
高通多媒体技术期刊 20150206



Qualcomm Technologies, Inc.

Confidential and Proprietary – Qualcomm Technologies, Inc.

机密和专有信息——高通技术股份有限公司



Confidential and Proprietary – Qualcomm Technologies, Inc.

Confidential and Proprietary – Qualcomm Technologies, Inc.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or web sites to: DocCtrlAgent@qualcomm.com. **禁止公开：**如在公共服务器或网站上发现本文档，请报告至：DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm or its affiliated without the express approval of Qualcomm's Configuration Management. **限制分发：**未经高通配置管理部门的明示批准，不得发布给任何非高通或高通附属及关联公司员工的人。 Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc. 未经高通技术股份有限公司明示的书面允许，不得使用、复印、复制、或修改全部或部分文档，不得以任何形式向他人透露其内容。

The user of this documentation acknowledges and agrees that any Chinese text and/or translation herein shall be for reference purposes only and that in the event of any conflict between the English text and/or version and the Chinese text and/or version, the English text and/or version shall be controlling. 本文档的用户知悉并同意中文文本和/或翻译仅供参考之目的，如英文文本和/或版本和中文文本和/或版本之间存在冲突，以英文文本和/或版本为准。 This document contains confidential and proprietary information and must be shredded when discarded. 未经高通明示的书面允许，不得使用、复印、复制全部或部分文档，不得以任何形式向他人透露其内容。本文档含有高通机密和专有信息，丢弃时必须粉碎销毁。 Qualcomm reserves the right to make changes to the product(s) or information contained herein without notice. No liability is assumed for any damages arising directly or indirectly by their use or application. The information provided in this document is provided on an "as is" basis. 高通保留未经通知即修改本档中提及的产品或信息的权利。本公司对使用或应用本文档所产生的直接或间接损失概不负责。本文档中的信息为基于现状所提供，使用风险由用户自行承担。

Qualcomm is a trademark of QUALCOMM Incorporated, registered in the United States and other countries. All QUALCOMM Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners. Qualcomm是高通公司在美国及其它国家注册的商标。所有高通公司的商标皆获得使用许可。其它产品和品牌名称可能为其各自所有者的商标或注册商标。

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited. 本文档及所含技术资料可能受美国和国际出口、再出口或转移出口法律的 限制。严禁违反或偏离美国和国际的相关法律。

Qualcomm Technologies, Inc. 5775 Morehouse Drive San Diego, CA 92121 U.S.A.
高通技术股份有限公司，美国加利福尼亚州圣地亚哥市莫豪斯路 5775 号，邮编 92121

Revision History

Revision	Date	Description
A	Feb 2015	Initial release

Note: There is no Rev. I, O, Q, S, X, or Z per Mil. standards.

Contents



- Performance
 - Android Performance Patterns
 - Android渲染优化
 - Android性能优化工具
 - 高通Android性能优化文档树
 - PerfLock API 概述
 - 应用systrace进行系统性能分析
 - 性能分析和调试的ADB命令
- Camera
 - 使用auto flicker detection功能出现banding, 如何调试
 - 如何解决HDR拍照时图片过曝
 - 推荐使用的Camera文档及视频更新



Performance

Android Performance Patterns

- Google于2015年始发布了一系列Android性能优化的视频，帮助开发者开发出速度更快，性能更优的Android应用程序，视频链接如下：
<https://www.youtube.com/playlist?list=PLWz5rJ2EKKc9CBxr3BVjPTPoDPLdPIFCE>
- 中文blog中也对这些视频进行了非常优秀的总结和梳理，相关链接：
<http://hukai.me/android-performance-patterns/>



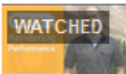
Android Performance Patterns

by Android Developers • 16 videos • 4,614 views • Last updated on Jan 7, 2015

Android Performance Patterns is a collection of videos focused entirely on helping developers write faster, more performant apps. On one side, it's about peeling back the layers of the Android System, and exposing how things are working under the hood. On the other side, it's about teaching you how the tools work, and what to look for in order to extract the right performance out of your app. But at the end of the day, Android Performance Patterns is all giving you the right resources, at the right time to help make the most awesome experience for your users. And that's the whole point, right? Less.

