

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，以获得出色的图像质量。本报告中提供的调优过程附带了使用 IMX219 传感器和 AM62A 入门套件 EVM 的示例。

内容

|                                |    |
|--------------------------------|----|
| 1 简介.....                      | 2  |
| 2 调优概述.....                    | 2  |
| 3 硬件要求.....                    | 2  |
| 4 软件要求.....                    | 3  |
| 4.1 Processor SDK Linux.....   | 3  |
| 4.2 TI 的参考成像软件.....            | 3  |
| 4.3 ISP 调优工具.....              | 4  |
| 5 传感器软件开发与集成.....              | 4  |
| 5.1 向 SDK 添加传感器驱动程序.....       | 4  |
| 5.2 更新 GStreamer 插件以支持传感器..... | 4  |
| 6 调优过程.....                    | 6  |
| 6.1 验证摄像头捕捉是否能够正常运行.....       | 6  |
| 6.2 使用初始 VPAC 配置启用摄像头流式传输..... | 6  |
| 6.3 调整摄像头安装.....               | 8  |
| 6.4 捕获原始图像并执行基本调优.....         | 8  |
| 6.5 执行微调.....                  | 17 |
| 7 总结.....                      | 17 |

商标

所有商标均为其各自所有者的财产。

## 1 简介

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出色图像质量，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理想 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理想 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 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/video2 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 ! \
kmssink driver-name=tidss sync=false
```

在上面的流水线中，GStreamer 元素 **tiovxisp** 包括 VPAC 硬件和 TI 用于 AE、AWB 和 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/dcc\_viss\_10b.bin**，用于提供调优的 ISP 参数
- **dcc-2a-file=/opt/imaging/imx219/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，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集成到系统中，并且可以捕获原始图像。向 GStreamer 插件添加对该传感器的支持。
4. **生成初始 ISP 配置**：为 VPAC 配置生成初始 DCC 配置文件。此配置可实现最低图像质量的视频流式传输，以便可以执行下一步。
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**：捕获原始图像并执行基本调优，以实现出色图像质量的 70%~80%。将初始 DCC 配置文件替换为调优结束时生成的新配置文件。
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 和 IMX219 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 IMX219 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。根据设置，可能需要不同尺寸的色卡。
- 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](#) 显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
```

此库具有 TI 用于 AE、AWB 和 ISP 控制或动态摄像头配置 (DCC) 的参考成像算法。它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/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/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。请联系当地的 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 传感器软件开发与集成

要使用 AM62A SK EVM 和 Linux SDK 来针对目标传感器进行 ISP 调优，需要开发软件，以便在 SDK 中支持相应传感器。传感器驱动程序实现超出了本文档的讨论范围。以下各节介绍了如何将传感器驱动程序添加到 SDK 中，以及如何修改 GStreamer 插件以支持传感器。

### 5.1 向 SDK 添加传感器驱动程序

假设已经实现了传感器的驱动程序，请访问 [AM62A Academy](#)，了解如何将传感器驱动程序添加到 SDK 中。在 Academy 主页上，转到“Develop Linux on TI EVM”->“Use Device Drivers”->“Use Camera”->“Enable a New CSI-2 Sensor”。

### 5.2 更新 GStreamer 插件以支持传感器

除了向 SDK 添加传感器驱动程序，还需要更新 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。

#### 5.2.1 更新 TIOVX 模块

##### 5.2.1.1 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 添加到函数 `tiovx_init_sensor()` 的列表中。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 流水线中应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)。

##### 5.2.1.2 重新编译模块

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

### 5.2.2 更新 GStreamer 插件

#### 5.2.2.1 源代码更改

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 IMX219 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_SONY_IMX219_RPI\n")
```

然后添加一个函数来配置传感器的自动曝光参数。例如，IMX219 具有以下函数：

```
static int32_t get_imx219_ae_dyn_params (IssAeDynamicParams *p_ae_dynPrms)
{
    int32_t status = -1;
    uint8_t count = 0;
```

