Application Note

AM6xA ISP 调优指南

Jianzhong Xu, Gang Hua, and Chau Le *Sitara MPU*

摘要

本应用报告介绍了以下工作流程，即为原始摄像头调优 TI AM6xA 视觉处理器系列上的 ISP，以获得尽可能最佳的图像质量。本报告中提供的调优过程在 AM62A 入门套件 EVM 上结合使用 IMX219 传感器（仅 RGB）和 OX05B1S 传感器 (RGB-IR)。

内容

1 引言....3
2 调优概述....3
3 硬件要求....4
4 软件要求....5
4.1 Processor SDK Linux®....5
4.2 TI 的参考成像软件....5
4.3 ISP 调优工具....5
5 传感器软件集成....5
5.1 图像流水线软件架构概述....5
5.2 向 SDK 添加传感器驱动程序....6
5.3 更新 TIOVX 模块....6
5.4 更新用于 VISS 的 GStreamer 插件....6
6 调优过程....9
6.1 验证摄像头捕捉是否能够正常运行....9
6.2 使用初始 VPAC 配置启用摄像头流式传输....9
6.3 调整摄像头安装....11
7 执行基本调优....12
7.1 启动调优工具并创建工程....12
7.2 调优顺序....14
7.3 黑电平消减....15
7.4 硬件 3A (H3A)....16
7.5 PCID....16
7.6 自动白平衡 (AWB)....18
7.7 颜色校正....21
8 执行微调....24
8.1 边缘增强 (EE)....24
8.2 噪声滤波器 4 (NSF4)....25
9 实时调优....27
9.1 要求....27
9.2 支持的功能....27
10 总结....29
11 修订历史记录....29

插图清单

图 3-1. 调优硬件和设备设置....4
图 5-1. AM6xA 图像处理流水线....6
图 7-1. 用于调优的传感器原始图像参数....12
图 7-2. DCC 调优工具的图像预览....13

图 7-3. DCC 调优工具插件.....14
图 7-4. 黑电平消减调优.....15
图 7-5. 黑电平消减前的图像.....16
图 7-6. 黑电平消减后的图像.....16
图 7-7. PCID 调优.....17
图 7-8. PCID 的原始图像输入.....18
图 7-9. Bayer 图形格式的 PCID 输出图像.....18
图 7-10. DCC 调优工具的实时捕获功能.....18
图 7-11. 使用 D50 照明时捕获的色卡图像.....19
图 7-12. 使用 D65 照明时捕获的色卡图像.....19
图 7-13. 使用 TL84 照明时捕获的色卡图像.....19
图 7-14. 使用 A 灯时捕获的色卡图像.....19
图 7-15. 自动白平衡调优.....20
图 7-16. 选择色卡的边角.....20
图 7-17. 自动白平衡调优结果.....21
图 7-18. 自动白平衡调优前的图像.....21
图 7-19. 自动白平衡调优后的图像.....21
图 7-20. 颜色校正调优.....22
图 7-21. 颜色校正调优输出.....22
图 7-22. 颜色校正调优前的图像.....23
图 7-23. 颜色校正调优后的图像.....23
图 8-1. EE Semi-Automatic 调优 GUI.....24
图 8-2. EE 调优之前的 YUV 图像输入.....25
图 8-3. EE 调优之后的 YUV 图像输出.....25
图 8-4. NSF4 调优 GUI.....26
图 9-1. EVM IP 地址.....27
图 9-2. RAW 捕捉.....27
图 9-3. YUV 捕捉.....28
图 9-4. DCC 更新.....28
图 9-5. 用于实时调优的曝光控制.....28
图 9-6. 用于实时调优的白平衡控制.....29

商标

1 引言

AM6xA 视觉处理器具有硬件加速图像信号处理器 (ISP)，也称为视觉预处理加速器 (VPAC)。VPAC 具有可配置的图像处理参数，旨在支持各种原始摄像头模块（典型的原始摄像头模块包括镜头、滤镜、原始图像传感器，有时还包括串行器）。为了在运行时获得特定原始摄像头模块的最佳图像质量，需要计算 VPAC 的参数，然后利用这些参数来逐帧处理原始传感器图像。为了实现这一点，工程师通常会在成像实验室的各种受控照明条件下准备合适的 VPAC 参数。然后在运行时，借助自动曝光 (AE)、自动白平衡 (AWB) 和动态 ISP 参数控制的软件成像算法，参考准备好的参数并通过插值来适应运行时照明环境。在成像实验室中准备合适的 VPAC 参数的过程在本应用报告中称为 ISP 调优。

本报告中所述的 ISP 调优过程适用于 AM6xA 视觉处理器系列中的所有 SoC，包括 AM62A、AM68A 和 AM69A。报告中提供了多个使用 AM62A 入门套件 EVM 的示例。

有关特定片上系统 (SoC) 上 ISP (VPAC) 的技术详细信息，请参阅该 SoC 的技术参考手册 (TRM)。

2 调优概述

AM6xA 系列 SoC 上的 ISP (VPAC) 通过动态摄像头配置 (DCC) 二进制文件进行配置。在基于 Linux® 的应用程序中，这些二进制文件通过常用的 GStreamer 流水线提供给 VPAC。VPAC 的处理块由 GStreamer 流水线元素（tiovxisp、tiovxldc 和 tiovxmultiscaler）封装，而 VPAC 的所有可配置参数都作为属性提供。

例如，以下 GStreamer 流水线将视频从 IMX219 摄像头流式传输到高清多媒体接口 (HDMI) 显示屏，并会在发送到显示屏前使用 VPAC 处理原始图像：

```
gst-launch-1.0 v4l2src device=/dev/video-imx219-cam0 io-mode=dmabuf-import ! \
video/x-bayer, width=1920, height=1080, framerate=30/1, format=rggb10 ! \
tiovxisp sink_0::device=/dev/v4l-imx219-subdev0 \
sensor-name="SENSOR_SONY_IMX219_RPI" \
dcc-isp-file=/opt/imaging/imx219/dcc_viss_10b.bin \
sink_0::dcc-2a-file=/opt/imaging/imx219/dcc_2a_10b.bin format-msb=9 ! \
video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! \
kmssink driver-name=tidss sync=false
```

在上面的流水线中，GStreamer 元素 **tiovxisp** 连接 VPAC 硬件和 TI 用于 AE 和 AWB (2A) 以及 ISP 参数控制成像算法的参考软件。IMX219 的 VPAC 配置通过以下两个二进制文件提供，这两个文件属于 **tiovxisp** 的属性：

- dcc-isp-file=/opt/imaging/imx219/linear/dcc_viss_10b.bin，用于提供调优的 ISP 参数
- dcc-2a-file=/opt/imaging/imx219/linear/dcc_2a_10b.bin，用于提供 AE 和 AWB 算法校准信息

这些二进制文件是 ISP 调优的输出，也称为动态摄像头配置 (DCC) 配置文件。

概括而言，AM6xA ISP 调优过程包括以下步骤（以 TI 的参考成像软件和调优工具为例）：

1. 硬件设置：准备和设置所有必要的硬件设备。
2. 软件设置：下载并安装所有必要的软件组件。
3. 传感器软件开发与集成：确保摄像头传感器驱动程序已正确与系统集成，并且可以捕获原始图像。添加自动曝光算法所需的曝光配置。向 GStreamer 插件添加对该传感器的支持。
4. 生成初始 ISP 配置：运行 Python® 脚本来为 VPAC 配置生成初始（默认或基准）DCC 配置文件。此配置可以实现具有足以运行下一步的良好图像质量的视频流。
5. 调整摄像头安装：使用上一步生成的 DCC 配置文件运行实时视频流式传输，并调整摄像头模块安装，以确保捕获位置、聚焦、照明等均良好的图像。
6. ISP 基本调优：采集原始图像并执行基本调优，以便在实验室条件下实现最佳图像质量的 70%~80%。
7. ISP 微调：使用在上一步中生成的新 DCC 配置文件再次运行实时流式传输。根据图像质量，确定需要改进的方面，并在必要时执行额外微调。

本应用手册使用 AM62A 入门套件 EVM 以及 IMX219 摄像头和 OX05B1S 摄像头来演示上述调优步骤。调优的原理和过程适用于任何定制板和原始摄像头。

3 硬件要求

对原始摄像头执行 ISP 调优所需的硬件设备包括：

- AM62A 入门套件 EVM (SK EVM) 或带有 AM62A SOC 的定制板。有关完整的 EVM 设置，请参阅 AM62A SK EVM 快速入门指南。
- 摄像头以及将摄像头连接到 AM62A SK EVM 所需的所有附件。有关如何将摄像头连接到 EVM 的信息，请参阅 AM62A Academy。在 Academy 主页上，转到 *Evaluating Linux → Tour of TI Linux → Camera*。
- 灯箱：
  - 带有经过校准的光源，通常至少在 2700K 至 6500K 之间
  - 如果需要更宽的色温范围，则需要更多的光源
- XRite 色卡。在某些情况下，根据设置需要不同的色卡大小。
- 用于监控和记录照明条件的光度计和/或色度计。
- 将摄像头固定在所需位置而需要的三脚架或支架。图 3-1 展示了此类设置的一个示例。

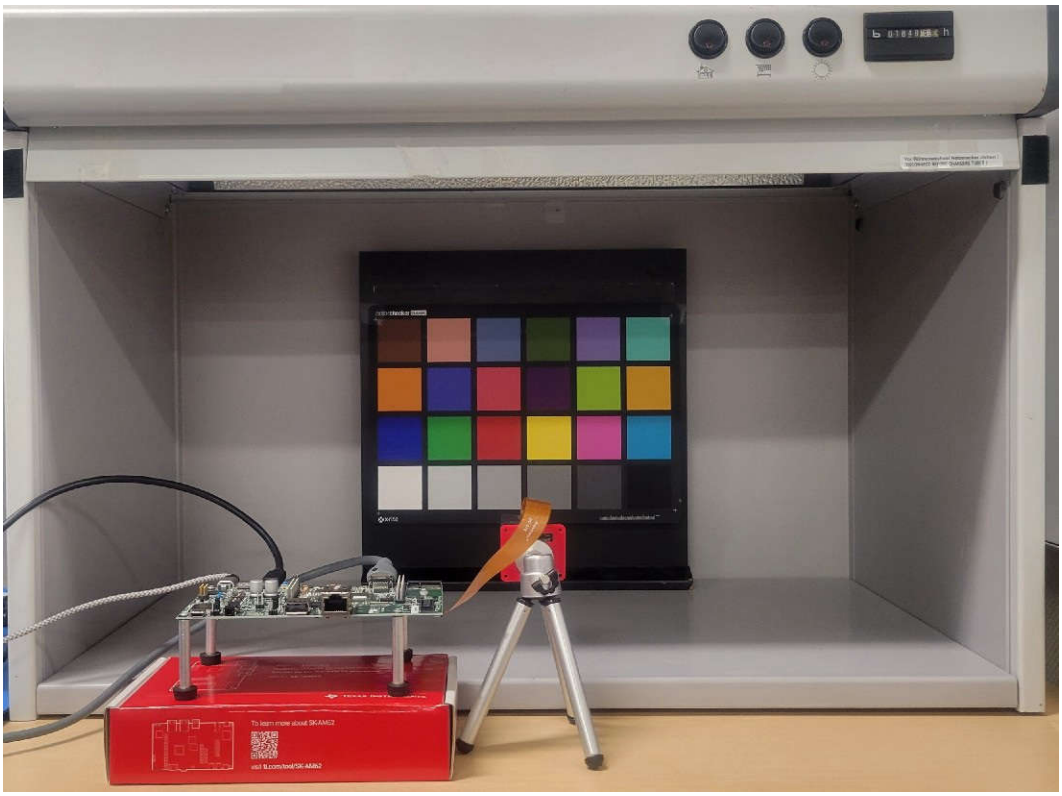


图 3-1. 调优硬件和设备设置

4 软件要求

4.1 Processor SDK Linux®

下载适用于 AM62A 的处理器软件开发套件 (SDK) Linux。

要将 EVM 引导至 Linux，请遵循 AM62A SK EVM 快速入门指南。

4.2 TI 的参考成像软件

将成像软件从 https://git.ti.com/cgit/processor-sdk/imaging/ 克隆到 Linux 主机 PC：

```
$ git clone git://git.ti.com/processor-sdk/imaging.git --branch main
```