[▶ Play all](#) [◀ Share](#) [+ Save](#)


1



Android Performance Patterns: Rendering Performance 101

Google Developers


2



Android Performance Patterns: Understanding Overdraw

Google Developers

3



Android Performance Patterns: Understanding VSYNC

Google Developers

Android渲染优化

- 该专辑包括16个视频，深入讨论了Android的渲染机制，通过各种工具分析和定位渲染性能瓶颈，并如何优化；同时视频还讨论了内存和电量优化
- 屏幕的刷新频率一般是60Hz, 为了取得流畅的用户体验，应用程序必须保持60fps的画面更新速率，当应用画面更新低于60fps，就会出现丢帧，用户会感受到卡顿，
- 保障应用程序流畅的关键是如果让CPU和GPU在16ms内处理完一帧画面。
- 开发者总是喜欢用最多的动画，更炫的特效，更多的图层等复杂界面来取得更好的用户体验，但是并不是所有手机平台都能处理如此复杂的应用程序，因此开发者必须对其应用程序进行优化

Android性能优化工具

- Google为Android性能优化提供了一系列工具
- 手机设置里面的开发者选项，选择Profile GPU Rendering，选中On screen as bars的选项，然后操作手机，你可以实时观察每帧渲染所需时间
 - Profile GPU Rendering的输出并不能完整的描述真实的问题，高通推荐用systreace, dumsys gfxinfo, 或者别的办法共同分析，后续期刊我们会重点讲解
- 如果应用程序每帧渲染时间都低于16ms，表明应用渲染很是完美，你不需要更多进一步工作；否则，你必须通过下面工具来定位你应用程序的瓶颈
- 手机设置里面的开发者选项，打开Show GPU Overdraw的选项，可以观察UI上的Overdraw情况
- 使用HierarchyViewer来查找Activity中的布局是否过于复杂
- 使用TraceView来观察CPU的执行情况，更加快捷的找到性能瓶颈
- 更多的内容请大家观看原视频文件，或者阅读相关博客
<http://hukai.me/android-performance-patterns/>

高通Android性能优化概述

- 前面谈到的是开发者如何从应用程序上优化性能，这些方法和工具适用于所有Android平台
- 高通对Android平台持续地进行性能的优化，确保客户基于高通平台的手机具有最佳的性能和功耗平衡。
- 高通性能优化文档树包括一系列文档（见下页），用来帮助客户对高通平台进一步优化，下面的章节对这些文档分别简要说明，请下载这些文档获取更多细节

高通Android性能优化文档树

Title	DCN	Description
MPCTL	80-NR256-2	介绍MPCTL Perflock API的具体实现，帮助客户更好的理解和应用Perflock API
CPU Governor Parameters	80-NR256-3	帮助客户理解interactive governor parameters and CPU Boost Parameters
UX - Guide to Analyse System issues	80-NR256-4	介绍如何应用systrace进行CPU相关的用户体验性能分析
UX – Guide to Analyse GPU and Display related issues	80-NR256-5	介绍如何应用systrace进行GPU和显示相关的用户体验性能分析
Memory	80-NR256-6	介绍如何生成Memory Map以及如何进行内存相关的性能调试
UX – Guide to Analyse Scrolling issues and Launch latencies	80-NR256-7	介绍如何应用systrace进行scrolling and launch latencies性能调试
ADB Commands	80-NR256-8	介绍常用的用于性能分析和调试的ADB命令

