HiSilicon PQ Tools

User Guide

Issue 12

Date 2016-10-28

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon Technologies Co., Ltd.

Address:	Huawei Industrial Base Bantian, Longgang Shenzhen 518129 People's Republic of China
Website:	http://www.hisilicon.com
Email:	support@hisilicon.com

About This Document

Purpose

This document instructs debugging personnel to adjust the picture quality and implement differentiated adjustment.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3516C	V100R001
Hi3518A	V100R001
Hi3518C	V100R001
Hi3518E	V100R001
Hi3516A	V100R001
Hi3516D	V100R001

Intended Audience

This document is intended for:

- Technical support engineers
- Software development engineers

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 12 (2016-10-28)

This issue is the twelfth official release, which incorporates the following changes:

Chapter 2 GUIs and Functions

Section 2.2.2, section 2.1.4, and section 2.4.4 are modified.

Chapter 3 Application Reference

Section 3.4 is added.

Issue 11 (2016-01-29)

This issue is the eleventh official release, which incorporates the following changes:

Chapter 1 Introduction to the HiSilicon PQ Tools

The title of section 1.2.3 is modified.

Chapter 2 GUIs and Functions

Figure 2-1, figure 2-2, figure 2-6, and figure 2-41 are modified.

Section 2.4.2 ismodified.

Section 2.6.1 and section 2.6.2 are modified.

In section 2.7.2, table 2-26 and table 2-41 are updated. Table 2-42 and table 2-60 are added.

Issue 10 (2015-12-18)

This issue is the tenth official release, which incorporates the following changes:

Chapter 1 Introduction to the HiSilicon PQ Tools

Section 1.2.4 is added.

Chapter 2 GUIs and Functions

In section 2.1.1, table 2-1 and table 2-2 are modified.

Chapter 3 Application Reference

The description of Note in section 3.2 is added, section 3.1.3 is deleted.

Issue 09 (2015-10-30)

This issue is the ninth official release, which incorporates the following changes:

Chapter 2 GUIs and Functions

Section 2.4.5 is modified. Table 2-72 is updated.

Section 2.1.5 and section 2.1.6 are added.

Issue 08 (2015-06-15)

This issue is the eighth official release, which incorporates the following changes:

Chapter 2 GUIs and Functions

The descriptions in sections 2.3.4.4−2.3.4.6 are modified.

Section 2.2 and section 2.3.5 are added.

The description in section 2.6 is modified.

Issue 07 (2015-03-30)

This issue is the seventh official release, which incorporates the following changes:

Chapter 2 GUIs and Functions

In section 2.3.3.1, figure 2-32 is updated.

The descriptions in section 2.3.3.3 and section 2.3.3.4 are updated.

Section 2.3.3.5 and section 2.3.3.6 are added.

In section 2.5.3.1, figure 2-45 is updated.

The description in section 2.5.3.2 is updated.

Issue 06 (2015-02-10)

This issue is the sixth official release, which incorporates the following changes:

Chapter 2 GUIs and Functions

In section 2.2.4, figure 2-13 is updated.

In section 2.3.4.2, figure 2-36 is updated.

In section 2.5.1.1, figure 2-41 is updated.

In section 2.6.1, table 2-3, table 2-5, and table 2-19 are updated.

Sections 2.2.6, 2.3.3.5, 2.5.2, 2.5.3, and 3.1.3 are added. Tables 2-24, 2-25, 2-28, 2-34, 2-56, and 2-57 are added.

Issue 05 (2014-12-20)

This issue is the fifth official release, which incorporates the following changes:

The contents related to the Hi3516D are added.

Chapter 2 GUIs and Functions

Figure 2-6 and Figure 2-8 are updated.

Section 2.1.4.7 is added.

The description in section 2.3.1.1 is updated.

Section 2.3.1.4 is added.

Issue 04 (2014-11-10)

This issue is the fourth official release, which incorporates the following changes:

Chapter 4 FAQs

In section 4.3, "cause analysis" is added.

Issue 03 (2014-09-25)

This issue is the third official release, which incorporates the following changes:

Chapter 2 GUIs and Functions

The description in sections 2.3.1 to 2.3.4 is updated.

Section 2.5 "Other Auxiliary Tools" is added.

In section 2.6.2, contents related to the parameters are updated.

Issue 02 (2014-09-14)

This issue is the second official release, which incorporates the following changes:

Contents related to the ITTP_Stream are deleted.

Issue 01 (2014-08-13)

This issue is the first official release, which incorporates the following changes:

The description of the Hi3516A is added.

Issue 00B03 (2014-04-30)

This issue is the third draft release.

Issue 00B02 (2014-02-26)

This issue is the second draft release.

Issue 00B01 (2013-12-30)

This issue is the first draft release.

Contents

Figures

Tables

1 Introduction to the HiSilicon PQ Tools

1.1 Overview

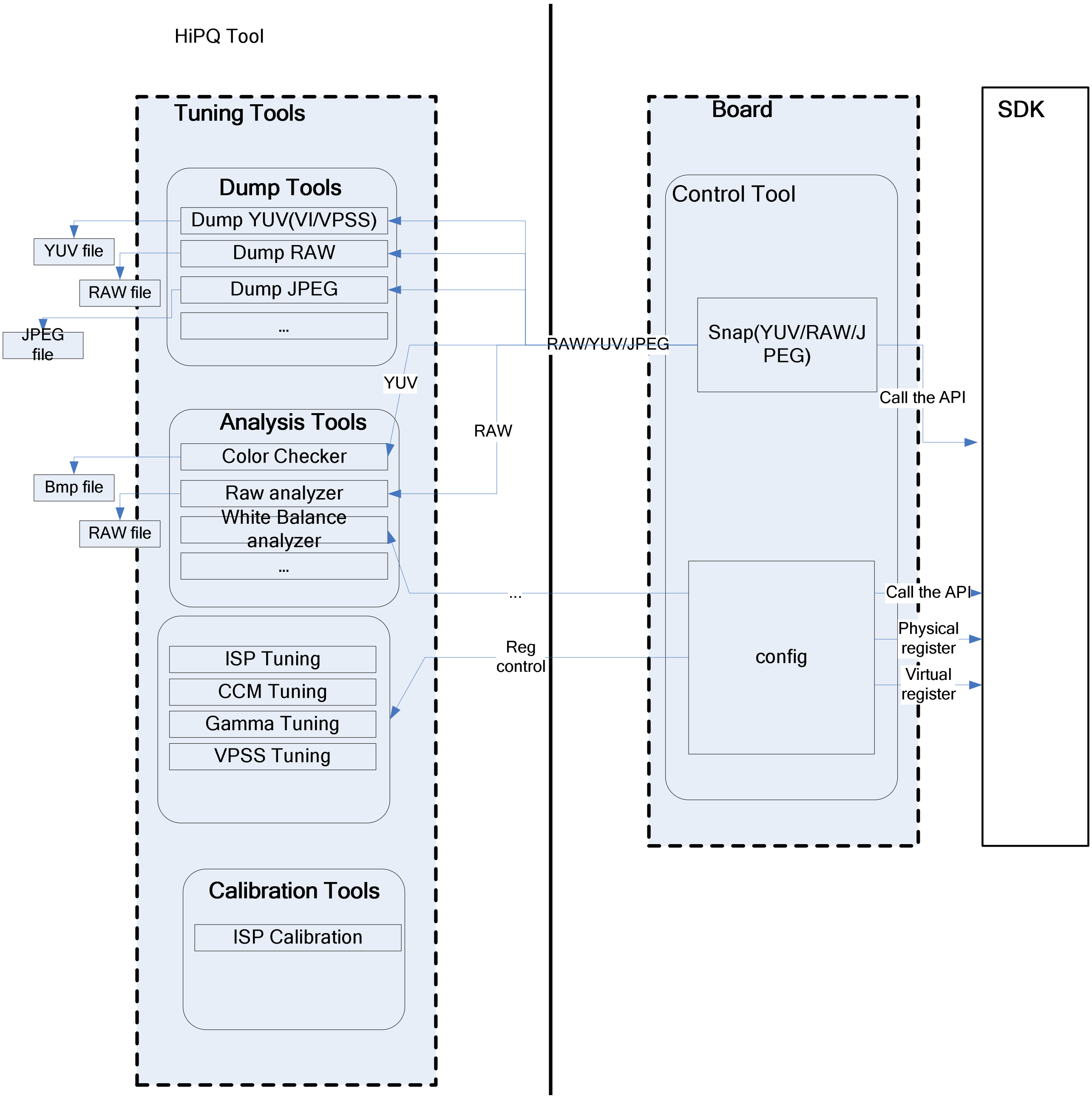
The HiSilicon PQ Tools provides a series of professional picture quality adjustment tools such as the tuning tools, stream tools, and analysis tools.

NOTE

Unless otherwise specified, this document applies to the Hi3516A and Hi3516D.

Figure 1-1 shows the architecture of the HiSilicon PQ Tools.

Figure 1-1 Architecture of the HiSilicon PQ Tools



The HiSilicon PQ Tools is divided into the following parts by usage:

- Calibration tools: Automatically generate initial configurations for picture quality adjustment and adjust the new sensor. This tool is independent currently.
- Online tuning tools: Fine-tune parameters or adjust parameters in differentiated mode. The adjustment takes effect in real time and the picture effect can be viewed by previewing pictures.
- Analysis tools: Work with the online tuning tools to provide common data and charts and support real-time analysis during adjustment.
- Stream tools: Take YUV, RAW, and JPEG snapshots.

- The HiSilicon PQ Tools is divided into PC tools and board tools by the tool delivery form. The HiSilicon PQ Tools include the online tuning tools and analysis tools in the form of plug-ins. The tools on the board include the ittb_control process that is used to adjust parameters online.

1.2 Environment Preparations

1.2.1 Software and Hardware Requirements

The following describes the software and hardware requirements for using the HiSilicon PQ Tools:

- Hardware requirements
  - Board with a Hi3516 (Hi3516A, Hi3516D) or Hi3518 (Hi3518A, Hi3518C, Hi3518E and Hi3516C) series chip, and Ethernet port
  - Desktop or portable computer
  - Network cable (the switches such as the router are required if the LAN is used)
  - Monitor of the computer running the HiSilicon PQ Tools with the resolution of 1024 (width) x 768 (height) or higher
- Software requirements

  Windows XP or later for the computer running the HiSilicon PQ Tools

1.2.2 Physical Connections

The HiSilicon PQ Tools is divided into client tools (tools on the PC) and server tools (tools on the board) that interact over the network. Physical connections can be established in either of the following ways:

- Connecting the computer to the board

  Connect the Ethernet port of the board to that of the computer by using the network cable.
- Using the local area network (LAN)
  - Connect the Ethernet port of the board to that of the router by using the network cable.
  - If a wired network is used, connect the Ethernet port of the computer to that of the router by using the network cable; if a wireless network is used, connect the computer to the wireless hotspot by following the current router settings or the network settings specified by the network administrator.

1.2.3 Installing and Running the HiSilicon PQ Tools on the Board in the Linux System

To burn the SDK to the chip and configure the running environment of board tools, perform the following steps (for details about the Hi3516A/Hi3516C/Hi3518, see the *Description of the Installation and Upgrade of the Hi3516A/Hi3516C/Hi3518 SDK* respectively):

Step 1 Decompress **Hi35XX_PQ_V**X.X.X.X**.tgz** in the SDK and copy it to the board or the server.

Step 2 Run the SDK services.

CAUTION

The HiSilicon PQ Tools is implemented based on the SDK video on demand (VOD) services. To ensure function integrity, ensure that the ISP, VPSS, and VENC services are successfully started before starting the HiSilicon PQ Tools.

Step 3 Run **./HiIspTool.sh** to start the board tool because the tool on the board is started based on the configuration in the **./HiIspTool.sh** script.

The script is divided into two parts:

- Preparations

```
killall ittb_control; #Kill the ittb_control process.
DLL_PATH=${LD_LIBRARY_PATH}:${PWD}/libs; #Specify a library path.
export LD_LIBRARY_PATH=${DLL_PATH}#Export the library.
```

NOTE

- **LD_LIBRARY_PATH** indicates the name of the Linux environment variable that is used to specify a path excluding the default path when the shared library (dynamic link library) is searched. The specified path is searched before the default path.
- The HiSilicon PQ Tools uses the compiled dynamic libraries that are stored in the **libs** folder of the release package. When a function is performed, the corresponding dynamic library is linked. If the dynamic library does not exist in the default path **/lib** or **/usr/lib**, **LD_LIBRARY_PATH** needs to be specified.

- Run the following script to run the executable program:

```
./ittb_control&
```

TIP

If **LD_LIBRARY_PATH** has been set for the external VOD tool, run ittb_control directly.

----End

1.2.4 Using the Board Software in the Huawei LiteOS System

In the Huawei LiteOS system, the software development kit (SDK) is in the format of library. The library is **lib_control.a**. The input function for the control service is int pq_control_main(int argc, char **argv), which is called by using the thread in the main program.

In addition, the configuration file **config.ini** and the current sensor configuration file are packed into a file system by using the tool and burnt into the flash. For details, see in the *Hi35xx Huawei LiteOS Development Environment User Guide* of the SDK.

NOTE

The packaging tool is stored in **osdrv\tools\pc** of the SDK.

1.3 Installing the HiSilicon PQ Tools on the PC

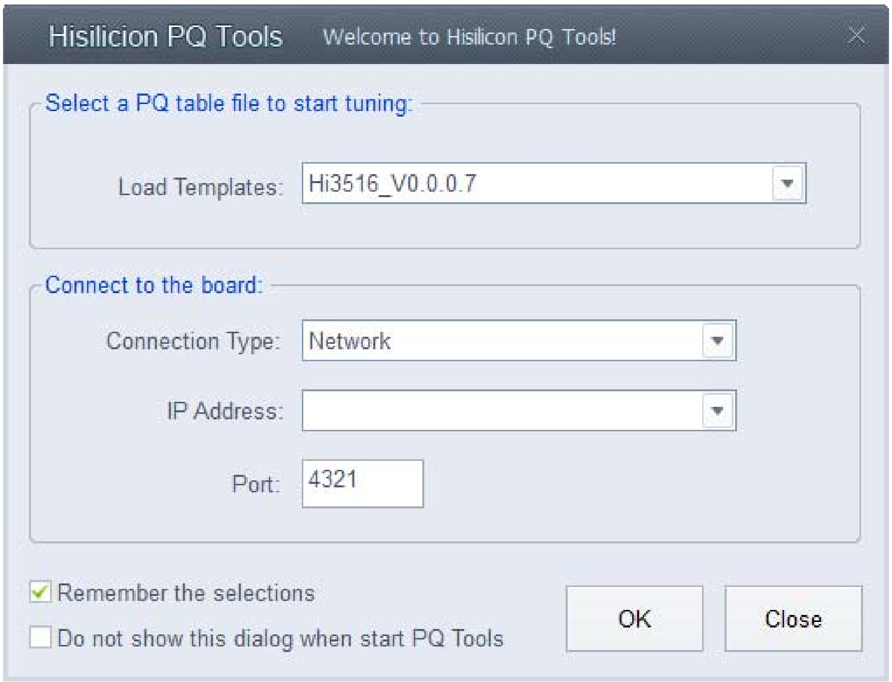
The HiSilicon PQ Tools for the PC is green software. Use the HiSilicon PQ Tools directly after decompressing the HiSilicon PQ Tools package in .zip format by using the WinRAR or WinZip to any directory with the write property.

1.4 Quick Start

1.4.1 Welcome Wizard

Each time you double-click **HiPQTools.exe**, the welcome wizard shown in Figure 1-2 is displayed when the HiSilicon PQ Tools starts, instructing you to rapidly create a picture quality (PQ) table and connecting the board.

Figure 1-2 Welcome wizard



If you want to adjust the picture quality on the welcome wizard, perform the following steps:

Step 1 Choose a PQ table template from **Load Templates** based on the name and version of the chip to be adjusted.

Step 2 Enter the IP address for the board in **IP Address**, enter the port number (**4321** by default) that is specified when the board program is running in **Port**, and click **OK**.

----End

The HiSilicon PQ Tools reads the selected template to generate a PQ table and automatically establishes a network connection between the computer and the board. If the network connection is successfully established, the HiSilicon PQ Tools automatically reads the values of adjustment items from the board.

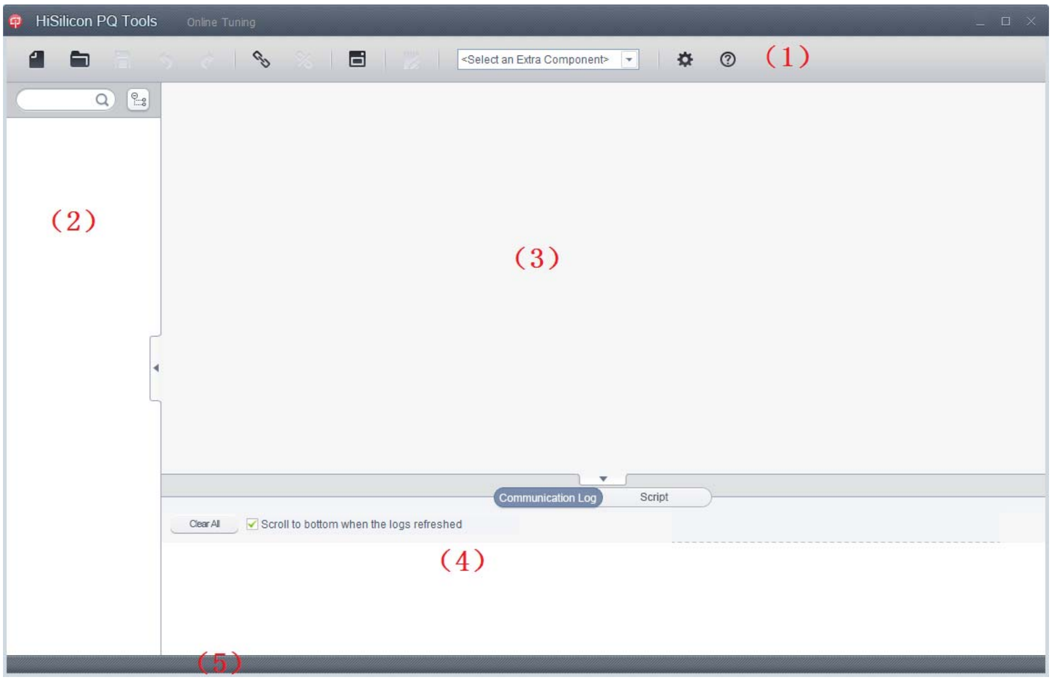
NOTE

- The HiSilicon PQ Tools remembers all entered information by default. If the information is used only for specified debugging and you do not want the HiSilicon PQ Tools to remember it, deselect **Remember the selections**.
- If you do not want the welcome wizard to be displayed with the HiSilicon PQ Tools, select **Do not show this dialog when start PQ Tools**. Then when the HiSilicon PQ Tools starts next time, the main graphical user interface (GUI) but not the welcome wizard is displayed.

