参考資料

Application Note

AM6xA ISP チューニング ガイド

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

概要

このアプリケーション レポートでは、未加工カメラで可能な限り最良の画質を得るために、TI の AM6xA ビジョン プロセッサ ファミリ上で ISP をチューニングするワークフローについて説明します。このレポートで提供するチューニング手順には、AM62A スタータ キット評価基板上で IMX219 センサ (RGB のみ) および OX05B1S センサ (RGB-IR) を使用した例が含まれています。

目次

図の一覧

商標

1 はじめに

AM6xA ビジョンプロセッサには、ハードウェア アクセラレーション形式の画像信号プロセッサ (ISP) が搭載されており、これはビジョン前処理アクセラレータ (VPAC) とも呼ばれます。設定可能な画像処理パラメータを備えた VPAC は、さまざまな未加工カメラ モジュールに対応できるよう設計されています (一般的な未加工カメラ モジュールは、レンズ、フィルタ、未加工イメージ センサ、場合によってはシリアライザを含みます)。特定の未加工カメラ モジュールにおいて実行時に最適な画質を得るためには、VPAC のパラメータを算出し、それをフレームごとに未加工センサ画像の処理に適用する必要があります。そのため、最適な VPAC パラメータは通常、さまざまな制御された照明条件下で、イメージング ラボにおいてエンジニアによって事前に準備されます。次に、実行時に、準備したパラメータを参照し、ランタイム照明環境に合わせて補間します。これは、自動露出 (AE)、自動ホワイト バランス (AWB)、および動的 ISP パラメータ制御のソフトウェア イメージング アルゴリズムを使用しています。イメージング ラボで最適な VPAC パラメータを準備する手順は、このアプリケーション レポートでは ISP チューニングと呼ばれます。

このレポートで説明する ISP チューニング手順は、AM62A、AM68A、AM69A を含む、AM6xA ビジョン プロセッサ ファミリのすべての SoC に適用されます。このレポートでは、AM62A スタータ キット評価基板を使用した例を示しています。

特定のシステム オン チップ (SoC) に関する ISP (VPAC) の技術的詳細については、その SoC のテクニカル リファレンス マニュアル (TRM) を参照してください。

2 チューニングの概要

AM6xA ファミリ SoC 上の ISP (VPAC) は、動的カメラ構成 (DCC) バイナリ ファイルを通じて構成されます。Linux® ベースのアプリケーションでは、これらのバイナリ ファイルは一般的な GStreamer パイプラインを通じて VPAC に提供されます。VPAC の各処理ブロックは、GStreamer パイプライン要素 (tiovxisp、tiovxldc、tiovxmultiscaler) によってカプセル化されており、VPAC のすべての設定可能なパラメータはプロパティとして提供されます。

例えば、以下の GStreamer パイプラインは、IMX219 カメラから高精細マルチメディア インターフェイス (HDMI) ディスプレイへ映像をストリーミングします。この際、表示前に VPAC を使用して未加工画像を処理します:

```
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
```

上記のパイプラインにおいて、GStreamer 要素 **tiovxisp** は、VPAC ハードウェアと、AE および AWB (2A) などのイメージング アルゴリズムや ISP パラメータ制御のための TI リファレンス ソフトウェアとのインターフェイスを提供します。IMX219 用の VPAC 設定は、**tiovxisp** のプロパティの一部として指定される、以下の 2 つのバイナリ ファイルを通じて提供されます:

- 調整された ISP パラメータの場合:dcc-isp-file=/opt/imaging/imx219/linear/dcc_viss_10b.bin
- AE および AWB アルゴリズムのキャリブレーション情報:dcc-2a-file=/opt/imaging/imx219/linear/dcc_2a_10b.bin

これらのバイナリ ファイルは ISP チューニングの出力であり、動的カメラ構成 (DCC) プロファイルとも呼ばれます。

概要として、AM6xA の ISP チューニング手順には、以下のステップが含まれます (例として、TI のリファレンス イメージング ソフトウェアおよびチューニング ツールを使用します):

1. ハードウェアの設定:必要なハードウェア機器をすべて準備し、セットアップします。
2. ソフトウェアの設定:必要なソフトウェア コンポーネントをすべてダウンロードし、インストールします。
3. センサ ソフトウェアの開発と統合:カメラ センサ ドライバがシステムへ適切に統合されており、未加工画像を取得可能であることを確認します。自動露出アルゴリズムに必要な露出設定を追加します。GStreamer プラグインにこのセンサのサポートを追加します。
4. ISP の初期設定の生成: Python® スクリプトを実行して、VPAC 構成用の初期 (デフォルトまたはベースライン) DCC プロファイルを生成します。この構成により、次のステップを実行するのに十分な画質でビデオ ストリーミングを有効にできます。

5. カメラ マウントの調整：前の手順で生成した DCC プロファイルを使用してライブ映像をストリーミングし、画像が適切な位置合わせ、フォーカス、照明条件などで取得されるようにカメラ モジュールの取り付けを調整します。
6. ISP の基本チューニング：未加工画像を取得し、ラボ環境下で最良画質の 70% ～ 80% を達成できるよう、基本的なチューニングを実施します。
7. ISP の微調整：前の手順で生成した新しい DCC プロファイルを使用して、再度ライブ ストリーミングを実行します。画質評価に基づき、改善が必要な項目を特定し、必要に応じて追加の詳細チューニングを実施します。

このアプリケーション ノートでは、AM62A スタータ キット評価基板に IMX219 カメラおよび OX05B1S カメラを接続し、前述のチューニング手順を実演します。このチューニングの原則および手順は、あらゆるカスタム ボードおよび未加工カメラに適用できます。

3 ハードウェア要件

未加工カメラの ISP チューニングを行うために必要なハードウェア機器は、以下のとおりです：

- AM62A スタータ キット評価基板 (SK EVM) または AM62A SOC を搭載したカスタム ボードです。評価基板の包括的なセットアップについては、AM62A SK 評価基板クイック スタート ガイドを参照してください。
- カメラ本体およびカメラを AM62A SK 評価基板に接続するために必要なすべての付属品。カメラを評価基板に接続する方法については、AM62A Academy を参照してください。Academy のメイン ページで、「Linux の評価」→「TI Linux のツアー」→「カメラ」を順に選択します。
- ライト ボックス：
  - 少なくとも 2700K から 6500K の範囲を持つ校正済み光源
  - より広い色温度範囲が必要な場合は、さらに多くの光源が必要です
- XRite カラー チェッカー チャート。場合によっては、設定に応じて異なるチャート サイズが必要になります。
- 照明の状態を監視および記録するためのライト メーター、クロマ メーター (またはその両方)。
- カメラを所定の位置に確実に固定するための三脚またはマウント (必要に応じて)。図 3-1 に、そのようなセットアップの例を示します。

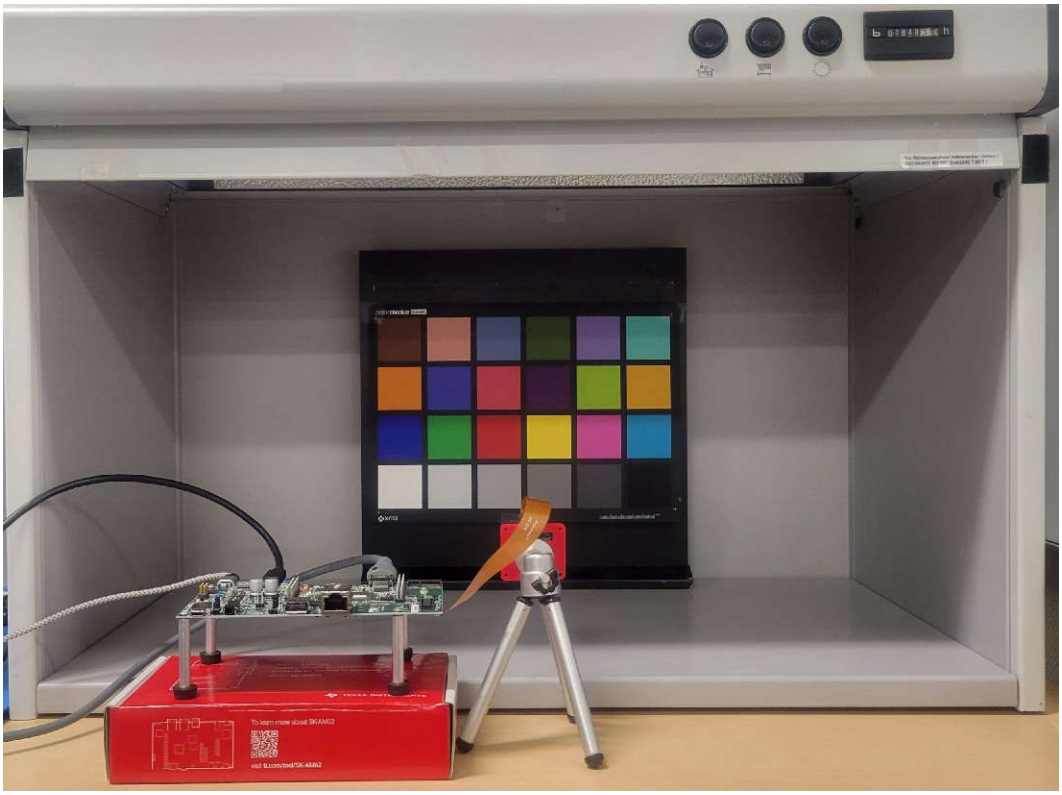

図 3-1. ハードウェアと機器のセットアップの調整

4 ソフトウェア要件

4.1 プロセッサ SDK Linux®

AM62A 向けプロセッサ ソフトウェア開発キット (SDK) Linux をダウンロードします。

評価基板を Linux にブートするには、AM62A SK 評価基板クイックスタート ガイドに従ってください。

4.2 TI のリファレンス画像処理ソフトウェア

イメージ作成ソフトウェアを https://git.ti.com/cgit/processor-sdk/imaging/ から Linux ホスト PC にクローンします:

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

このレポートには、AE、AWB、および ISP 制御または DCC 向けの TI のリファレンス イメージング アルゴリズムが含まれています。この情報には、チューニング プロセスで必要となる Python スクリプトも含まれています。

4.3 ISP チューニング ツール

ISP チューニング ツールおよびイメージング アルゴリズム (例えば、AE、AWB、および ISP 制御) は、複数のイメージング系サードパーティから提供されており、量産向けのサポートを受けられる場合があります。TI の TDA4 および AM6xA 向け DCC チューニング ツールおよびイメージング アルゴリズムは、設計リファレンスとしても利用可能であり、このレポートでは例として使用しています。このツールを入手するには、お近くの TI FAE にお問い合わせください。TI のサードパーティ製チューニング ツールも同様に利用可能ですが、このレポートの対象範囲には含まれていません。

5 センサ ソフトウェアの統合

AM6xA Linux SDK を使用してターゲット センサ向けに ISP をチューニングするには、SDK 内でセンサをサポートするソフトウェアを開発する必要があります。センサ ドライバの開発については、このレポートでは説明しません。以下のセクションでは、まず AM6xA における画像パイプラインのソフトウェア アーキテクチャの概要を示し、その後、センサ ドライバを SDK に追加する方法、およびセンサ対応のために GStreamer プラグインを修正する方法について説明します。