PerfLock API 概述

- 高通对CPU governor, mpdecision缺省都进行了系统级的优化，确保其参数对整个手机平台性能最优
- 针对某些特定应用，我们可能还需要进一步优化。比如在触摸和启动应用程序的时刻，或者滑动浏览器，用户期待迅速提升CPU频率，取得更加流畅的用户体验
- 而对视频播放等场景，用户期待更好的功耗表现
- 高通提供PerfLock API，客户可以对某些特定应用进一步进行性能和功耗优化。
- [80-NT384-1 A PerfLock API Overview](#) 介绍了PerfLock API的结构和用法
- [80-NR256-2 A MPCCTL Feature](#)则介绍了PerfLock API的具体实现

应用systrace进行系统性能分析

- 高通常用Systrace进行性能分析，当客户报告任何性能问题，通常我们需要客户第一时间提供systrace给我们
- 下面几篇文档都介绍了如何抓取systrace，并分别给出了一些简单的事例分析
- [80-NR256-4 A Analyzing UX-Related System Issues](#)
- [80-NR256-5 A Analyzing UX-Related GPU and Display-Related Issues](#)
- [80-NR256-7 A Analyzing UX-Related Scrolling Issues and Launch Latencies](#)

性能分析和调试的ADB命令

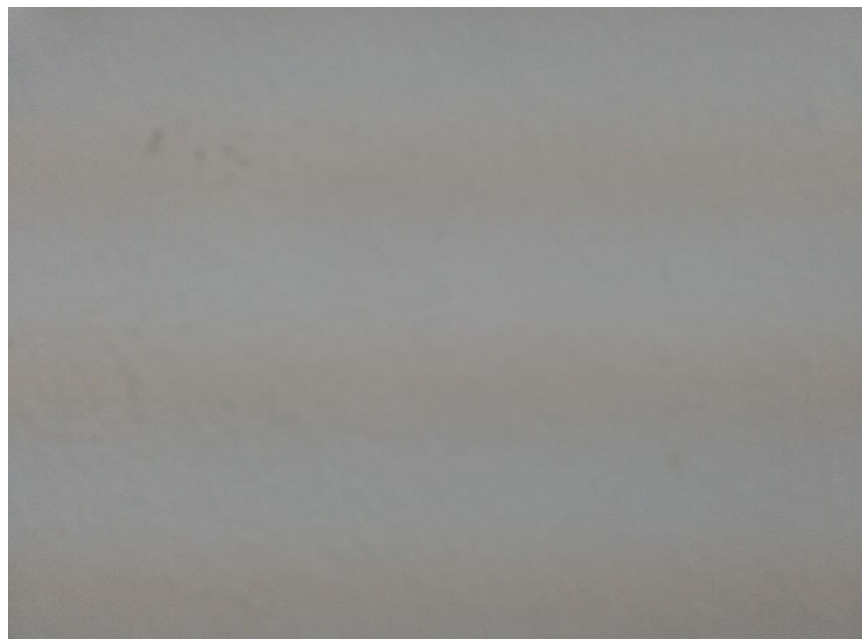
- 当我们怀疑某些性能问题与CPU调度有关时，我们建议把CPU调到 Performance Mode作最初步的定位分析
- 相应的，如果我们怀疑某些性能与GPU有关，我们也会把GPU调到 Performance Mode作最初步的定位分析
- 如果我们怀疑某些性能与显示相关，我们会建议客户去检查fps，同时查看当前应用有多少layers参与合成，合成方式是否正确
- 所有这些调试都可以通过ADB命令完成，具体命令请参考 [80-NR256-8_A_ADB_Commands_Debug_Performance_Issues](#)