1.4.2 Main GUI

Figure 1-3 shows the main GUI of the HiSilicon PQ Tools.

Figure 1-3 GUI



As shown in Figure 1-3, the main GUI is divided into the following areas by function:

(1) Toolbar: Provides shortcuts of common operations.

(2) PQ table panel: Displays all available adjustment items in the opened PQ table.

(3) Adjustment area: Displays the adjustment page of the tree node you clicked on the left.

(4) Advanced function area: Displays communication logs and allows you to enter and run scripts.

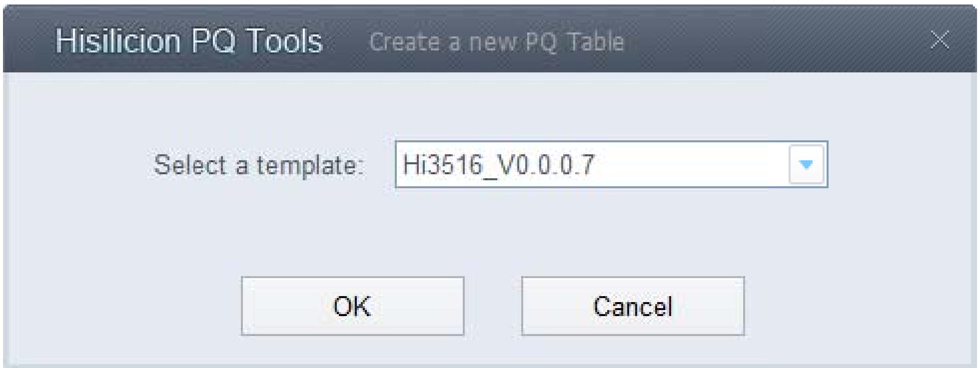
(5) Status bar: Displays the information about operations.

1.4.3 Common Operations

1.4.3.1 Creating a PQ Table

To create a PQ table for picture quality adjustment, click  on the toolbar, choose a PQ table template from the **Select a template** drop-down box in the displayed **Create a new PQ Table** dialog box, and click **OK**.

Figure 1-4 Create a new PQ Table dialog box



After a PQ table is created, its tree structure is displayed in the PQ table panel. For details, see chapter 2 "GUIs and Functions."

1.4.3.2 Saving a PQ Data File

To save the current PQ table and the PQ data read on the board to a file, click on the toolbar, set the file name and save path in the displayed **Save As** dialog box, and click **Save**.

Then a .sav file is generated in the specified path. The file contains the structure of the PQ table.

1.4.3.3 Opening a PQ Data File

To open a saved PQ data file, click on the toolbar, select a file in the displayed **Open** dialog box, and click **Open**.

When the file is opened, its PQ table structure is read and then displayed in the PQ table panel.

1.4.3.4 Undoing and Redoing an Operation

To undo a debugging operation, click on the toolbar; to redo a canceled operation, click .

1.4.3.5 Connecting the Computer to a Board

To connect the computer to a board, click on the toolbar, enter the IP address for a board or choose the IP address for the board connected before from the **IP Address** drop-down box, enter a port ID in **Port**, and click **Connect**. See Figure 1-5.

Figure 1-5 Connect to the Board dialog box

Enter the IP address for a board or choose the IP address for the board connected before from the **IP Address** drop-down box, enter a port ID in **Port** text box, and click **Connect**. Then the HiSilicon PQ Tools attempts to connect to the board. See Figure 1-5.

If the board is successfully connected,  is unavailable and  is available.

1.4.3.6 Disconnecting the Computer from a Board

To disconnect the computer from a board, click  on the toolbar.

1.4.3.7 Opening an Extra Plug-in or Program

The 

drop-down box lists all available extra plug-ins and programs. To open an extra plug-in or program, choose the required one from the drop-down box.

CAUTION

- The opened plug-in is closed when the HiSilicon PQ Tools is exited.
- Some extra programs require the network connection. If you open such a program and the HiSilicon PQ Tools does not connect to the board, the open operation is blocked.

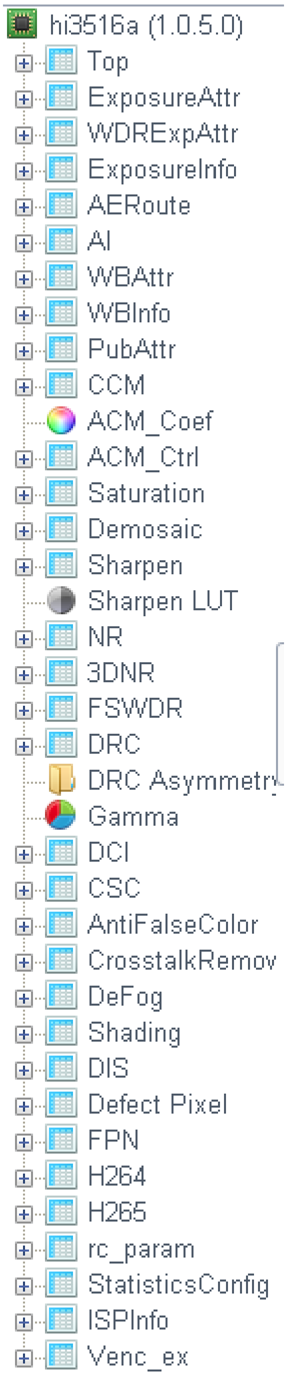
2 GUIs and Functions

2.1 GUIs and Functions of Online Tuning Tools

2.1.1 Opening the Tuning Tools GUI

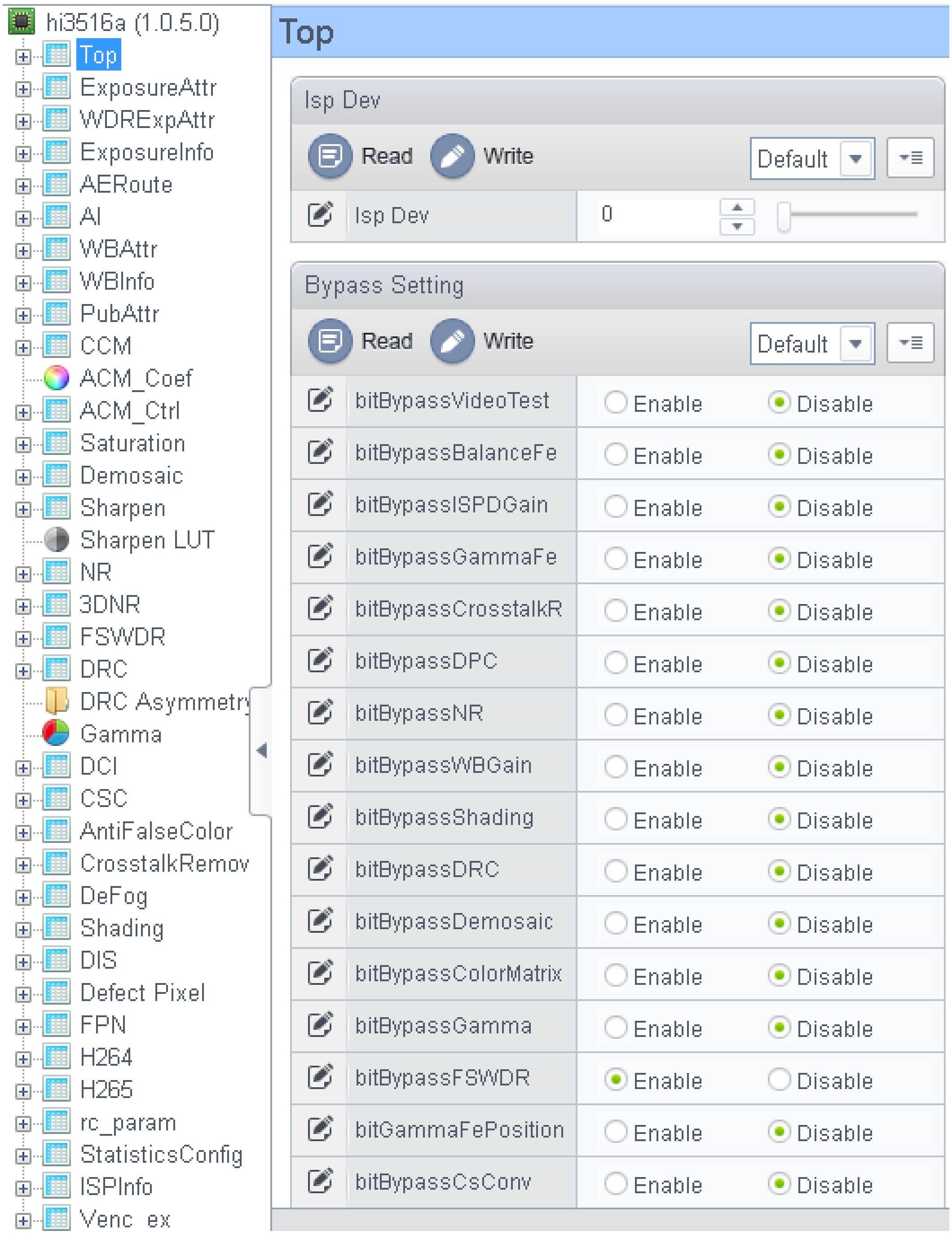
After a PQ table is created or a PQ data file is opened, the structure of the current PQ table is displayed in the PQ table panel on the left. See Figure 2-1.

Figure 2-1 PQ table for the Hi3516A



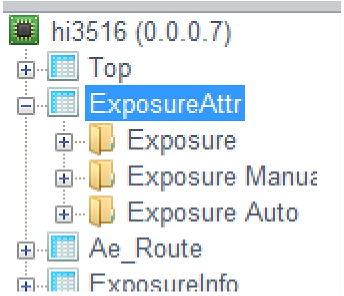
As shown in Figure 2-1, the nodes below **hi3516a (1.0.0.3)** indicate multiple adjustment pages in the PQ table that are organized by the PQ function. When you click a node, its contents are displayed in the adjustment area on the right. See Figure 2-2.

Figure 2-2 Opening an adjustment page of the PQ table



As shown in Figure 2-3, the search box allows you to rapidly search for adjustment items by keyword. Enter a keyword (non-case-sensitive) in the search box, and press **Enter** or click the magnifier icon. Then the HiSilicon PQ Tools locates the next field that contains the keyword and opens the corresponding adjustment page.

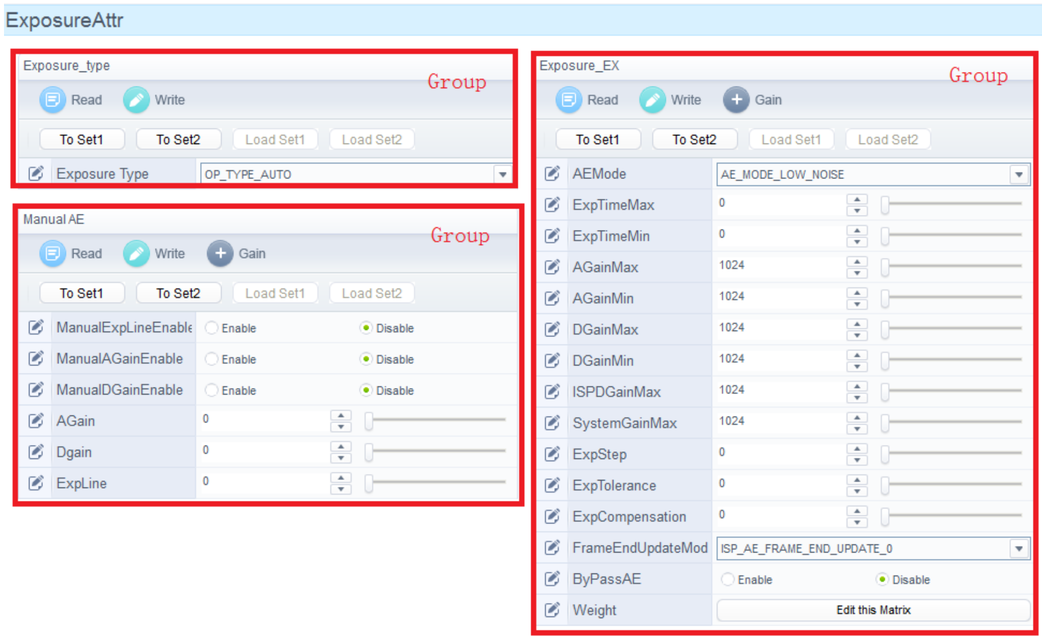
Figure 2-3 Search function



2.1.2 Adjusting Register and Algorithm Parameters

In the PQ table tree, the adjustment pages marked with indicate the register/algorithm parameter adjustment pages. Figure 2-4 shows the PQ GUI of the register/algorithm parameter adjustment page related to exposure.

Figure 2-4 PQ GUI of the register/algorithm parameter adjustment page related to exposure

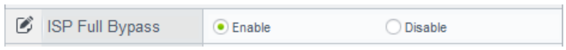
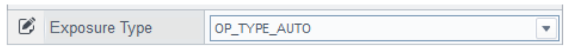
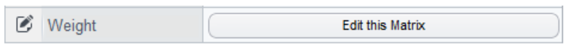
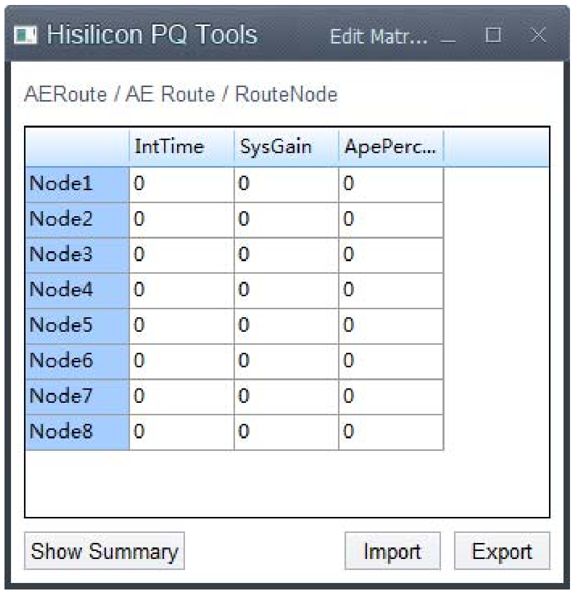


The algorithm parameters in each PQ GUI are grouped by function.

2.1.2.2 Viewing and Modifying Register and Algorithm Parameters

Each group contains register and algorithm parameters. The parameter marked with indicates a parameter with the write property, and the parameter marked with is read-only. The parameters are classified into five types by the internal form and value range and are displayed as different controls. See Table 2-1.

Table 2-1 Types of register and algorithm parameters

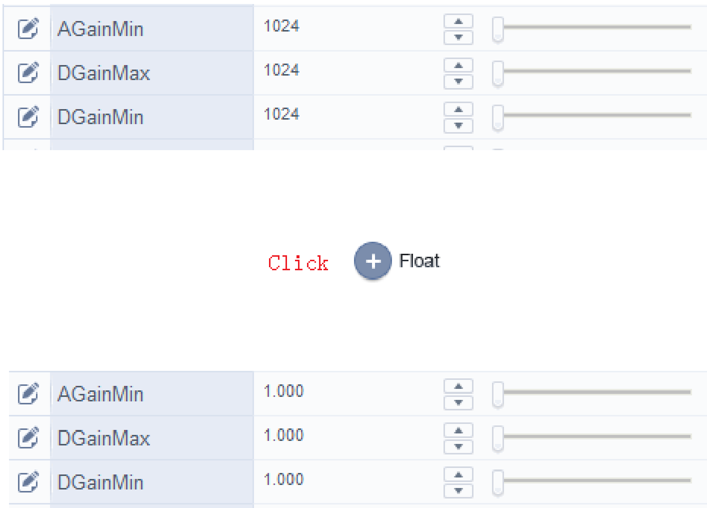
Type	Description	Control Operation
Value	The parameter is a real number with a specific value range.	The current value can be viewed in the text box. The value can be changed in any of the following ways: 1) Enter the value in the text box. 2) Click the arrow on the right of the text box. 3) Drag the slide label on the right. 
Boolean	The value is one of the two parameter values.	The parameter is viewed and set by using the check box. 
Enumeration	The parameter is an enumeration with three or more options.	The parameter is viewed and set by using the drop-down box. 
Matrix	The parameter is a multi-byte sequence.	The following button is displayed on the adjustment page:  After you click the **Edit this Matrix** button, the matrix values are listed in the table control in the displayed dialog box.  After changing the values, click **OK** to close the dialog box. If the matrix is read-only, the values in the table control cannot be changed, and the **View this Matrix** button is displayed.