5.1 イメージ パイプライン ソフトウェア アーキテクチャの概要

このセクションでは、で AM6xA 上の ISP およびイメージ パイプライン ソフトウェア アーキテクチャの概要を説明します。これは、後述するチューニング手順を理解するのに役立ちます。

AM6xA デバイス上の ISP は、ピクセル レベルでの画像データ処理のための基本的なビジョン プリミティブ機能を提供します。ISP には、次の 3 つのサブモジュールがあります:ビジョン イメージング サブシステム (VISS)、レンズ歪み補正 (LDC)、マルチ スカラ (MSC)。

- **ビジョン イメージング サブシステム (VISS):**VISS は未加工データに対して画像処理を実行します。これには、ワイド ダイナミック レンジ (WDR) マージ、欠陥ピクセル補正 (DPC)、レンズ シェーディング補正 (LSC)、グローバルおよびローカルの明るさとコントラスト強調 (GLBCE)、デモザイク処理、色変換、およびエッジ強調 (EE) が含まれます。このブロックは未加工画像を入力として受け取り、YUV 出力を生成します。
- **レンズ歪み補正 (LDC):**LDC エンジンは YUV ドメインプロセッサであり、遠近法と幾何学的変換を実行するように設計されています。これは、レンズ歪み補正、エピポーラ整列、汎用的な透視変換など、さまざまな効果を生成するために使用できます。
- **マルチスカラ (MSC):**MSC は、指定された入力から最大 10 個のスケーリング出力を生成でき、さまざまなスケーリング比 (1 × ～ 0.25 × の範囲) に対応します。10 個の各スケーリング処理は、ピラミッド スケールまたはインター オクターブ スケール生成のいずれかを実行するよう構成できます。

これら 3 つのサブモジュールのうち、VISS と LDC はチューニングが必要であり、VISS には個別にチューニングする必要がある複数の処理ブロックが含まれています。

図 5-1 は、カメラ キャプチャ サブシステムおよび ISP を含む AM6xA の画像処理パイプラインのソフトウェア アーキテクチャを示しています。カメラ キャプチャ ドライバは A53 または A72 コア上の Linux で動作し、ISP ドライバは R5 コア上の RTOS で動作します。異なる CPU コア上で動作していますが、カメラ キャプチャと ISP は、GStreamer または TIOVX の同一フレームワークによってインターフェイスおよび統合が可能です。

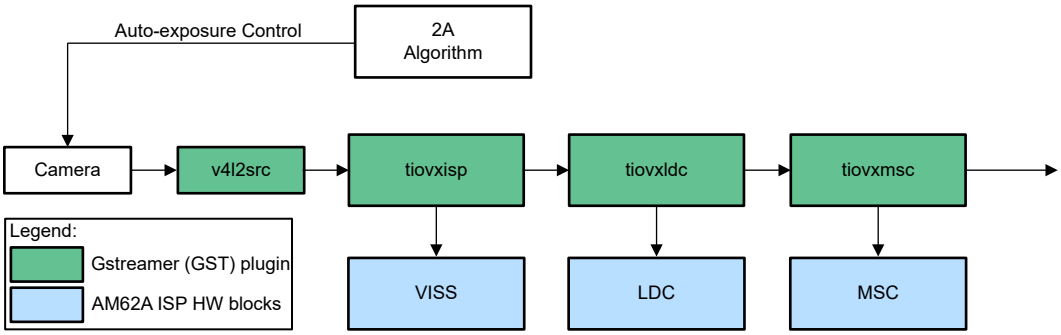

図 5-1. AM6xA 画像処理パイプライン

このレポートでは、画像パイプラインの各コンポーネントを統合するための GStreamer のサンプル例を示します。カメラ キャプチャ サブシステムは、GStreamer プラグイン v4l2src を介してアクセスされます。3 つの各 ISP サブモジュールは、それぞれ対応する GStreamer プラグイン (tiovxisp、tiovxldc、tiovxmsc) を介してアクセスされます。

5.2 SDK にセンサ ドライバを追加

SDK にセンサ ドライバを追加する手順については、AM62A Academy を参照してください。Academy のメイン ページから、「TI EVM で Linux を開発」→「デバイス ドライバを使用」→「カメラを使用」→「新しい CSI-2 センサを有効化」の順に進みます。

SDK にセンサ ドライバを追加することに加え、センサをサポートするために ISP 用の GStreamer プラグインを更新する必要があります。これには、評価基板上のターゲット ファイル システム内の 2 つのコンポーネントを変更および再ビルドする作業が含まれます。詳細は以下のセクションで説明します。

5.3 TIOVX モジュールの更新

5.3.1 ソース コードの変更

評価基板上のターゲット ファイル システムで、/opt/edgeai-tiovx-modules / src /tiovx_sensor_module.c に移動し、センサの名前と id を tiovx_init_sensor() 関数のリストに追加します。たとえば、IMX219 には次の機能があります:

```
// 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;
}
```

ここでは、IMX219 カメラのセンサ名と DCC ID の両方を定義します。GStreamer パイプラインでは正確な名前文字列 (SENSOR_SONY_IMX219_RPI) を使用し、今後のステップでは DCC センサ ID (219) をチューニング ツールで使用します。

5.3.2 モジュールの再構築

ソース コードの変更が完了したら、/opt/edgeai-gst-apps/scripts/install_tiovx_modules.sh スクリプトを実行してモジュールを再構築し、再インストールします。

5.4 VISS 用 GStreamer プラグインの更新

ISP で新しいセンサをサポートするには、VISS 用 GStreamer プラグインを更新し、更新内容には、以下が含まます:

- プラグインのプロパティ内のセンサ リストに該当するセンサを追加
- 図 5-1 に示すように、自動露出と自動ホワイト バランシング (2A) アルゴリズムで必要なセンサの露出設定を追加します。

5.4.1 VISS プラグインのプロパティを更新

評価基板のターゲット ファイル システムで、/opt/edgeai-gst-plugins/ext/tiovx/gsttiovxisp.c にアクセスし、センサの名前をリストに追加します (例として以下に示す OX05B1S):

```
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 2A アルゴリズムに露出設定を追加

図 5-1 に示すように、カメラ センサの露出は 2A アルゴリズムによって自動的に制御されます。2A アルゴリズムは、ISP から提供される H3A 統計情報に基づいて露出を調整します。露出時間とゲインの両方を 2A アルゴリズムで調整できます。すべてのセンサには、内部で動作可能な最小および最大の露光時間とゲインが存在します。これらの仕様はデータシートに記載されており、VISS プラグインを介して 2A アルゴリズムに渡す必要があります。この情報を 2A アルゴリズムに提供するため、新しい関数を追加する必要があります。この関数は通常、gsttiovxisp.c で get_<sensor>_ae_dyn_params() という名前になっています。

5.4.2.1 ゲイン

最小および最大ゲインは通常、データシートに Nx ゲインおよび Mx ゲインとして記載されています。ここで、N は最小値、M は最大値を表します。対応するゲイン レジスタの値も通常、記載されています。例えば、あるセンサでは次のように規定されている場合があります:

- 最小ゲイン:1× の場合、gain_register の値は 16 です
- 最大ゲイン:15.5× の場合、gain_register の値は 248 です

DCC ライブ チューニング ツールでは、1× ゲインを 1024 として浮動小数点でゲインを計算し、表示するため、この規約に従ってゲインを設定し、2A とセンサ ドライバ間でゲインをマッピングすることが推奨されます。関数 get_<sensor>_ae_dyn_params() の最小および最大ゲインは、以下の設定で決定されます:

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

したがって、2A アルゴリズムで使用されるゲイン値は、この特定のセンサで使用されるゲイン値の実質 64 倍となります。2A アルゴリズムがセンサに設定すべきゲイン値を返した後、値はセンサへ送信する前に 64 で除算する必要があります。このマッピングは、関数 gst_tiovx_isp_map_2A_values() で行われます:

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

5.4.2.2 露出時間

露光時間については、最小値および最大値は通常、行周期の数で指定されます。たとえば、特定のセンサの仕様は次のとおりです:

- 最小露出時間:6 行の期間
- 最大露光時間:(フレーム長 – 30) 行期間

解像度 2592 × 1944 を例にすると、フレーム長は 1944 行に垂直ブランキングを加えた値になります。垂直ブランキングが 184 であると仮定すると、フレーム長は 2128 行になります。したがって、最大露光時間は 2128 − 30 = 2098 行周期となります。センサ ドライバを確認し、データシートに従って露光時間がレジスタに書き込まれていることを確認します。

DCC ライブ チューニング ツールでは露光時間がマイクロ秒で表示されるため、2A 側でも露光時間をマイクロ秒で設定し、2A とセンサ ドライバ間でマッピングを行うのが推奨されます。マッピングはフレーム サイズとフレーム レートによって異

なります。例えば、解像度 2592 × 1944、60fps の場合、最小および最大露光時間は、関数 get_<sensor>_ae_dyn_params() 内で以下のように設定できます:

```
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 */
```

したがって、マイクロ秒から行周期数 (センサ ドライバに渡される値) へのマッピングは、gst_tiovx_isp_map_2A_values() 内で提供されます:

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

5.4.2.3 その他のパラメータ

2A アルゴリズムで必要なその他のパラメータには、次のデフォルト値が使用されます:

```
/* 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 プラグインの再構築

ソース コード /opt/edgeai-gst-plugins/ext/tiovx/gsttiovxisp を変更した後、/opt/edgeai-gst-apps/scripts/install_gst_plugins.sh スクリプトを実行して GStreamer プラグインを再構築し、再インストールします。

5.4.4 GStreamer プラグインで新しいセンサを検証

上記の変更をすべて行った後、GStreamer プラグイン tiovxisp のプロパティに新しいセンサ名が表示されていることを確認します。たとえば、OX05B1S は以下のようになります:

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

6 チューニング手順

このセクションでは、ISP のチューニングの手順の詳細について説明します。要約すると、この手順には次の手順が含まれます:

1. カメラが正しく動作していることを確認します
2. 自動化スクリプトを使用して、初期 (デフォルト、またはベースライン) の ISP 構成プロファイルを作成します
3. 制御された照明条件下で、ラボ チューニングを実施します。チューニングは基本的なチューニングと微調整の 2 つのステップに分けることができます。
4. センサが実際に展開される実環境のシーンで、量産チューニングを実施します。この手順は、このレポートでは対象外です。

6.1 カメラ キャプチャの機能動作を確認

カメラドライバが SDK に統合済みであり、AM62A SK 評価基板が Linux を起動してカメラを認識できることを前提とします。v4l2-ctl コマンドと media-ctl コマンドの両方で、次のように予期される出力が表示されることを確認します (例として IMX219 を使用):

```
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)
```

注

4-0010 media-ctl 出力は、センサの I2C バス アドレスであり、この値は別の SDK リリースで異なる場合があります。

次に、カメラを特定のフォーマットに設定できること、および GStreamer パイプラインを使用して未加工画像をキャプチャできることを確認します。media-ctl コマンドによって上記のように 4-0010 が表示されていると仮定した場合の例は、次のとおりです:

```
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"
```