```
g_return_val_if_fail (p_ae_dynPrms, status);

p_ae_dynPrms->targetBrightnessRange.min = 40; // min brightness on 0~255 scale
p_ae_dynPrms->targetBrightnessRange.max = 50; // max brightness on 0~255 scale
p_ae_dynPrms->targetBrightness = 45; // target brightness
p_ae_dynPrms->threshold = 5; // difference between target and min/max
p_ae_dynPrms->enableBlc = 0; // not used

p_ae_dynPrms->exposureTimeStepSize = 1; // step size for adjusting exposure time in microseconds
p_ae_dynPrms->exposureTimeRange[count].min = 100; // min exposure time in microseconds
p_ae_dynPrms->exposureTimeRange[count].max = 33333; // max exposure time in microseconds

p_ae_dynPrms->analogGainRange[count].min = 1024; // min analog gain (1024 is for 1.0x gain)
p_ae_dynPrms->analogGainRange[count].max = 8192; // max analog gain (8192 is for 8.0x gain)
p_ae_dynPrms->digitalGainRange[count].min = 256; // digital gain is not used
p_ae_dynPrms->digitalGainRange[count].max = 256; // digital gain is not used
count++;

p_ae_dynPrms->numAeDynParams = count;
status = 0;
return status;
}
```

最后，添加曝光时间和模拟增益映射。例如，IMX219 具有以下映射：

```
if (g_strcmp0 (self->sensor_name, "SENSOR_SONY_IMX219_RPI") == 0)
{
    double multiplier = 0;
    // map AE output exposure time in microseconds to the exposure register value
    *exposure_time_mapped = (1080 * exposure_time / 33);

    // map AE output analog gain in Q10 integer format to the analog gain register value
    multiplier = analog_gain / 1024.0;
    *analog_gain_mapped = 256.0 - 256.0 / multiplier;
}
```

### 5.2.2.2 重新编译插件

完成源代码更改后，通过运行脚本 `/opt/edgeai-gst-apps/scripts/install_gst_plugins.sh` 来重新编译和重新安装插件。

### 5.2.2.3 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 `tiouvixisp` 的属性中是否列出了新传感器名称。例如，IMX219 会如下所示：

```
root@am62axx-evm:~# gst-inspect-1.0 tiouvixisp
...
sensor-name : TIOVX camera sensor string ID.Below are the supported sensors
              SENSOR_SONY_IMX219_RPI
```

## 6 调优过程

本节详细介绍了调优过程。

### 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/video2
    /dev/video3
...
    /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

请注意，“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，它可能会有所不同。

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 -p” 命令显示的内容（如上所示）：

```
root@am62axx-evm:~# media-ctl -V '"imx219 4-0010":0 [fmt:SRGB10_1X10/1920x1080 field:none]'
```

```
root@am62axx-evm:~# gst-launch-1.0 -v v4l2src num-buffers=5 device=/dev/video2 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可能与捕获这些图像的照明环境不匹配。

### 6.2 使用初始 VPAC 配置启用摄像头流式传输

确认摄像头及其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。实时流式传输还将简化后续步骤，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，如以下各节所述。

#### 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。

需要在此配置文件中指定以下参数：

- **原始图像格式信息**
  - WIDTH：传感器图像宽度
  - HEIGHT：传感器图像高度
  - BIT\_DEPTH：传感器像素位深度
  - BAYER\_PATTERN：Bayer 模式，其中 0=RGGB；1=GRBG；2=GBRG；3=BGGR

### 备注

请注意，上述配置信息必须与媒体控制命令中用于配置摄像头数据格式的信息相匹配。例如，如果 IMX219 摄像头的媒体控制命令使用 “[fmt:SRGGB10\_1X10/1920x1080 field: none]”，则应在此配置文件中使用时使用以下值：

- WIDTH 1920
- HEIGHT 1080
- BIT\_DEPTH 10
- BAYER\_PATTERN 0
- 摄像头传感器支持
  - 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[更新传感器的 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_drv/src` 下。

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

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

生成的 xml 文件将位于配置文件中 `PRJ_DIR` 指定路径下的 `dcc_xmls` 文件夹中。此文件夹包含两组 .xml 文件，其中一组用于线性模式，另一组用于 WDR 模式。每个 xml 文件夹中还将生成一个脚本文件“`generate_dcc.sh`”。例如，以下是线性模式下的 xml 文件夹内容（对于 IMX219）：

```
$PRJ_DIR/dcc_xmls/linear$ 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脚本 `generate_dcc.sh`，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。

### 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 中至少具有以下预构建的二进制文件：