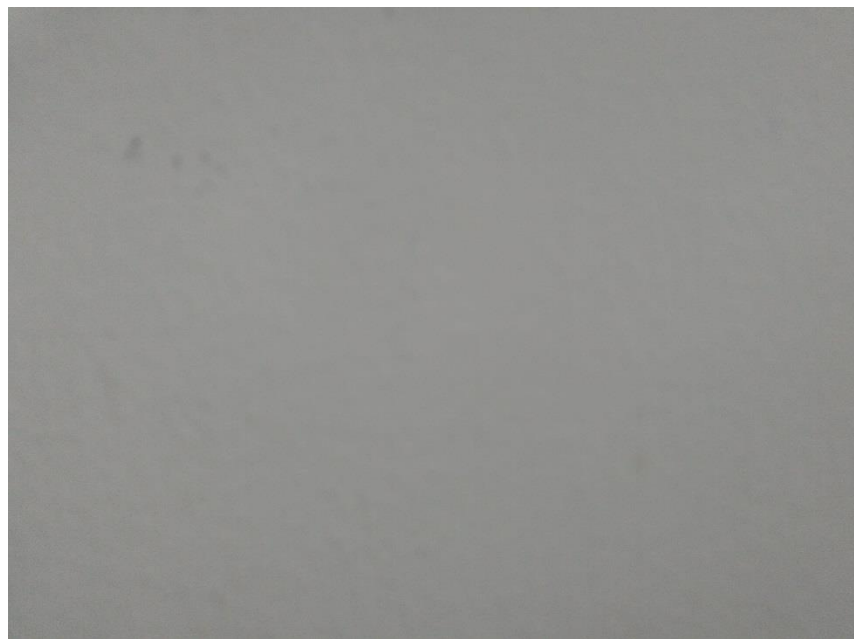
Camera

使用auto flicker detection功能出现banding, 如何调试

问题：使用自动50/60Hz侦测功能，发现预览界面和拍照图片出现banding,如下左图：



有banding图片



无banding图片

使用auto flicker detection功能出现banding, 如何调试

解决：

1. 如下图，通过菜单，设置到50Hz, 在相同的环境下，测试是否有banding, 如果有，则可能是驱动问题，需要check曝光时间的设置；



Tip: 测试环境必须有日光灯，且不能太亮，如果环境太亮，AE曝光时间会跑到10毫秒以下，此时出现banding是正常的。

使用auto flicker detection功能出现banding, 如何调试

解决：

2. 确定不是driver问题，请使用下面命令，开启auto flicker detection log，抓取复现问题的 log 分析；

`adb shell setprop persist.camera.stats.debug.mask 17`

3. 以“afd_process_pack_output” 为关键字搜索log, 我们会找到类似以下的log:

afd_process_pack_output: afd_enable 1, monitor 0, flicker 0, freq 50, afd_state 5, afd_atb 2

afd_process_pack_output: afd_enable 1, monitor 0, flicker 0, freq 50, afd_state 5, afd_atb 2

afd_process_pack_output: afd_enable 1, monitor 0, flicker 0, freq 50, afd_state 5, afd_atb 2

红色字体部分，表示侦测结果；

4. 以“afd actual peak” 为关键字搜索log, 我们会找到类似以下的log:

afd numpeaks 4 afd actual peak 10

afd numpeaks 4 afd actual peak 9

afd numpeaks 4 afd actual peak 9

afd numpeaks 4 afd actual peak 9

afd numpeaks 4 afd actual peak 9

afd numpeaks 4 afd actual peak 11

actual peak 表示在当前场景，找到的明暗变化的条数。这个值如果大于我们默认设置的6，就会认为场景中，出现banding,需要调整banding设置。

使用auto flicker detection功能出现banding, 如何调试

解决：

5. 找到复现问题的场景，当场景被识别为50Hz时候，预览画面中没有banding, 此时可以通过如下log动态侦测“actual peak”的值，如果 actual peak 在画面中，没有banding的情况下，仍然出现大于6的值，导致了50Hz到60Hz的转换，那么就需要调参数了；

```
adb logcat | find/I "actual peak"
```