2.1.2.3 Changing the Floating Point Format

Some register/algorithm parameters (such as the gain parameter in the exposure parameter group) support the floating point format, which is more accurate than the value format.

Clicking  implements switching between the floating point format and the value format. See Figure 2-5.

Figure 2-5 Changing the floating point format



When parameters are in the floating point format, if you click , these parameters will be displayed in the value format.

2.1.2.4 Reading Data from the Board

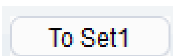
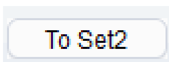
If the board is connected and  in a group is clicked, the HiSilicon PQ Tools reads the values of all parameters in the group from the board and then displays the values.

2.1.2.5 Writing Data to the Board

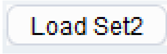
If the board is connected and  in a group is clicked, the HiSilicon PQ Tools writes the values of all parameters with the write property in the group to the board.

2.1.2.6 Caching and Restoring Data

The HiSilicon PQ Tools provides two groups of data cache spaces for each adjustment item group. You can temporarily store adjusted values in the spaces to perform restoration or comparison operations.

- Click  or  to temporarily store the adjusted values of a group.

- Click 

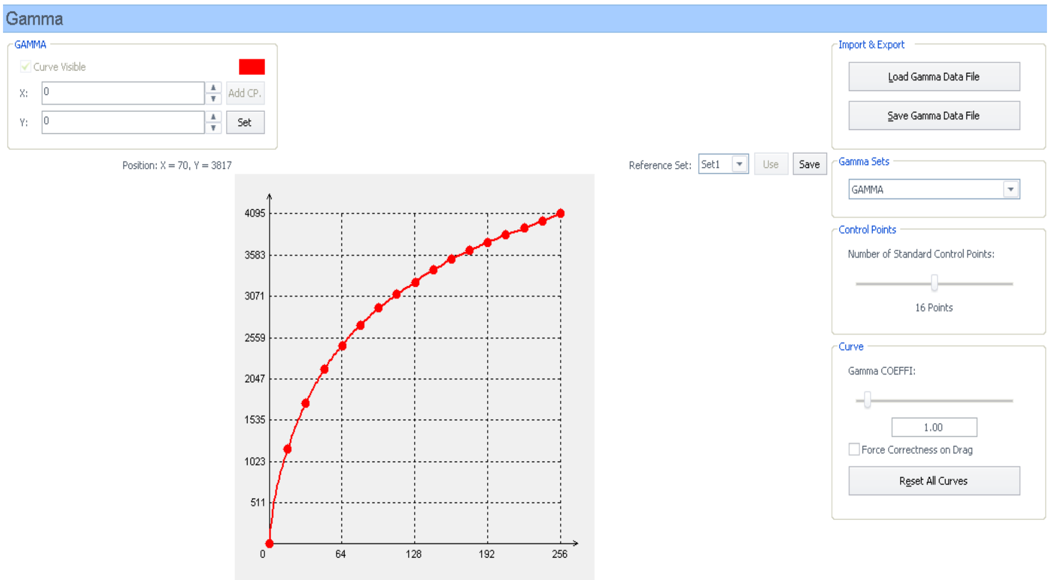
or 

to restore the values temporarily stored to a group.

2.1.3 Adjusting Visual Gamma Parameters

In the PQ table tree, the adjustment page marked with  indicates the visual gamma adjustment page. Figure 2-6 shows the GUI of the visual gamma adjustment page.

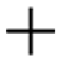
Figure 2-6 GUI of the visual gamma adjustment page



2.1.3.1 Adjusting the Gamma Curve

As shown in Figure 2-6, there is an RGB component curve in the gamma coordinate system in the lower part of the visual gamma PQ GUI. The curve has several control points for changing the curve trend.

When the pointer is moved onto a control point, the pointer is changed to  . In this case, dragging the point upwards or downwards changes the vertical position of the point.

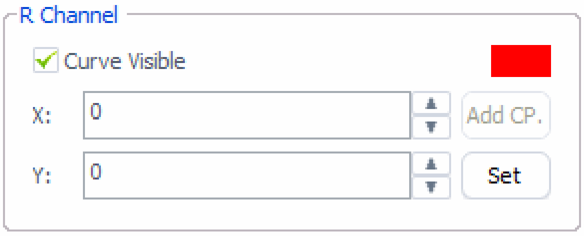
When the pointer is moved onto the curve and is changed to , clicking the curve maps the current pointer position as a point on the curve and changes the point to a control point.

When the pointer is moved onto a control point, right-clicking the control point changes it to a common point. Note that the start and end control points of a curve cannot be changed to common points.

2.1.3.2 Advanced Control Functions of the Gamma Curve

As shown in Figure 2-6, there are curve group boxes in the upper part of the visual gamma PQ GUI for accurately controlling the curve. Figure 2-7 shows the R channel group box for controlling the gamma curve.

Figure 2-7 R channel group box



The following describes the options in Figure 2-7:

- Selecting **Curve Visible** displays the curve in the coordinate system, and deselecting **Curve Visible** hides the curve in the coordinate system.
- After entering a value in the **X** text box, you can view the vertical coordinate of the point on the curve corresponding to the horizontal coordinate.
- If the point defined by the values entered in the **X** and **Y** text boxes is not a control point, clicking **Add CP.** changes the point to a control point.
- If the point defined by the values entered in the **X** and **Y** text boxes is a control point, clicking **Set** after entering a new value in the **Y** text box changes the vertical coordinate of the current control point to the new value.

2.1.3.3 Switching the Gamma Dataset

The HiSilicon PQ Tools provides multiple gamma datasets for each gamma page. You can select a dataset from the **Data Sets** group box on the right. The operations such as editing, saving, or reading the curve, reading data from the board, or writing data to the board are valid only for the current dataset.

2.1.3.4 Reading, Writing, and Saving Gamma Data

You can save, read, or write gamma data in the **Import & Export** group box on the right.

- Clicking **Save Gamma Data File** saves the adjusted gamma data to a local file.
- Clicking **Load Gamma Data File** reads gamma data from a file and displays it as a curve based on the selected dataset.

Data can be saved as a file in floating-point or hexadecimal format. For each floating-point data segment, six decimal places are used.

2.1.3.5 Reading/Writing Gamma Data from/to the Board

You can read/write gamma data from/to the board in the **Data Transfer** group box on the right.

- Clicking 

 reads gamma data from the board and displays them as a curve based on the selected dataset.
- Clicking 

 writes the current gamma curve data to the board.

2.1.3.6 Auxiliary Gamma Functions

The gamma adjustment page shown in Figure 2-6 provides a series of auxiliary functions for facilitating gamma curve adjustment.

- If **Sync Tuning** is selected, when any of the three curves of the R/G/B components is dragged, the other two and their control points change synchronously, that is, the three curves are the same.
- Dragging the slide label in the **Control Points** group box sets the number of control points on each curve.
- Dragging the slide label or entering a coefficient in the text box of the **Gamma COEFFI** group box adjusts the gamma curve. The coefficient ranges from 0.01 to 20.00.
- If **Force Correctness on Drag** is selected, each time a control point is dragged, its maximum vertical coordinate must be less than or equal to the vertical coordinate of its right control point, and its minimum vertical coordinate must be greater than or equal to the vertical coordinate of its left control point.
- Clicking **Reset All Curves** resets all curves to the standard gamma curves with the coefficient of 1.00.
- The **Reference Set** in the upper right part of the curve can be used to temporarily save the adjusted curves. To save a curve, select a reference group ID, and click **Save**. To fetch a temporarily saved curve, select the reference group ID corresponding to the curve, and click **Use**.

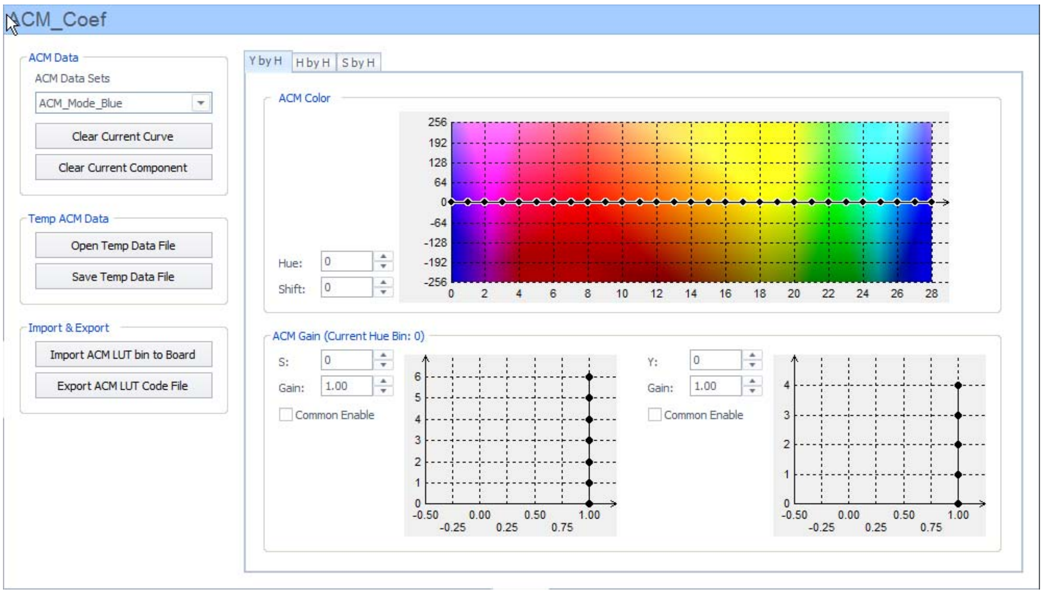
2.1.4 Adjusting Visual ACM Parameters

In the PQ table tree, the adjustment page marked with  indicates the visual automatic color management (ACM) adjustment page. Figure 2-8 shows the PQ GUI of the visual ACM adjustment page.

CAUTION

- Currently this function applies only to the picture quality adjustment of the Hi3516A and Hi3516D chips.
- This function facilitates single-color adjustment, and intuitively reduces the saturation and noise in low illumination.
- The visual ACM adjustment tool can import only the two-dimensional matrix data file exported from it. It cannot import or display the LUT generated by the ACM calibration tool.

Figure 2-8 ACM adjustment page



2.1.4.1 Adjusting the ACM Hue

There is a hue adjustment diagram on the tab page of each hue component (Y by H, H by H, S by H). The control points distributed in the diagram correspond to the hue index values. Dragging the control point changes the corresponding hue index value under the hue component.

In the hue adjustment diagram, right-clicking the control point selects the control point (marked in sky blue). When any of the control points is dragged, all the selected control points move synchronously.

Right-clicking a selected control point changes it to unselected status. Double-right-clicking the hue adjustment diagram resets all the control points to unselected status.

2.1.4.2 Adjusting ACM Gain Value

On the tab page of each hue component, there are also adjustment diagrams of saturation gain and luminance gain, which can be found in the **ACM Gain** group box.

Note that each hue index value in ACM corresponds to a pair of saturation gain value and luminance gain value. Therefore, before adjusting the gains of saturation and luminance, you need to determine the gain value of which hue index is to be adjusted. The hue index can be selected in either of the following ways:

- Enter the index value in the **Hue** text box, which is on the left of the hue adjustment diagram.
- In the hue adjustment diagram, move the pointer onto the control point of corresponding index value.

After the hue index is selected, the HiSilicon PQ Tools displays the gains under this hue index value, and the **ACM Gain** group box prompts the selected hue index value.

In the adjustment diagram of saturation gain and luminance gain, dragging the control points changes the gains of each index value.

If **Common Enable** is selected, when any of the control points is dragged in the gain diagram, other control points move synchronously.

If there are selected points in the hue adjustment diagram when the gains of one hue index value are modified, the gains of all the selected points change synchronously with the gains being modified.

In **ACM Color**, the color is divided into 29 hues. Under **ACM Gain**, the default values of the saturation (S) gain and the luminance (Y) gain are **1**. Select a hue and adjust it up or down. Then the adjustment value of this hue, seven saturation gains, and five luminance gains are obtained. The adjustment value indicates that this hue has the same adjustment amount under the 7 x 5 combinations of the saturation and luminance.

Changing the gain generates different values on the basis of this adjustment value. That is, one combination of the saturation and luminance corresponds to an adjustment amount. In this way, the saturation of this hue is reduced under the specific luminance while is retained under other luminance values.

2.1.4.3 Switching the ACM Dataset

The HiSilicon PQ Tools provides multiple ACM data sets for each ACM page. You can select a dataset from the drop-down box in the **ACM Data** panel on the left. The operations such as editing, saving, or reading the curve, and transmitting data, are valid only for the current dataset.

2.1.4.4 Importing and Exporting ACM Two-Dimensional Matrix Data

To open the ACM two-dimensional matrix data file and restore the data to the GUI, click **Open ACM Data File** on the left. To save the current ACM two-dimensional matrix configuration in the data file, click **Save ACM Data File**.

The two-dimensional matrix data saved and imported is valid only for the selected dataset.

Clicking **Save Temp Data File** saves the current ACM two-dimensional matrix setting to a data file. Clicking **Open Temp Data File** opens the ACM two-dimensional matrix data file and restores the data in the file to the GUI. In this way, you can continue the adjustment on the basis of the last adjustment result.

The visual ACM adjustment tool can import only the two-dimensional matrix data file exported from it. It cannot import or display the LUT generated by the ACM calibration tool.

The saved and imported two-dimensional matrix data is valid only for the currently selected data set.

2.1.4.5 Writing ACM Data to the Board

You can write ACM data to the board by using the read and write buttons. For details, see section 2.1.7 "Reading and Writing Adjusted Data in Batches and Automatically Writing Data".

Clicking 

converts the current ACM configuration to a lookup table and writes it to the board.

2.1.4.6 Importing ACM Lookup Table to the Board

To import the data file of ACM lookup table to HiSilicon PQ Tools, click **Import ACM LUT to Board** on the left of the PQ GUI. After reading the data file, the HiSilicon PQ Tools delivers the data of the lookup table directly to the board.

This function must work with the ACM offline correction function. For introduction to ACM correction plug-in, see section 2.5.1 "Calibration Importer."

2.1.4.7 Exporting Three-Dimensional Matrix Data

You can export the ACM configuration on the GUI as an .h file for use in the code by clicking the **Export ACM LUT Code File** button on the left of the GUI.

The converted three-dimensional matrix lookup table is saved in the .h file.

2.1.4.8 Other Auxiliary Functions

Clicking **Clear Current Curve** in the **ACM Data** group box resets the ACM data in the selected dataset to default status.

Clicking **Clear Current Component** in the **ACM Data** group box resets the data of adjusted ACM component (depends on the selection of the component tab page) in the selected dataset to default status.

2.1.5 Debugging the Sharpen Curve

In the PQ table tree, the PQ page marked with  indicates the PQ page of the visual sharpen curve. Figure 2-9 shows the corresponding GUI of the PQ page for the sharpen curve.

Figure 2-9 PQ page for the sharpen curve

When the sharpen parameters (**LUT Core**, **Strength**, and **Magnitude**) in the left part are adjusted, the sharpen LUT curve in the right part changes accordingly.

After an appropriate curve is obtained through parameter adjustment, click **Apply To Board** to apply all the parameters in the left part to the board. If the automatic write switch is enabled, the tool automatically applies data to the board each time the parameters are modified.

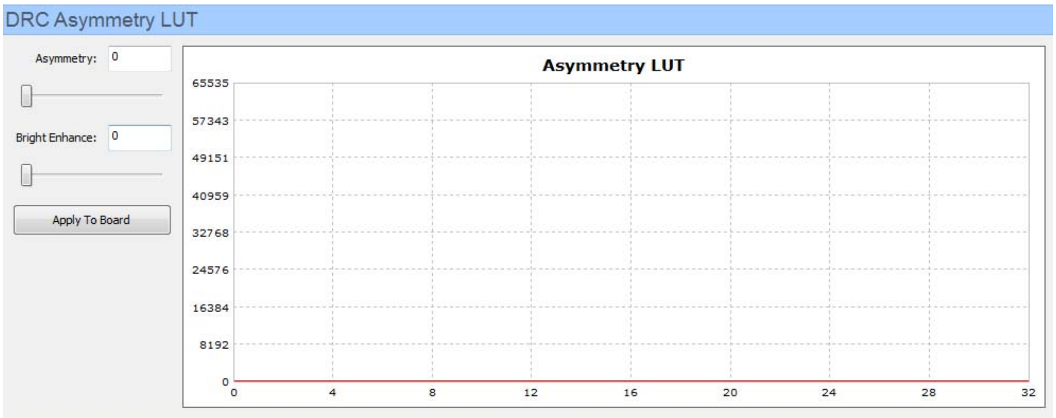
CAUTION

The three configurable parameters on the **Sharpen LUT** page are repeated with those on the sharpen register page, and cannot be updated associatively. Therefore, the parameter write operation is not triggered on the **Sharpen LUT** page when you click **All Write** on the main GUI. The values on the sharpen register page prevail as the written data.

2.1.6 Debugging the DRC Curve

In the PQ table tree, the PQ page marked with indicates the PQ page of the visual DRC curve. Figure 2-10 shows the corresponding GUI of the PQ page for the DRC curve.

Figure 2-10 PQ page for the DRC curve



When the DRC parameters (**Asymmetry** and **Bright Enhance**) in the left part are adjusted, the asymmetry LUT curve in the right part changes accordingly.

After an appropriate curve is obtained through parameter adjustment, click **Apply To Board** to apply all the parameters in the left part to the board. If the automatic write switch is enabled, the tool automatically applies data to the board each time the parameters are modified.