```
root@am62axx-evm:~# ls /opt/imaging/imx219
dcc_2a_10b.bin dcc_viss_10b.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/video2 io-mode=dmabuf-import ! \
video/x-bayer, width=1920, height=1080, framerate=30/1, format=rggb10 ! \
tiouvisp 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 ! \
kmssink driver-name=tidss sync=false
```

即使使用初始 VPAC 配置时图像质量并未经过全面调优，这也可以验证适用于 VPAC 的 GStreamer 插件在摄像头中运行正常。输出视频应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

设置光箱和色卡。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野 (FOV) 范围。

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，在分辨率为 1920x1080 时，IMX219 驱动程序会裁剪图像，图像会在显示屏上看起来放大。在这种情况下，使用较小尺寸的色卡可能效果更好。

### 6.4 捕获原始图像并执行基本调优

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和/或 YUV 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/AM62A 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。

#### 6.4.1 启动调优工具并创建工程

在 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 6-1 显示了 IMX219 在 1080p 流模式下的示例。

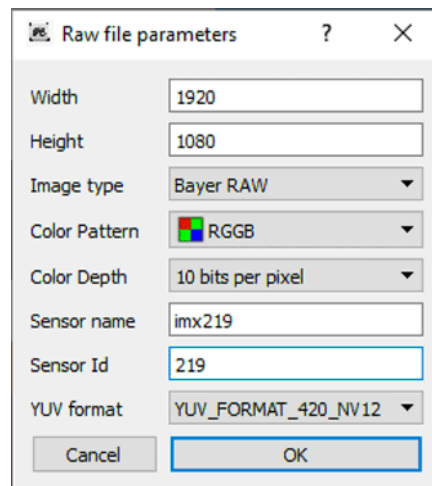


图 6-1. 用于调优的传感器原始图像参数

#### 备注

请注意，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[摄像头捕获验证](#)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，该图像随即应会显示在“Preview”窗口中。例如，下面是一个 1920x1080 且每像素 10 位的 IMX219 原始图像显示在调优工具中。

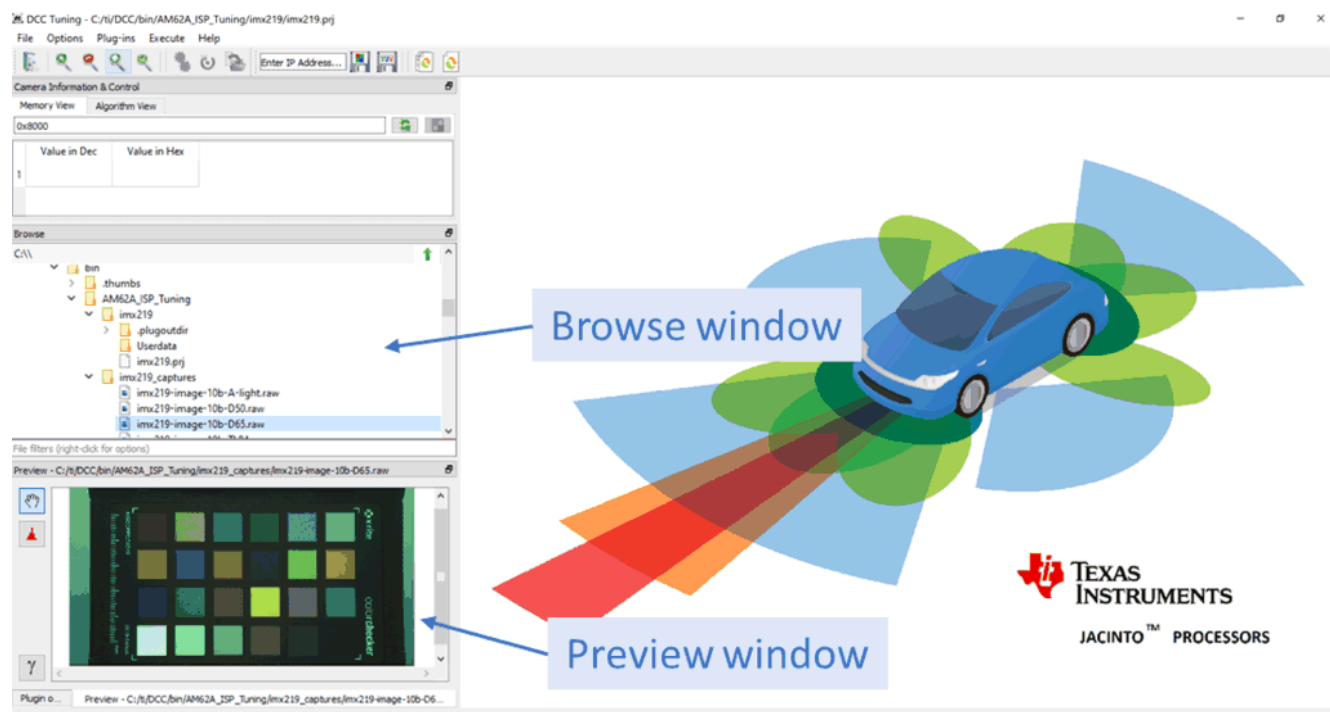


图 6-2. DCC 调优工具的图像预览

#### 备注

在调优工具中只能预览以每像素 10/12/16 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，用户可以继续执行调优，如以下各节所述。

## 6.4.2 调优顺序

AM6xA ISP (VPAC) 由多个功能块组成。原始图像由这些块逐个处理。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，如调优工具菜单“Plug-ins”中所示：

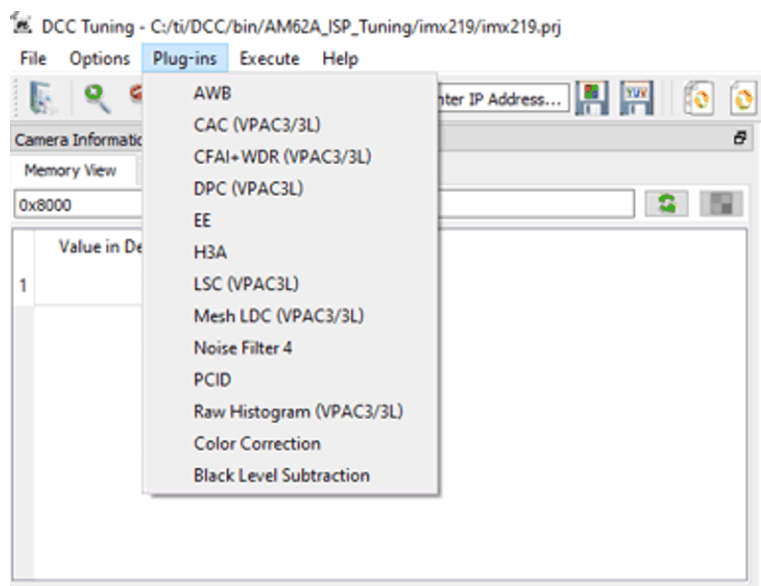


图 6-3. DCC 调优工具的插件

### 备注

每个插件都有自己的调优指南，该指南可从“Help”->“Documentation”下拉菜单中访问。调优期间应参考这些指南。

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

1. 黑电平消减：数据台阶电平（传感器黑电平）通常由传感器驱动器指定，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

在调整每个插件后，将生成一组用于 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，从而逐步提高实时流媒体的图像质量。