本报告具有 TI 用于 AE、AWB 和 ISP 控制或 DCC 的参考成像算法。此信息还包含调优过程中需要的 Python 脚本。

4.3 ISP 调优工具

ISP 调优工具和成像算法（例如 AE、AWB 和 ISP 控制）可从多个成像第三方获得，这些第三方可能为客户的生产提供支持。TI 的 TDA4 和 AM6xA DCC 调优工具和成像算法也可用于设计参考，并在本报告中用作示例。可联系当地的 TI FAE 以获取此工具。TI 的第三方调优工具也可以采用类似的方式使用，但本报告不对此进行介绍。

5 传感器软件集成

要使用 AM6xA Linux SDK 来针对目标传感器进行 ISP 调优，需要开发软件以在 SDK 中支持相应的传感器。传感器驱动程序开发不在本文档的讨论范围内。后面各小节首先概述 AM6xA 上的图像流水线软件架构，然后介绍如何将传感器驱动程序添加到 SDK 以及如何修改 GStreamer 插件以支持传感器。

5.1 图像流水线软件架构概述

本节概述了 AM6xA 上的 ISP 和图像流水线软件架构，有助于理解后面介绍的调优过程。

AM6xA 器件上的 ISP 提供通用视觉基元功能，用于在像素级执行的图像数据处理。ISP 中有三个子模块：视觉成像子系统 (VISS)、镜头失真校正 (LDC) 和多标量 (MSC)。

- **视觉成像子系统 (VISS)：**VISS 对原始数据进行图像处理，其中包括宽动态范围 (WDR) 合并、缺陷像素校正 (DPC)、镜头阴影校正 (LSC)、全局和局部亮度和对比度增强 (GLBCE)、去马赛克、颜色转换和边缘增强 (EE)。该块采用原始图像作为输入并生成 YUV 输出。
- **镜头失真校正 (LDC)：**LDC 引擎是一款 YUV 域处理器，设计用于执行透视和几何转换。此引擎可用于创建多种效果，例如镜头失真校正、对极校正和通用透视变换。
- **多标量 (MSC)：**MSC 可以根据给定输入以各种缩放比例（1 × 和 0.25 × 之间）生成最多 10 个缩放输出。10 个缩放操作均可配置为执行金字塔缩放或倍频程间的缩放生成。

在这三个子模块中，需要对 VISS 和 LDC 进行调优，而 VISS 包含多个需要单独调优的处理块。

图 5-1 展示了 AM6xA 的图像处理流水线软件架构，包括摄像头捕捉子系统和 ISP。摄像头捕捉驱动程序在 Linux 中 A53 或 A72 内核上运行，ISP 驱动程序在 RTOS 中 R5 内核上运行。尽管在不同的 CPU 内核上运行，摄像头捕捉和 ISP 可以通过与 GStreamer 或 TIOVX 的相同框架连接和集成。

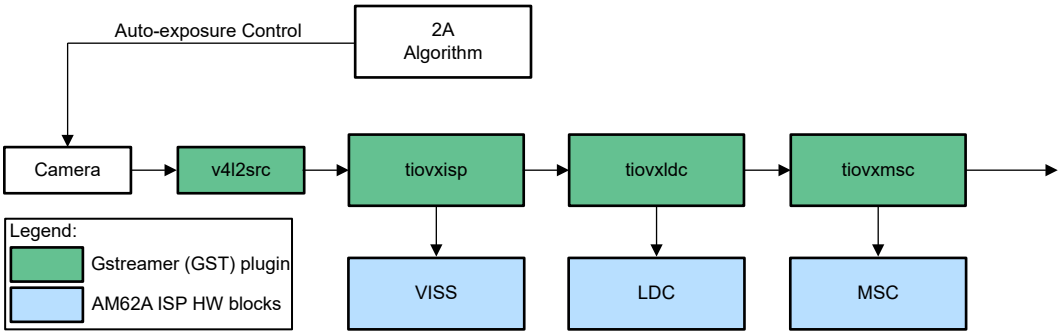


图 5-1. AM6xA 图像处理流水线

在本报告中，提供了用于集成图像流水线组件的 GStreamer 示例。摄像头捕捉子系统可通过 GStreamer 插件 v4l2src 访问。可通过相应的 GStreamer 插件访问这三个 ISP 子模块：tiovxisp、tiovxldc 和 tiovxmsc。

5.2 向 SDK 添加传感器驱动程序

有关向 SDK 添加传感器驱动程序的说明，请参阅 AM62A Academy。在 Academy 主页上，转到 *Develop Linux on TI EVM → Use Device Drivers → Use Camera → Enable a New CSI-2 Sensor*。

除了向 SDK 添加传感器驱动程序之外，还需要更新用于 ISP 的 GStreamer 插件以支持传感器。这涉及到修改和重建 EVM 上目标文件系统中的两个组件（在后续小节中说明）。

5.3 更新 TIOVX 模块

5.3.1 源代码更改

在 EVM 上的目标文件系统中，转到 `/opt/edgeai-tiovx-modules/src/tiovx_sensor_module.c`，并将传感器的名称和 ID 添加到 `tiovx_init_sensor()` 函数中的列表。例如，IMX219 对应的情况如下所示：

```
// This line defines a new TIOVX camera sensor string ID for IMX219
else if(strcmp(sensorObj->sensor_name, "SENSOR_SONY_IMX219_RPI") == 0)
{  // This line assigns a new numeric DCC ID to IMX219
   sensorObj->sensorParams.dccId=219;
}
```

此处定义了 IMX219 摄像头的传感器名称和 DCC ID。GStreamer 流水线中使用确切的名称字符串 *SENSOR_SONY_IMX219_RPI*，而在后续步骤中，调优工具会使用 DCC 传感器 ID (219)。

5.3.2 重新编译模块

完成源代码更改后，通过运行 `/opt/edgeai-gst-apps/scripts/install_tiovx_modules.sh` 脚本，重新编译并重新安装模块。

5.4 更新用于 VISS 的 GStreamer 插件

为了使 ISP 支持新的传感器，需要更新用于 VISS 的 GStreamer 插件，其中包括：

- 将该传感器添加到插件属性的传感器列表中
- 为该传感器添加曝光设置，自动曝光和自动白平衡 (2A) 算法需要此设置，如图 5-1 所示。

5.4.1 更新 VISS 插件属性

在 EVM 上的目标文件系统中，转到 /opt/edgeai-gst-plugins/ext/tiovx/gsttiovxisp.c，并将传感器名称添加到列表中（下面以 OX05B1S 为例）：

```
g_object_class_install_property (gobject_class, PROP_SENSOR_NAME,
    g_param_spec_string ("sensor-name", "Sensor name",
        "TIOVX camera sensor string ID. Below are the supported sensors\n"
...
        "SENSOR_OX05B1S\n"
...
```

5.4.2 添加 2A 算法的曝光设置

如 图 5-1 所示，摄像头传感器的曝光是通过 2A 算法自动控制的。2A 算法根据 ISP 提供的 H3A 统计数据调整曝光。曝光时间和增益均可通过 2A 算法进行调整。每个传感器都有一个可以从内部操作的最短/最长曝光时间和最小/最大增益。这些规格可在数据表中找到，需要通过 VISS 插件传递给 2A 算法。需要添加一个新函数来将此信息提供给 2A 算法。该函数在 gsttiovxisp.c 中通常命名为 get_<sensor>_ae_dyn_params()。

5.4.2.1 增益

最小和最大增益通常在数据表中被指定为 Nx 增益和 Mx 增益，其中 N 是最小值，M 是最大值。通常还会在增益寄存器中提供相应的值。例如，某个传感器可能会指定以下参数：

- 最小增益：1 ×，gain_register 值为 16
- 最大增益：15.5 ×，gain_register 值为 248

由于 DCC 实时调优工具使用 1024 表示 1 × 增益来以浮点数计算和显示增益，因此最好在此转换中设置增益并在 2A 与传感器驱动程序之间映射增益。函数 get_<sensor>_ae_dyn_params() 具有以下最小和最大增益设置：

```
p_ae_dynPrms->analogGainRange[count].min = 1024; /* 1x gain */ p_ae_dynPrms-
>analogGainRange[count].max = 15872; /* 15.5x gain */
```

因此，2A 算法使用的增益值实际上是此特定传感器所用增益值的 64 倍。在 2A 算法返回要为传感器设置的增益后，该增益值需要除以 64，然后才能被发送到传感器。此映射在函数 gst_tiovx_isp_map_2A_values() 中完成：

```
*analog_gain_mapped = analog_gain / 64;  /* 64 = 1024 / 16 */
```

5.4.2.2 曝光时间

对于曝光时间，最小值和最大值通常指定为行周期数。例如，某个传感器可能具有以下规格：

- 最短曝光时间：6 行周期
- 最长曝光时间：（帧长度 – 30）行周期

以分辨率 2592 × 1944 为例，帧长度为 1944 行加垂直消隐。假设垂直消隐为 184，则帧长度为 2128 行。因此，最长曝光时间为 2128 – 30 = 2098 行周期。检查传感器驱动程序，并确保根据数据表将曝光时间写入寄存器。

由于 DCC 实时调优工具以微秒为单位显示曝光时间，因此推荐的做法是设置 2A 的曝光时间（以微秒为单位）并在 2A 与传感器驱动程序之间执行映射。映射取决于帧大小和帧速率。例如，对于 2592 × 1944 分辨率和 60fps，最短和最长曝光时间可在函数 get_<sensor>_ae_dyn_params() 中按如下所示设置：

```
p_ae_dynPrms->exposureTimeRange[count].min = 47; /* 6*16.67/2128*1000 micro sec */ p_ae_dynPrms-
>exposureTimeRange[count].max = 16435; /* (2128-30)*16.67/2128*1000 micro sec */
```

因此，函数 gst_tiovx_isp_map_2A_values() 中提供了从微秒到行周期数（传递给传感器驱动程序）的映射：

```
*exposure_time_mapped = (int) ((double)exposure_time * 2128 * 60 / 1000000 + 0.5);
```

5.4.2.3 其他参数

2A 算法所需的其他参数可以使用以下默认值：

```
/* setting brightness target and range: range is always [target-threshold, target+threshold]. -
numbers in 0~255 range
*/
p_ae_dynPrms->targetBrightnessRange.min = 40; /* lower bound of the target brightness range */
p_ae_dynPrms->targetBrightnessRange.max = 50; /* upper bound of the target brightness range */
p_ae_dynPrms->targetBrightness = 45;  /* target brightness */
p_ae_dynPrms->threshold = 5;  /* maximum change above or below the target brightness */
p_ae_dynPrms->enableBlc = 0;  /* not used */

/* setting exposure and gains */
p_ae_dynPrms->exposureTimeStepSize = 1;  /* step size of automatic adjustment for exposure time */
p_ae_dynPrms->digitalGainRange[count].min = 256;  /* digital gain not used */
p_ae_dynPrms->digitalGainRange[count].max = 256;  /* digital gain not used */
```

5.4.3 重新编译插件

更改源代码 `/opt/edgeai-gst-plugins/ext/tiovx/gsttiovxisp` 后，通过运行 `/opt/edgeai-gst-apps/scripts/install_gst_plugins.sh` 脚本重新编译并重新安装 GStreamer 插件。

5.4.4 在 GStreamer 插件中验证新传感器

完成上述所有更改后，验证 GStreamer 插件 tiovxisp 的属性中是否列出了新传感器名称。例如，OX05B1S 如下所示：

```
root@am62axx-evm:~# gst-inspect-1.0 tiovxisp
...
  sensor-name : TIOVX camera sensor string ID. Below are the supported sensors
                          SENSOR_OX05B1S
```