キャプチャされた未加工画像は、ヘッダや圧縮のない純粋な Bayer パターン配列フォーマット (IMX219 センサでは RGGB) です。これらの未加工画像は、未加工画像ビューアや ffmpeg などのツールで表示できます。この段階では、センサのデフォルトの露光時間およびゲインが撮影環境の照明条件と必ずしも一致しないため、未加工画像は露出オーバーまたは露出不足になる可能性があります。

6.2 初期 VPAC 構成でカメラ ストリーミングを有効化

カメラおよびドライバが機能的に動作していること、ならびに VPAC 用の GStreamer プラグインが新しいセンサをサポートしていることを確認した後、この手順では初期 VPAC 構成でライブ ストリーミングを実行します。この手順の目的は、VPAC 用の GStreamer プラグインが正しく動作することを確認することです。ライブ ストリーミングを行うことで、次の手順が容易になります。例えば、カメラの取り付け位置の調整がしやすくなります。

初期 DCC バイナリは、センサーの解像度およびカラー形式に合わせて VPAC と自動露出 (AE) プログラムを適切に設定するため、Python スクリプトにより自動生成されます。TI のリファレンス画像処理ソフトウェアは、セクション 6.2.1 からセクション 6.2.3 で説明するように、初期 VPAC 構成 (または DCC プロファイル) を生成するために必要なツールを提供します。

6.2.1 構成ファイルの生成

TI のリファレンス画像処理ソフトウェア、imaging/tools/default_DCC_profile_gen/configs に移動し、デフォルトのカメラ プロパティ用の構成ファイルを作成します。このフォルダ内の既存の構成ファイルを参照できます。

この構成ファイルで次のパラメータを指定します:

- **カメラ センサ情報**
  - SENSOR_ID:センサの DCC ID (VISS 向け GStreamer プラグインの更新に記載されているとおり、**tiovx_sensor_module.c 内でハードコードされている「dccID」値と一致している必要があります**)
  - SENSOR_NAME:センサの名称 (これは情報表示用のみであり、ツールでは使用されません)
  - SENSOR_DCC_NAME:対応する DCC 名 (これは情報専用で、ツールでは使用されません)
- **XML 出力フォルダ**
  - PRJ_DIR:生成された .xml ファイルを格納するフォルダ。このフォルダは、imaging/sensor_Srv/sor src の下にある必要があります (例:../../../sensor_drv src /<sensor name>)。
- **未加工画像フォーマット情報**
  - SENSOR_WIDTH:センサ イメージ幅
  - SENSOR_HEIGHT:センサ イメージの高さ
  - COLOR_PATTERN:Bayer パターン (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:センサ モード。0 = リニア、1 = wdr
  - BIT_DEPTH:センサ ピクセルのビット深度
  - WDR_BIT_DEPTH:デコンパンド後の WDR 未加工センサ画像のビット深度 (通常は 20 または 24)
  - WDR_KNEE_X および WDR_KNEE_Y:WDR デコンパンドのニー ポイント (カンマ区切りで記述し、間にスペースを入れないこと)
  - BLACK_PRE:デコンパンド前に減算するセンサのブラック レベル
  - BLACK_POST:デコンパンド後に減算するセンサのブラック レベル
  - GAMMA_PRE:20 ビットおよび 24 ビットの WDR 未加工を 16 ビットの ISP 内部ビット幅に圧縮するための GAMMA 値。通常、24 ビット WDR センサでは約 50 (0.5)、20 ビット センサでは約 70 (0.7) です
  - H3A_INPUT_LSB:H3A 入力ビット範囲の LSB 位置 (bit-H3A_INPUT_LSB から bit-H3A_INPUT_LSB+9 まで)

注

上記の設定情報は、実際のセンサ形式と一致している必要があります。

IMX219 の構成例を次に示します。WDR 関連のパラメータは使用されませんが、スクリプトを実行するには含める必要があります。

```
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
```

構成ファイルを作成したら、以下のとおり imaging/tools/default_DCC_profile_gen/scripts にある ctt_def_xml_gen.py Python スクリプトを実行します:

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

生成された xml ファイルは、$PRJ_DIR/dcc_xml フォルダに保存されます。スクリプト ファイル generate_dcc.sh も xml フォルダに生成されます。例えば、以下は IMX219 (リニア モード) の xml フォルダの内容です:

```
$PRJ_DIR/dcc_xmls/linear$ 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 DCC バイナリ ファイルの生成

前のステップで生成された .xml ファイルが含まれているフォルダ (リニアまたは wdr) に移動し、次のようにフォルダ内の generate_dcc.sh スクリプトを実行します:

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

このスクリプトは、PRJ_DIR で指定されたパス配下の dcc_bins フォルダに、デフォルト構成用の DCC バイナリ ファイルを生成します。

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

6.2.3 初期設定でビデオをストリーミング

前の手順で生成した DCC バイナリ ファイルを、AM62A SK 評価基板の /opt/imaging/<camera> フォルダのファイル システムにコピーします。例えば、IMX219 には SDK 内に次のプリビルド バイナリ ファイルが用意されています:

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

次に、media-ctl コマンドを使用して、初期設定ファイルのプロパティと整合性のあるカメラ形式を設定し、ライブ ストリーミングを実行します:

```
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
```

初期設定を使用した場合、画質は十分にチューニングされていない可能性がありますが、これにより新しいセンサに対して ISP 用の GStreamer プラグインが正しく動作していることを確認できます。出力された映像は、(まだチューニングは完了していないものの) 正しい色が得られており、適切に露出された映像となっています。これにより、次のステップでカメラの取り付けを簡単に調整できます。

6.3 カメラ マウントの調整

ライト ボックスとカラー チェッカを設定します。ストリーミング中にカメラをカラー チェッカに向けます。カラー チェッカがカメラの視野角 (FOV) 内に正確に収まるよう、カメラの位置を調整します。

特定の解像度では、センサ ドライバが画像を ISP へ送信する前にクロップできます。たとえば、解像度 1920 × 1080 の場合、IMX219 ドライバは画像をトリミングし、画像がディスプレイ上で拡大表示されます。この場合、より小さいサイズのカラー チェッカの方が適しています。

7 ベーシック チューニングの実行

一般に、ISP チューニングでは、まず制御された照明条件下で未加工画像または YUV 画像 (あるいはその両方) を一式取得し、チューニング ツールを使用して ISP パラメータおよび自動ホワイト バランシング (AWB) のキャリブレーションを調整する必要があります。このセクションでは、VPAC チューニング手順を説明するために、TI の TDA4 および AM6xA 向け DCC チューニング ツールを使用します。イメージング向けには、TI のサードパーティ製チューニング ツールも利用可能な場合があり、同様の機能に加えて、より高度な機能や量産レベルのサポートを提供していることもあります。

7.1 チューニング ツールの起動およびプロジェクトの作成

Microsoft® Windows®、<installation folder>\bin\DCC.exe で、インストールされているチューニング ツールの実行可能ファイルを探して実行します。正しい VPAC バージョンの選択を求められた場合は、VPAC3L (AM62A) を選択します。ツールが起動したら、プロジェクトを作成し、未加工画像プロパティを入力します。図 7-1 に、1080p ストリーミング モードの IMX219 の例を示します。

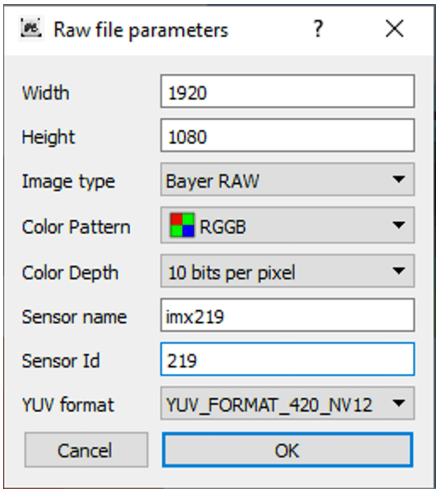

図 7-1. チューニング用センサ未加工画像パラメータ

注

画像のプロパティは、未加工画像の取得時に使用した設定と一致している必要があります。また、センサ ID は GStreamer プラグイン内にハード コードされている値と一致している必要があります。

新しいプロジェクトを作成すると、チューニング ツールを通じて未加工画像をプレビューできます。カメラ キャプチャ検証用コマンドを使用して、良好な照明条件でカラー チェッカーの未加工画像をキャプチャします。チューニング ツールの「参照」ウィンドウでこの画像ファイルを選択すると、プレビュー ウィンドウに画像が表示されます。例えば、以下はチューニング ツール上に表示された、1920 × 1080、10 ビット / ピクセルの IMX219 の未加工画像です。

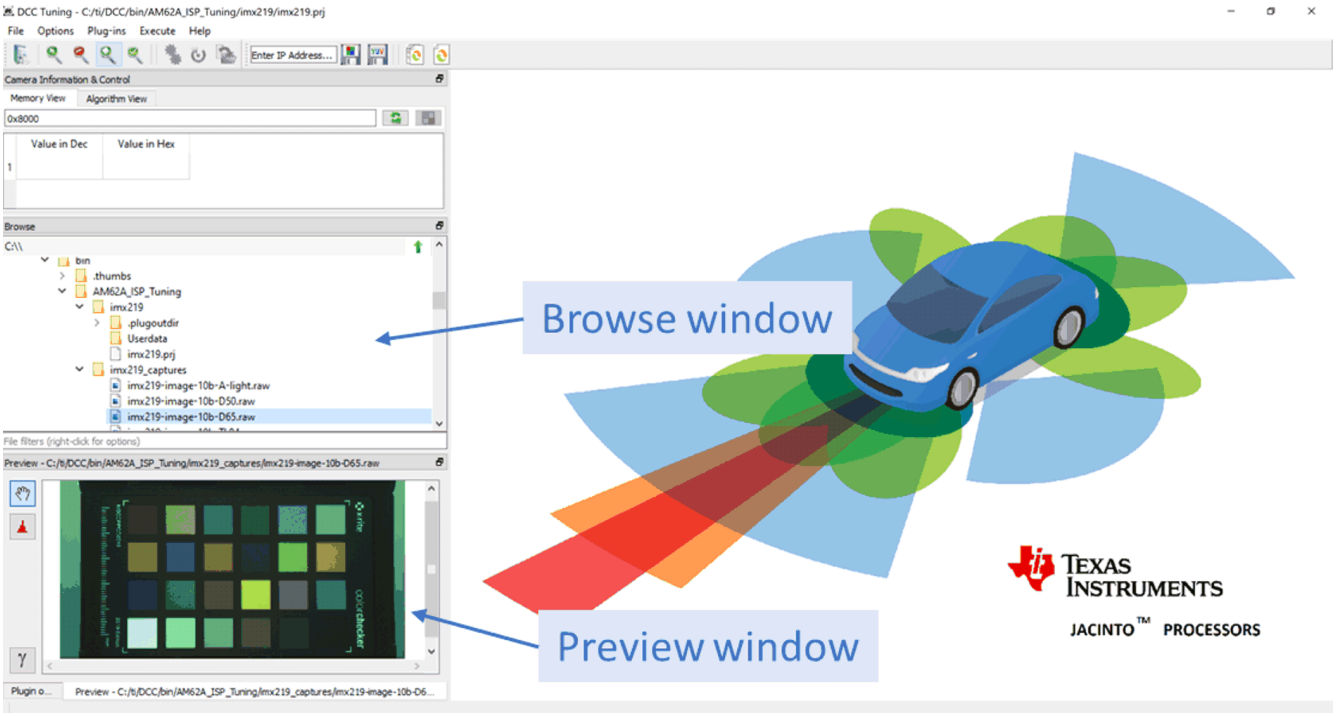

図 7-2. DCC チューニング ツールによる画像プレビュー

注

チューニング ツールでプレビューできるのは、10 ビット / ピクセル、12 ビット、または 16 ビットで取得された未加工画像のみです。1 ピクセルあたり 8 ビットで取得した場合は、まず 16 ビット / ピクセルに変換してから、チューニング ツールでプレビューおよび使用できます。

すべての準備が整ったので、以下のセクションで説明する手順に従ってチューニングを実施します。

7.2 チューニングの順序

AM6xA の ISP (VPAC) は、複数の機能ブロックで構成されています。未加工画像は、これらのブロックによって順次処理されます。このチューニング ツールでは、ISP ブロックを独立したグループ単位でチューニングできます。各グループには一つ以上の ISP ブロックが含まれます。チューニング グループは、チューニング ツールのメニュー「プラグイン」に示されているとおり、プラグインと呼ばれます:

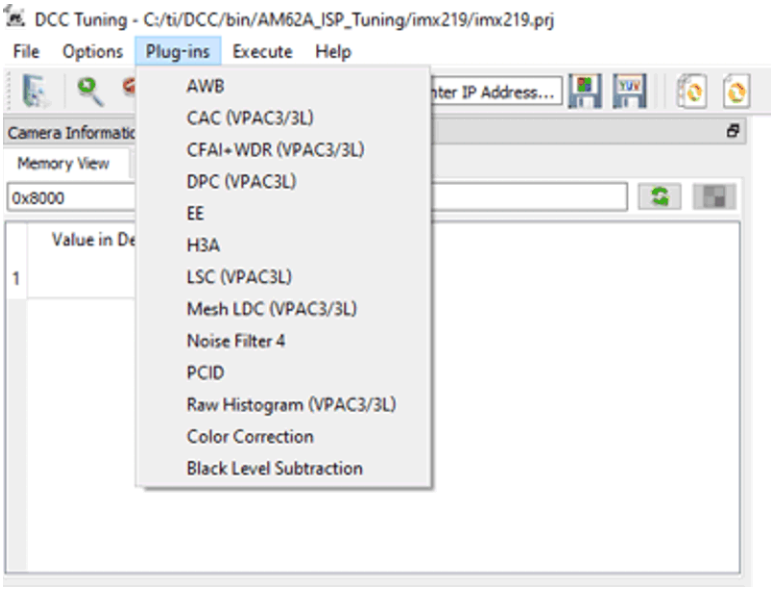

図 7-3. DCC チューニング ツール プラグイン

注

各プラグインにはチューニング ガイドが用意されており、「ヘルプ」→「文書化」のドロップダウン メニューから参照できます。チューニング中にこれらのガイドを参照してください。

一般に、これらのプラグインは、未加工画像を処理する際の ISP ブロックと同じ順序でチューニングしてください。以下に、推奨されるチューニングの順序を示します:

1. ブラック レベル減算 (BLC):データ ペデスタル (センサのブラック レベル) は通常センサ ドライバによって指定され、ここで検証のために算出できます。
2. H3A:ハードウェア 3A (自動露出、自動フォーカス、自動ホワイト バランス) 統計情報
3. PCID:パターン変換および IR デモザイキング (RGB + IR センサのみ)
4. AWB:自動ホワイト バランシング
5. 色補正
6. EE:エッジの強化
7. ノイズ フィルタ 4
8. メッシュ LDC:レンズ歪み補正
9. CFAI + WDR:カラーフィルタ アレイ補間 + ワイド ダイナミック レンジ
10. LSC:レンズ シェーディング補正

RGB 専用センサと RGB-IR センサでは、チューニングにわずかな違いがあります:

- RGB のみのセンサの場合:BLC、H3A、AWB、CCM、NSF4、EE など。
- RGB + IR センサの場合:H3A、PCID、AWB、CCM、NSF4、EE など。

各プラグインのチューニング後、VPAC 構成用の新しい XML ファイル一式を生成できます。これらの新しい XML ファイルは、初期設定から生成されたものを置き換えることで、画質を徐々に向上させることができます。

このアプリケーション ノートでは、PCID のチューニングを説明するために OX05B1S センサを使用し、ブラック レベル、AWB、カラー補正など残りのプラグインのチューニングを説明するために IMX219 センサを使用しています。チューニングの詳細については、チューニング ツールのヘルプ メニューのプラグイン ガイドを参照してください。TI のサードパーティによる他バージョンの ISP チューニング ツールも、おおむね同様の手順に従います。

7.3 ブラック レベル減算

ブラック レベル減算 (BLC) プラグインのチューニングは、RGB のみのセンサに対してのみ必要です。RGB-IR センサの場合は、PCID によって IR の減算処理が行われるため、このプラグインを調整する必要はありません。

IMX219 を含むリニア センサの場合は、ISP 内で後段のゲイン処理 (例えばホワイト バランス用のゲイン) を適用する前に、未加工画像ピクセルからブラック レベル またはペデスタルを減算します。ペデスタル値はセンサ ドライバ内にコード化されていますが、センサ ドライバがサポートする各動作モードについて、実際の値を測定します。例えば、IMX219 カメラでは、10 ビット モードではブラック レベルの実測値がおよそ 63 (下図を参照) であり、8 ビット モードでは 16 です。

ターゲットのセンサに対してブラック レベル減算を調整するには、次の手順に従います:

1. カメラ レンズを完全に覆い、黒の RAW 画像を取得します。
2. 「プラグイン」ドロップダウン メニューから Black Level Subtraction を選択します。
3. 「参照」ウィンドウで RAW 画像を選択すると、その画像は「プレビュー」ウィンドウに黒く表示されるはずです。
4. 「RAW ファイル」ウィンドウに未加工イメージを指定し、次のように「プロセス プラグイン」をクリックします。
5. 測定された黒レベルは、右上のウィンドウの「高度パラメータ」タブに表示されます。

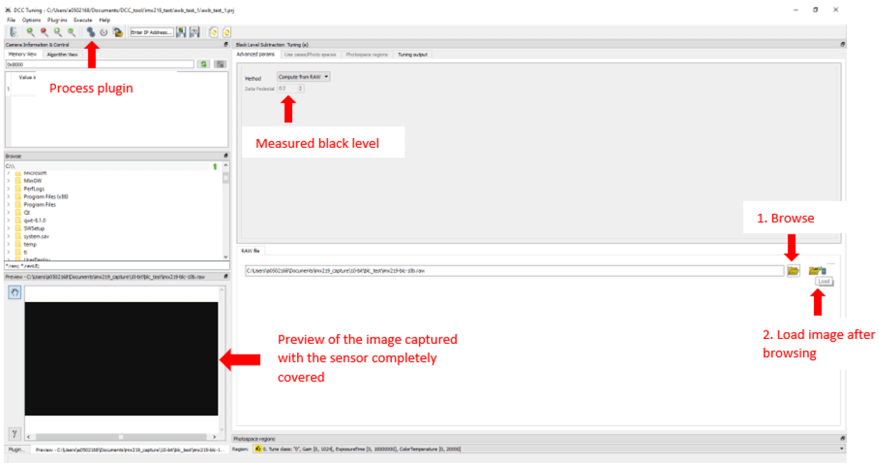

図 7-4. ブラック レベル減算チューニング

WDR センサの場合、ブラック レベル減算は通常、WDR のデコンパンディングおよび再コンパンド処理と組み合わせて実行され、単一の参照テーブル (LUT) を生成します。これはチューニング ツールの CFA + WDR プラグインで実現できます。WDR センサに関する詳細は、当該プラグインのユーザー ガイドを参照してください。

プラグインのチューニングが完了したら、「DCC プロファイル バイナリをエクスポート」ボタンをクリックして、このプラグイン用の出力 XML ファイルを生成します。生成された XML ファイルは、セクション 7.1 で作成したプロジェクト フォルダ内の .plugoutdir\XML フォルダに保存されます。Black Level Subtraction の場合、出力 XML ファイルは imx219_viss_blc.xml だけです。初期設定から生成された同じ XML ファイルを置き換えます。次に、Python スクリプトを再実行して、DCC バイナリ ファイルの生成の説明に従って、新しい DCC バイナリ ファイルを生成します。次の手順では、新しく生成された DCC バイナリを用いてストリーミング品質を改善します。以下の各セクションで各プラグインのチューニングを行った後、この操作を完了します。

各プラグインのチューニング後の画質向上を確認するには、新しく生成された DCC バイナリ ファイルを使用して、ISP 処理済みの静止画像を取得します。例えば、IMX219 用のブラック レベル減算プラグインをチューニングした後は、新しいバイナリ ファイルを使用して、以下の GStreamer パイプラインを実行します:

```
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"
```

図 7-5 はブラック レベル減算のチューニング前 (初期設定を使用) の取得画像例を示しており、図 7-6 はブラック レベル減算のチューニング後の画像を示しています (黒の見え方を比較してください)。

図 7-5. ブラック レベル減算前の画像

図 7-6. ブラック レベル減算後の画像

7.4 ハードウェア 3A (H3A)

H3A は、自動露出 (AE)、自動ホワイト バランス (AWB)、自動フォーカス (AF) アルゴリズム用の画像統計情報を収集するためのハードウェア IP ブロックです。IMX219 や OX05B1S のような固定焦点カメラでは、AE と AWB のみが対象となります。初期設定の生成に使用される ctt_def_xml_gen.py スクリプトは、すでに AE および AWB アルゴリズム用に H3A を適切に設定しています。したがって、固定焦点カメラでは H3A に関して追加のチューニング手順は不要です。

注

Python スクリプトは、H3A、AE、および AWB を適切に設定できるように、BIT_DEPTH を入力パラメータとして受け取ります。したがって、このパラメータは正しく設定する必要があります。

Python スクリプトは、リニア センサ向けに、以下の方法でセンサまたは H3A のデータ フローを設定します:

1. デフォルトでブラック レベルが 0 であると仮定します (上記のブラック レベル減算ステップでは、実際の黒レベルを測定して更新する必要があります)
2. センサのビット深度に基づき、入力ピクセルを 16 ビット ISP 内部フォーマットの MSB 側にシフトします
3. リニア センサのピクセルの上位 (最大 10 ビット) を H3A へ送信します
4. 指定されたセンサ分解能に従って H3A を構成します
5. AE および AWB が正しく動作できるよう、同じ H3A 構成を AE、AWB アルゴリズムに提供します i

H3A 設定の微調整が必要な場合は、SoC の TRM およびチューニング ツールのユーザー ガイドに従ってください。

7.5 PCID

OX05B1S を含む RGB-IR センサの場合は、まず PCID プラグインをチューニングし、AWB、CCM、NSF4 などのキャリブレーションに使用する出力 16 ビット Bayer 未加工画像を生成します。

PCID をチューニングするには、異なる照明条件下で未加工画像を取得する必要があります。これらの照明条件の詳細については、異なる照明条件での未加工画像の取得を参照してください。以下は、このプラグインの調整の概要です。詳細については、ヘルプ メニューの PCID プラグイン ガイドを参照してください。

- 16 ビットの未加工入力画像を読み込みます
- 適切なパラメータ値を入力します
  - 入力形式と出力形式を選択する際は、出力形式が設定済みのパイプラインと一致していることを確認します。例えば、OX05B1S のカラー フォーマットは BGGR です。入力形式は「2:BGGIr」に設定し、出力形式は「1:IR 位置で R を補間し、R 位置で B を補間」を選択し、GStreamer パイプラインは BGGR として構成する必要があります。
- プラグインを実行して、IR 減算を適用した 16 ビットの出力 Bayer パターンを生成します
- この手順をすべての照明条件について繰り返します
- DCC プロファイル バイナリをエクスポートして、XML ファイルとバイナリ ファイルを生成します

注

RGB-IR センサをチューニングする場合、ブラック レベル値を必要とするプラグインを調整する際は、ブラック レベルを 0 に設定してください。これは IR 減算の段階ですでにブラック レベルが除去されているためです。

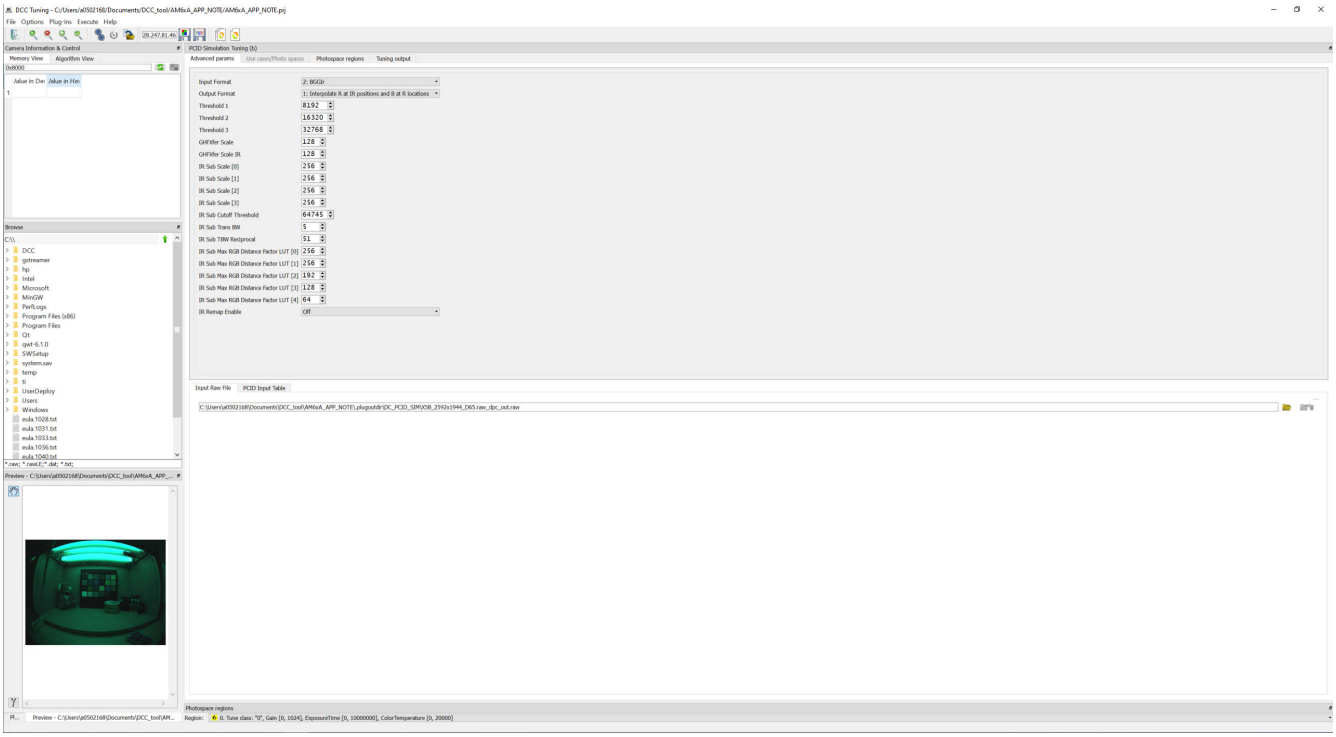

図 7-7. PCID チューニング

以下に、OX05B1S を使用した GStreamer パイプラインの例を示します:

```
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
```

図 7-8 および図 7-9 は、ISP でまったく処理されていないキャプチャの未加工画像の例と、PCID チューニング後の Bayer パターン形式の出力画像の例を示しています。

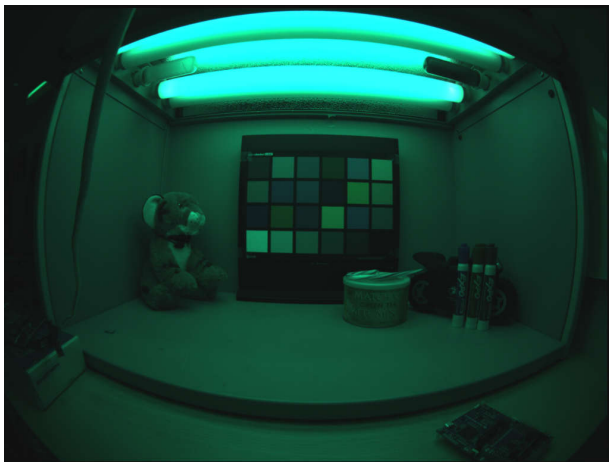

図 7-8. PCID への未加工画像入力

図 7-9. Bayer パターンでの PCID 出力画像

7.6 自動ホワイト バランシング (AWB)

7.6.1 異なる照明条件での未加工画像の取得

AWB をチューニングするには、異なる照明条件下で未加工画像を取得する必要があります。これらの画像は、次のカラー補正チューニングのステップでも再利用されます。必要な画像をキャプチャする手順を以下にまとめます (詳細については、AWB プラグイン ガイドを参照してください):

- 照明条件を次のように設定します: D65、D50、TL84、および A 光源 (一つずつ)
- ライト ボックス内で、カラー チェッカー チャートを正立させ、カメラの FOV の中央に配置します
  - 一行目の左上がブラウン パッチ、最終行の右下がブラック パッチになるように配置されていることを確認します
- 各照明条件で、露出した未加工画像をキャプチャします
  - 自動露出調整が安定するまで待ちます (モニターのライブ映像が十分に明るく、白パッチに明らかな飽和やクリッピングがない状態。この処理は通常数秒で行われます。)
  - 初期設定でライブ ビデオ ストリーミングを実行します
  - 未加工画像は、次の 2 つの方法でキャプチャできます:
    - ストリーミング用の GStreamer パイプラインを停止します。その後、新しい GStreamer パイプラインを実行して未加工画像を取得します (GStreamer の実行間でセンサの露出設定は変更されません)。
    - AM62A EVM と PC が同じイーサネットに接続されている場合、チューニング ツールはイーサネット経由で評価基板から未加工画像を直接取得することもできます。DCC チューニング ツールのツール バーから評価基板の IP アドレスを入力し、以下に示すように未加工画像を取得します。詳細については、チューニング ツールのユーザー ガイドを参照してください。

図 7-10. DCC チューニング ツールのライブ キャプチャ機能

図 7-11 から図 7-14 に、上記のすべての照明条件で適切に露出したキャプチャの例を示します (照明の色によって生じる色の違いに注意してください):

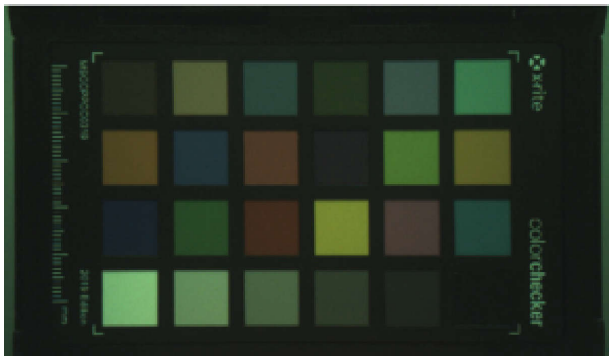

図 7-11. D50 照明下で撮影したカラー チャート画像

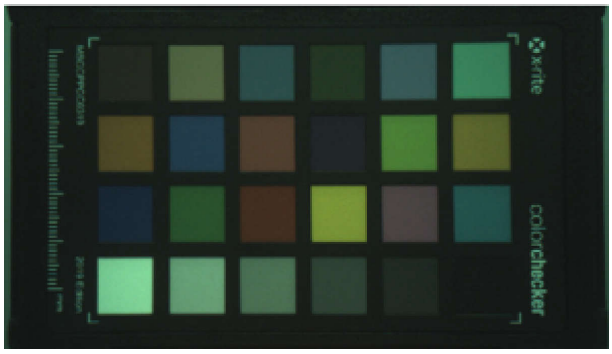

図 7-12. D65 照明下で撮影したカラー チャート画像

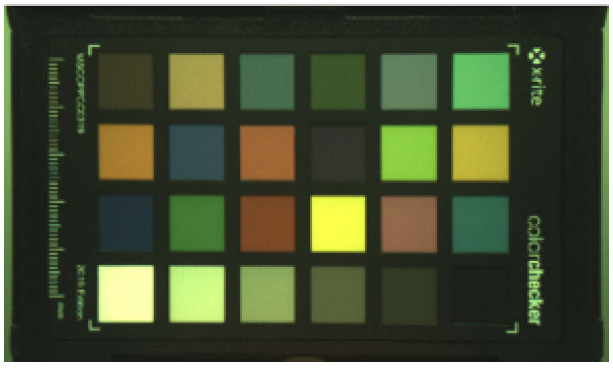

図 7-13. TL84 照明下で撮影したカラー チャート画像

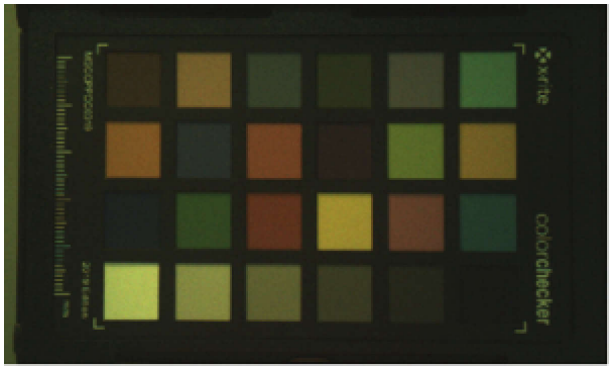

図 7-14. A 光源下で撮影したカラー チャート画像

7.6.2 AWB の調整

未加工画像がキャプチャされたら、プラグイン ドロップダウン メニューから AWB チューニングを開始します。「参照ファイル」タブで未加工画像を一枚ずつ読み込み (詳細は AWB プラグイン ガイドを参照)、その後、画像ごとにパラメータ値を入力します:

- 色温度:未加工画像を撮影する際に使用した値である必要があります。たとえば、D65 照明条件の場合、値は 6500 です。
- 露出、ゲイン、および絞り:これらの値は AWB では使用されません。そのため、無視してかまいません。
- ブラック レベル:ブラック レベル減算プラグインで測定したペデスタル値を設定する必要があります。この例では、IMX219 の測定したブラック レベルは 10 ビット モードで 63 です。

カラー チェッカー チャートの四隅を選択する際には、特に注意してください:

- 左上の角から開始し、カラー チェッカー チャートの四隅を時計回りにクリックします。
- 四隅を選択すると、ツールが 24 個のパッチを自動的に認識し、以下に示すように各パッチの選択状態を表示します。

図 7-15 から図 7-16 に、AWB チューニング用に一つの未加工画像をインポートする例を示します。

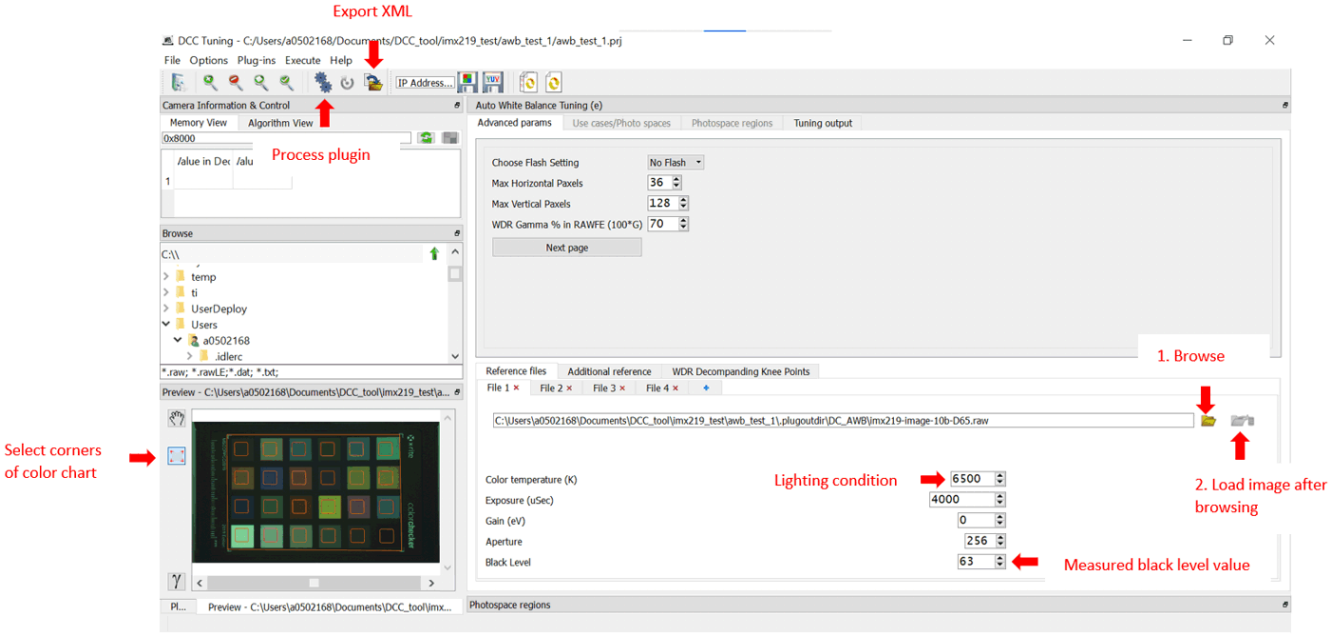

図 7-15. 自動ホワイト バランスのチューニング

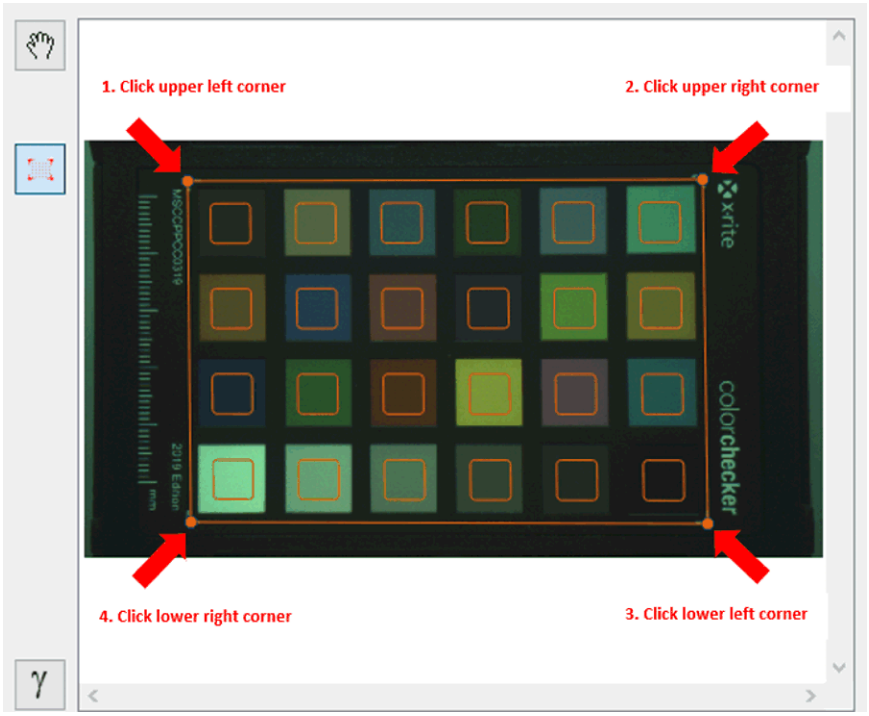

図 7-16. カラー チェッカー チャートのコーナーを選択

すべての未加工画像を読み込んだ後、AWB プラグイン ガイドに従ってチューニングを行います。チューニングが成功すると、参照用の Cb-Cr プロット図が表示されます。以下は、上記に示した未加工画像を使用した結果のプロットです。

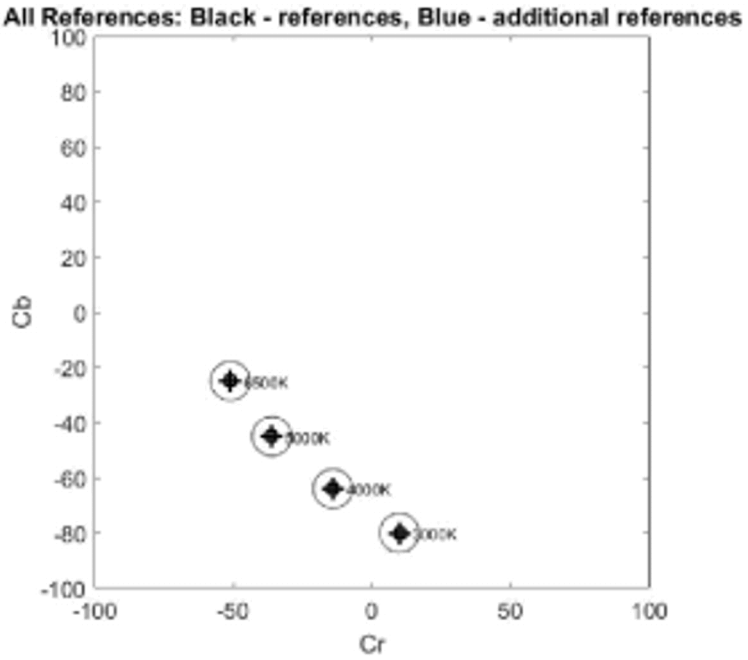

図 7-17. 自動ホワイト バランスのチューニング結果

調整が完了したら、ブラック レベル減算の調整に従って、新しい出力 XML ファイルと新しいバイナリ ファイルを生成します。その後、新たに生成された DCC バイナリ ファイルを使用してストリーミングおよびキャプチャを行います。図 7-18 と図 7-19 は、それぞれ AWB チューニング前と AWB チューニング後に撮影した画像を示しています (AWB チューニング後の画像では、すべてのグレー パッチがニュートラルに表示されます)。

図 7-18. 自動ホワイト バランス調整前の画像

図 7-19. 自動ホワイト バランス調整後の画像

7.7 色補正

正しい色出力を得るには、「色補正プラグイン」をチューニングします。AWB チューニング用に取得した未加工画像は、このプラグインのチューニングに使用できます。以下に、このチューニング手順の概要を示します。詳細については、ヘルプ メニューの「色補正プラグイン ガイド」を参照してください。