本应用手册以采用 1080p 模式的 IMX219 传感器为示例，说明了如何使用 TI 的 TDA4/AM6xA DCC 调优工具对黑电平、AWB 和颜色校正进行调优和校准。有关调优的更多详细信息，请参阅调优工具帮助菜单中的插件指南。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。

### 6.4.3 黑电平消减

对于包括 IMX219 在内的线性传感器，应当先从原始图像像素中消减黑电平或台阶电平，然后再在 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，但最好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，该图像应该在“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。

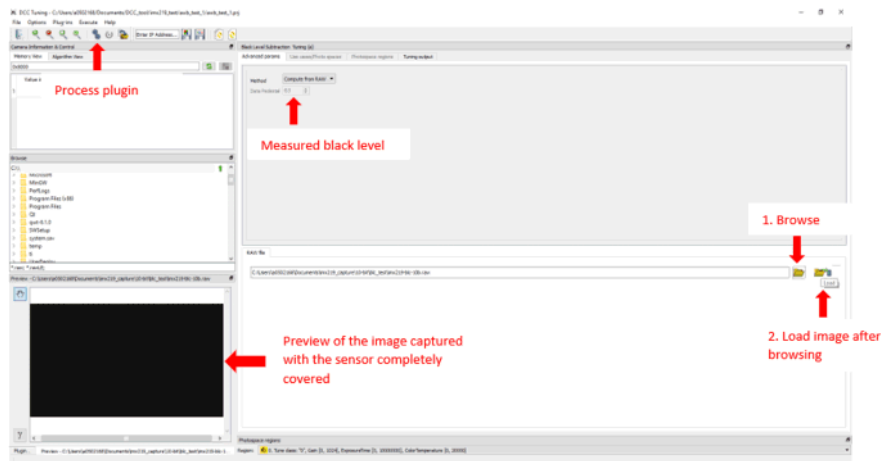


图 6-4. 黑电平消减调优

对于 WDR 传感器，黑电平消减通常会与 WDR 解压缩和重新压缩相结合，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。有关 WDR 传感器的更多详细信息，请参阅插件的用户指南。

完成插件调优后，点击“Export DCC profile binary”按钮来为该插件生成输出 XML 文件。XML 文件位于节 6.4.1 中创建的项目文件夹下的“.plugoutdirXML”文件夹内。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。这应该在以下部分中调整每个插件后完成。

要在调整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，在调整 IMX219 的 Black Level Subtraction 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/video2 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 6-5 显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 6-6 则显示了黑电平消减调优后的图像（比较黑色外观）。