6 调优过程

本节详细介绍 ISP 调优过程。总而言之，该过程包括以下步骤：

1. 验证摄像头是否正常运行
2. 使用自动脚本创建初始（默认或基线）ISP 配置文件
3. 在受控的照明条件下执行实验室调优。调优可分为 2 个步骤：基本调优和微调。
4. 在要部署传感器的实际场景中执行生产调优。对于此步骤，本报告不再做介绍。

6.1 验证摄像头捕捉是否能够正常运行

假设摄像头驱动程序已集成到 SDK 中，并且 AM62A SK EVM 引导至 Linux 并可以探测摄像头。确认 `v4l2-ctl` 和 `media-ctl` 命令都显示了如下所示的预期输出（以 IMX219 为例）：

```
root@am62axx-evm:~# v4l2-ctl --list-devices
j721e-csi2rx (platform:30102000.ticsi2rx):
        /dev/video3
        /dev/video4
...
        /dev/media0

root@am62axx-evm:~# media-ctl -d /dev/media0 -p | grep imx219
                <- "imx219 4-0010":0 [ENABLED,IMMUTABLE]
- entity 13: imx219 4-0010 (1 pad, 1 link, 0 route)
```

备注

media-ctl 输出中的 **4-0010** 是传感器的 I2C 总线地址，对于不同的 SDK 版本，此值可能会有所不同。

然后，确认摄像头可以配置为特定格式，并且可以使用 GStreamer 流水线捕获原始图像。下面是一个示例，其中假设 4-0010 是 `media-ctl` 命令显示的内容（如上所示）：

```
root@am62axx-evm:~# media-ctl -V '"imx219 4-0010":0 [fmt:SRGGB10_1X10/1920x1080 field:none]'
root@am62axx-evm:~# gst-launch-1.0 -v v4l2src num-buffers=5 device=/dev/video3 io-mode=dmabuf ! \
video/x-bayer, width=1920, height=1080, framerate=30/1, format=rggb10 ! \
multifilesink location="imx219-image-%d.raw"
```

捕获的原始图像采用纯 Bayer 图形阵列格式（IMX219 传感器为 RGGB）并且没有任何标头或压缩。这些原始图像可以通过原始图像查看器或其他工具（如 ffmpeg）查看。在此阶段，原始图像可能会过度曝光或曝光不足，因为传感器中的默认曝光时间和增益不一定与捕获这些图像的照明环境匹配。

6.2 使用初始 VPAC 配置启用摄像头流式传输

确认摄像头和驱动程序工作正常且用于 VPAC 的 GStreamer 插件支持新传感器后，此步骤是使用初始 VPAC 配置运行实时流式传输。此步骤的目的是确保用于 VPAC 的 GStreamer 插件正常工作。实时流式传输将简化后续步骤，例如调整摄像头安装位置。

初始 DCC 二进制文件由 Python 脚本自动生成，以便针对传感器分辨率和颜色格式正确设置 VPAC 和自动曝光 (AE) 程序。TI 的参考成像软件提供了生成初始 VPAC 配置（或 DCC 配置文件）所需的工具，如节 6.2.1 至节 6.2.3 所述。

6.2.1 生成配置文件

转到 TI 的参考成像软件 `imaging/tools/default_DCC_profile_gen/configs`，并创建默认摄像头属性的配置文件。此文件夹中的现有配置文件可以作为参考。

在此配置文件中指定以下参数：

- **摄像头传感器信息**
  - SENSOR_ID：传感器的 DCC ID（**这必须与 `tiovx_sensor_module.c` 中的硬编码“dccID”值匹配**，如更新用于 VISS 的 GStreamer 插件中所述）
  - SENSOR_NAME：传感器的名称（仅供参考，该工具并不使用）
  - SENSOR_DCC_NAME：对应的 DCC 名称（仅供参考，该工具并不使用）
- **XML 输出文件夹**
  - PRJ_DIR：用于存储所生成 .xml 文件的文件夹。此文件夹必须位于 imaging/sensor_srv/src 下，例如，../../../sensor_drv/src/<传感器名称>。
- **原始图像格式信息**
  - SENSOR_WIDTH：传感器图像宽度
  - SENSOR_HEIGHT：传感器图像高度
  - COLOR_PATTERN：Bayer 图形，其中 0=RGGB，1=GRBG，2=GBRG，3=BGGR，4=MONO，10=RGGI，11=GRIG，12=BGGI，13=GBIG，14=GIRG，15=IGGR，16=GIBG，17=IGGB
  - WDR_MODE：传感器模式。0=线性，1=WDR
  - BIT_DEPTH：传感器像素位深度
  - WDR_BIT_DEPTH：解析后 WDR 原始传感器图像位深度，通常为 20 或 24
  - WDR_KNEE_X 和 WDR_KNEE_Y：WDR 解析拐点（必须用逗号分隔并且两者之间没有空格）
  - BLACK_PRE：解析前要消减的传感器黑电平
  - BLACK_POST：解析后要消减的传感器黑电平
  - GAMMA_PRE：将 20 位和 24 位 WDR 原始图像压缩为 16 位 ISP 内部位宽度的伽马值。24 位 WDR 传感器通常约为 50 (0.5)，20 位传感器通常约为 70 (0.7)
  - H3A_INPUT_LSB：H3A 输入位范围的 LSB 位置（从 bit-H3A_INPUT_LSB 到 bit-H3A_INPUT_LSB+9）

备注

上述配置信息必须与实际的传感器格式匹配。

下面是 IMX219 的配置示例。不使用与 WDR 相关的参数，但必须包含这些参数，脚本才能运行。

```
SENSOR_ID 219
SENSOR_NAME IMX219
SENSOR_DCC_NAME SENSOR_IMX219_RPI
PRJ_DIR ../../../sensor_drv/src/imx219_output
SENSOR_WIDTH 1920
SENSOR_HEIGHT 1080

COLOR_PATTERN 0
WDR_MODE 0

BIT_DEPTH 10

WDR_BIT_DEPTH 20

WDR_KNEE_X 0,512
WDR_KNEE_Y 0,2048

BLACK_PRE  0
BLACK_POST  0

GAMMA_PRE 70

H3A_INPUT_LSB 0
```

创建配置文件后，运行 imaging/tools/default_DCC_profile_gen/scripts 中的 ctt_def_xml_gen.py Python 脚本，如下所示：

```
imaging/tools/default_DCC_profile_gen/scripts$ python ctt_def_xml_gen.py ../configs/<configuration file>
```

生成的 xml 文件位于 $PRJ_DIR/dcc_xml 文件夹中。xml 文件夹中还会生成一个脚本文件 generate_dcc.sh。例如，以下是 IMX219（线性模式下）的 xml 文件夹内容：

```
$PRJ_DIR/dcc_xmls/linear$ ls -l generate_dcc.sh imx219_awb_alg_ti3_tuning.xml imx219_cfa_dcc.xml
imx219_h3a_aewb_dcc.xml imx219_h3a_mux_luts_dcc.xml imx219_linear_decompand_dcc.xml
imx219_mesh_ldc_dcc.xml imx219_rgb2rgb_dcc.xml imx219_viss_blc.xml imx219_viss_nsf4.xml
```

6.2.2 生成 DCC 二进制文件

转到包含上一步所生成 .xml 文件的文件夹（线性模式或 wdr 模式），然后运行该文件夹中的 generate_dcc.sh 脚本，如下所示：

```
$PRJ_DIR/dcc_xmls/linear$ chmod +x ./generate_dcc.sh
$PRJ_DIR/dcc_xmls/linear$ ./generate_dcc.sh
```

此脚本会在 PRJ_DIR 指定路径下的 dcc_bins 文件夹中生成默认配置的 DCC 二进制文件。

```
$PRJ_DIR/dcc_bins$ls-l
dcc_2a.bin
dcc_ldc.bin
dcc_viss.bin
```

6.2.3 使用初始配置流式传输视频

将上一步生成的 DCC 二进制文件复制到 AM62A SK EVM 上文件系统的 /opt/imaging/<camera> 文件夹中。例如，IMX219 在 SDK 中具有以下预构建的二进制文件：

```
root@am62axx-evm:~# ls /opt/imaging/imx219/linear
dcc_2a_10b_1920x1080.bin dcc_viss_10b_1920x1080.bin
```

然后使用 *media-ctl* 命令设置与初始配置文件中属性一致的摄像头数据格式，并运行实时流式传输：

```
root@am62axx-evm:~# media-ctl -V '"imx219 4-0010":0 [fmt:SRGGB10_1X10/1920x1080 field:none]'

root@am62axx-evm:~# gst-launch-1.0 v4l2src device=/dev/video-imx219-cam0 io-mode=dmabuf-import ! \
video/x-bayer, width=1920, height=1080, framerate=30/1, format=rggb10 ! \
tiovxisp sink_0::device=/dev/v4l-imx219-subdev0 \
sensor-name="SENSOR_SONY_IMX219_RPI" \
dcc-isp-file=/opt/imaging/imx219/dcc_viss_10b.bin \
sink_0::dcc-2a-file=/opt/imaging/imx219/dcc_2a_10b.bin format-msb=9 ! \
video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! \
kmssink driver-name=tidss sync=false
```

即使在使用初始配置时图像质量并未经过全面调优，这也可以验证用于 ISP 的 GStreamer 插件是否可以在新传感器上正常运行。输出视频显示正确的输出颜色（尽管并未调优）和正确的视频曝光。这样一来，也更容易在下一步中调整摄像头安装。

6.3 调整摄像头安装

设置灯箱和色卡。在流式传输过程中，将摄像头对准色卡。调整摄像头位置，确保色卡刚好填满摄像头的视场 (FOV)。

对于特定分辨率，传感器驱动程序可能会在将图像发送到 ISP 之前裁剪图像。例如，当分辨率为 1920 × 1080 时，IMX219 驱动程序会裁剪图像，图像在显示屏上看起来被放大。在这种情况下，使用较小尺寸的色卡可能效果更好。

7 执行基本调优

通常，ISP 调优需要用户首先在受控照明条件下捕获一组原始图像或 YUV 图像（或者两种类型的图像），并使用调优工具调整 ISP 参数和 AWB 校准。本节采用 TI 的 TDA4 和 AM6xA DCC 调优工具来说明 VPAC 调优过程。TI 第三方也可能提供其他成像调优工具，这些工具具有相似功能，甚至更高级的功能和生产级支持。

7.1 启动调优工具并创建工程

在 Microsoft® Windows® 中，找到并运行安装的调优工具可执行文件：<安装文件夹>\bin\DCC.exe。当系统提示输入正确的 VPAC 版本时，选择“VPAC3L (AM62A)”。工具启动后，创建工程并输入原始图像属性。图 7-1 展示了 IMX219 在 1080p 流模式下的示例。

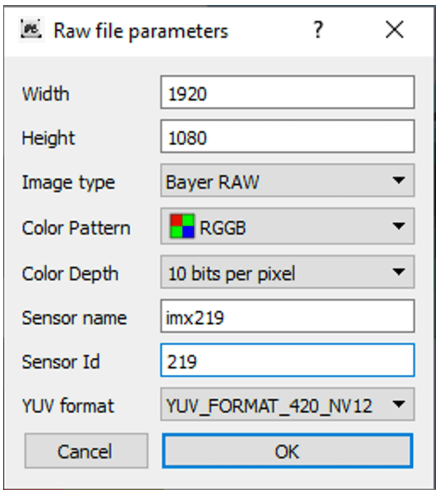


图 7-1. 用于调优的传感器原始图像参数

备注

图像属性必须与用于捕获原始图像的属性匹配，并且传感器 ID 必须与 GStreamer 插件中的硬编码值匹配。

创建新工程后，可以通过调优工具预览原始图像。使用摄像头捕捉验证命令，在良好的光照条件下捕获色卡的原始图。在调优工具的 *Browse* 窗口中选择此图像文件，该图像应会显示在 *Preview* 窗口中。例如，下面是一个 1920 x 1080 且每像素 10 位的 IMX219 原始图像显示在调优工具中。