- 入力した未加工画像を読み込み、必要に応じてリファレンス画像も読み込みます。チューニング ツールにはデフォルトのリファレンス画像が用意されていますが、ユーザーは任意のリファレンス画像を使用できます。
- パラメータ値を入力し、AWB チューニングと同様にカラー チェッカー チャートの四隅を選択します。
- リファレンス画像に対して同じプロセスを繰り返します。
- プラグインを処理し、「DCC プロファイル バイナリのエクスポート」を実行して、図 7-20 および図 7-21 に示すように XML ファイルとバイナリ ファイルを生成します。

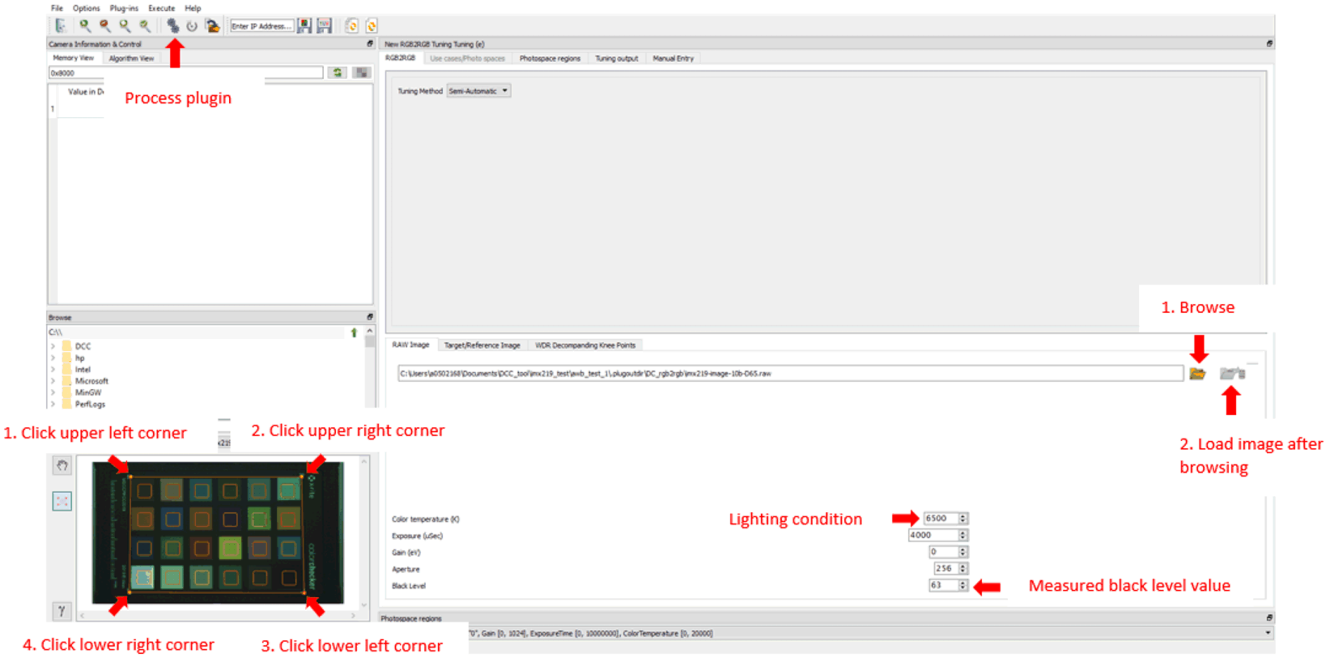

図 7-20. 色補正調整

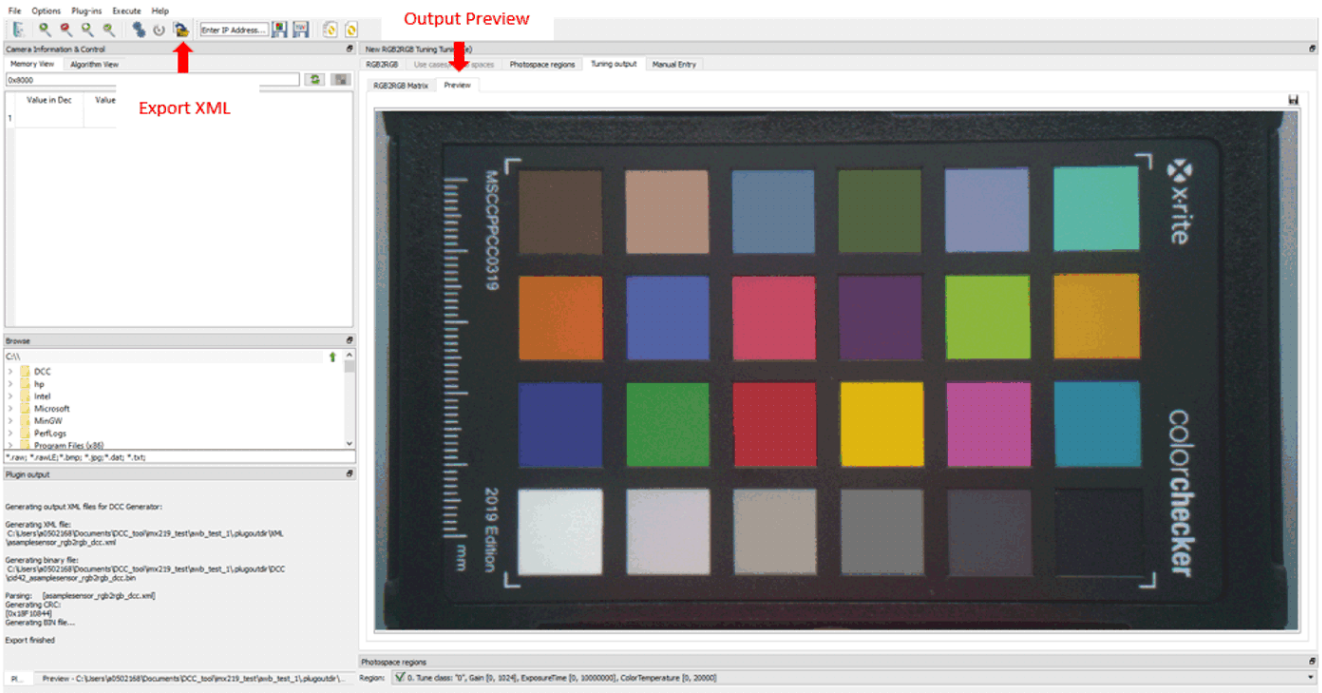

図 7-21. 色補正調整出力

AWB チューニングと同様に、新たに生成した DCC バイナリを使用して画像をキャプチャします。図 7-22 および図 7-23 に、色補正調整の前後の画像を示します (色補正調整後の画像の修正された色に注意してください)。

図 7-22. 色補正前の画像調整

図 7-23. 色補正後の画像調整

8 微調整の実行

ブラック レベル減算、ハードウェア 3A、PCID (RGB-IR センサ向け)、自動ホワイトバランス、および色補正が適切にチューニングされると、ライト ボックス照明条件下において、ISP は最良画質の約 70% ～ 80% を達成できます。画質をさらに向上させるため、以下の追加プラグインをチューニングできます:

- 出力画像のシャープさを向上するためのエッジ強調 (EE)
- 暗所条件でのノイズ抑制のためのノイズ フィルタ 4 (NSF4)
- レンズの歪みを除去するためのメッシュ レンズ歪み補正 (LDC)
- モザイク除去処理および WDR センサ向けのカラー フィルタ アレイ補間 + 広ダイナミック レンジ (CFAI + WDR)
- レンズ シェーディング除去のためのレンズ シェーディング補正 (LSC)

8.1 エッジ拡張 (EE)

EE モジュールのパラメータおよび LUT は、ユーザーが入力したスレッショルドとスロープに基づき、プラグインによって自動生成されます。

注

EE モジュールを AM6xA で有効にするには、次のように ee_mode をイネーブルにします:

- /opt/edgeai-tiovx-modules /tiovx/tiovx_viss_module.c を開き、ee_mode を見つけます
- 以下のように ee_mode を設定して、エッジ強化を有効化します:
  - obj->params.fcp[0].ee_mode = TIVX_VPAC_VISS_EE_MODE_Y8;

- カラー補正の調整後に YUV 画像をキャプチャします。この画像をチューニング ツールの EE プラグインにロードします。
- 「チューニング方法」を「半自動」に設定します
- プラグインを処理して EE チューニング GUI に切り替えます。図 8-1 に EE 調整 GUI および各制御の詳細を示します
- 「有効化」セクションの「オン」または「オフ」ラジオ ボタンを使用して、エッジ エンハンサを有効または無効にします
- 「ゲイン」を「0」に設定して、エッジ シャープナを無効にします
- EE の効果を確認するには、「セミオートマチック チューニング GUI」で「イメージ ソース」のいずれかを選択し、「適用」ボタンを選択します。

図 8-2 および図 8-3 に、EE チューニング前後のチューニング ツール出力の例を示します。

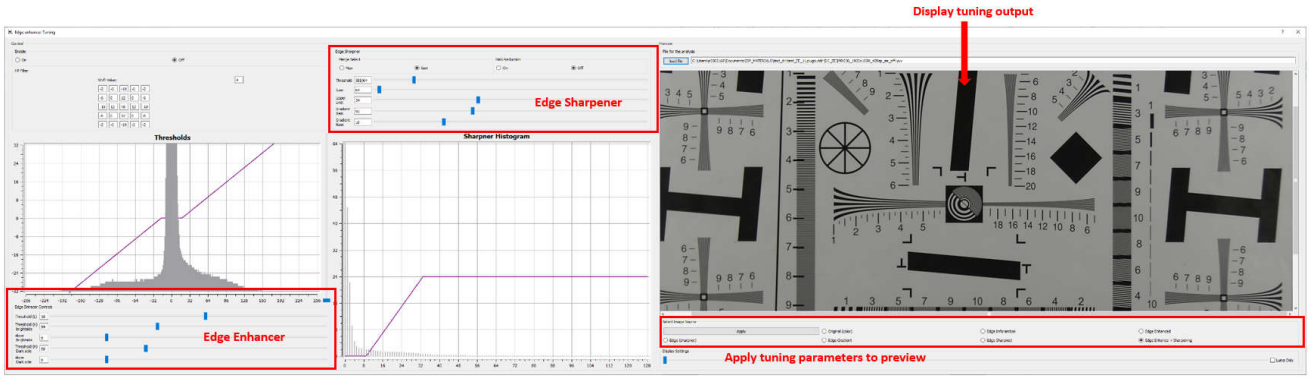

図 8-1. EE セミオートマチック チューニング GUI

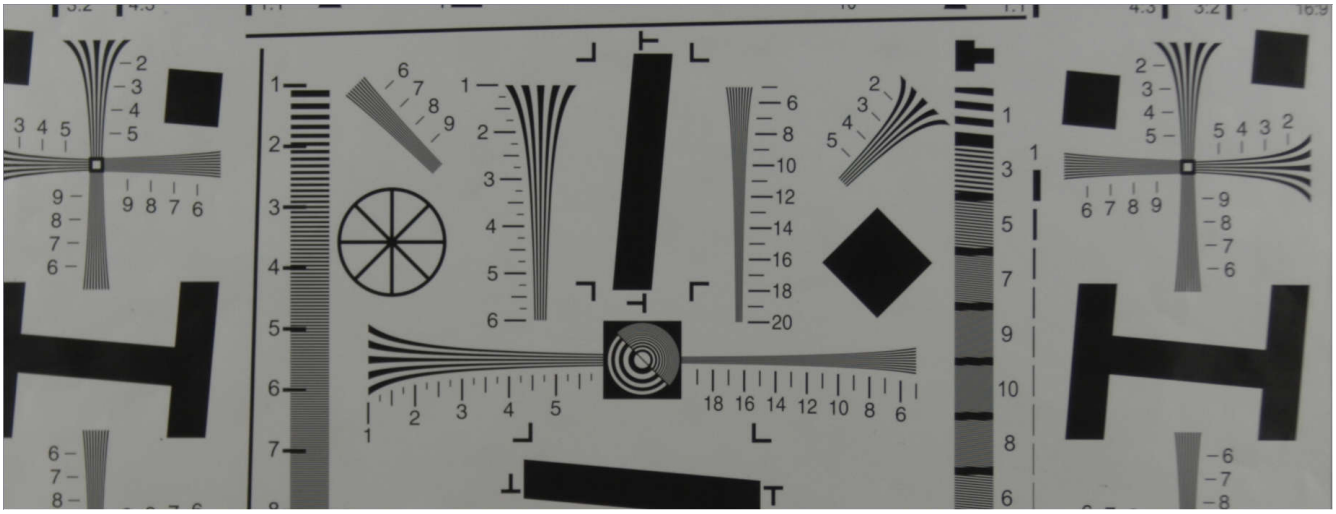

図 8-2. EE 調整前の YUV 画像入力

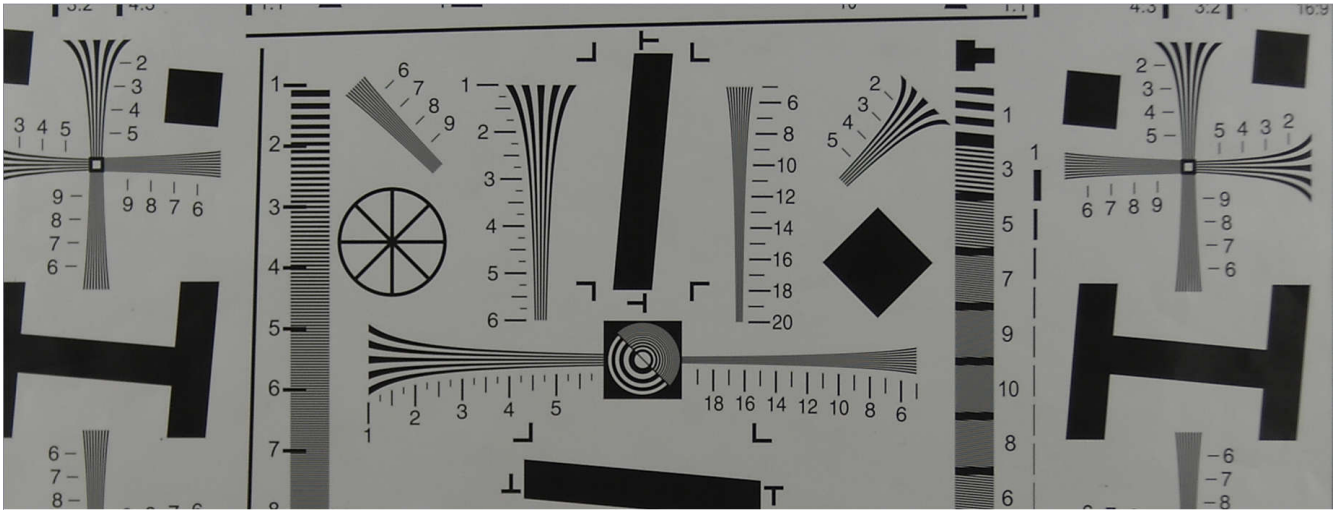

図 8-3. EE 調整後の YUV 画像出力

8.2 ノイズ フィルタ 4 (NSF4)

ノイズ フィルタリングは、画像の細部やその他の重要な特徴をぼかすことなく、画像内のノイズを低減または除去します。以下に、このプラグインのチューニング概要を示します。詳細については、「ヘルプ」メニューの NSF4 プラグイン ガイドを参照してください。

- 未加工画像を読み込みます (例:AWB チューニングで使用した未加工画像のいずれか)
- 「チューニング方法」を「半自動」に設定します
- AWB チューニングで行ったとおり、カラー チェッカ チャートの四隅を選択します (左上の角から開始し、時計回りに選択)
- RAWFE デコンパンダリング LUT を読み込みます。lut_rawfe_pwl_vshort.txt スクリプトで構成ファイルを生成すると、XML 出力フォルダ内に lut_rawfe_pwl_vshort.txt が出力されます
- RAWFE デコンパンダリング LUT を 12 ビット LUT で On に設定します
- プラグインを処理し、NSF4 チューニング GUI へ切り替えます。図 8-4 に、NSF4 調整 GUI および各制御の詳細を示します
- ディスプレイ コントロールでパラメータを調整します
  - X/Y 範囲:プロット領域を拡大します

- ノイズ チャートの「NF チューニングプロット」には、ノイズ サンプルが小さな円として表示されます。青色の曲線は、THR チューニングで設定した現在の X および Y の THR 値に基づいてプロットされます。スライド バーを動かすか数値を入力して、ノイズ サンプルの円と青色の曲線が一致するように THR (Y) 値を手動で調整します。
- 全体の強度を 1.00 より大きくしないでください。線がぼやける場合は、ノイズ フィルタリングが過度に強い可能性があるため、全体の強度を下げます。
- NFS4 の効果を確認するには、「NF を実行」を選択し、「出力画像」オプションを有効にすると結果が表示されます。
- 出力が好みに合わせて調整されるまで、前の手順を繰り返します。