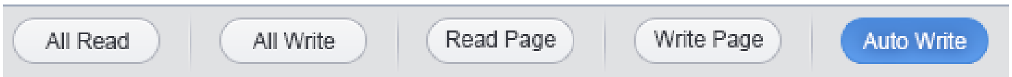
CAUTION

The two configurable parameters on the **DRC Asymmetry LUT** page are repeated with those on the DRC register page, and cannot be updated associatively. Therefore, the parameter write operation is not triggered on the **DRC Asymmetry LUT** page when you click **All Write** on the main GUI. The values on the DRC register page prevail as the written data.

2.1.7 Reading and Writing Adjusted Data in Batches and Automatically Writing Data

Each time a PQ table is created or a data file is opened, the read and write buttons shown in Figure 2-11 are displayed on the corresponding GUI.

Figure 2-11 Read and write buttons



All read and write operations are valid only when the board is connected. The following describes the read and write buttons:

- **All Read** button ( ): Clicking this button reads the current values of all adjustment items in the PQ table from the board. After a PQ table is created (the automatic read function is disabled) or a .bin file is imported, an all read operation is recommended.
- **All Write** button ( ): Clicking this button writes the temporarily stored values of all adjustment items in the PQ table to the board. When a .cfg data file is imported, you can use this function to import the data in the file to the board for viewing the picture effect.
- **Auto Write** button (indicates the disabled status, and indicates the enabled status): If the auto write function is enabled, the value is written to the board each time after an adjustment item with the write property is modified. You are advised to enable this function to ensure that the adjusted value takes effect immediately.
- **Read Page** button ( ): Clicking this button reads the current values of all adjustment items in the active GUI from the board. If the active GUI is a special adjustment page such as the gamma page, clicking this button reads data on this page.
- **Write Page** button ( ): Clicking this button writes the temporarily stored values of all adjustment items in the active GUI to the board. If the active GUI is a special page such as the gamma page, clicking this button writes data to this page.

2.2 Calibration Tool

2.2.1 Overview

The effect of the picture obtained by the instrument may deviate from the expected picture effect due to interference of external factors. The calibration tool takes the picture obtained by the current instrument and the expected picture as inputs and assists the instrument in getting the final output close to the expected effect.

The Matlab is used for some operations of the calibration tool. Therefore, you need to pre-install **MCR_R2012a_win32_installer.exe** before using the calibration tool.

2.2.2 ACM Calibration

2.2.2.1 Basic Principle

The ACM calibration tool sets up the 3D color mapping table of the input picture and the target picture based on the 24-color chart or 140-color chart provided by users. The ACM algorithm adjusts the luminance, hue, and saturation of the picture by pixel based on the mapping table to satisfy user preference.

2.2.2.2 ACM Calibration Process

To implement ACM calibration, perform the following steps:

Step 1 Confirm that the AWB configuration for the camera to be calibrated and the target camera is correct, and the luminance and gray scale of the camera to be calibrated are the same as those of the target camera before ACM calibration, because the picture luminance and gamma curve affect the ACM calibration accuracy.

Step 2 Focus the camera to be calibrated and the target camera on the 24-color chart or 140-color chart and take BMP or JPG pictures at the illumination of about 600 Lux under the D50 or D65 light box.

Step 3 Calibrate the color chart, and generate the basic 3D color mapping table.

Step 4 Fine-tune some colors, generate the final 3D color mapping table (the adjustment of each color is independent), and save the calibration result (.h and .dat files are generated).

Step 5 Verify the calibration effect in various scenarios. If you are unsatisfied with a color, you can fine-tune this color for multiple times by importing the calibration parameters (the previous calibration effect is retained during fine-tuning and is cleared during color chart calibration).

----End

2.2.2.3 GUIs and Functions of the ACM Calibration Tool

ACM Main GUI

Figure 2-12 shows the main GUI of the ACM calibration tool.

Figure 2-12 Main GUI of the ACM calibration tool

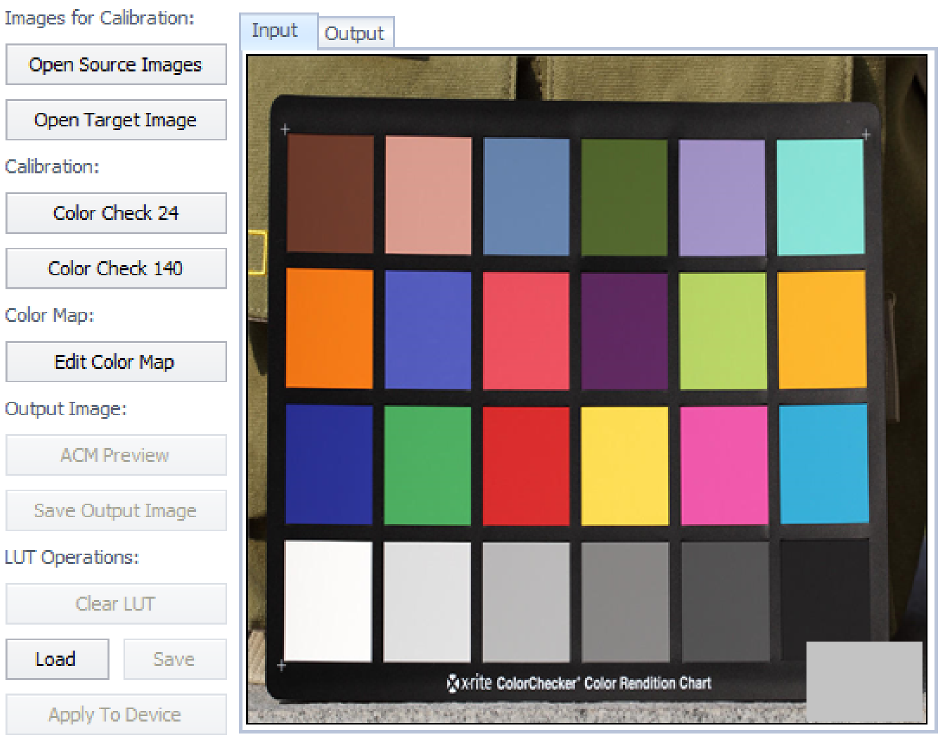
Click the **ACM** tab. Then the GUI for ACM calibration is displayed. As shown in Figure 2-12, the ACM calibration tool consists of the control area, list area, and display area, which are represented by (1), (2), and (3) respectively.

- Control area: Contains the controls of major ACM calibration functions.
- List area: Shows information about the opened input picture.
- Display area: Displays the input picture, target picture, and the output picture obtained after ACM parameters take effect.

Opening the Input Picture

Figure 2-13 shows the GUI when the input picture is opened.

Figure 2-13 Opening the input picture



Click **Open Source Images** in the control area, choose a 24-color JPG or BMP picture as the input picture, and open the picture. Then this picture is added to the list area and information about this picture is displayed in the list area. The opened picture is also displayed in the display area on the **Input** tab page.

Opening the Target Picture

Figure 2-14 shows the GUI when the target picture is opened.

Figure 2-14 Opening the target picture

Click **Import Target Image** in the control area, choose a 256-color JPG or BMP picture as the target picture, and open the picture. The target picture is displayed in the display area on the **Target** tab page.

Calibrating the Color Chart

NOTE

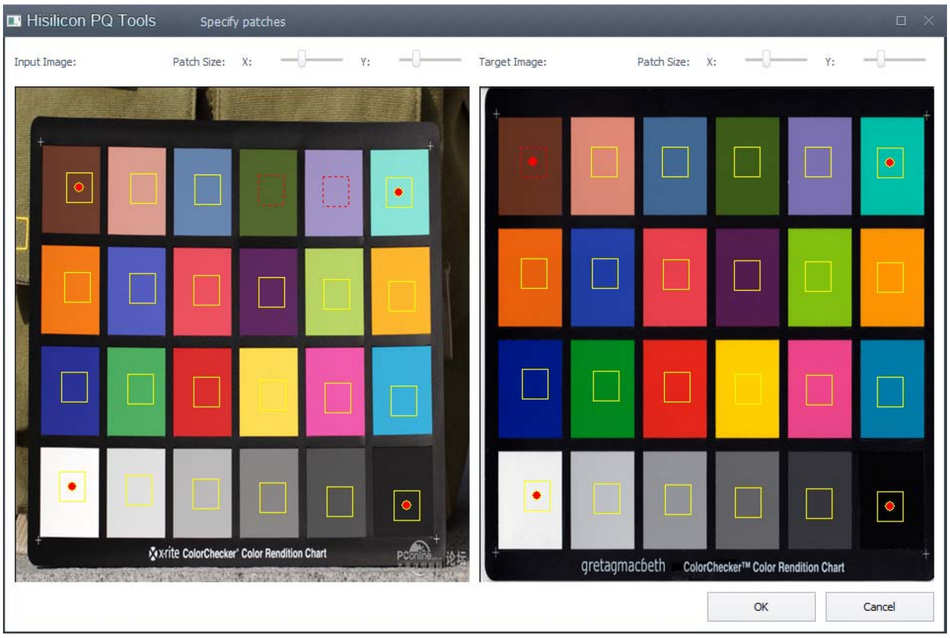
Only one group of ACM calibration parameters needs to be maintained for the calibration tool. The calibration parameters generated after color chart calibration will overwrite the previous calibration parameters, which is equivalent to reconfiguring the previous calibration parameters.

Step 1 Click the corresponding button under **Calibration** based on the scenario of the input picture. To be specific, click **Color Check 24** if the input picture is obtained in the 24-color chart scenario, and click **Color Check 140** if the input picture is obtained in the 140-color chart scenario.

Step 2 Click to select the colors in the displayed dialog box.

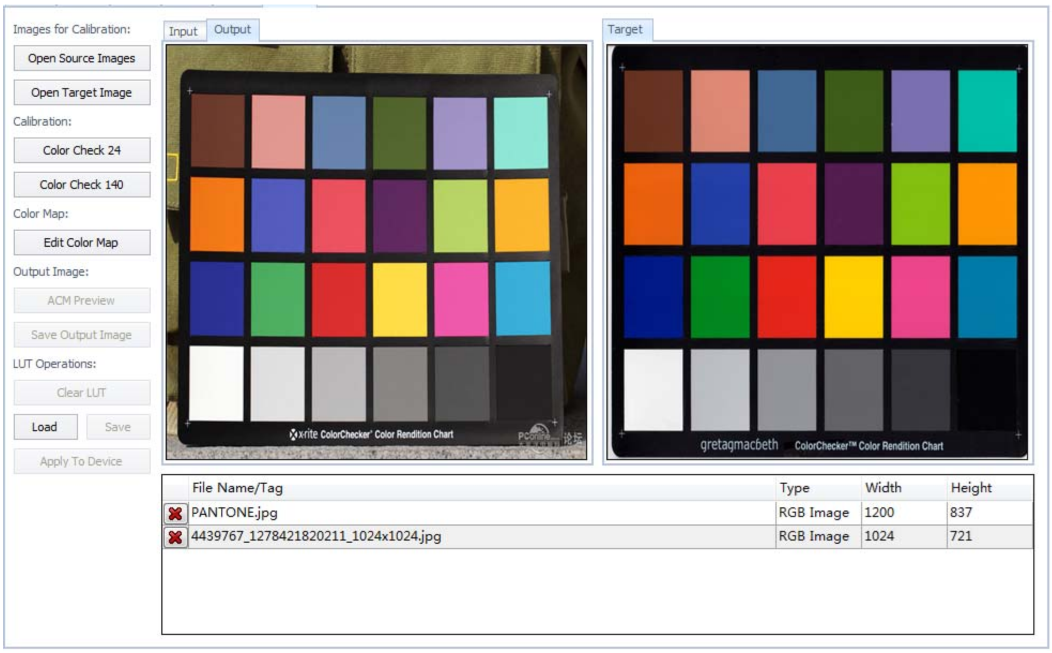
1. Click the red point in the color frame to drag the color frame and adjust the color.
2. Drag the slide labels to adjust the width and height of the color frame. A larger color frame indicates more obtained color block pixels, more accurate calibration result, but more time consumption.
3. Click any position in the color frame. Then the borders of the color frame become dashed lines, indicating that the color chosen by this color frame is not involved in the operation of color chart calibration.

Figure 2-15 Selecting 24-color regions



Step 3 Click **OK** to start color chart calibration.

Figure 2-16 Implementing ACM color chart calibration



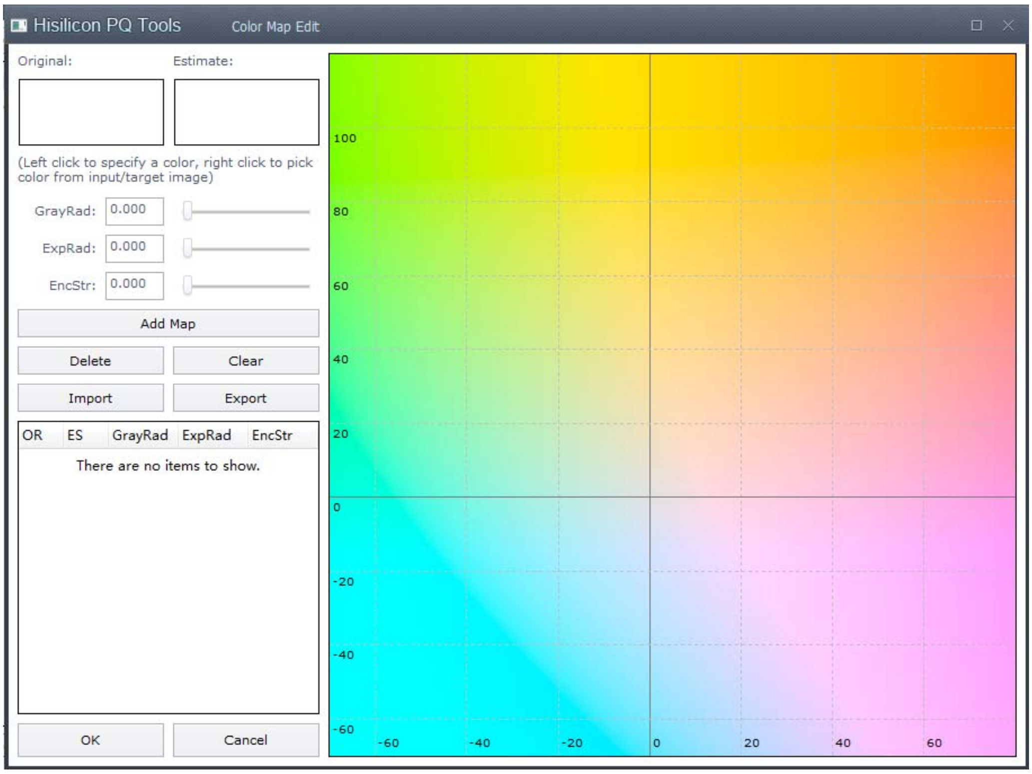
Step 4 After color chart calibration is complete, the ACM calibration parameters are generated in the calibration tool and they take effect on the input picture to generate the output picture. Then the tab is switched to **Output**, and the output picture is displayed in the display area on the **Output** tab page.

----End

Recalibrating Colors

In addition to color calibration with color chart, you can specify the ACM color mapping. You can click the **Edit Color Map** on the ACM calibration page to open a dialog box to fine-tune colors, as shown in Figure 2-17.

Figure 2-17 Recalibrating a color



In the operation area on the left side, the color blocks of the original color (Original) and the expected color (Estimate) are displayed. You can use the color blocks to adjust the color mapping combination:

- Click the **Original** or **Estimate** color block. You can select one color as a new original or expected color using the Windows palette.
- Right-click the **Original** color block. You can select a color region and take its background color as a new original color on the input image on the ACM calibration main page.
- Right-click the **Estimate** color block. You can select a color region and take its background color as a new expected color on the input image on the ACM calibration main page.

After the color mapping combination is determined, the following calibration parameters need to be configured:

- **GrayRad**: Indicates the radius of the gray area. The default value is 1.0. The ACM calibration protects the gray color. When **GrayRad** is set to **0**, the gray color is not protected and the gray area is affected if ACM is implemented on the colors with low saturation. A larger value indicates a larger protected area. When **GrayRad** is set to the maximum value, the ACM does not take effect.

- **ExpRad**: Indicates the radius of the circle that centered on the input color. The default value is 1.0. The ACM calibration ensures that the similar colors are adjusted to the same direction. The colors in the circle are involved in the ACM while the colors outside the circle are not. To enhance one color, increase **ExpRad**. The recommended value range is [1.0, 2.0], which ensures that the ACM effect is more natural. To enhance several colors, decrease **ExpRad**. The recommended value range is [0.5, 1.0], which ensures that the colors are not overlaid.
- **EncStr**: Indicates the strength of the ACM enhancement. The default value is 1.0. A larger value indicates that the ACM enhancement is more obvious.

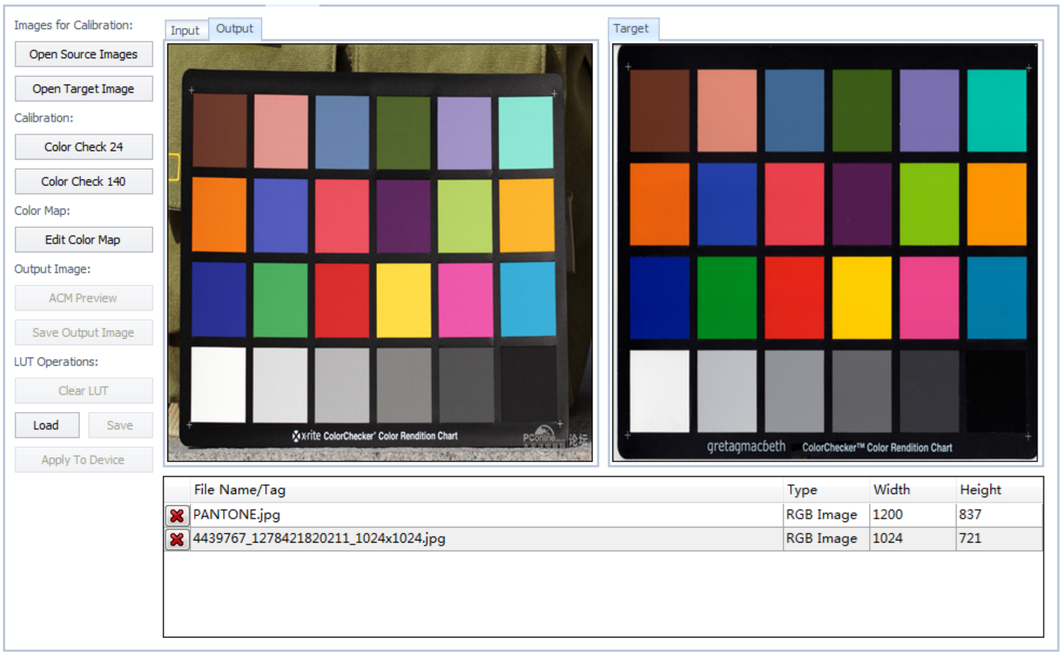
After the color combination and parameters are edited:

- Click **Add Map** if you need to add the color combination to the list below. Click the mapping directly in the list and adjust the color block and parameter editing control on the top side if you need to modify the mapping.