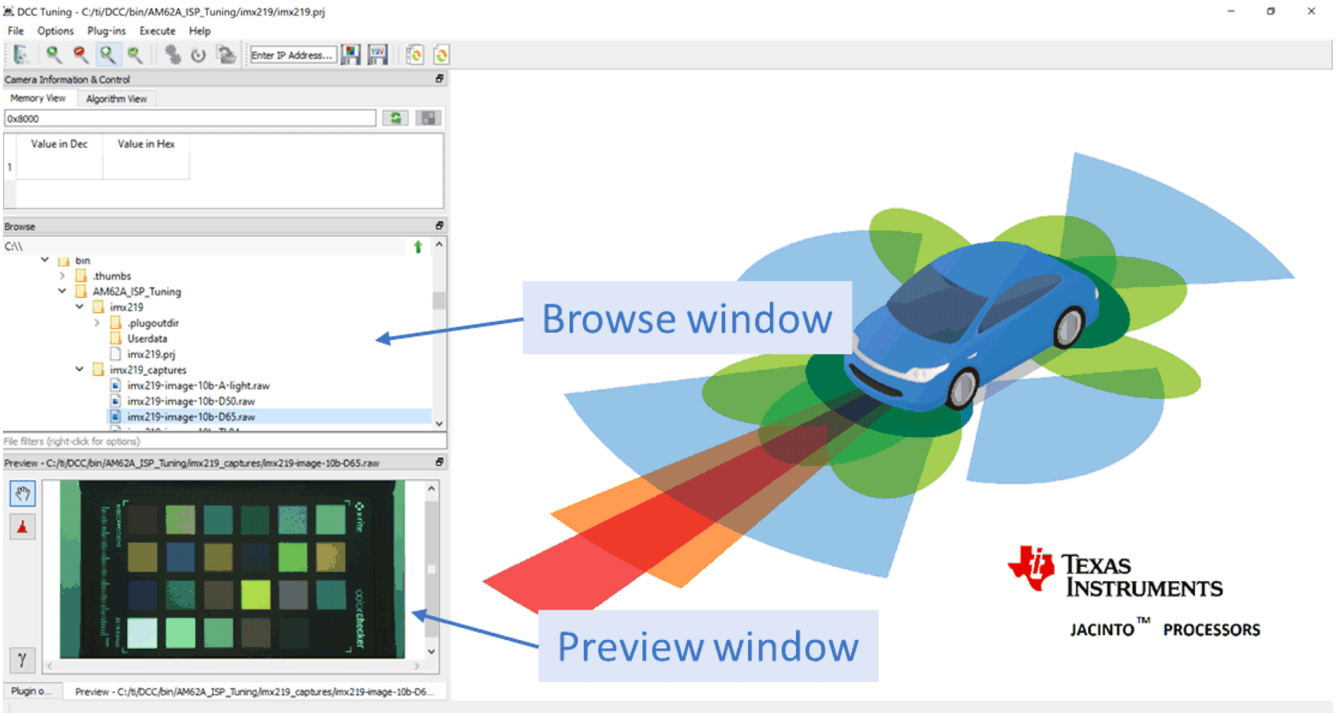


图 7-2. DCC 调优工具的图像预览

备注

在调优工具中只能预览以每像素 10、12 或 16 位捕获的原始图像。如果图像是以每像素 8 位捕获的，可以首先将图像转换为每像素 16 位，然后再在调优工具中预览图像并使用。

现在一切都已准备就绪，可以继续执行调优，如以下各节所述。

7.2 调优顺序

AM6xA ISP (VPAC) 由多个功能块组成。原始图像由这些功能块逐个处理。调优工具允许对独立组中的 ISP 块进行调优，每个组包含一个或多个 ISP 块。调优组称为插件，如调优工具 *Plug-ins* 菜单中所示：

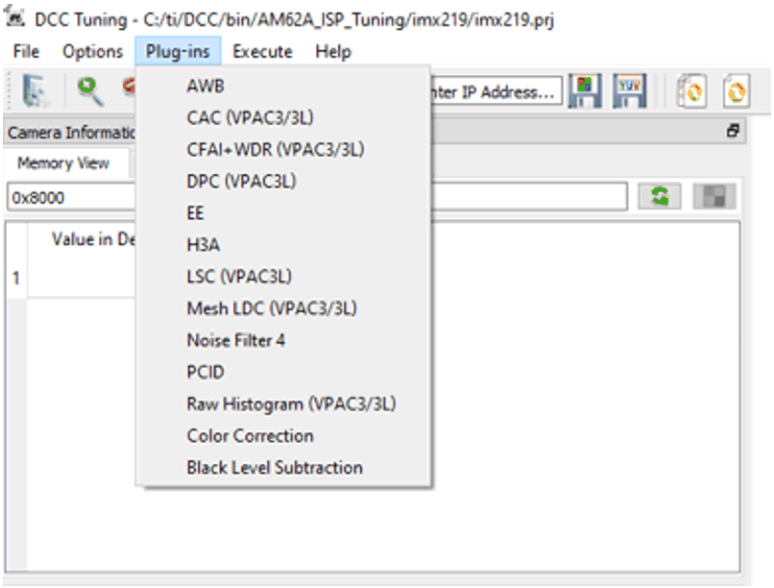


图 7-3. DCC 调优工具插件

备注

每个插件都有调优指南，该指南可从“Help”→“Documentation”下拉菜单中访问。在调优期间，请参考这些指南。

通常，在处理原始图像时，须按照与 ISP 块相同的顺序对这些插件进行调优。以下是推荐的调优顺序：

1. 黑电平消减 (BLC)：数据台阶电平（传感器黑电平）通常由传感器驱动程序指定，可在此处计算以进行验证。
2. H3A：硬件 3A（自动曝光、自动对焦、自动白平衡）统计信息
3. PCID：图形转换和红外去马赛克（仅适用于 RGB+IR 传感器）
4. AWB：自动白平衡
5. 颜色校正
6. EE：边缘增强
7. 噪声滤波器 4
8. 网格 LDC：镜头失真校正
9. CFAI + WDR：色彩滤波阵列插值 + 宽动态范围
10. LSC：镜头阴影校正

调整仅 RGB 传感器和 RGB-IR 传感器时略有不同：

- 对于仅 RGB 传感器：BLC、H3A、AWB、CCM、NSF4、EE 等等。
- 对于 RGB-IR 传感器：H3A、PCID、AWB、CCM、NSF4、EE 等等。

在调整每个插件后，可以生成一组用于 VPAC 配置的新 XML 文件。这些新的 XML 文件可以替换从初始配置生成的文件，以逐步提高图像质量。

本应用手册使用 OX05B1S 传感器演示 PCID 的调优，使用 IMX219 传感器演示其他插件的调优，包括黑电平、AWB、颜色校正等。有关调优的更多详细信息，请参阅调优工具 *Help* 菜单中的插件指南。TI 第三方提供的其他版本 ISP 调优工具遵循大致相同的过程。

7.3 黑电平消减

黑电平消减 (BLC) 插件调优只需要对仅 RGB 传感器执行。对于 RGB-IR 传感器，由于 PCID 执行了红外消减，因此，此插件无需调优。

对于包括 IMX219 在内的线性传感器，先从原始图像像素中消减黑电平或台阶电平，然后在 ISP 中应用任何增益（例如白平衡增益）。尽管台阶电平值在传感器驱动程序中进行了编码，但在传感器驱动程序支持的每个传感器工作模式下测量其实际值。例如，IMX219 摄像头在 10 位模式下（如下图所示）测得的黑电平约为 63，在 8 位模式下测得的黑电平约为 16。

按照以下步骤为目标传感器调优黑电平消减：

1. 完全盖住摄像头镜头并捕捉黑色原始图像。
2. 从 *Plug-ins* 下拉菜单中选择“Black Level Subtraction”。
3. 在 *Browse* 窗口中选择相应原始图像，该图像应当在 *Preview* 窗口中显示为黑色。
4. 在 *RAW file* 窗口中提供原始图像，然后点击 *Process Plugin*，如下所示。
5. 测得的黑电平将显示在右上方窗口的 *Advanced params* 选项卡中。

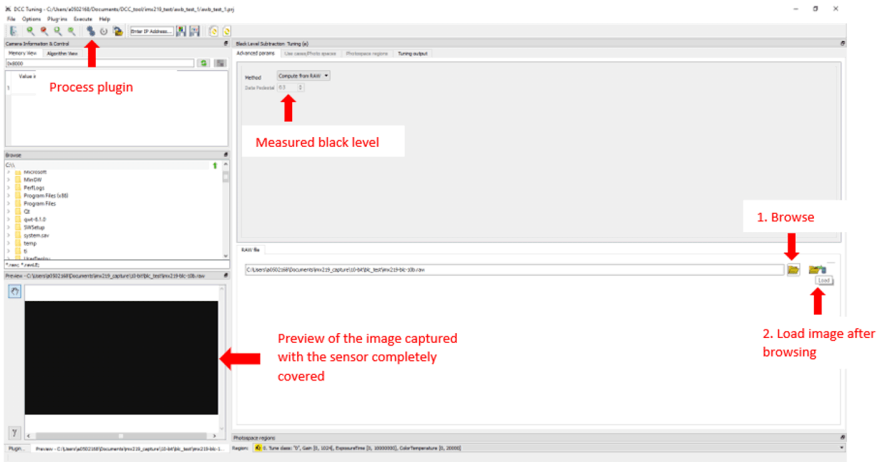


图 7-4. 黑电平消减调优

对于 WDR 传感器，黑电平消减通常会与 WDR 解析和重新压缩相结合，以生成单个查找表 (LUT)。这可以通过调优工具中的 *CFA + WDR* 插件来实现。对于 WDR 传感器，请参阅插件的用户指南以了解更多详细信息。

完成插件的调优后，点击 *导出 DCC 配置文件二进制文件* 按钮以生成该插件的输出 XML 文件。XML 文件位于在节 7.1 中创建的项目文件夹下的 `.plugoutdir\XML` 文件夹中。对于黑电平消减，只有一个输出 XML 文件：imx219_viss_blc.xml。替换从初始配置生成的相同 XML 文件。然后重新运行 Python 脚本以生成新的 DCC 二进制文件，如生成 DCC 二进制文件中所述。在下一步中使用新生成的 DCC 二进制文件来提高流式传输质量。按照后续小节中的说明调优每个插件后完成此操作。

要在调优每个插件后查看图像质量的改善情况，请使用新生成的 DCC 二进制文件捕获 ISP 处理的静态图像。例如，在调优 IMX219 的黑电平消减插件后，将以下 GStreamer 流水线与新的二进制文件配合使用：

```
gst-launch-1.0 -v v4l2src num-buffers=5 device=/dev/video3 io-mode=dmabuf-import ! \
video/x-bayer, width=1920, height=1080, framerate=30/1, format=rggb10 ! \
tiovxisp sink_0::device=/dev/v4l-subdev2 \
sensor-name="SENSOR_SONY_IMX219_RPI" \
dcc-isp-file=/opt/imaging/imx219/dcc_viss_10b.bin \
sink_0::dcc-2a-file=/opt/imaging/imx219/dcc_2a_10b.bin format-msb=9 ! \
video/x-raw, format=NV12, width=1920, height=1080, framerate=30/1 ! \
jpegenc ! multifilesink location="imx219-image-%d.jpg"
```

图 7-5 展示了在黑电平消减调优（使用初始配置）之前捕获的图像示例，图 7-6 则显示了黑电平消减调优之后捕获的图像（比较黑色外观）。

图 7-5. 黑电平消减前的图像

图 7-6. 黑电平消减后的图像

7.4 硬件 3A (H3A)

H3A 是一个硬件 IP 块，用于为自动曝光 (AE)、自动白平衡 (AWB) 和自动对焦 (AF) 算法收集图像统计数据。对于 IMX219 和 OX05B1S 等定焦摄像头，仅 AE 和 AWB 是相关的。生成初始配置时所用的 `ctt_def_xml_gen.py` 脚本已经为 AE 和 AWB 算法正确配置了 H3A。因此，对于定焦摄像头，无需执行其他 H3A 调优步骤。

