具

ITS测试工具位于CTSVerifier压缩包的android-cts-verifier\CameraITS目录中，是基于python的测试脚本。

# 测试环境

ITS测试工具支持Mac OS X，Linux和Windows运行，以下为ubuntu环境下安装过程：

（Windows环境下配置较复杂，容易出现缺少支持包问题，不建议使用）

1. 安装Android SDK，配置adb可用
2. 安装Python 2.7和numpy堆栈

sudo apt-get install pythonnumpy

sudo apt-get install pythonscipy

sudo apt-get install pythonmatplotlib

sudo apt-get install pythonopencv

如果这些指令不能正常工作或有其他异常，可以改使用捆绑的Python 2.7发行版Anaconda已通过官方验证，可与ITS基础结构一起使用，

1. 还需要的额外设备包括：灯箱，三脚架，测试图例（灯箱和三脚架都是网上自己买的，测试图例是用测试工具包里面的pdf文档自己打印的）

# 设备端配置

设备端需要安装CTSVerifier.apk,进入Camera ITS Test测试项（如果设备的CameraHAL支持版本为1.x，此项会直接自动pass），然后通过USB数据线连接设备和测试电脑。

灯箱通电打光，将设备放置于三脚架上，再将三脚架放置于灯箱内。

# 测试流程

1.测试电脑端通过shell窗口进入到android-cts-verifier\CameraITS目录中。

2.执行source build/envsetup.sh命令，初始化测试环境。

3.然后执行 python tools/run\_all\_tests.py 即可启动测试。

测试从后摄开始，依次执行各个场景。然后执行前摄测试，顺序一致。测试过程中需要将正在进行测试的设备摄像头（前/后）指向灯箱内，灯箱内背景图需要根据不同场景切换不同的测试图例。

ITS测试一共包含7个场景测试，详情见[附录A](#_附录A)介绍。

在每个场景测试前会有检测环节，会在测试电脑的/tmp/目录生成对应文件夹并且拍摄一张测试图例像，通过查看图片可以确认设备位置是否正确，如果正确，在电脑shell窗口输入Y继续执行测试，如果不正确（不居中，不清晰等）请调整设备相对灯箱位置，在电脑shell窗口输入N或直接回车，会再次执行检测环节，直到调整到合适位置，在输入Y继续执行测试。

4.在完整执行完后摄/前摄各个场景测试后，可以针对fail项来进行单项测试，单项测试以场景为单位，就是说场景1其中有一条fail，整个场景1就需要重新测试。

测试命令如下：

python tools/run\_all\_tests.py camera=1 scenes=1

其中，camera取值包括 0：后摄，1：前摄

Scenes取值包括 0，1，2，3，4，5，sensor\_fusion

更详细参数介绍，请参考[附录B](#_附录B)介绍。

测试结果分析  
 ITS测试没有单独的测试报告生成，如果所有项目测试完成，设备端CtsVerifier

中的Camera ITS Test项会自动变绿pass。测试结果可参考显示于测试电脑的shell窗口的内容，示例如下：

截取于camera1前摄，scene1场景1测试结果

Press Enter after placing camera 1 to frame the test scene: scene1

The scene setup should be: A grey card covering at least the middle 30% of the scene

Running vendor 3A on device

Capture an image to check the test scene

Capturing 1 frame with 1 format [yuv]

Please check scene setup in /tmp/tmpDRDobM/1/scene1.jpg

Is the image okay for ITS scene1? (Y/N)Y

Start running ITS on camera 1, scene1

......

PASS scene1/test\_capture\_result [9.0s]

PASS scene1/test\_crop\_region\_raw [9.8s]

SKIP scene1/test\_crop\_regions [2.0s]

FAIL scene1/test\_dng\_noise\_model [5.9s]

FAIL\* scene1/test\_ev\_compensation\_advanced [9.9s]

PASS scene1/test\_ev\_compensation\_basic [9.9s]

FAIL scene1/test\_exposure [15.3s]

......

27 / 31 tests passed (87.1%), 7 tests skipped

(\*) tests are not yet mandated

其中：

PASS 项表示测试通过

SKIP 项表示设备不支持此项测试，直接跳过，不算fail

FAIL 项表示测试失败，需要改变测试条件或修改软件进一步测试

FAIL\* 项表示测试失败，但目前Google认证没有强制要求通过，但一般都强烈建议修改可通过。

# 附录

## 附录A

分别介绍如下（以9.0\_r9测试包为例）：

● 场景0：没有设置要求。包含11项测试。

●场景1：设备在三脚架上，灯箱背景需要贴灰卡图片（请参考android-cts-verifier\CameraITS\tests\scene1\scene1.pdf 或scene1\_0.67\_scaled.pdf 背景需要灰色卡和白色边缘）此项包含36项，测试项较多，容易fail，不用急，后面可以重新测试场景。

● 场景2：此项用于测试人脸检测，设备在三脚架上，灯箱背景需要人脸图片（请参考android-cts-verifier\CameraITS\tests\scene2\scene2.pdf 或scene2\_0.67\_scaled.pdf），此项包含2项测试，比较容易通过

● 场景3：此项用于测试图像清晰度，设备在三脚架上，灯箱背景需要ISO 12233图表（请参考android-cts-verifier\CameraITS\tests\scene3\scene3.pdf 或scene3\_0.67\_scaled.pdf），此项包含6项测试。

● 场景4：此项用于测试场景的宽高比。设备在三脚架上，灯箱背景需要一个包含一个黑色圆圈和一个方框的静态测试页（请参考android-cts-verifier\CameraITS\tests\scene4\scene4.pdf 或scene4\_0.67\_scaled.pdf），此项包含2项测试。

● 场景5：此项用于测试镜头阴影和颜色均匀性。原文档介绍需要散光片，实测用一张白纸挡在设备摄像头前，正对光源即可，此项包含1项测试

● 场景fusion：此场景用于测试对焦模式，是在设备带有陀螺仪项目可能会开启的功能。目前已经测试的设备都未支持此项，后续补充。

## 附录B

测试指令详细参数使用方法

Command line arguments:

camera: the camera(s) to be tested. Use comma to separate multiple

camera Ids. Ex: "camera=0,1" or "camera=1"

device: device id for adb

scenes: the test scene(s) to be executed. Use comma to separate

multiple scenes. Ex: "scenes=scene0,scene1" or

"scenes=0,1,sensor\_fusion" (sceneX can be abbreviated by X

where X is a integer)

chart: [Experimental] another android device served as test chart

display. When this argument presents, change of test scene

will be handled automatically. Note that this argument

requires special physical/hardware setup to work and may not

work on all android devices.

result: Device ID to forward results to (in addition to the device

that the tests are running on).

rot\_rig: [Experimental] ID of the rotation rig being used (formatted as

"<vendor ID>:<product ID>:<channel #>" or "default")

tmp\_dir: location of temp directory for output files

skip\_scene\_validation: force skip scene validation. Used when test scene

is setup up front and don't require tester validation.

dist: [Experimental] chart distance in cm.