图 6-5. 黑电平消减前的图像



图 6-6. 黑电平消减后的图像

#### 6.4.4 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 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 IMX219 的输入参数，以便可以正确对 H3A/AE/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线性传感器的传感器/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。

#### 6.4.5 自动白平衡 (AWB)

##### 6.4.5.1 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。这些图像将在颜色校正调优的下一步中再次使用。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）
- 将色卡直立置于光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应足够明亮，且白色色片上没有出现明显饱和/剪裁。这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中的任何一种进行捕获
- 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 6-7. DCC 调优工具的实时捕获功能

图 6-8 至图 6-11 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 6-8. 使用 D50 照明时捕获的色卡图像



图 6-9. 使用 D65 照明时捕获的色卡图像



图 6-10. 使用 TL84 照明时捕获的色卡图像

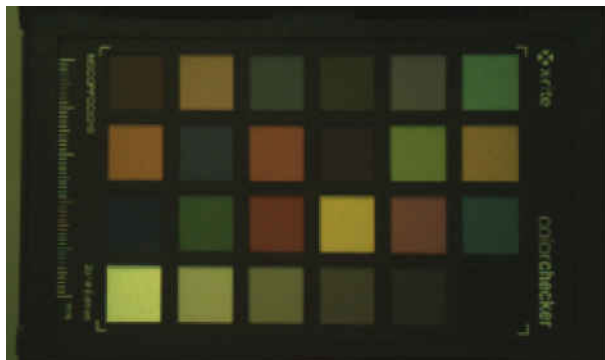


图 6-11. 使用 A 灯时捕获的色卡图像