备注

该 Python 脚本将 BIT_DEPTH 用作输入参数，以便可以正确对 H3A、AE 和 AWB 进行编程。因此，必须正确设置此参数。

该 Python 脚本通过以下方式配置传感器或线性传感器的 H3A 数据流：

1. 默认情况下假设黑电平为 0（在上面的*黑电平消减*步骤中，必须测量并更新实际黑电平）
2. 在给定传感器位深度的情况下，将输入像素移入 16 位 ISP 内部格式的 MSB
3. 将线性传感器像素的（多达）前 10 位发送到 H3A
4. 根据给定的传感器分辨率配置 H3A
5. 为 AE、AWB 算法提供相同的 H3A 配置，以便 AE、AWB 能够正常工作

如果需要微调 H3A 设置，请遵循 SoC 的 TRM 和调优工具中的用户指南。

7.5 PCID

对于包括 OX05B1S 在内的 RGB-IR 传感器，首先调整 PCID 插件以生成输出 16 位 Bayer 原始图像，以用于校准 AWB、CCM、NSF4 等。

要调优 PCID，需要在不同的照明条件下捕获原始图像。捕获不同照明条件下的原始图像列出了这些照明条件的具体细节。下面是对此插件进行调优的总结。有关更多详细信息，请参阅“Help”菜单中的 PCID 插件指南。

- 加载一个输入 16 位原始图像
- 输入合适的参数值
  - 选择“Input Format”和“Output Format”时，请验证“Output Format”是否与配置的流水线匹配。例如，OX05B1S 的颜色格式为 BGGR。“Input Format”设置为“2:BGGIr”，"Output Format”设置为“1: Interpolate R at IR positions and B at R locations”，并且 GStreamer 流水线必须配置为 BGGR。
- *处理* 插件以生成具有 16 位 IR 消减的输出 Bayer 图形
- 对所有照明条件重复此过程
- *导出 DCC 配置文件二进制文件* 以生成 XML 和二进制文件

备注

调优 RGB-IR 传感器时，请将黑电平设置为 0，同时对需要黑电平值的插件进行调优，因为已在 IR 消减中移除了黑电平。

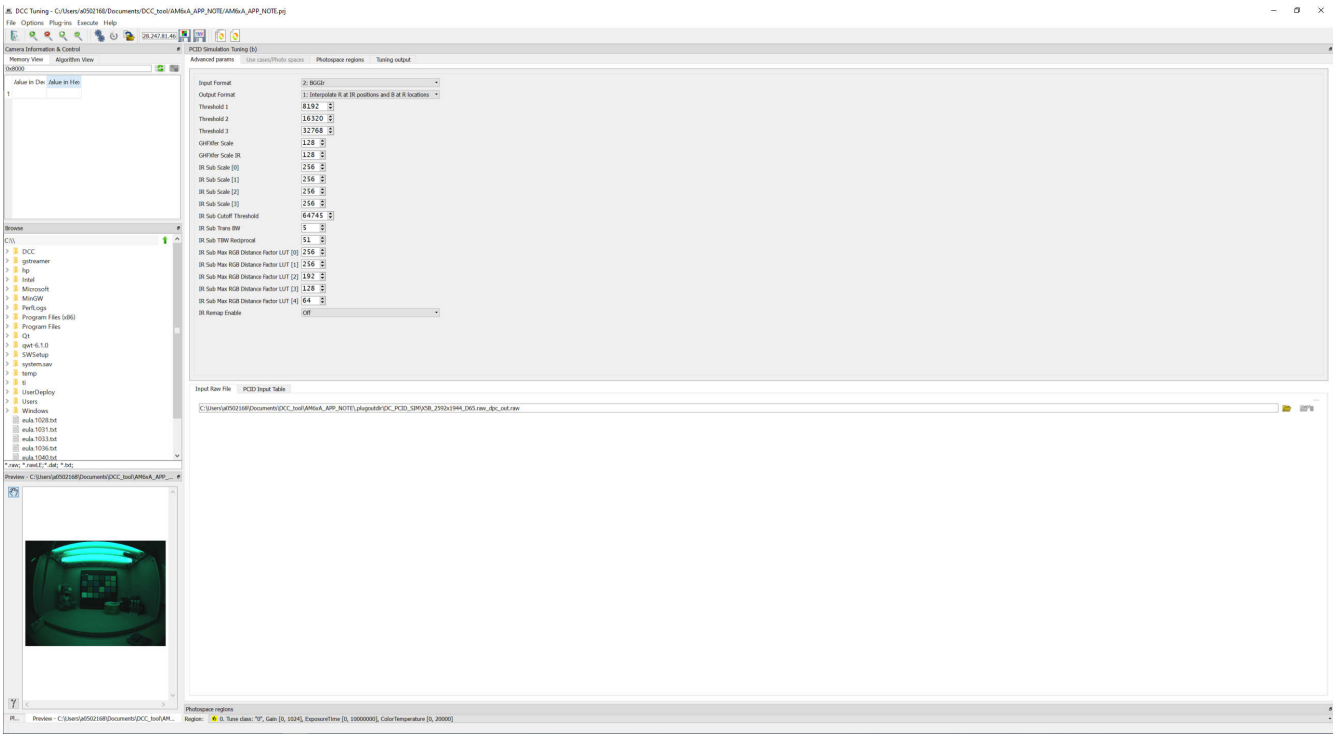


图 7-7. PCID 调优

使用 OX05B1S 的 GStreamer 流水线示例如下：

```
gst-launch-1.0 -v v4l2src device=/dev/video4 io-mode=dmabuf-import ! \ video/x-bayer, width=2592,
height=1944, framerate=30/1, format=bggi10 ! queue ! \ tiovxisp sink_0::pool-size=4
sink_0::device=/dev/v4l-subdev2 sensor-name="SENSOR_OX05B1S" \ dcc-isp-file=/opt/imaging/ox05b1s/
linear/dcc_viss.bin sink_0::dcc-2a-file=/opt/imaging/ox05b1s/linear/dcc_2a.bin format-msb=9 !
queue ! \ tiovxldc dcc-file=/opt/imaging/ox05b1s/linear/dcc_ldc.bin sensor-name=SENSOR_OX05B1S ! \
video/x-raw, format=NV12, width=2592, height=1944 ! queue ! \ tiovxmultiscaler src_0::pool-size=4
target=1 ! video/x-raw, format=NV12, width=1920, height=1080 ! queue ! \ tiperfoverlay
location=perf_logs num-dumps=10 ! queue ! kmssink driver-name=tidss force-modesetting=true
sync=false
```

图 7-8 和图 7-9 展示了 ISP 完全未处理的捕获原始图像示例以及 Bayer 图形格式的 PCID 调优输出图像。

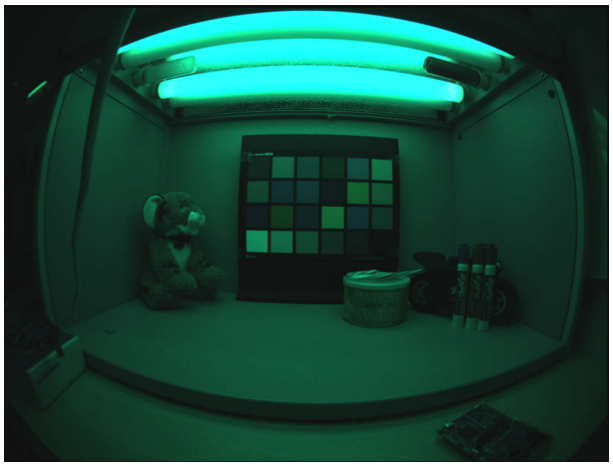


图 7-8. PCID 的原始图像输入

图 7-9. Bayer 图形格式的 PCID 输出图像

7.6 自动白平衡 (AWB)

7.6.1 捕获不同照明条件下的原始图像

要调优 AWB，需要在不同的照明条件下捕获原始图像。这些图像将在下一步颜色校正调优中再次使用。以下总结说明了如何捕获所需图像（有关详细信息，请参阅 AWB 插件指南）：

- 将照明条件设置为：D65、D50、TL84 和 A 灯（一次一个）
- 将色卡直立置于灯箱中并使其位于摄像头 FOV 的中间
  - 确保棕色色片位于第一行的左上角，黑色色片位于最后一行的右下角
- 针对每种照明条件捕获曝光良好的原始图像
  - 等待自动曝光调整稳定下来（监视器上的实时视频输出足够明亮，且白色色片上没有出现明显饱和或剪裁。此过程通常在几秒钟内完成。）
  - 使用初始配置运行实时视频流式传输
  - 原始图像可以通过以下两种方式之一进行捕获：
    - 停止用于流式传输的 GStreamer 流水线。然后运行新的 GStreamer 流水线以捕获原始图像（传感器曝光设置在 GStreamer 运行之间保持不变）。
    - 如果 AM62A EVM 和 PC 连接到同一以太网，调优工具还可以直接通过以太网从 EVM 捕获原始图像。在 DCC 调优工具的工具栏中，输入 EVM 的 IP 地址并捕获原始图像，如下所示。有关更多详细信息，请参阅调优工具的用户指南。

图 7-10. DCC 调优工具的实时捕获功能

图 7-11 至图 7-14 是所有上述照明条件下曝光良好的捕获示例（请注意由照明颜色引起的颜色差异）：

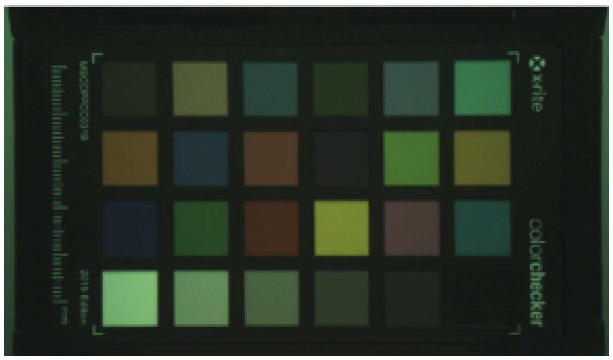


图 7-11. 使用 D50 照明时捕获的色卡图像

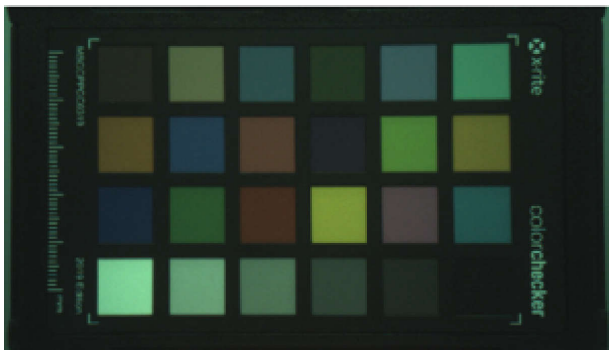


图 7-12. 使用 D65 照明时捕获的色卡图像

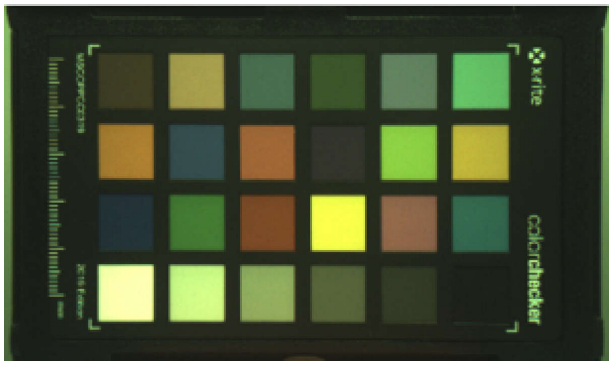


图 7-13. 使用 TL84 照明时捕获的色卡图像

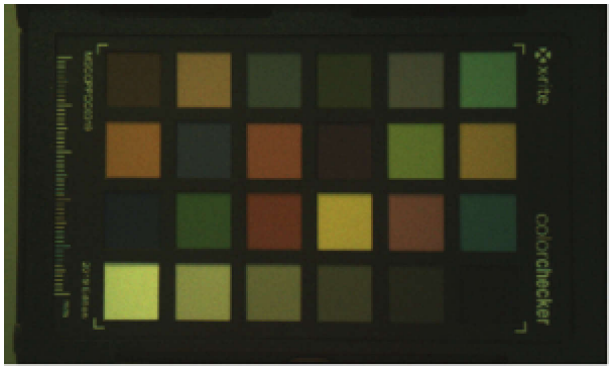


图 7-14. 使用 A 灯时捕获的色卡图像

7.6.2 AWB 调优

捕获原始图像后，从 *Plug-ins* 下拉菜单开始 AWB 调优。在 *Reference files* 选项卡中逐个导入原始图像（有关详细信息，请参阅 AWB 插件指南），然后为每个图像输入参数值：