6. 调整header文件中，以下参数：

```
90, /* Difference Threshold */ ,
```

此参数可理解为当画面中的亮暗程度相差多大时候，会被认为是一个亮暗的peak. 所以这种情况下，我们需要增加这个参数值，使被认为一个peak的条件变严格。

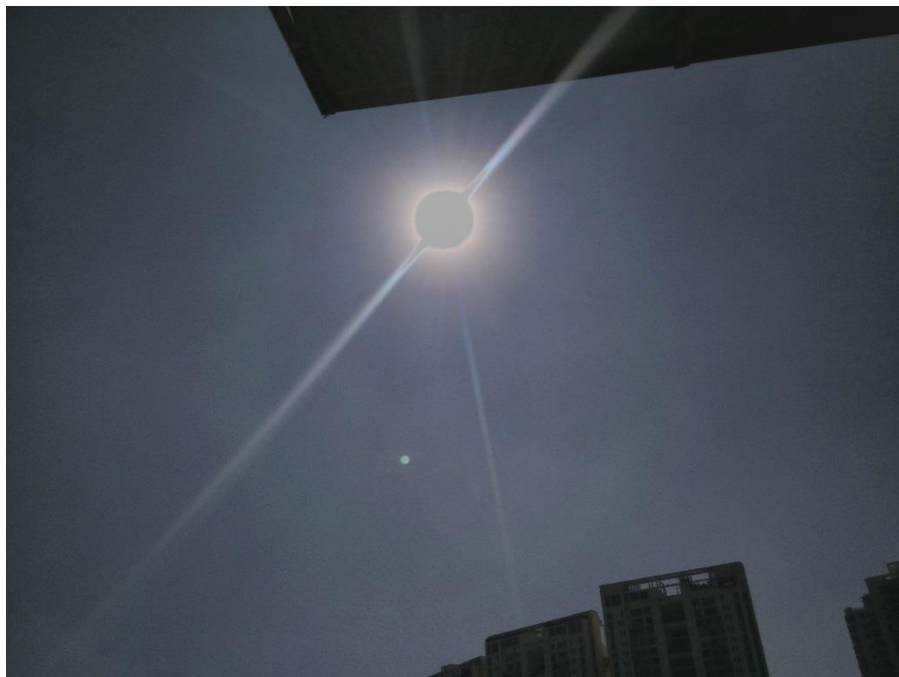
7. 调整参数，使camera在50Hz，对着白纸或者白墙时候，actual peak的值不大于6，即能使auto flicker detection不误判。

Tip: 关于auto flicker detection方面更多的内容，大家可以参考

[80-VF110-1_C_Qcamera_Train_Automatic_Flicker_Detect.pdf](#) 这份文档。

开启HDR拍照，有一张图片过曝

问题：开启HDR对着高亮处拍照，其中一张图片严重过曝，如下图：



正常图片



严重过曝图片

开启HDR拍照，有一张图片过曝

问题图片2：



正常图片



严重过曝图片

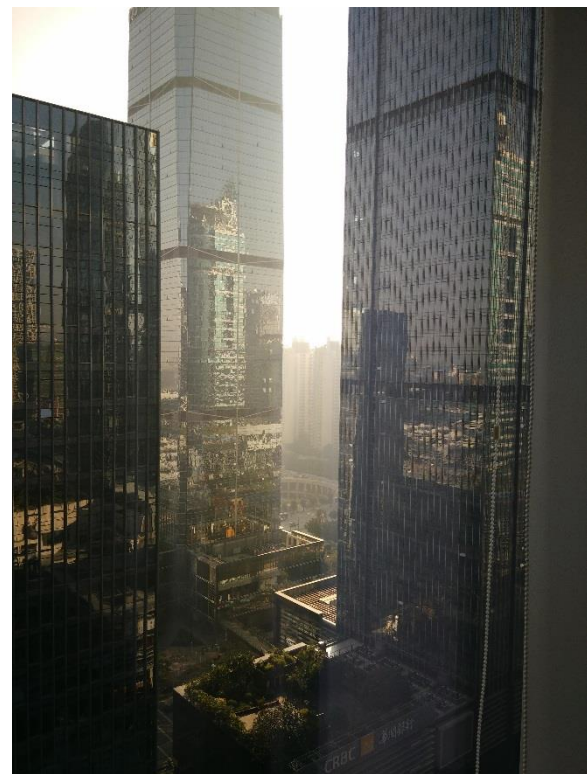
开启HDR拍照，有一张图片过曝

解决：

1. 使用以下命令，将用于合成HDR的三张原始照片，保存出来；

```
adb shell setprop persist.camera.imglib.hdr.dump in out
```