  You can configure whether to validate the color mapping added to the list in the first column (En, Enable) of the list (the added color mapping takes effect when it is selected). For the color mapping that does not take effect, its diagram will not be displayed in the LAB graph on the right, and will be not written into the ACM matrix when the fine-tuning calibration algorithm is running.

  - Click **Deselect** if you need to deselect the currently selected item in the mapping list. After the item is deselected, users' modifications on the mapping colors and mapping parameters will not affect the existing mapping in the list.
  - Click **Delete** if you need to delete the current color mapping in the list.
  - Click **Clear** if you need to clear all the color mappings.
  - Click **Export** if you need to export the current color mapping table to local hard disk.
  - Click **Import** if you need to restore the color mapping restored in the hard disk to the interface.
- Click **OK** in the dialog box. Then recalibration is started. After recalibration, the tab is switched to **Output**, and the output picture is displayed in the display area on the **Output** tab page. As shown in Figure 2-18, you can observe the effect after fine-tuning.

Figure 2-18 Comparing the re-calibrated color in the output picture with the corresponding color in the target picture



Note the following:

- The current ACM calibration result is generated in overlaying mode. The colors in the list are applied from top to bottom and saved in the ACM matrix of the tool.
- When color A is similar to color B, the calibration tool applies the distance priority principle to ensure that the adjustment on color B does not affect that on color A. Therefore, it is normal that the adjustment on color B does not take effect.
- The following steps are recommended to perform color fine-tuning: add only one color mapping each time you open the **Edit Color Map** dialog box. After the effect is conformed, add another color mapping.

Applying Calibration Parameters

Step 1 If you click **ACM Preview**, the ACM calibration parameters in the calibration tool are applied to the currently displayed input picture to generate the output picture.

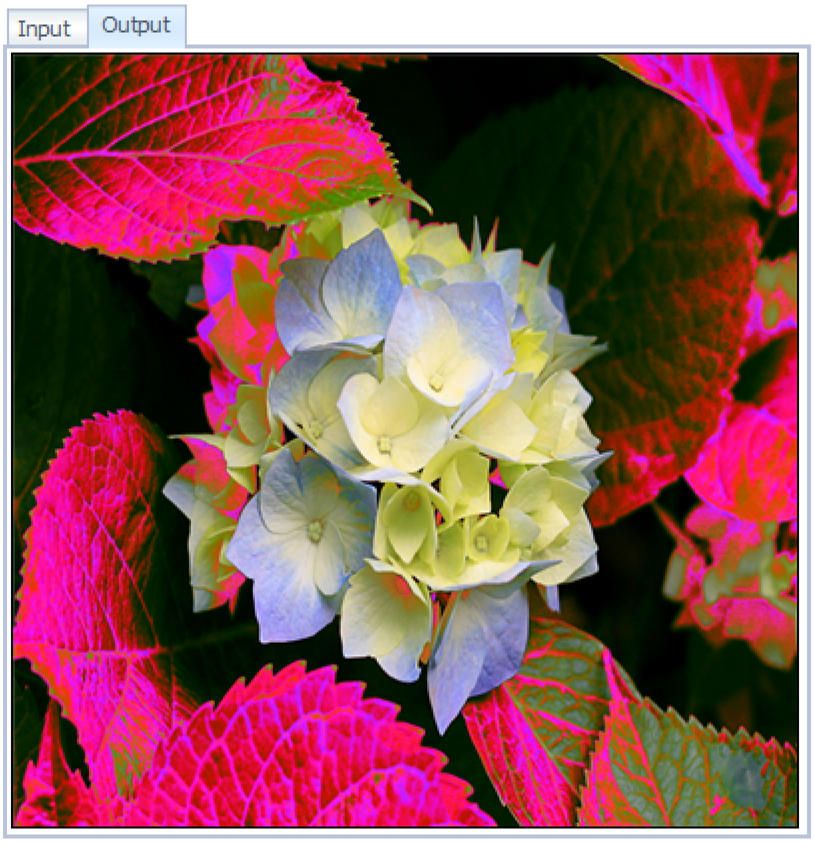
Figure 2-19 shows the input picture before ACM calibration parameters are applied.

Figure 2-19 Input picture before ACM calibration parameters are applied

Step 2 Open a picture to which you want to apply the calibration parameters, and click **ACM Preview**. Then the output picture is generated.

Figure 2-20 shows the output picture generated after ACM calibration parameters are applied.

Figure 2-20 Output picture generated after ACM calibration parameters are applied



Step 3 You can switch between the **Input** tab and **Output** tab to compare the input picture with the output picture.

The comparison of Figure 2-19 and Figure 2-20 shows the ACM capability. However, you are advised not to click **ACM Preview** in applications because this operation may result in noises. It is recommended that ACM be used only to fine-tune colors.

---End

Saving the Output Picture

You can save the displayed output picture as a JPG or BMP picture by clicking **Save Output Image**. The BMP format is recommended because compression loss occurs when the picture is saved in the JPG format.

Saving Calibration Parameters

You can save calibration parameters into a .lut file (which can be imported next time) by clicking **Save** under **LUT Operations**. It is inconvenient to view parameters in a .lut file. Therefore, an .h file with the same name is also generated to make it easy to view parameters.

The saved LUT cannot be displayed on the visual ACM adjustment page and cannot be used for continued adjustment.

Importing Calibration Parameters

You can select the previously saved .lut file and import calibration parameters by clicking **Load** under **LUT Operations**. Then you can fine-tune some colors.

Resetting Calibration Parameters

You can reset calibration parameters to default values by clicking **Clear LUT**.

Switching the Input Picture

You can click the file in the list area to open the corresponding picture and use it as the input picture.

Deleting the Input Picture

You can click the red cross button in the first column of the list to delete the corresponding input picture from the list.

Sending the Calibration Result to the Board

After ACM calibration is complete, click **Apply To Device** to directly apply the generated ACM three-dimensional matrix and gains to the board. The sent result takes effect immediately.

2.3 Advanced Functions

2.3.1 Register Modifier

You can change the register value at a specific address by using the register modifier. The register modifier allows you to read and modify the 32-bit physical or virtual registers.

Clicking  on the toolbar opens the register modifier shown in Figure 2-21.

Figure 2-21 Register modifier

To read the value of a register at a specific address, perform the following steps:

Step 1 Enter an address in the **Address** text box.

Step 2 Select **Physical** or **Virtual**.

Step 3 Click **Read**.

If the register value is successfully read, a value is displayed in the **Value** text box, and the bitwise register modification area on the right is marked based on the value.

----End

To write a value to a register at a specific address, perform the following steps:

Step 1 Enter an address in the **Address** text box.

Step 2 Select **Physical** or **Virtual**.

Step 3 Enter the value to be written in the **Value** text box or mark the bitwise register modification area on the right based on the value to be written.

Step 4 Click **Write**.

----End

CAUTION

- The register address must be an integral multiple of 4; otherwise, it cannot be read and written.
- If an address has been used or defined, it can be read or written. If an address is not defined for a register type or its function is unknown, use this address with caution. After an address is filled, the register type (physical or virtual) must be selected. For example, 0x000A0004 is both a virtual register address and a valid physical address. However, the register at this physical address is invalid or the physical address is specified for another important function. If you select the physical register type and then read and write to this address, the program may crash or other exceptions may occur. Therefore, use such addresses with caution.

2.3.2 Communication Logs

After the board is connected, all data related to the interaction between the computer and the board is recorded as logs and displayed in the **Communication Logs** window. See Figure 2-22.

Figure 2-22 Communication Logs window

The logs that are recorded when the computer and board interact include the following information:

- Log created time
- Communication mode and parameters
- Communication contents (such as the adjustment item being read or written)
- Interaction status (including session startup and session answer)
- Error information if communication errors occur

Clicking **Clear Logs** clears displayed communication logs.

If the opened PQ table or the connection between the computer and the board is abnormal, a communication error may occur. In this case, locate and resolve the issues by following Table 2-2.

Table 2-2 Common communication errors

Error Message	Description
Cannot connect to the board.	Symptom: The board fails to be connected. Solution: Check the connection between the computer and the board.
Device not matched.	Symptom: The device corresponding to the opened PQ table mismatches the connected board. Solution: Change a PQ table.
Version not matched.	Symptom: The device corresponding to the opened PQ table matches the board, but the version of the board program mismatches the PQ table. Solution: Change a PQ table.
Failed to receive response from the board.	Symptom: No response is received from the board. Solution: Check the network connection and the running status of the board program.

2.3.3 Scripts and Batch Processing

The HiSilicon PQ Tools allows you to set register and algorithm parameters in batches by running scripts. Click **Scripts** to open the **Scripts** window. See Figure 2-23.

Figure 2-23 Scripts window

You can enter a script in a line in the text box in the following format:

```
set (<Adjustment page name> / <Adjustment group name> / <Adjustment item
name>) <Value>
```

CAUTION

- For the register and algorithm parameters in non-matrix form, enter only one decimal or hexadecimal value in **<Value>**.
- For the register and algorithm parameters in matrix form, enter multiple required decimal or hexadecimal values in **<Value>** and use commas (,) to separate them.

After entering scripts, click **Run Scripts**. Then the HiSilicon PQ Tools runs the scripts line by line and set the specified register and algorithm parameters.

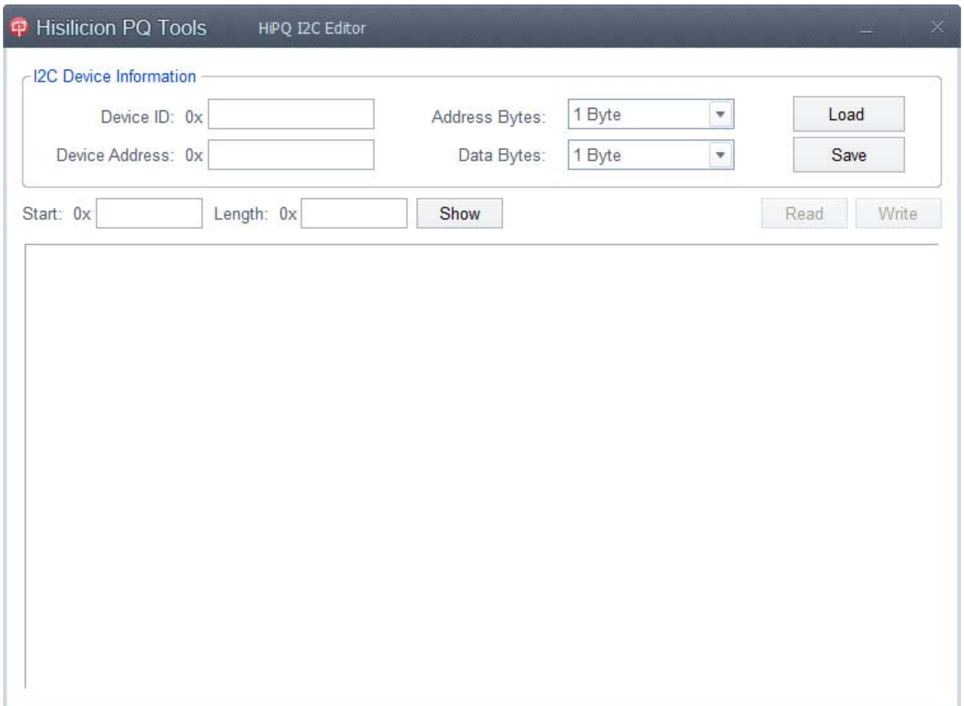
If **Record** is selected, scripts are automatically recorded, and a script in the format of set (<Adjustment page name> / <Adjustment group name> / <Adjustment item name>) <Value> is generated and displayed in the text box each time a register or algorithm parameter is modified.

2.3.4 I²C Editor

You can use the I²C editor to read or write to parameters of some peripheral components.

Choose **HiPQ I²C Editor** from the **Select an Extra Component** drop-down box on the toolbar to open the I²C editor window.

Figure 2-24 I²C editor



I²C Device Information is used for configuring I²C components.

To read the I²C value at a specific address segment, perform the following steps:

Step 1 Enter the ID and address of the I²C device in **Device ID** and **Device Address** respectively.

Step 2 Select address bit width and data bit width in **Address Bytes** and **Data Bytes** respectively.

Step 3 Enter the start address and length in the **Start** box and the **Length** box respectively.

Step 4 Click **Show**. Then the editing box of the register that matches the address is shown below.

Step 5 Click **Read**.

----End

If the I^2C value is successfully read, the HiSilicon PQ tools displays the value of corresponding address in each editing box.

To write the I^2C value at a specific address segment, perform the following steps:

Step 1 Enter the ID and address of the I^2C device in **Device ID** and **Device Address** respectively.

Step 2 Select address bit width and data bit width in **Address Bytes** and **Data Bytes** respectively.

Step 3 Enter the start address and length in the **Start** box and the **Length** box respectively.

Step 4 Click **Show**. Then the editing box of the register that matches the address is shown below.

Step 5 Edit the written I^2C address and modify its value.

Step 6 Click **Write**.

----End

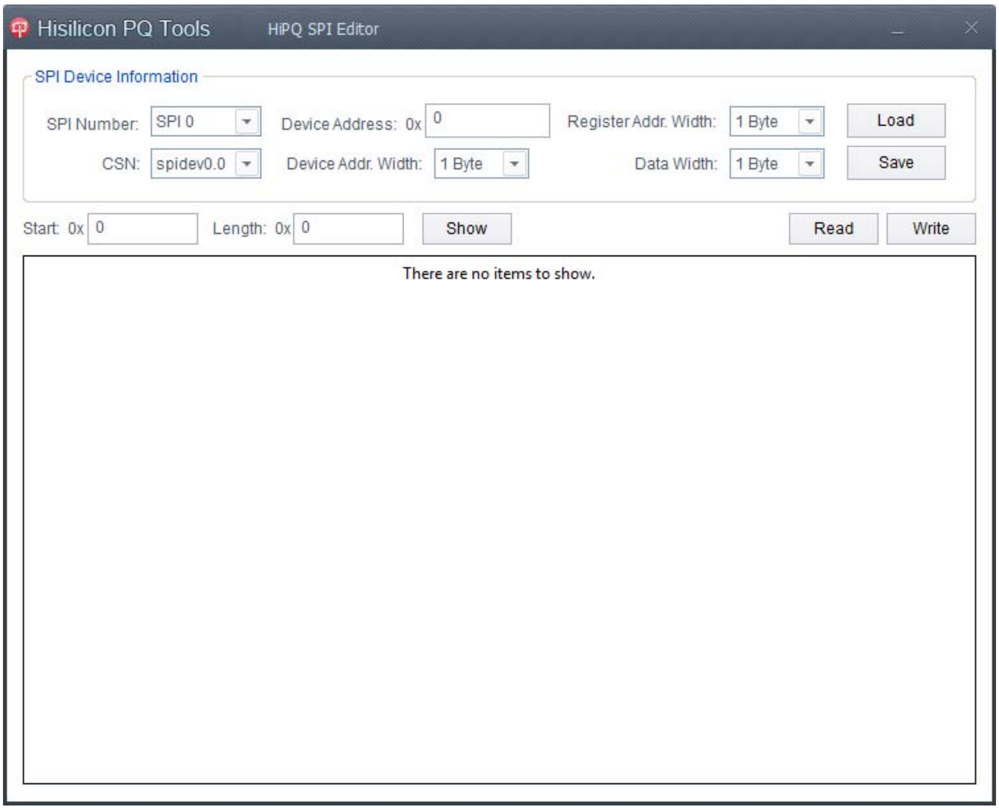
In addition, to import or save the configuration of I^2C components, click **Load** or **Save** respectively.

2.3.5 SPI Editor

You can use the SPI editor to read or write to registers on the peripheral components based on the serial peripheral interface (SPI).

Choose **HiPQ SPI Editor** from the **Select an Extra Component** drop-down box on the toolbar to open the SPI editor window.

Figure 2-25 SPI Editor



SPI Device Information is used for configuring SPI components.

To read the SPI value at a specific address segment, perform the following steps:

Step 1 Select the corresponding device port in **SPI Number** and **CSN**.

Step 2 Enter the device address in **Device Address** and select the address bit width of the device in **Device Addr. Width**.

Step 3 Select the address bit width and data bit width of the register in **Register Addr. Width** and **Data Width** respectively.

Step 4 Enter the start address and length in **Start** and **Length** respectively.

Step 5 Click **Show**. Then the editing box of the register that matches the address is shown below.

Step 6 Click **Read**.

----End

To write the SPI value at a specific address segment, perform the following steps:

Step 1 Select the corresponding device port in **SPI Number** and **CSN**.

Step 2 Enter the device address in **Device Address** and select the address bit width of the device in **Device Addr. Width**.