- 色温：这必须是捕获原始图像时使用的温度。例如，在 D65 照明条件下，该值为 6500。
- 曝光、增益和光圈：这些值并不用于 AWB；因此可以忽略这些值。
- 黑电平：这必须是在黑电平消减插件中测量的台阶电平值。在此示例中，IMX219 在 10 位模式下测得的黑电平为 63。

选择色卡的边角时，应特别注意：

- 从左上角开始，按顺时针点击色卡的四个角。
- 选择四个角后，该工具会自动识别 24 个色片并显示每个色片的选择，如下所示。

图 7-15 至图 7-16 展示了导入一个原始图像以进行 AWB 调优的示例。

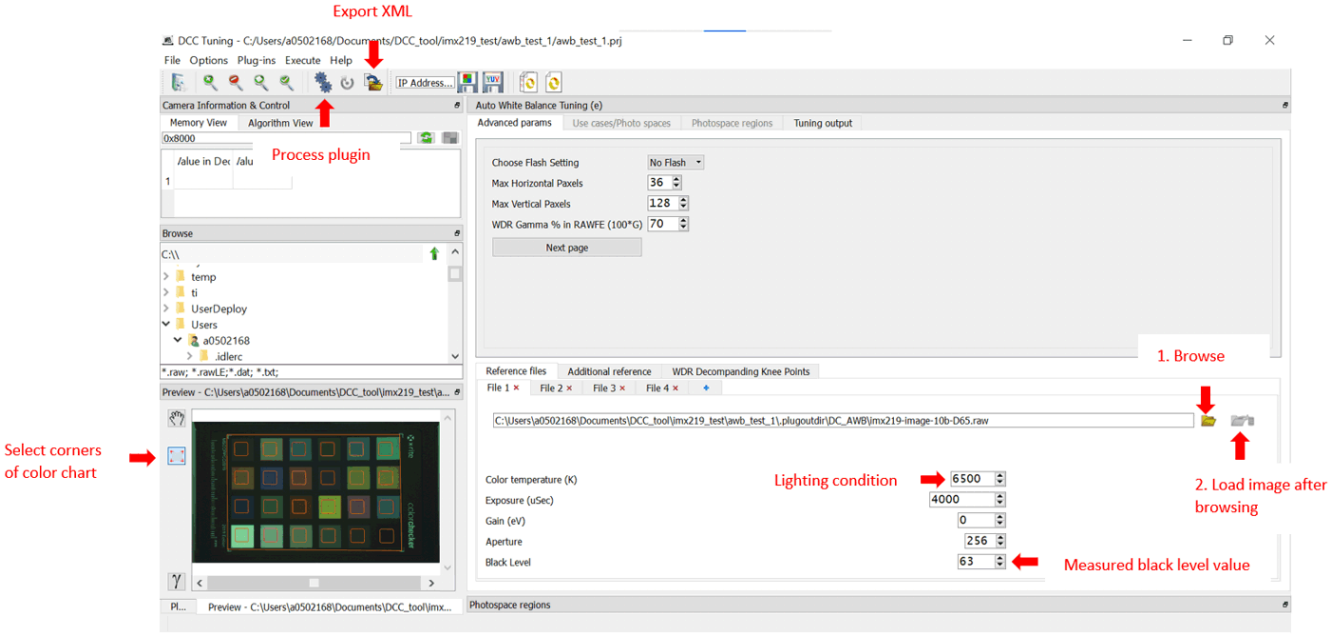


图 7-15. 自动白平衡调优

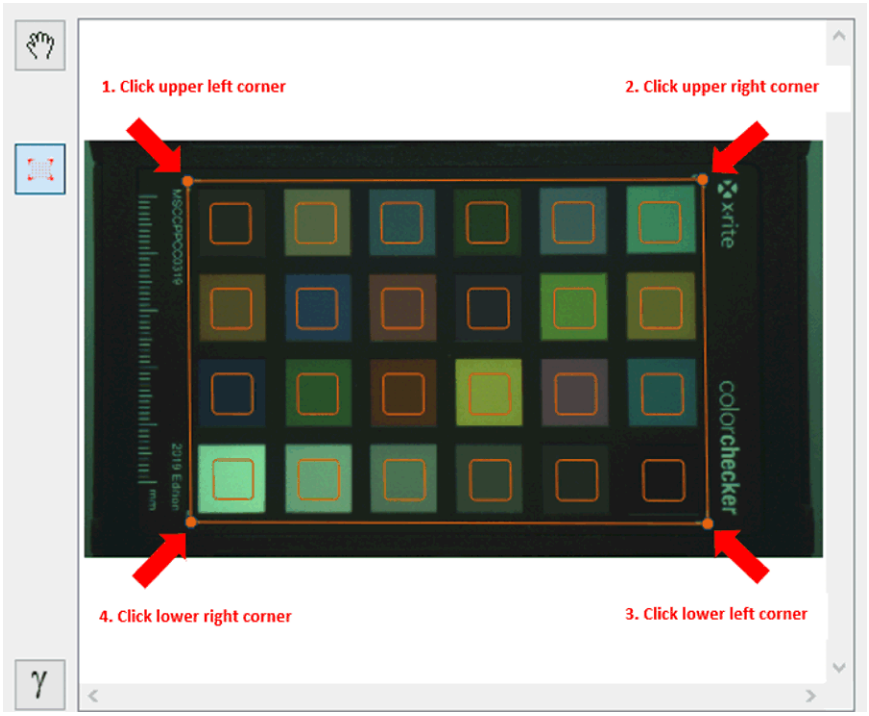


图 7-16. 选择色卡的边角

导入所有原始图像后，按照 AWB 插件指南进行调优。如果调优成功，会显示参考 Cb-Cr 绘图方案。下面是使用上面所示原始图像时的结果图。

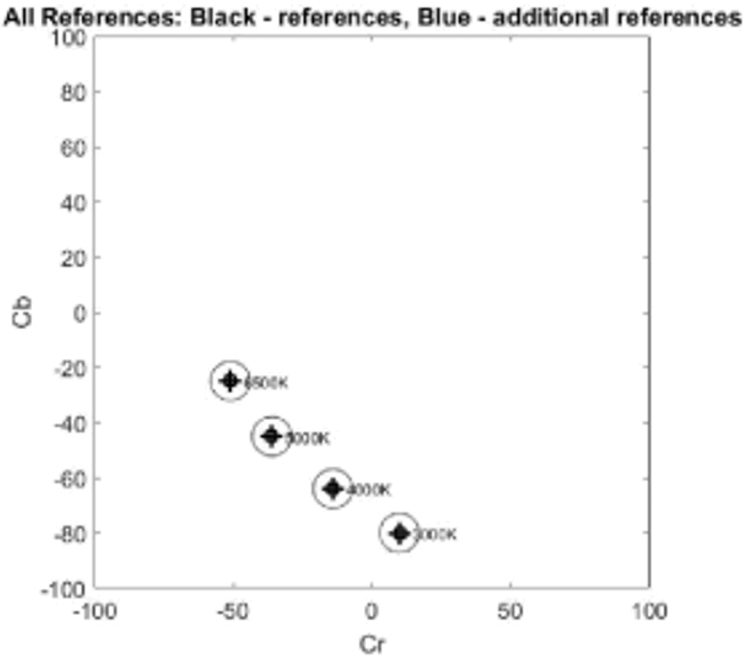


图 7-17. 自动白平衡调优结果

完成调优后，如黑电平消减调优中所述，生成新的输出 XML 文件和新的二进制文件。然后使用新生成的 DCC 二进制文件进行流式传输和捕获。图 7-18 和图 7-19 展示了在 AWB 调优前后捕获的图像（AWB 调优后，图像中的所有灰阶色片都显示为中性）。

图 7-18. 自动白平衡调优前的图像

图 7-19. 自动白平衡调优后的图像

7.7 颜色校正

为了获得正确的颜色输出，需要对颜色校正插件进行调优。可使用捕捉用来进行 AWB 调优的原始图像来此对插件进行调优。下面是此调优步骤的摘要。有关更多详细信息，请参阅 Help 菜单中的颜色校正插件指南。

- 加载输入原始图像和可选的参考图像。调优工具提供了默认参考图像，但用户可以使用任何参考图像。
- 输入参数值并选择色卡的四个角，如 AWB 调优中所做那样。
- 对参考图像重复相同的过程。
- *处理* 插件并点击*导出 DCC 配置文件二进制文件*以生成 XML 和二进制文件，如图 7-20 和图 7-21 所示。