2. 保存下来的图片，为YUV格式，请使用chromatix打开查看；



开启HDR拍照，有一张图片过曝

解决：

3. 可以看到第一张图片是过曝的，通过“ae-bracket”搜索问题复现的log，会找到以下log，显示出三张图片的real gain 和 line count:

```
sensor_set_vfe_sof:635 ae-bracket:idx[0] of 3 real_gain=1.007812, linecount=60
sensor_set_vfe_sof:637 ae-bracket: sof_counter=3, applying exposure, idx=0
sensor_set_vfe_sof:635 ae-bracket:idx[1] of 3 real_gain=1.000000, linecount=30
sensor_set_vfe_sof:637 ae-bracket: sof_counter=2, applying exposure, idx=1
sensor_set_vfe_sof:635 ae-bracket:idx[2] of 3 real_gain=1.007812, linecount=84
sensor_set_vfe_sof:637 ae-bracket: sof_counter=1, applying exposure, idx=2
```

4. 第一张图片的line count比第三张图片小，而亮度却比第三张图片亮非常多，不太合理，怀疑sensor HDR延时不够，导致第一张图片过亮；

5. 修改sensor driver 中的 .sensor_num_HDR_frame_skip 设置，从1改为2，问题解决。

推荐使用的Camera文档及视频更新

敬爱的客户，

如你所知，我们在最近几个月中针对您常遇到的问题，增加改进了文档的内容。我们强烈建议你和你的团队研究以下重点培训材料。这些材料应该可以回答您经常提到的问题。完整的文档列表可以参见决方案[00028470](#)（软件）和[00028513](#)（图像质量）。

推荐使用的camera软件培训材料

- [80-NU323-2](#): Multimedia Driver Development and Bringup Guide – Camera
- [80-NL239-33](#): Linux Camera Debugging Application Note
- VD80-NA157-22 ([View](#)): Video: MSM8974/APQ8074/MSM8X26/APQ8084 Linux Camera Overview
- VD80-NF499-4 ([View](#)): Video: Qcamera Server And Mctl Code Walk Through For MSM8974/APQ8074/MSM8X26 Linux Camera SW
- VD80-NF499-5 ([View](#)): Video: Sensor Module Code Walkthrough For MSM8974/APQ8074/MSM8X26 Linux Camera Software
- VD80-NF499-2 ([View](#)): Video: Camera FrontEnd Code Walkthrough For MSM8974/APQ8074/MSM8X26 Linux Camera Software
- VD80-NF499-3 ([View](#)): Video: Kernel Code Walkthrough For MSM8974/APQ8074/MSM8X26 Linux Camera Software
- [80-NL239-36](#): Camera Workflow
- [80-NC193-10](#) (English), [80-NC193-10SC](#) (Chinese): Accessing PVL Camera Drivers from Global Components Database
-

推荐使用的camera图像质量调试的培训材料

- [80-NK872-5](#): 3A Tuning Guide
- [80-NK872-2](#): Chromatix 6 Camera Tuning
- [80-NK872-4](#): Chromatix 6 Camera Tuning For MSM8994/APQ8084
- [80-NK872-6](#): Typical Issues Tuning Guide
- [80-NK872-1](#): Chromatix 6 User Guide
- [80-N5126-1](#): Camera Module Selection And Calibration Data
- [80-NN869-1](#): Sensor PLD And AF Gating Process Overview

Questions?

<https://support.cdmatech.com>