Step 3 Select the address bit width and data bit width of the register in **Register Addr. Width** and **Data Width** respectively.

Step 4 Enter the start address and length in **Start** and **Length** respectively.

Step 5 Click **Show**. Then the editing box of the register that matches the address is shown below.

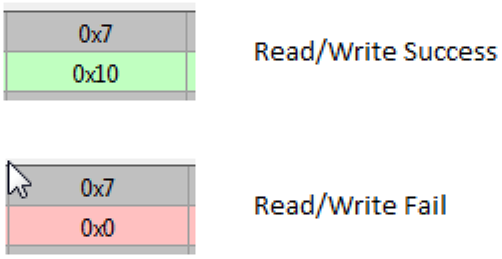
Step 6 Change the register value.

Step 7 Click **Write**.

----End

If the SPI value is successfully read or written, the background color of the cell indicating the register value is changed to green. Otherwise, the background color of the cell indicating the register value is changed to red, as shown in Figure 2-26.

Figure 2-26 Results of the SPI Editor



In addition, to import or save the configuration of SPI components, click **Load** or **Save** respectively.

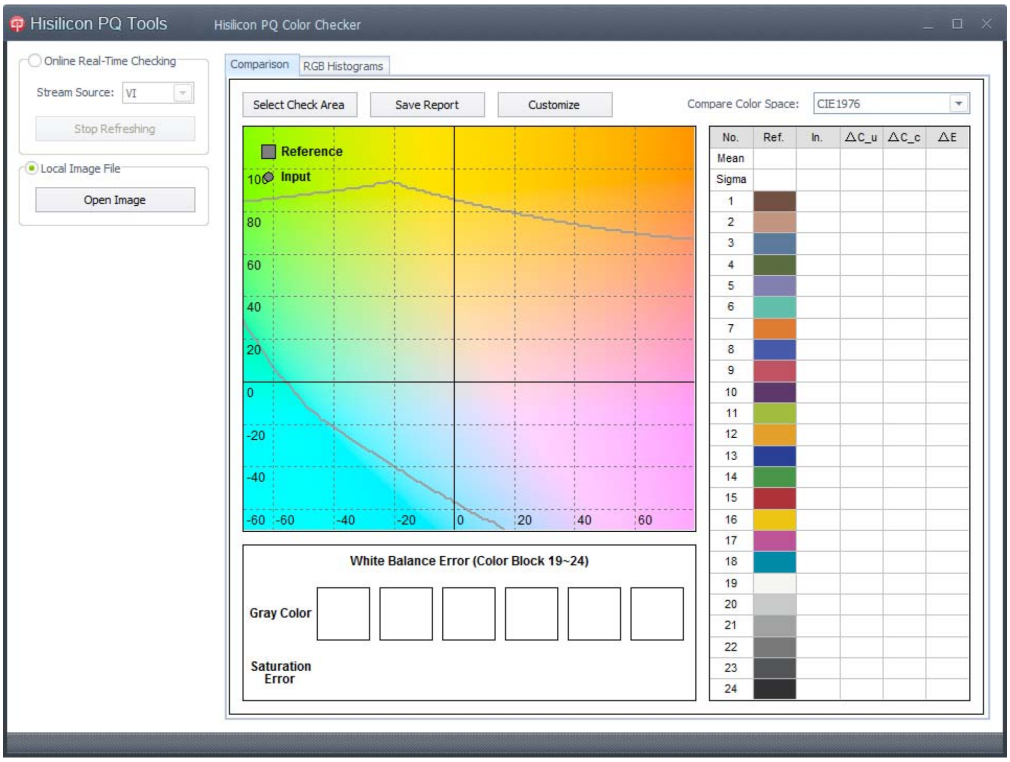
2.4 GUIs and Functions of Analysis Tools

2.4.1 Color Checker

2.4.1.1 GUI

Choose **HiPQ Color Checker** from the **Select an Extra Component** drop-down box in the main GUI of the HiSilicon PQ Tools. The **HiSilicon PQ Color Checker** window shown in Figure 2-27 is displayed.

Figure 2-27 HiSilicon PQ Color Checker



The **Color Checker** window displays the following analysis data:

- Comparison chart of the position and distance of coordinates in the L*a*b* matrix for the standard color and sample color
- L*a*b* color space distance (ΔC*ab, including the value after the saturation value is corrected) obtained by using the CIE1976 algorithm when the luminance is not considered, and the L*a*b* color space distance (ΔE*ab) when the luminance is considered
- L*a*b* color space distance (ΔC00, including the value after the saturation value is corrected) obtained by using the CIEDE2000 algorithm when the luminance is not considered, and the L*a*b* color space distance (ΔE00) when the luminance is considered
- White balance error analysis
- Average values of the R, G, and B components for the sampled color
- Y component statistics histogram (only for the online mode)
- RGB component statistics histogram, including the RGB histogram and histogram of each component.

2.4.1.2 Selecting a Data Source

The color checker supports online color analysis when the board is connected and can read a local picture for color analysis.

To implement online color analysis, perform the following steps:

Step 1 Focus the camera on a standard PANTONE 24-color chart.

Step 2 On the main GUI of the HiSilicon PQ Tools, open a PQ table (ensure that the chip model and version match the board).

Step 3 Connect the HiSilicon PQ Tools to the board.

Step 4 On the main GUI of the color checker, select **Online Real-Time Checking**.

Step 5 Select a data source (**VI, VPSS** or **VO**) from the **Stream Source** drop-down box.

----End

To analyze the color of a local picture, perform the following steps:

Step 1 On the main GUI of the color checker, select **Local Image File**, and click **Open an Image File** to select a local picture.

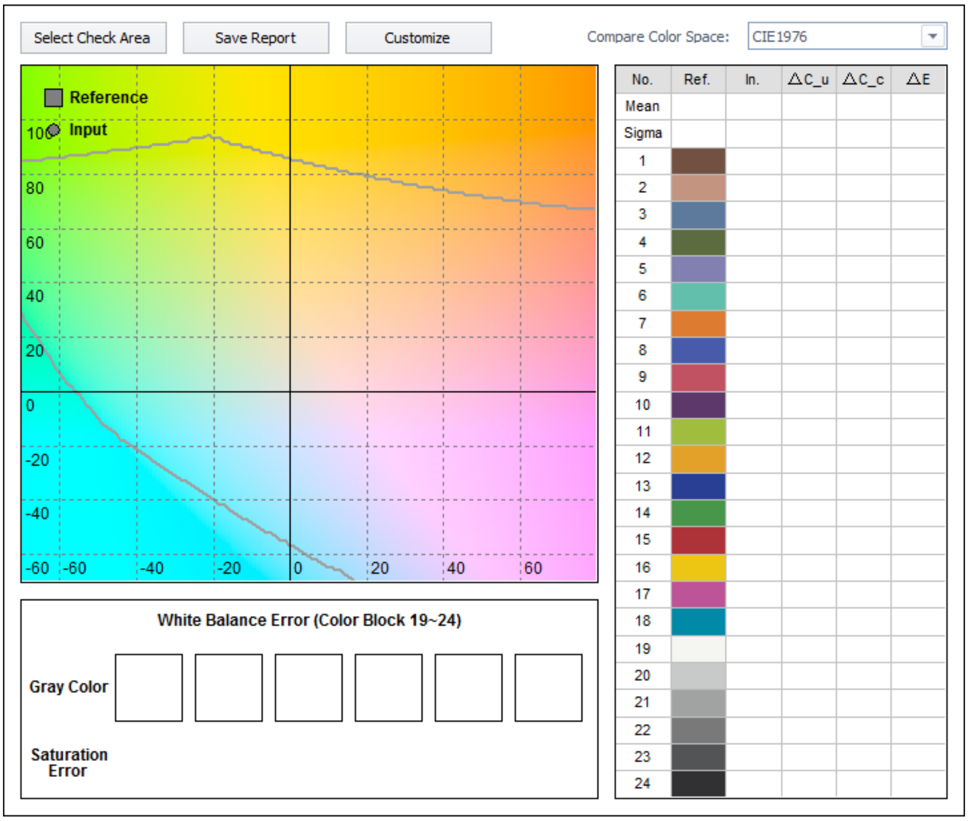
Step 2 Select the picture of a PANTONE 24-color chart.

----End

2.4.1.3 Using the 24-Color Comparison Function

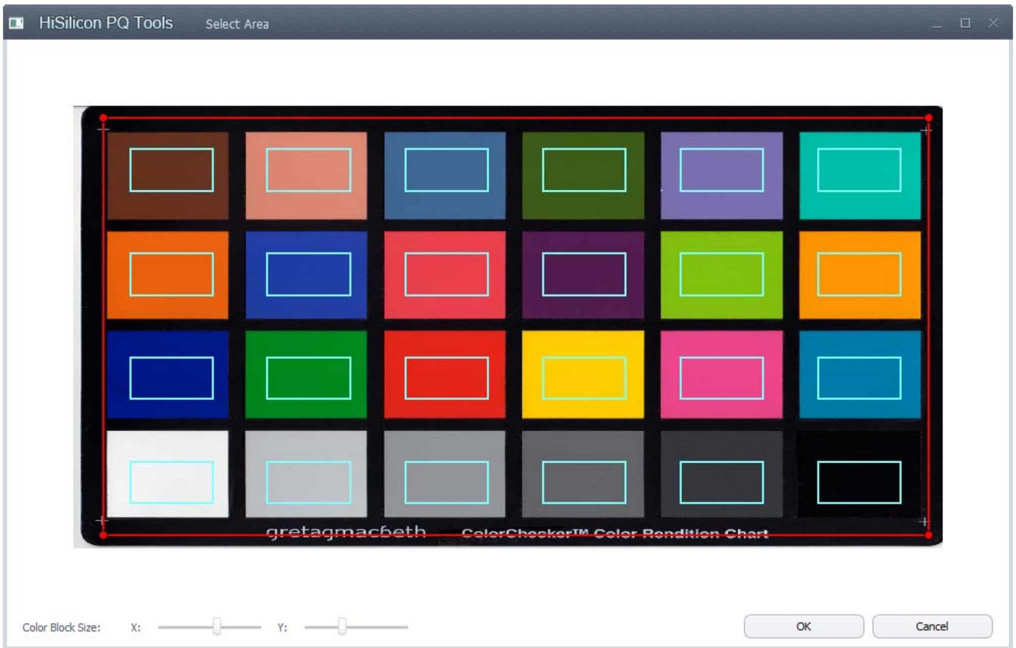
When a picture is obtained from the board or a local picture is opened, a message is displayed on the status bar, indicating that the image is ready. Now the 24-color comparison function can be used. See Figure 2-28.

Figure 2-28 Window for the 24-color comparison function



On the **Comparison** tab page, clicking **Select Check Area** opens the dialog box of 24-color sampling.

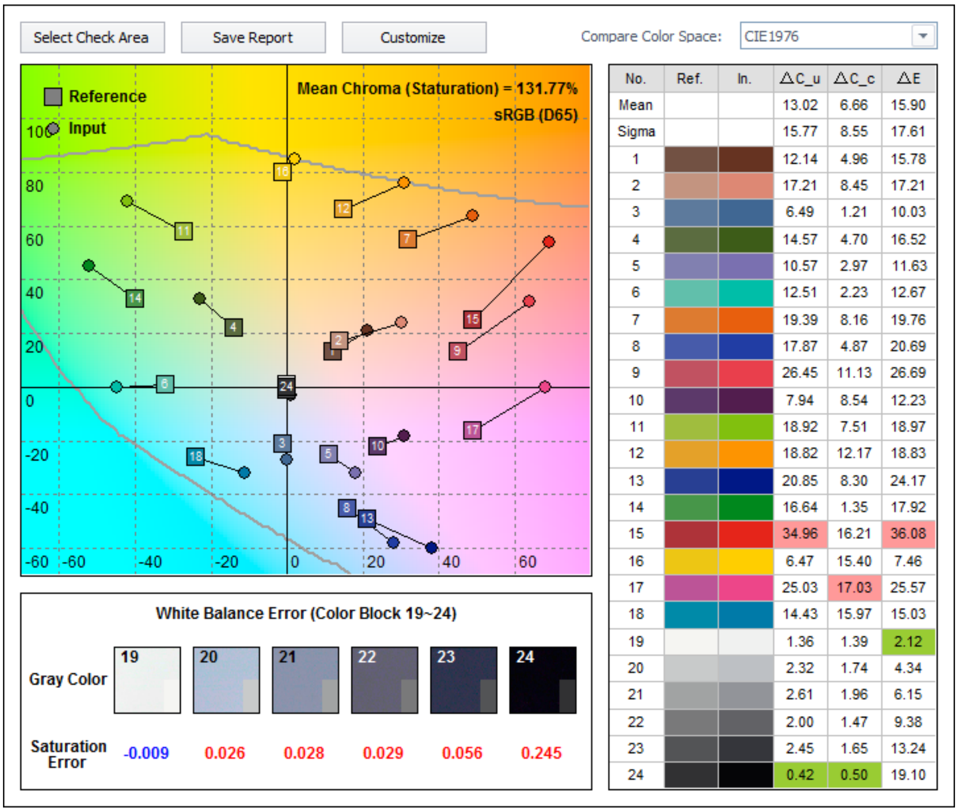
Figure 2-29 Window for selecting the sampling areas of 24 colors



Drag the red rectangle until the small blue rectangles fall within the 24 color blocks, which implements the optimal color sampling effect. In **Color Block Size** on the lower part, dragging the slide label in **X** adjusts the length of the blue rectangles, and dragging the slide label in **Y** adjusts the width of the blue rectangles.

After adjustment, click **OK**. Then the color analysis results shown in Figure 2-30 are displayed.

Figure 2-30 Color analysis results



Compare Color Space can be used to compare components listed in the table on the right of Figure 2-30. Currently the conversion between CIE1976, CIEDE2000 and RGB is supported.

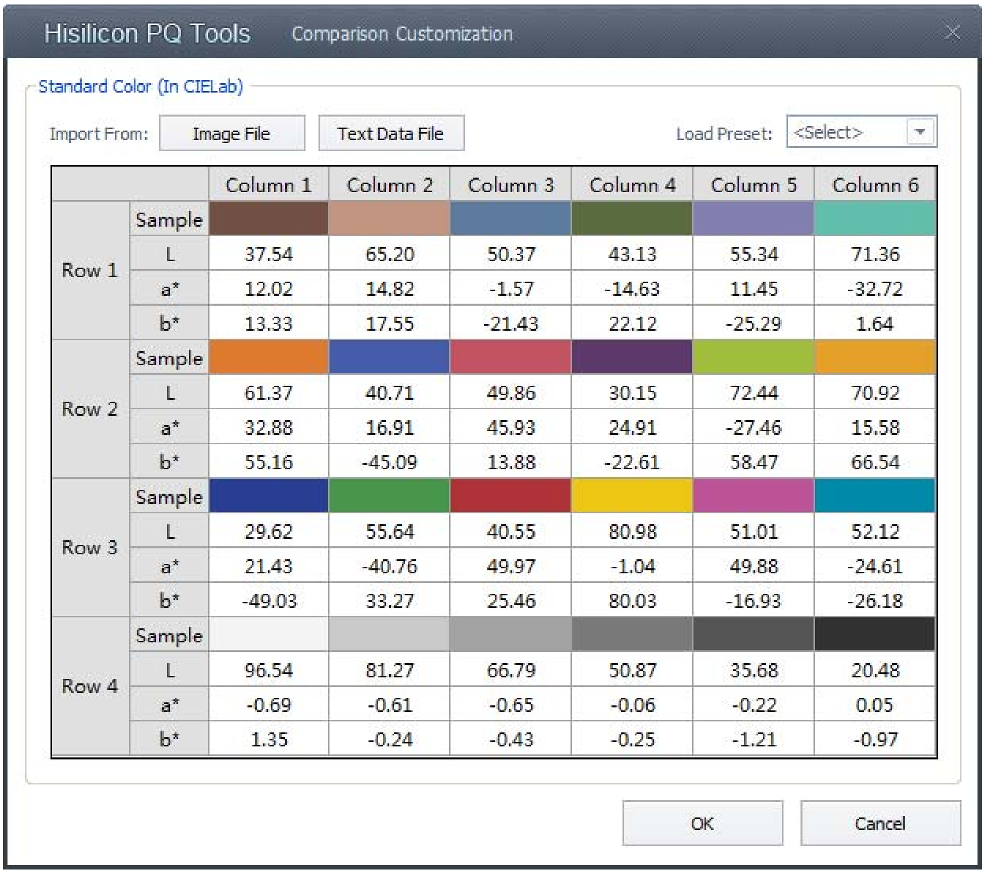
To select the sampling areas of 24 colors again, click **Select Check Area** in the **Checking Configuration** group box on the left of the color checker.

2.4.1.4 Customizing Standard Colors

The color checker provides a group of default standard color chart values expressed by the components in the L*a*b* color space. This group of color chart values is approximate to the X-rite standard values obtained under the D65 illuminant from a 2° view angle of the observer.

You can customize the standard color chart values. Click the **Customize** button on the **Comparison** tab page to open the standard color chart configuration dialog box, as shown in Figure 2-31.

Figure 2-31 Standard color chart configuration dialog box



		Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
Row 1	Sample						
	L	37.54	65.20	50.37	43.13	55.34	71.36
	a*	12.02	14.82	-1.57	-14.63	11.45	-32.72
	b*	13.33	17.55	-21.43	22.12	-25.29	1.64
Row 2	Sample						
	L	61.37	40.71	49.86	30.15	72.44	70.92
	a*	32.88	16.91	45.93	24.91	-27.46	15.58
	b*	55.16	-45.09	13.88	-22.61	58.47	66.54
Row 3	Sample						
	L	29.62	55.64	40.55	80.98	51.01	52.12
	a*	21.43	-40.76	49.97	-1.04	49.88	-24.61
	b*	-49.03	33.27	25.46	80.03	-16.93	-26.18
Row 4	Sample						
	L	96.54	81.27	66.79	50.87	35.68	20.48
	a*	-0.69	-0.61	-0.65	-0.06	-0.22	0.05
	b*	1.35	-0.24	-0.43	-0.25	-1.21	-0.97

To change the standard color chart values, enter the values in the corresponding cells in the table.