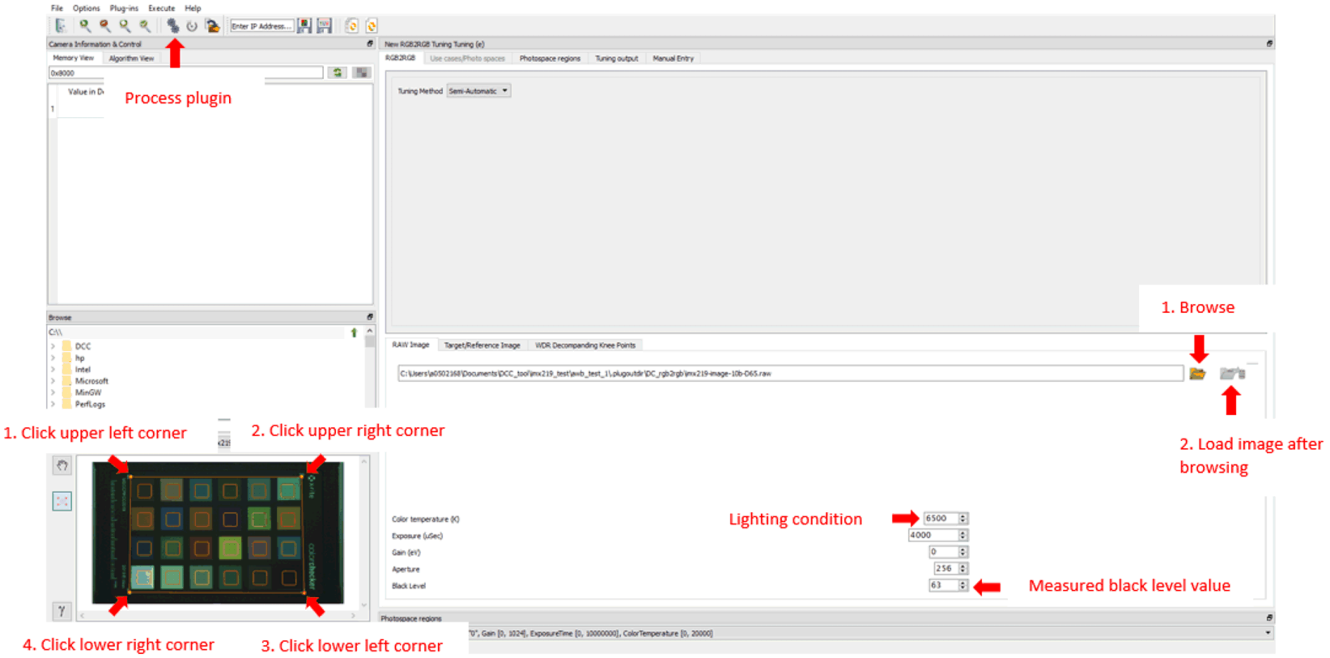


图 7-20. 颜色校正调优

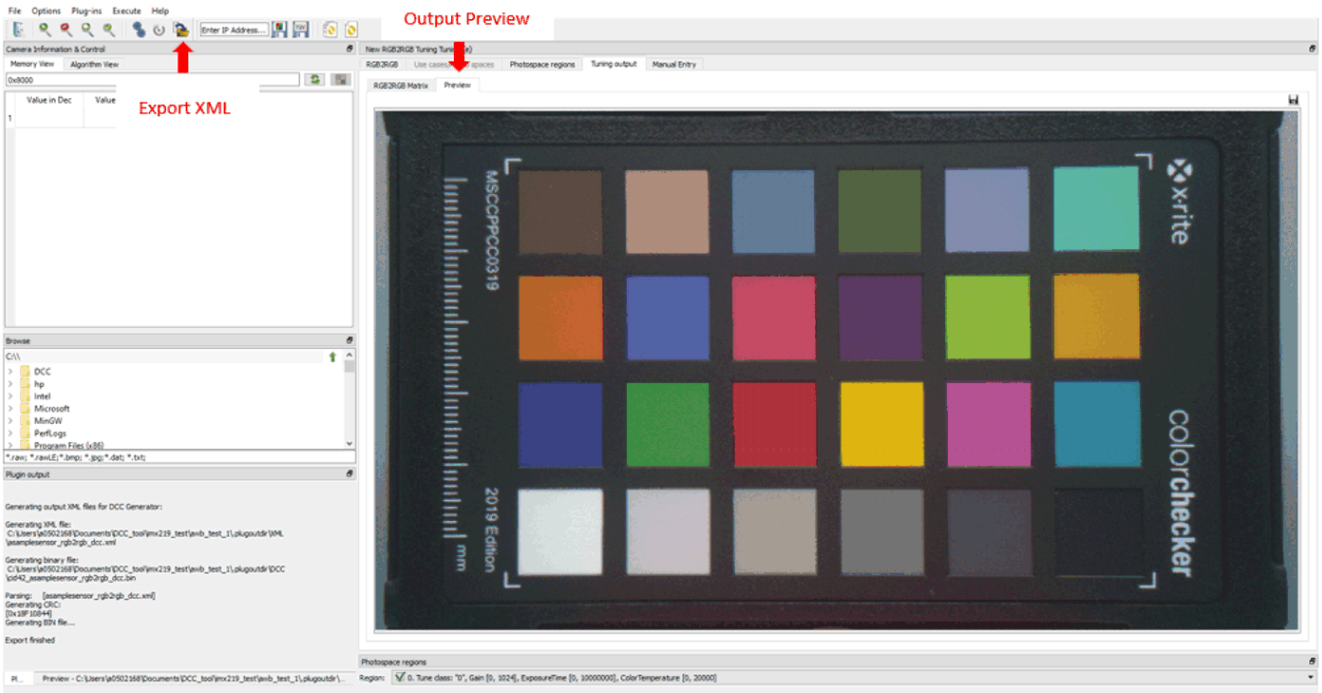


图 7-21. 颜色校正调优输出

如 AWB 调优中所做那样，使用新生成的 DCC 二进制文件捕获图像。图 7-22 和图 7-23 展示了颜色校正调优前后的图像（请注意颜色校正调优后图像中校正的颜色）。

图 7-22. 颜色校正调优前的图像

图 7-23. 颜色校正调优后的图像

8 执行微调

在正确完成黑电平消减、硬件 3A、PCID（适用于 RGB-IR 传感器）、自动白平衡和颜色校正调优后，ISP 可在各种灯箱照明条件下实现最佳图像质量的 70%~80%。为了进一步改善图像质量，还可以对其他插件进行额外调优，其中包括：

- 边缘增强 (EE)，可提高输出图像的清晰度
- 噪声滤波器 4 (NSF4)，用于抑制黑暗条件下的噪点
- Mesh 镜头失真校正 (LDC)，用于消除镜头失真
- 色彩滤波阵列插值 + 宽动态范围 (CFAI + WDR)，用于去马赛克和 WDR 传感器
- 镜头阴影校正 (LSC)，用于消除镜头阴影

8.1 边缘增强 (EE)

EE 模块参数和 LUT 由插件根据用户输入的阈值和斜率自动生成。

备注

为了使 EE 模块在 AM6xA 上生效，请启用 *ee_mode*，如下所示：

- 打开 `/opt/edgeai-tiovx-modules/src/tiovx_viss_module.c` 并找到 *ee_mode*
- 通过设置 *ee_mode* 来启用边沿增强，如下所示：
  - obj->params.fcp[0].ee_mode = TIVX_VPAC_VISS_EE_MODE_Y8;

- 在调优颜色校正后捕捉 YUV 图像。将此图像加载到调优工具的 EE 插件中。
- 将 Tuning Method 设置为 Semi-Automatic
- *处理* 插件以切换到 EE 调优 GUI。图 8-1 展示了 EE 调优 GUI 和每个控件的详细信息
- 使用 *Enable* 部分中的 *On* 或 *Off* 单选按钮启用或禁用 *Edge Enhancer*
- 通过将 *Gain* 设置为 '0' 来禁用 *Edge Sharpener*
- 要看到 EE 生效，请在 *Semi-Automatic* 调优 GUI 中选择其中一个 *Image Sources*，然后选择 *Apply* 按钮。

图 8-2 和图 8-3 分别显示了 EE 调优*之前* 和*之后* 的调优工具输出示例。

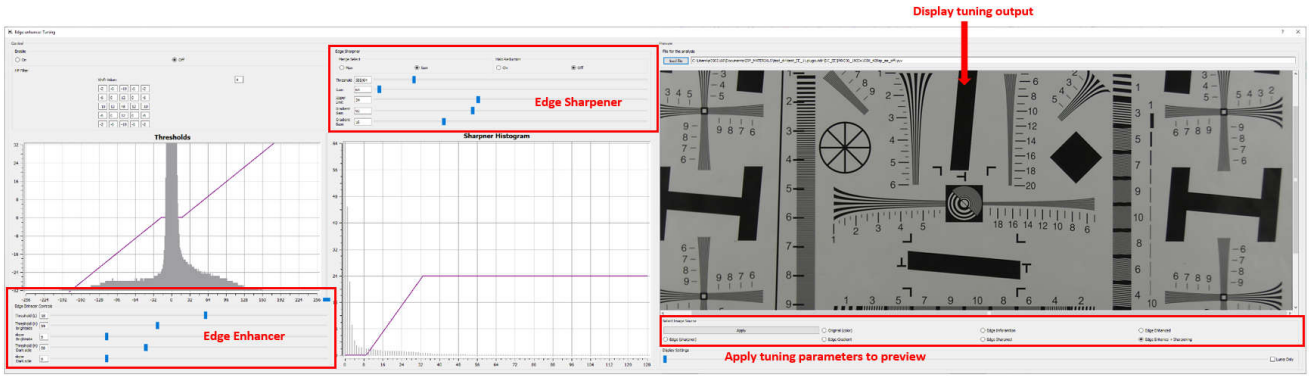


图 8-1. EE Semi-Automatic 调优 GUI

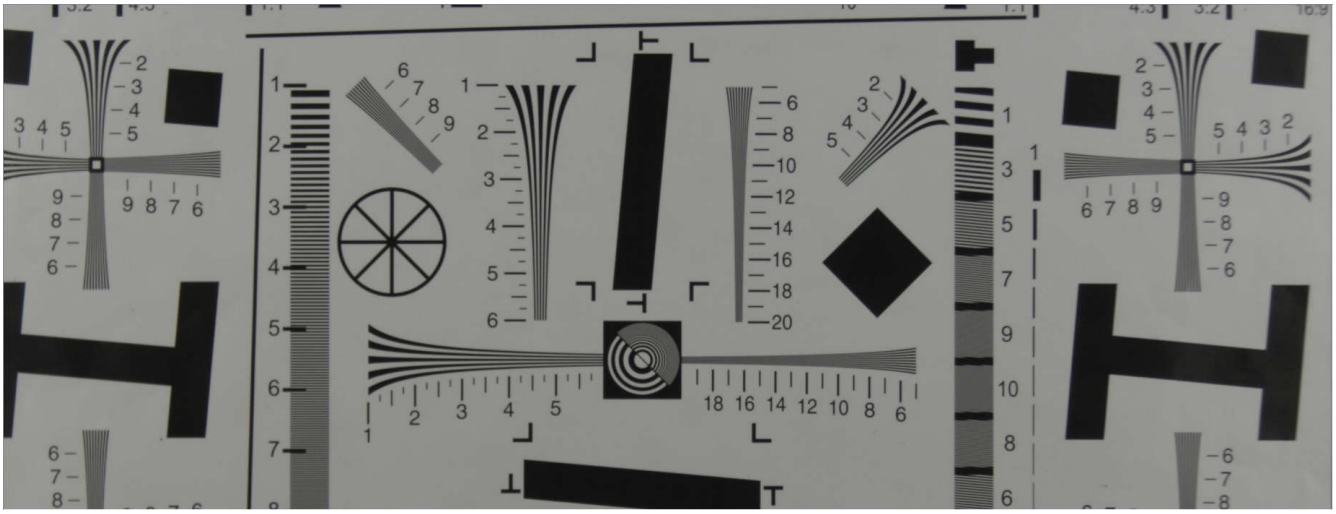


图 8-2. EE 调优之前的 YUV 图像输入

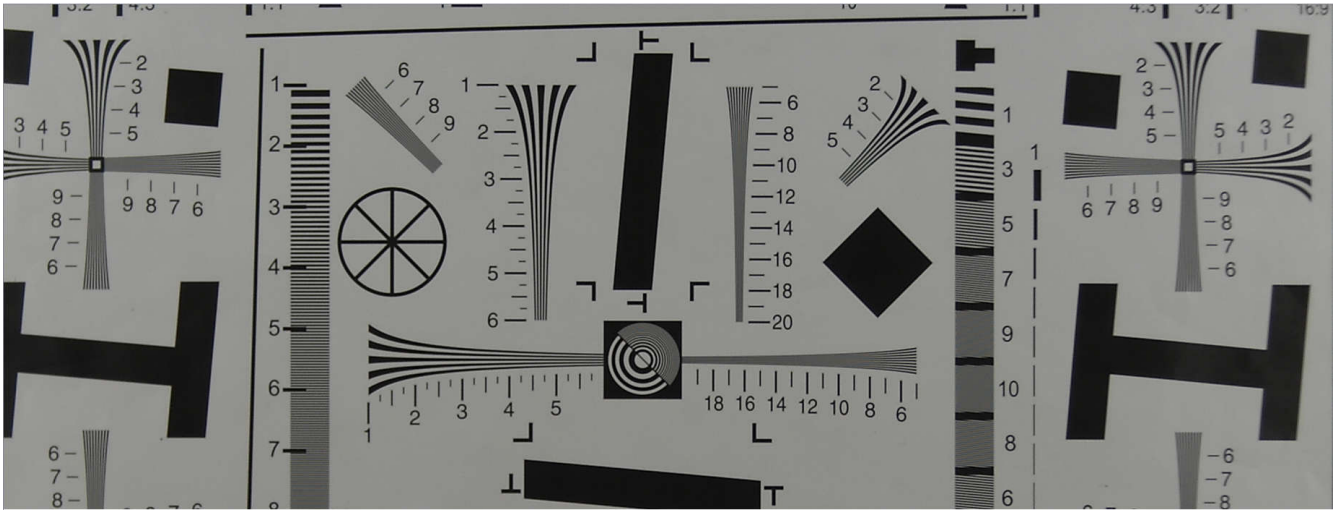


图 8-3. EE 调优之后的 YUV 图像输出

8.2 噪声滤波器 4 (NSF4)

噪声滤波可减少或消除图像中的噪声，而不会使细节或其他重要图像特征变得模糊。下面是对此插件进行调优的总结。有关更多详细信息，请参阅 *Help* 菜单中的 NSF4 插件指南。