#### 6.4.5.2 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，该值应为 6500。
- 曝光、增益和光圈：这些值实际上并不用于 AWB，因此可以忽略。
- 黑电平：这必须是在 Black Level Subtraction 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，请特别注意：

- 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 6-12 至图 6-13 显示了导入一个原始图像以进行 AWB 调优的示例。

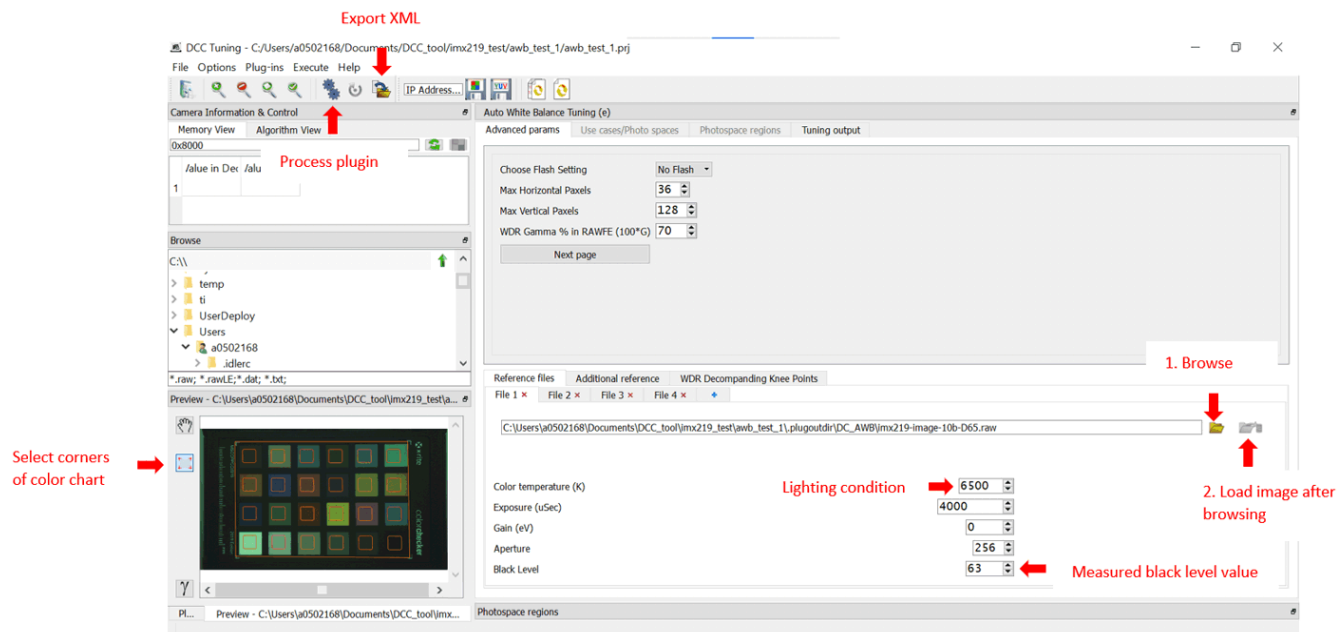


图 6-12. 自动白平衡调优

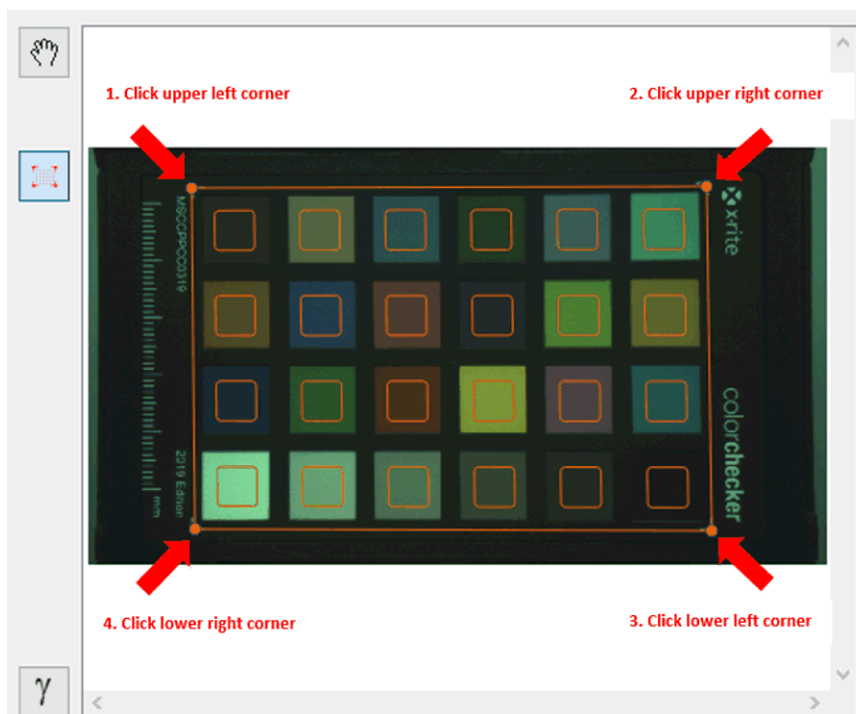


图 6-13. 选择色卡的边角

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，将显示参考 Cb-Cr 绘图方案。下面是使用上面所示原始图像时的结果图。

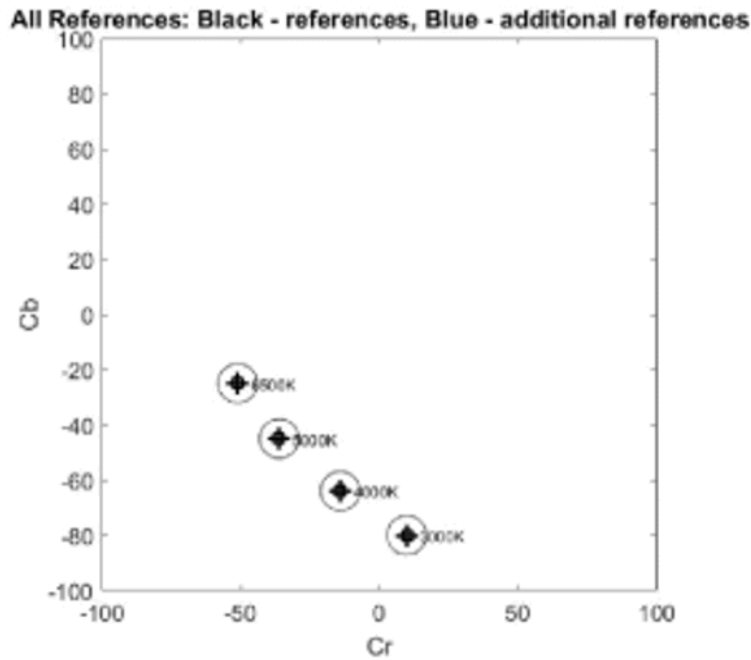


图 6-14. 自动白平衡调优结果

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 6-15 和图 6-16 显示了在 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 6-15. 自动白平衡调优前的图像



图 6-16. 自动白平衡调优后的图像

#### 6.4.6 颜色校正

要获得正确的颜色输出，应调整颜色校正插件。为 AWB 调优捕获的原始图像可用于调优此插件。下面是此调优步骤的摘要。有关更多详细信息，请参阅帮助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，但用户可以使用自己选择的参考图像。
- 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“Export DCC profile binary”以生成 XML 和二进制文件，如下图所示。