You can select a standard color chart image by clicking the **Image File** button, and use the image as the standard color chart after a 24-color region is specified. Then the color checker automatically calculates the L*a*b* value of the color chart based on the specified 24-color region.

You can select a data file containing the standard color chart values by clicking the **Text Data File** button, and import the values as the standard color chart values of the color checker. The CSV files generated by the Imatest 3.4 can be imported. You can import the ideal values (standard value) or the meas values (input values).

You can select a group of preconfigured values from the **Load Preset** drop-down box in the upper right part of the dialog box as the standard color chart values. Only the approximate X-rite D65 values can be selected as the preconfigured values currently.

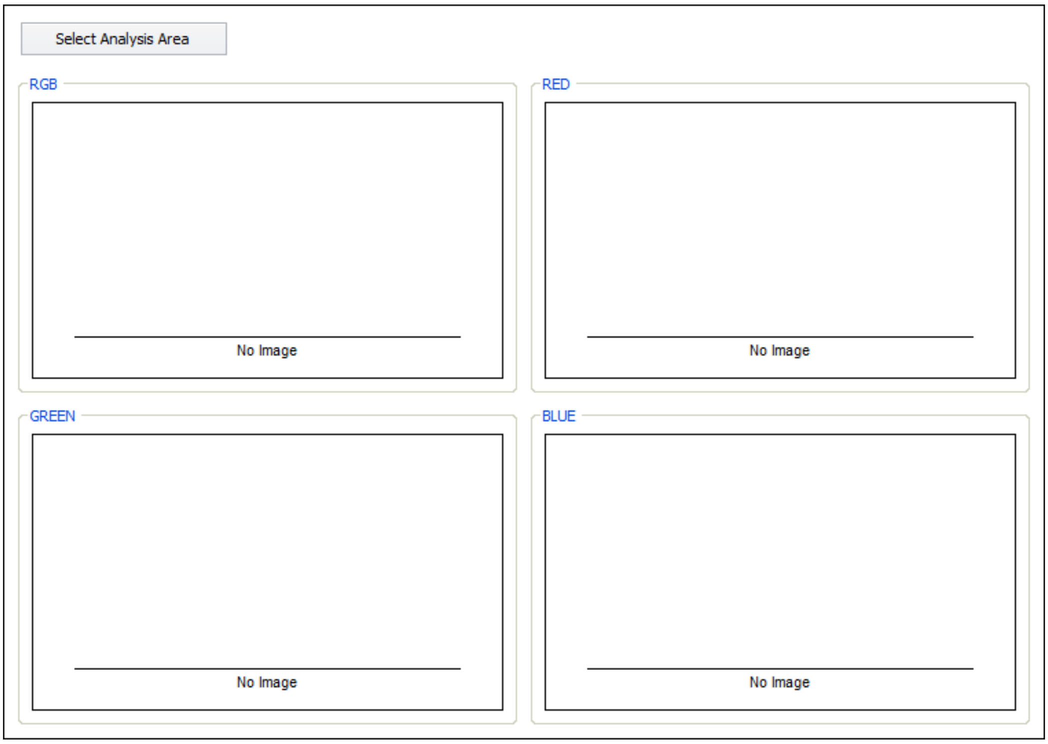
CAUTION

If you need to import the data files generated by the Imatest into the standard color chart configuration dialog box, do not modify the CSV files exported from the Imatest. Otherwise, the **Color Checker** may fail to identify the files.

2.4.1.5 Viewing the RGB Statistics Histogram

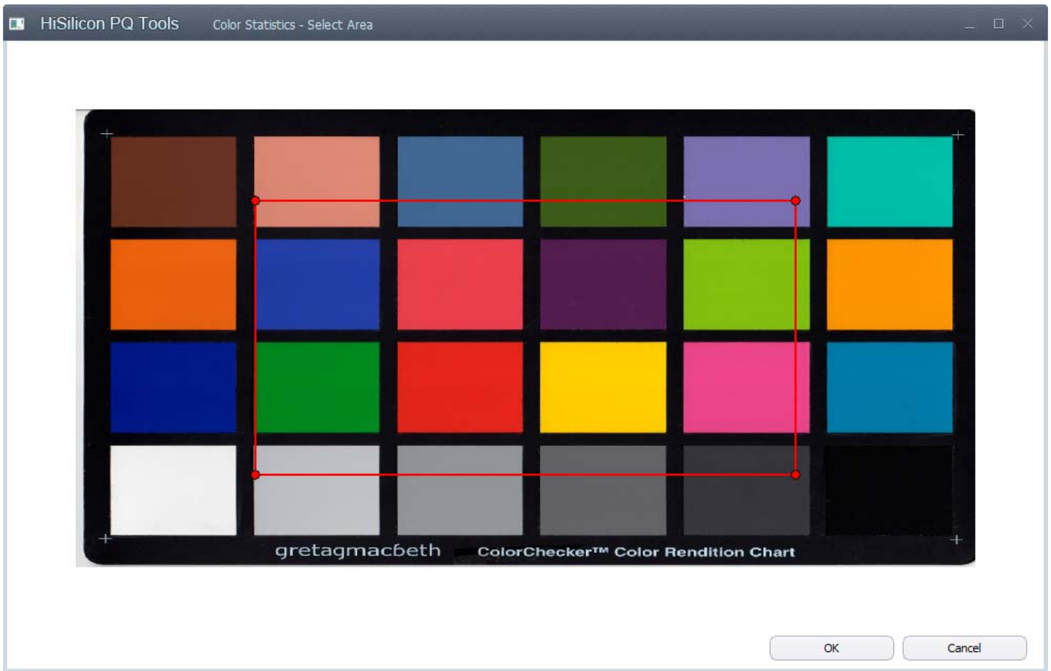
When a picture is obtained from the board or a local picture is opened, the RGB statistics histogram function can be used.

Figure 2-32 GUI of statistics histogram function



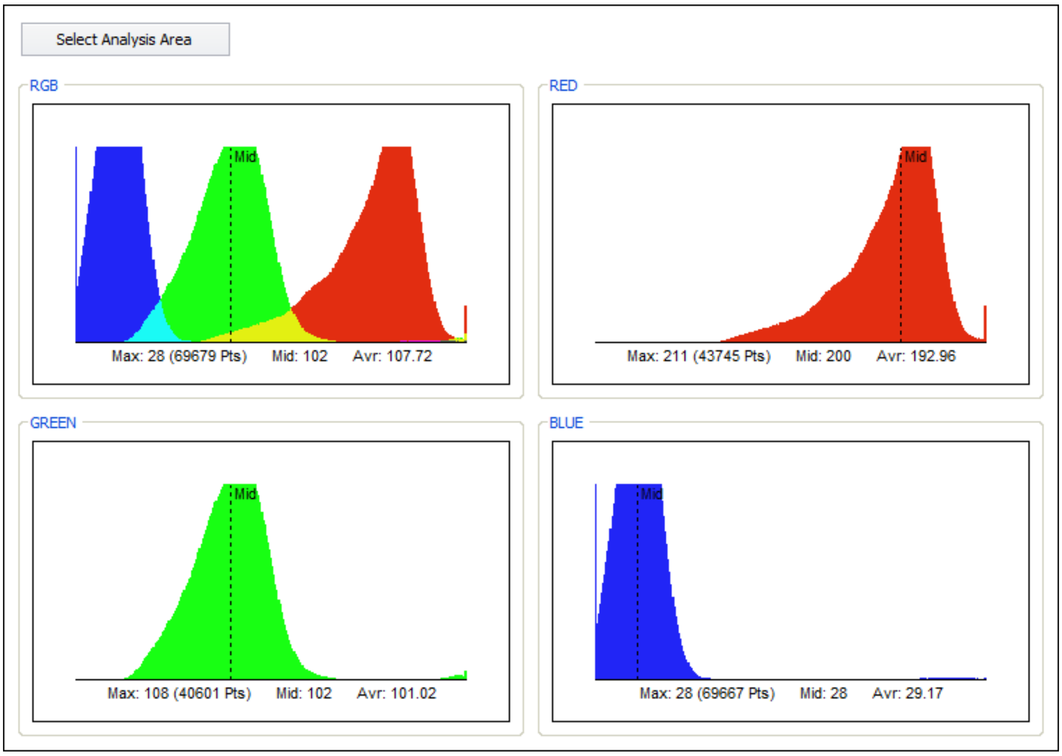
You must select a statistics area before viewing the statistics histogram. Activate the **YRGB Statistics** tab page. Click **Select Analysis Area** on the tab page. The **Select Area** dialog box shown in Figure 2-33 is displayed.

Figure 2-33 Select Area dialog box



Drag any corner of the red rectangle, and click **OK**. Then the component statistics histograms of the selected area are displayed, as shown in Figure 2-34. When a local picture is used as the data source, the Y component histogram is not displayed.

Figure 2-34 Displaying statistics histograms



In each statistics histogram, the x-axis indicates the component value (0 to 255 from left to right), and the y-axis indicates the number of points corresponding to a specific component value (if too many points correspond to a maximum component value, the picture is cropped). The colorful texts under histograms are special values.

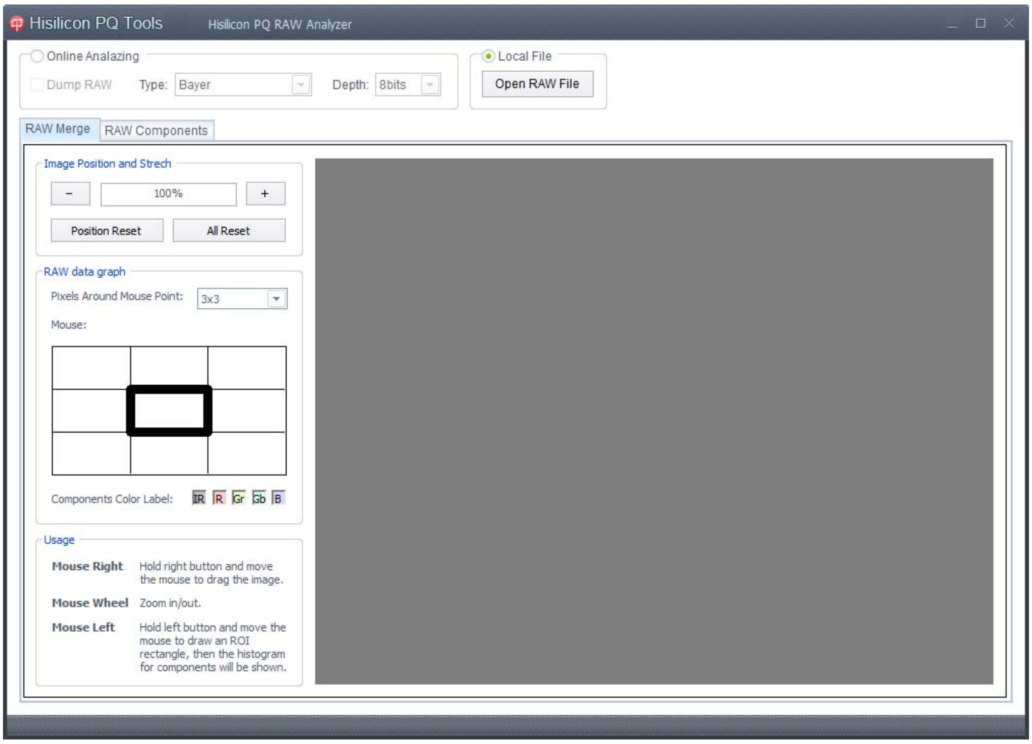
- Maximum value (**Max**): component value corresponding to the maximum number of points
- Median value (**Mid**): The number of points corresponding to the component values that are greater than or equal to (or less than or equal to) the median value are about 50% of the total points.
- Average value (**Avr**): average value of all points of a component

2.4.2 RAW Analyzer

2.4.2.1 GUI

Choose **HiPQ RAW Analyzer** from the **Select an Extra Component** drop-down box in the main GUI of the HiSilicon PQ Tools. The **HiSilicon PQ RAW Analyzer** window shown in Figure 2-35 is displayed.

Figure 2-35 HiSilicon PQ RAW Analyzer



The RAW analyzer allows you to view the component and luminance of each point on the opened picture as well as the luminance distribution histogram. To view an entire picture, click **RAW Merge**; to view a component, click **RAW Components**.

2.4.2.2 Selecting a Data Source

The RAW analyzer supports online data analysis when the board is connected and can read a local .raw file for data analysis. The HiSilicon .raw files and common .raw files are supported.

To implement online data analysis, perform the following steps:

Step 1 On the main GUI of the HiSilicon PQ Tools, open a PQ table (ensure that the chip model and version match the board).

Step 2 Run the ISP/VI/VPSS service on the board.

Step 3 Connect the HiSilicon PQ Tools to the board.

Step 4 On the main GUI of the RAW analyzer, select **Online Analyzing**. Set the depth and type (bayer, RGBIR, or IR only) of the captured picture (which can also be modified during online analysis).

Step 5 Select the **Dump RAW** check box.

----End

CAUTION

The online capture function of the RAW analyzer captures the raw picture data in linear mode.

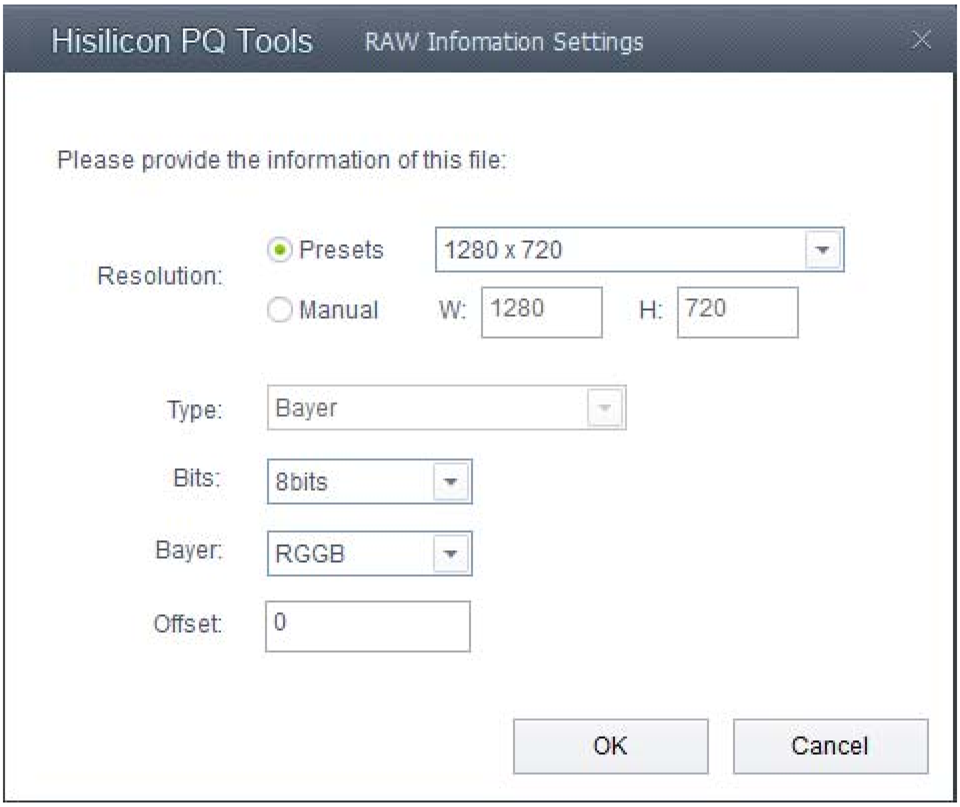
To open a local .raw file, perform the following steps:

Step 1 On the main GUI of the HiSilicon PQ Tools, select **Local File Analyzing**.

Step 2 Click **Open RAW File** to select a .raw file.

Step 3 If the RAW analyzer detects the picture information contained in the file name, it automatically parses the file. Otherwise, a dialog box is displayed, asking you to set information about a common .raw file. In the displayed **RAW Information Settings** dialog box, specify the resolution, depth, type, component pattern, IR component replacement position, and header offset of the file.

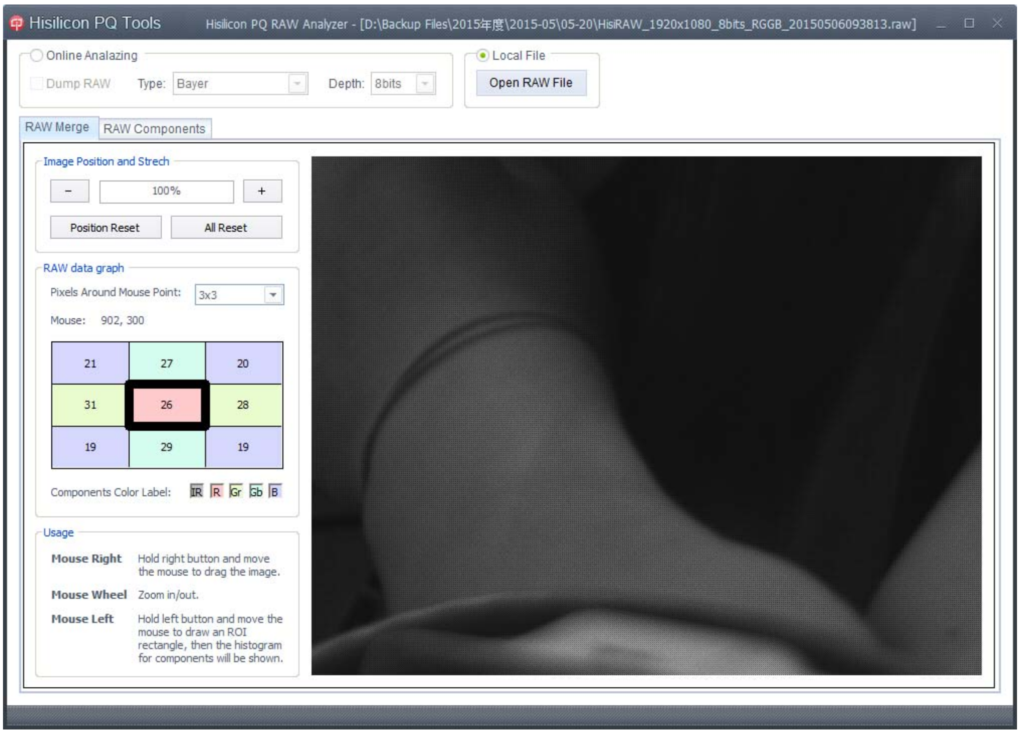
Figure 2-36 Setting a common .raw file



Step 4 Click **OK**.