- 加载原始图像（例如，用于 AWB 调优的原始图像之一）
- 将 Tuning Method 设置为 Semi-Automatic
- 选择色卡的四个角（从左上角开始，顺时针继续），如 AWB 调优中所做那样
- 加载 RAWFE 解析 LUT。使用 python 脚本来生成配置文件时，可以在 XML 输出文件夹中找到 `lut_rawfe_pwl_vshort.txt`
- 在 12 位 LUT 条件下，将 RAWFE 解析 LUT 设置为 On
- *处理* 插件以切换到 NSF4 调优 GUI。图 8-4 展示了 NSF4 调优 GUI 和每个控件的详细信息
- 调整 *Display Control* 中的参数
  - *X/Y range*：放大绘图区域
- 噪声图 *NF 调优图* 以小圆圈显示噪声样本。蓝线曲线是根据在 *THR 调优* 中设置的当前 X 和 Y THR 值绘制的。通过移动滑动条或键入数字手动调整 THR (Y) 值，以使噪声样本圆圈和蓝色曲线匹配。
- 不允许 *Overall Strength* 大于 1.00。如果线条变得模糊，请降低总体强度，因为噪声滤波过强。
- 若要查看 NFS4 是否生效，请选择 *Run NF*，一旦启用 *Output Image* 选项，便会显示结果。
- 重复前面的步骤，直至调整后的输出符合您的喜好。

- 如果对结果感到满意，点击 *Finish* 按钮保存调优 *导出 DCC 配置文件二进制文件*，以生成输出 XML 和二进制文件

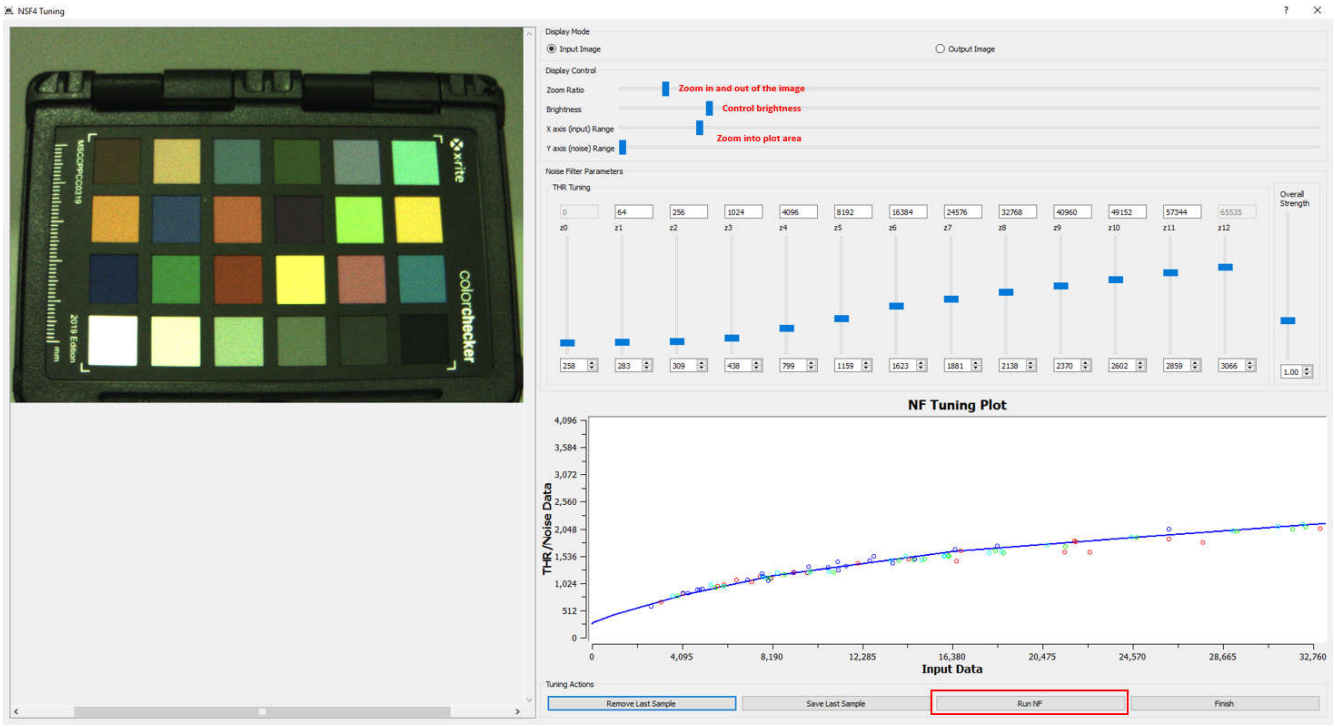


图 8-4. NSF4 调优 GUI

可以执行额外调优，但在此版本中不对此进行介绍：

- Mesh LDC：请参阅此常见问题解答
- CFAI + WDR
- LSC

9 实时调优

实时调优功能允许调优工具在运行时通过以太网接口与目标系统交互。

完成以下操作以针对 AM6xA 启用实时调优：

1. 在 EVM 上的目标文件系统中，导航至 `/opt/edgeai-tiovx-modules/CMakeLists.txt`
2. 修改 **ENABLE_DCC_TOOL** *Enable DCC Tuning Tool Support OFF → ON*
3. 通过运行 `/opt/edgeai-gst-apps/scripts/install_tiovx_modules.sh` 脚本重新编译并重新安装模块

9.1 要求

- 必须通过以太网将 AM6xA EVM 连接到网络。
- 必须将 PC 和 EVM 连接到同一个网络。PC 必须能够 ping EVM。
  - VPN 连接可能会干扰此连接。在尝试实时调优之前，请断开 PC 与 VPN 的连接。
  - 在 EVM Linux 控制台上，使用 *ifconfig* 找到 EVM IP 地址并将其输入到下图所示的框中。

图 9-1. EVM IP 地址

  - 运行 GStreamer 流水线，以便 (1) 传感器将原始图像流式传输到 AM6xA；(2) GStreamer 中的 AE、AWB 和调优服务器也正在运行。

9.2 支持的功能

以下各节列出并说明支持的功能。

9.2.1 RAW 捕捉

RAW 捕捉会保存来自捕捉节点输出的原始图像。此图像是图像传感器的输出，完全未经 J7、AM6xA ISP 处理。宽度、高度和位深度由传感器驱动程序决定。

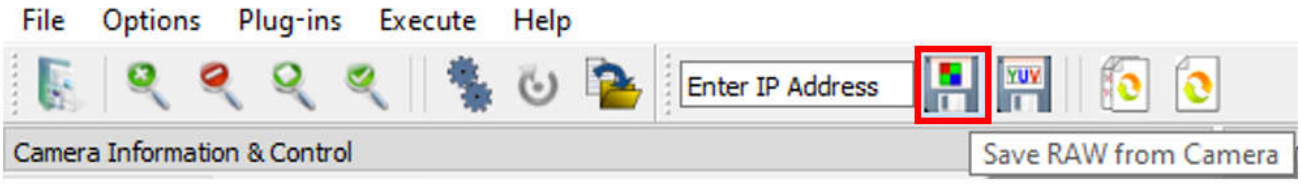


图 9-2. RAW 捕捉

9.2.2 YUV 捕捉

YUV 捕捉可保存来自 VISS 节点输出的 YUV 图像。此图像是 LDC 或标量之前 ISP 的输出。宽度、高度和位深度由 VISS 节点决定。

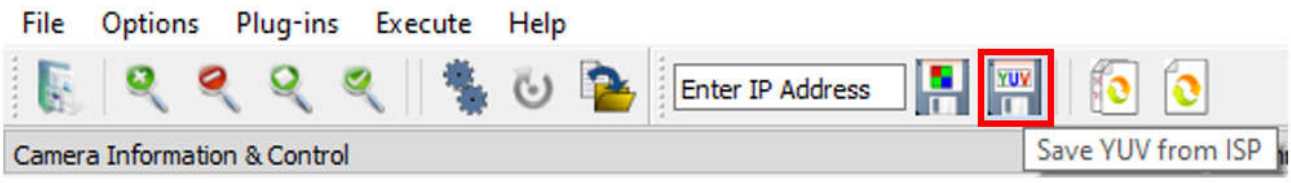


图 9-3. YUV 捕捉

9.2.3 实时 DCC 更新

实时 *DCC 更新* 包括以下功能：

- 将当前在 GUI 工具中打开的插件的 DCC 二进制文件发送到 EVM。在连接到 EVM 的显示器上，可以即时看到对图像质量的影响。
- 新参数立即生效，无需更改任何代码或重新启动应用程序。

图 9-4. DCC 更新

9.2.4 曝光控制

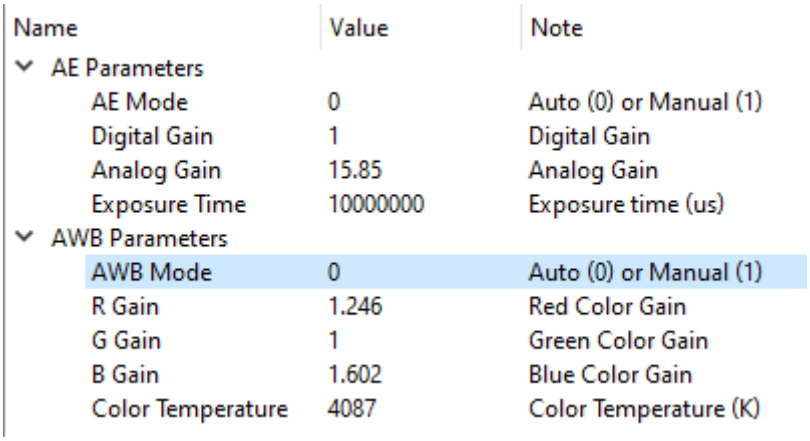
- 在“Algorithm View”选项卡中，读取通过 AE 算法计算所得的曝光和增益
- 通过设置 AE Mode = Manual (1) 绕过 AE
  - 根据传感器驱动程序限制设置手动曝光和增益值（参数在这个小节中设置）。

Name	Value	Note
AE Parameters		
AE Mode	0	Auto (0) or Manual (1)
Digital Gain	1	Digital Gain
Analog Gain	15.85	Analog Gain
Exposure Time	10000000	Exposure time (us)
AWB Parameters		
AWB Mode	0	Auto (0) or Manual (1)
R Gain	1.246	Red Color Gain
G Gain	1	Green Color Gain
B Gain	1.602	Blue Color Gain
Color Temperature	4087	Color Temperature (K)

图 9-5. 用于实时调优的曝光控制

9.2.5 白平衡控制

- 读取通过 AWB 算法计算得出的增益和色温
- 通过设置 AWB Mode = Manual (1) 绕过 AWB
- 手动设置增益和色温，它们受传感器和 VISS 驱动程序限制约束



Name	Value	Note
AE Parameters		
AE Mode	0	Auto (0) or Manual (1)
Digital Gain	1	Digital Gain
Analog Gain	15.85	Analog Gain
Exposure Time	10000000	Exposure time (us)
AWB Parameters		
AWB Mode	0	Auto (0) or Manual (1)
R Gain	1.246	Red Color Gain
G Gain	1	Green Color Gain
B Gain	1.602	Blue Color Gain
Color Temperature	4087	Color Temperature (K)

图 9-6. 用于实时调优的白平衡控制

9.2.6 传感器寄存器读取/写入

传感器寄存器读取/写入功能可以针对图像传感器寄存器执行读取或写入操作。

备注

此功能仅在 IMX219 上受支持。

10 总结

本应用手册介绍了 AM6xA ISP 调优工作流程，在流程中使用了 TI 的 AM62A Processor SDK、成像软件和 DCC 调优工具。使用 IMX219 摄像头描述仅 RGB 调优过程，使用 OX05B1S 摄像头描述特定于 RGB-IR 的调优过程。

11 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision * (March 2023) to Revision A (May 2024) **Page**

- 添加了 2A 算法的曝光设置 7
- 添加了 RGB-IR 传感器的调优指南 14
- 添加了边缘增强的调优指南 24
- 添加了噪声滤波器的调优指南 25
- 添加了实时调优功能的指南 27

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款或 ti.com 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265