- 調整結果に問題がなければ、「完了」ボタンをクリックしてチューニングを保存します。「DCC プロファイル バイナリのエクスポート」を実行すると、出力用の XML ファイルおよびバイナリ ファイルが生成されます

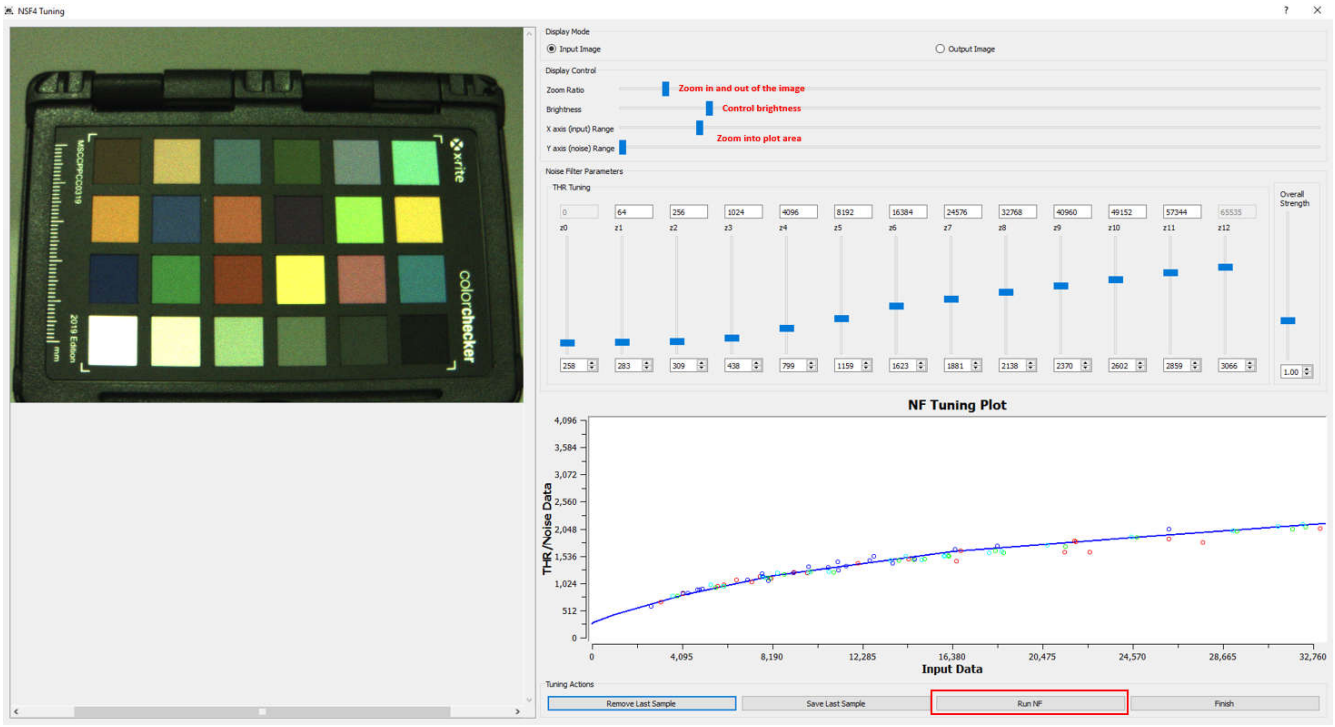

図 8-4. NSF4 チューニング GUI

今回の改訂では取り上げていませんが、追加で実施可能なチューニング項目は以下のとおりです:

- メッシュ LDC: この FAQ をご覧ください
- CFAI + WDR
- LSC

9 ライブ チューニング

ライブ チューニング機能により、チューニング ツールはイーサネット インターフェイス経由で実行時にターゲット システムと相互作用できます。

AM6xA でのライブ チューニングを有効にするには、以下の手順を完了します:

1. 評価基板のターゲット ファイル システムで、/opt/edgeai-tiovx-modules/CMakeLists.txt に移動します
2. **ENABLE_DCC_TOOL** を「OFF」から「ON」に変更し、DCC チューニング ツールのサポートを有効にします
3. /opt/edgeai-gst-apps/scripts/install_tiovx_modules.sh スクリプトを実行して、モジュールを再構築して再インストールします

9.1 要件

- AM6xA EVM は、イーサネット経由でネットワークに接続する必要があります。
- PC と評価基板は同一のネットワークに接続されている必要があります。PC は評価基板に対して ping を実行できる必要があります。
  - VPN 接続はこれに干渉する可能性があります。ライブ チューニングを実行する前に、PC を VPN から取り外します。
  - 評価基板 Linux コンソールで ifconfig を使用して評価基板の IP アドレスを見つけ、以下のようにボックスに入力します。

図 9-1. 評価基板の IP アドレス

  - (1) センサが未加工画像を AM6xA へストリーミングし、(2) GStreamer 内の自動露出 (AE)、自動ホワイト バランシング (AWB)、およびチューニング サーバーも動作するように、GStreamer パイプラインを実行します。

9.2 サポートされている機能

サポートされている機能を以下のセクションで説明します。

9.2.1 RAW キャプチャ

RAW キャプチャは、キャプチャ ノードの出力から未加工画像を保存します。この画像はイメージ センサの出力で、J7、AM6xA ISP では完全に処理されていません。幅、高さ、ビット深度はセンサ ドライバによって決定されます。

図 9-2. RAW キャプチャ

9.2.2 YUV キャプチャ

YUV キャプチャは、VISS ノードの出力から YUV 画像を保存します。この画像は、LDC またはスケーラ処理前の ISP 出力です。幅、高さ、ビット深度は、VISS ノードによって決まります。

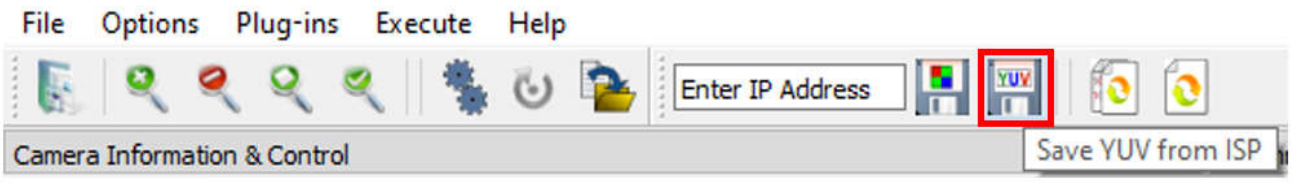

図 9-3. YUV キャプチャ

9.2.3 ライブ DCC の更新

ライブ DCC 更新には、以下の機能が含まれます:

- 現在 GUI ツールで開いているプラグインの DCC バイナリを評価基板に送信します。評価基板に接続されたディスプレイには、画質への瞬間的な影響が見られます。
- コードの変更やアプリケーションの再起動を必要とせずに、新しいパラメータを即座に反映します。

図 9-4. DCC の更新

9.2.4 露出制御

- 「アルゴリズム表示」タブで、AE アルゴリズムによって計算された露出とゲインを読み取ります
- AE モードを手動 (1) に設定することで、AE をバイパスします
  - センサ ドライバの制限に基づいて、手動露出とゲインの値を設定します (このセクションでパラメータを設定します)。

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)

図 9-5. ライブ チューニングの露出制御

9.2.5 ホワイト バランス制御

- AWB アルゴリズムで計算されたゲインと色の温度を読み取ります
- AWB モード = 手動 (1) を設定して、AWB をバイパスします
- センサおよび VISS ドライバの制限に応じて、手動ゲインと色の温度を設定します

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)

図 9-6. ライブ チューニング用のホワイト バランス制御

9.2.6 センサ レジスタ (読み取り / 書き込み)

センサ レジスタの読み取り / 書き込み機能は、イメージ センサ レジスタの読み取りまたは書き込みが可能です。

注

この機能は IMX219 でのみサポートされます。

10 まとめ

このアプリケーション ノートでは、TI の AM62A プロセッサ SDK、画像処理ソフトウェア、DCC チューニング ツールを使用した、AM6xA ISP のチューニングのワークフローについて説明します。RGB のみのチューニング手順は IMX219 カメラを用いて説明し、RGB-IR 特有のチューニングは OX05B1S カメラを用いて説明します。

11 改訂履歴

資料番号末尾の英字は改訂を表しています。その改訂履歴は英語版に準じています。

Changes from Revision * (March 2023) to Revision A (May 2024)	Page
• 2A アルゴリズムの露出設定を追加	7
• RGB-IR センサ向けのチューニング ガイドを追加	15
• エッジ拡張のチューニング ガイドを追加	25
• ノイズ フィルタのチューニング ガイドを追加	26
• ライブ チューニング機能のガイドを追加	28

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、 TI の販売条件 、 TI の総合的な品質ガイドライン 、 ti.com または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。 TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

最終更新日：2025 年 10 月