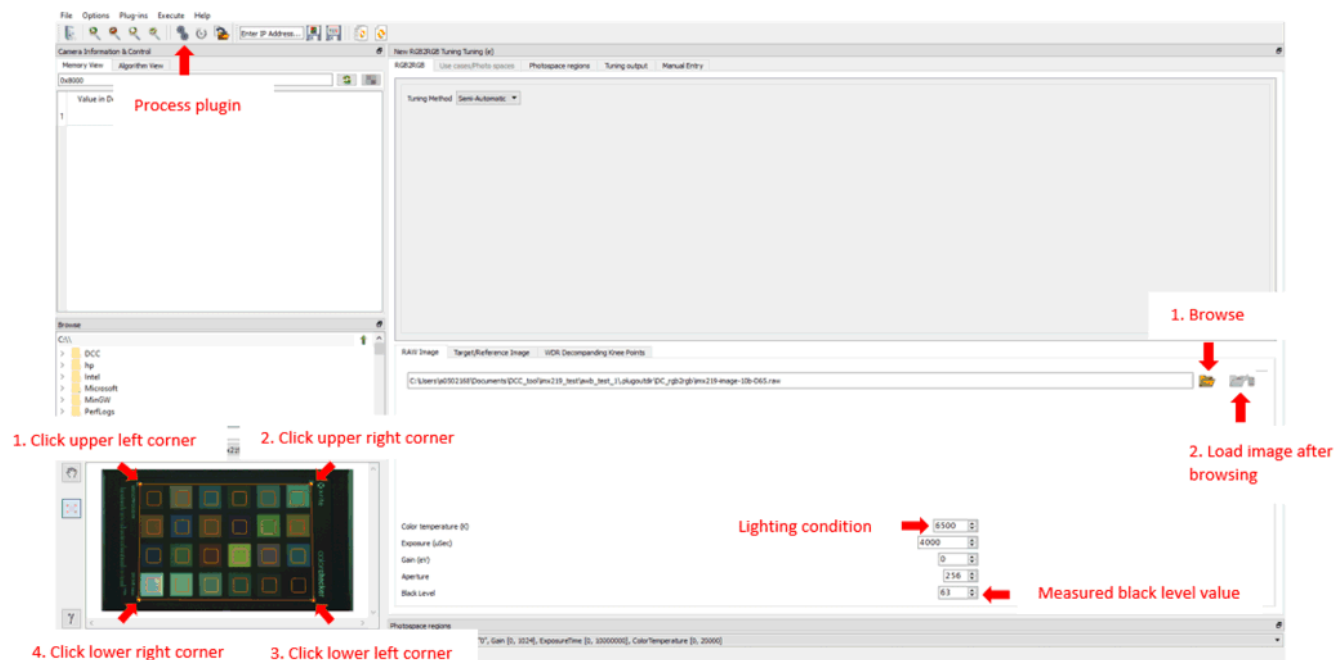


图 6-17. 颜色校正调优

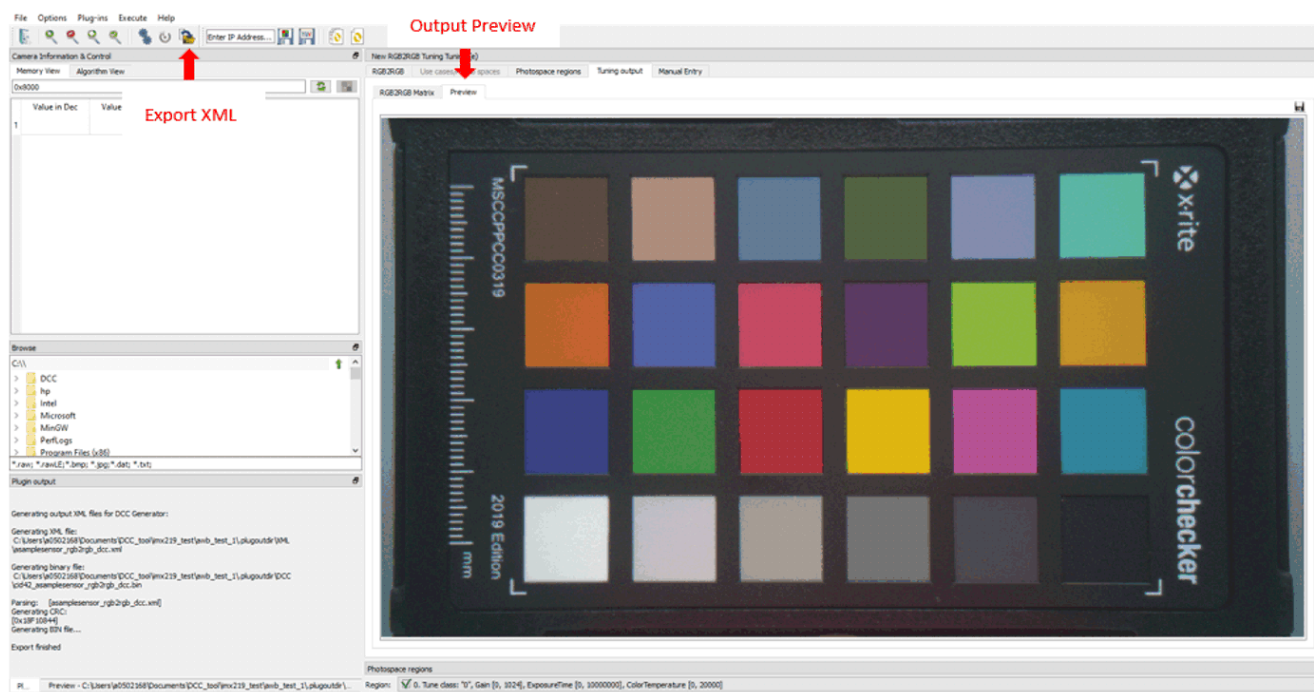


图 6-18. 颜色校正调优输出

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 6-19 和图 6-20 显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 6-19. 颜色校正调优前的图像



图 6-20. 颜色校正调优后的图像

## 6.5 执行微调

对于 IMX219 摄像头，在正确完成黑电平消减、硬件 3A、自动白平衡和颜色校正调优后，ISP 应在各种灯箱照明条件下实现出色图像质量的 70%~80%。接着还可以根据 DCC 调优工具中提供的各个插件的调优指南，对其他插件进行调优。例如，可以对以下额外插件进行调优以实现更好的图像质量：

- EE（边缘增强）插件，可提高输出图像的清晰度
- 噪声滤波器 4 插件，用于抑制黑暗条件下的噪点
- Mesh LDC（镜头失真校正）插件，用于消除镜头失真
- CFAI + WDR（色彩滤波阵列插值 + 宽动态范围）插件，用于去马赛克和 WDR 传感器 LSC
- （镜头阴影校正）插件，用于消除镜头阴影
- 适用于 RGBIR 4x4 传感器的图形转换和红去马赛克 (PCID) 插件

## 7 总结

本应用手册介绍了 AM6xA ISP 调优的工作流程。此版本中使用 TI 的 AM62A Processor SDK、成像软件和 DCC 调优工具，逐步对 IMX219 摄像头进行 ISP 调优以获得出色的输出颜色。

以下内容将在未来的修订版本中加以介绍：

- 使用 TI DCC 调优工具中的更多插件来针对其他图像质量因素进行调优
- TI DCC 调优工具的实时调优功能
- 使用第三方的成像算法和调优工具进行 ISP 调优

## 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

Copyright © 2023，德州仪器 (TI) 公司