The RAW analyzer displays the .raw picture, as shown in Figure 2-37.

Figure 2-37 Displaying the .raw picture



----End

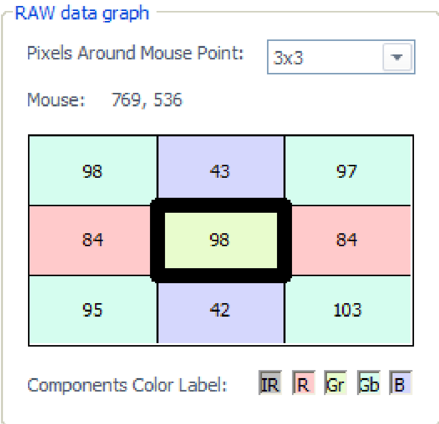
2.4.2.3 Obtaining Analysis Data

After a .raw picture is displayed, if you move the pointer onto the picture, you can view the component and luminance values of the pointer position (called target point) and surrounding points. If points are difficult to select due to the default picture size, zoom in or zoom out on the picture in any of the following ways:

1. Hold down the right button and move the mouse to drag the picture.
2. Click the picture to zoom in on the picture (the point clicked is centered in the enlarged picture) or right-click the picture to zoom out on the picture.
3. In the **Zoom** group box, click + or – to zoom in or zoom out on the picture based on the configured scaling ratio.
4. In the text box of the **Zoom** group box, enter a scaling ratio.

The analysis data is displayed in the left **Raw Data Graph** group box shown in Figure 2-38.

Figure 2-38 Analysis data graphic



The data in the black square indicates the luminance of the target point, and the data in other squares indicates the luminance of the points around the target point.

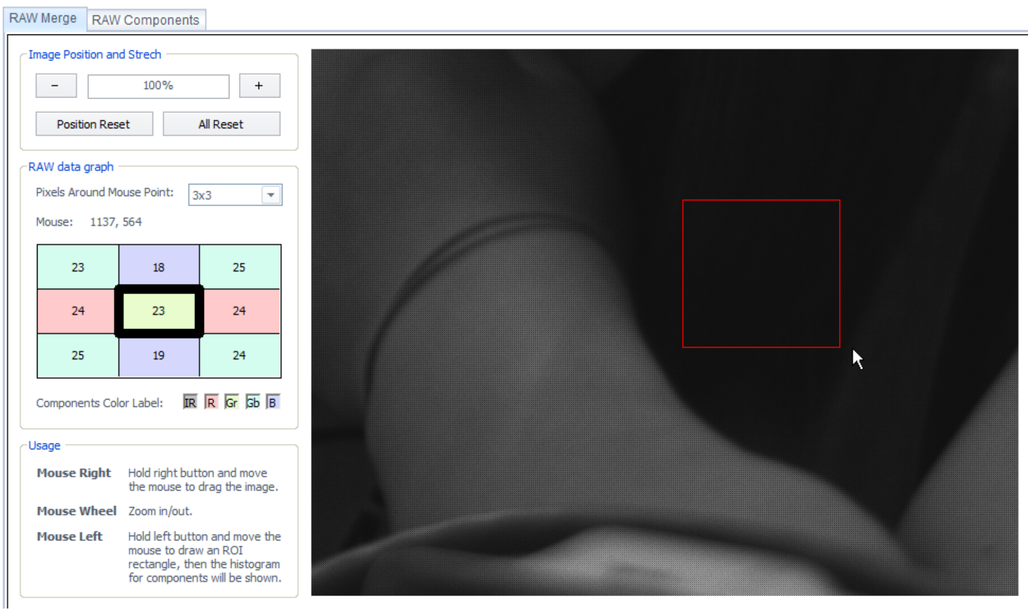
The background color of each square indicates the component to which a point belongs, which is illustrated above the analysis data graphic.

To set the number of points around the target point in the graphic, choose a value from the **Analyze Size** drop-down box (The numbers can be 3 x 3, 5 x 5, or 7 x 7.).

2.4.2.4 Viewing the Luminance Statistics Histogram

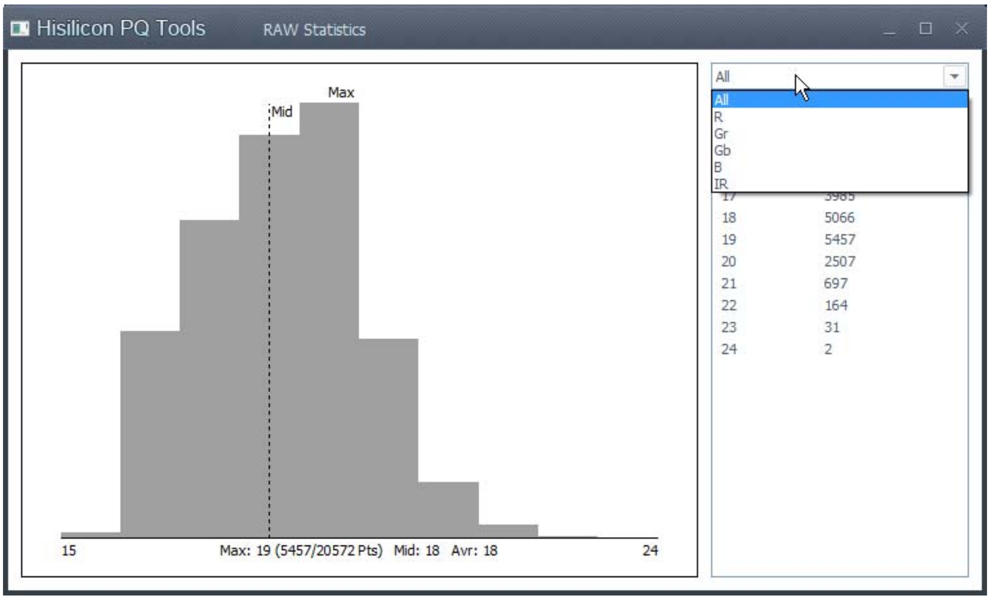
To view the luminance statistics histogram of a specific region, draw a rectangular region in the displayed picture as the region to be analyzed using the mouse, as shown in Figure 2-39.

Figure 2-39 Selecting a luminance analysis area



After a statistics area is selected, click **Show Statistics**. Then the luminance statistics histogram shown in Figure 2-40 is displayed.

Figure 2-40 Viewing the luminance statistics histogram and data



In the statistics histogram, the x-axis indicates the component value, and the y-axis indicates the number of points corresponding to a specific luminance. The colorful texts under histograms are special values.

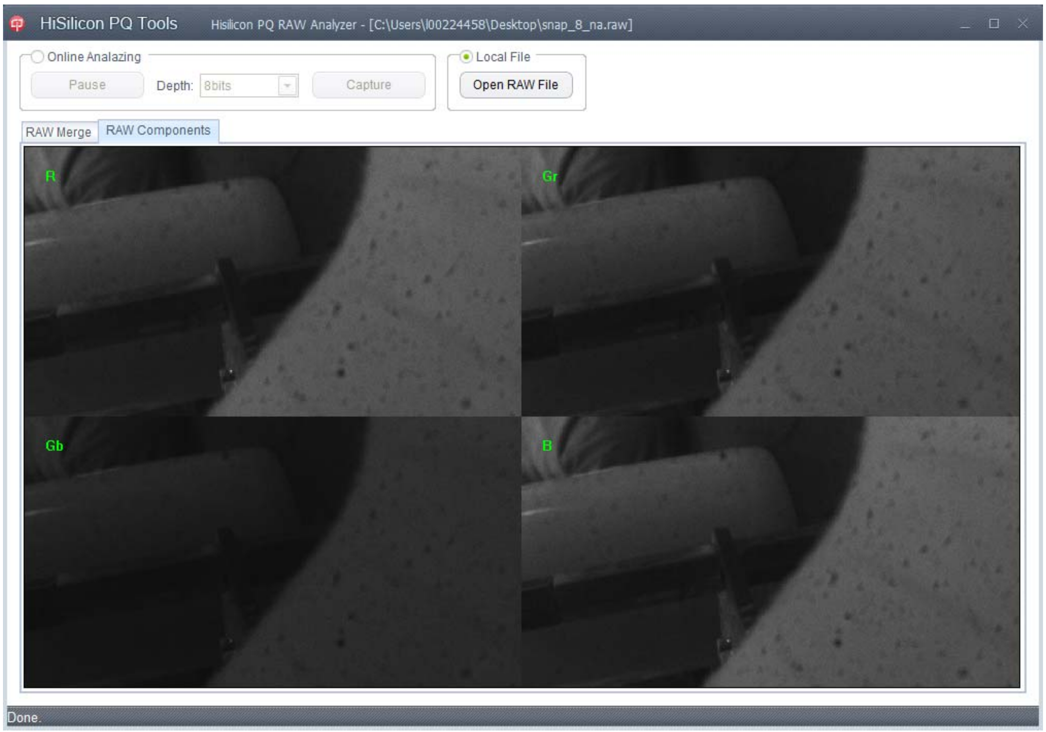
- Maximum value (**Max**): component value corresponding to the maximum number of points
- Median value (**Mid**): The number of points corresponding to the luminance values that are greater than or equal to (or less than or equal to) the median value are about 50% of the total points.
- Average value (**Avr**): average luminance value of all points
- The list on the right provides the number of points corresponding to each luminance value.

The drop-down box in the upper right corner of the window shown in Figure 2-40 allows you to view the overall histogram and histogram of each component.

2.4.2.5 Viewing Picture Components

On the main GUI of the RAW analyzer, click **RAW Components** to view picture components. See Figure 2-41.

Figure 2-41 Viewing picture components



2.4.3 White Balance Analyzer

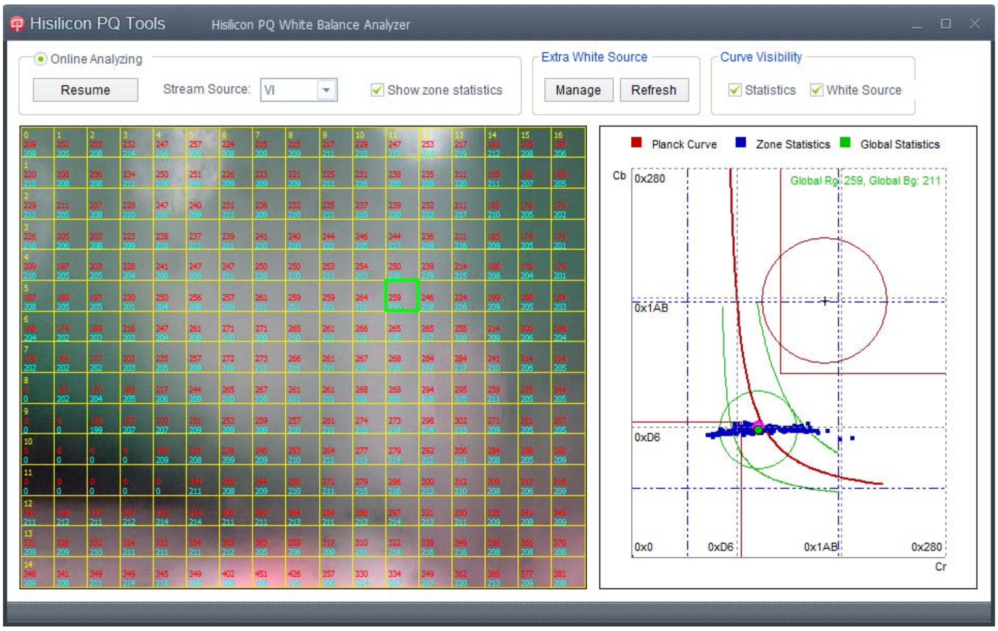
The HiSilicon PQ Tools provides the white balance analyzer for the picture quality debugging personnel to view the color temperature curves and white balance statistics.

The white balance analyzer applies only to the picture quality adjustment of the Hi3518 series chips.

2.4.3.1 GUI

Choose **HiPQ White Balance Analyzer** from the **Select an Extra Component** drop-down box in the main GUI of the HiSilicon PQ Tools. The **Hisilicon PQ White Balance Analyzer** window shown in Figure 2-42 is displayed.

Figure 2-42 Hisilicon PQ White Balance Analyzer



The white balance analyzer provides white balance statistics for the picture quality personnel.

2.4.3.2 Online Analysis by Connecting to the Board

The white balance analyzer supports only online analysis by connecting the computer to the board. Perform the following steps:

Step 1 On the main GUI of the HiSilicon PQ Tools, open a PQ table (ensure that the chip model and version match the board).

Step 2 Connect the HiSilicon PQ Tools to the board.

Step 3 Focus the camera on the target analysis area.

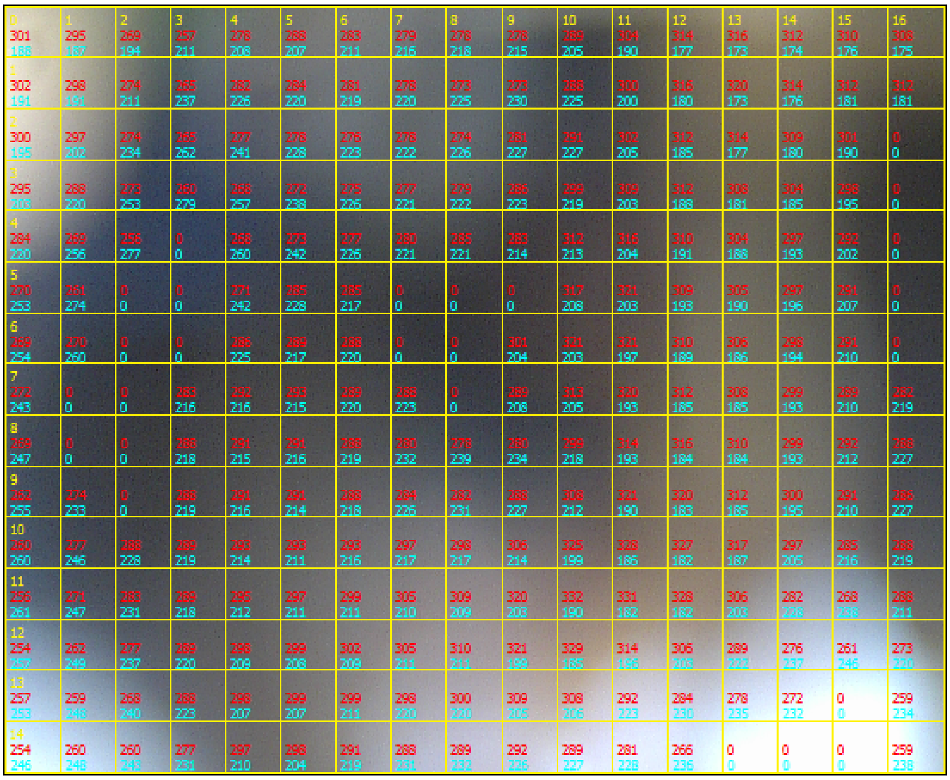
----End

After the board is connected, the white balance analyzer requests the picture and statistics from the board.

2.4.3.3 Viewing the Statistics of the Real-Time Picture and Zones

After being connected to the board, the HiSilicon PQ Tools obtains white balance data from the board and displays statistics of the current picture and zones in a picture, as shown in Figure 2-43.

Figure 2-43 Statistics of the picture and zones



A picture is divided into 17 x 15 zones. Each zone displays the statistics of the current picture. The data in red indicates **ZoneRg**, and the data in blue indicates **ZongRb**.

Selecting or deselecting **Show zone statistics on video view** shows or hides zoned statistics.

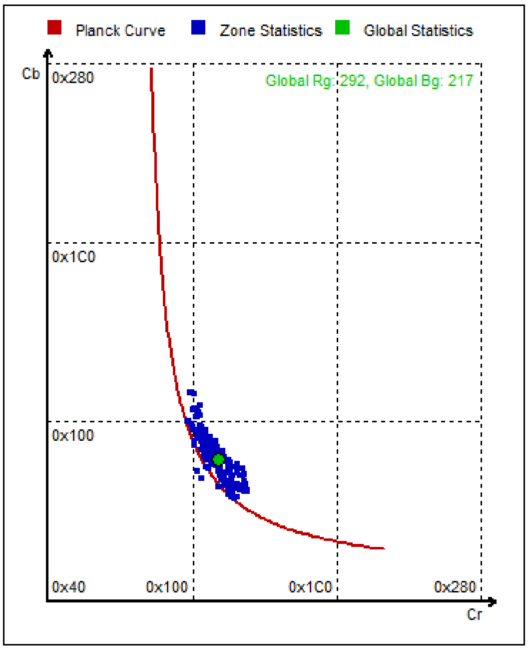
If a zone is selected in the picture on the left of Figure 2-42, the points corresponding to the zone in the coordinate graph on the right of Figure 2-42 are highlighted.

You can double-click a zone and configure the mapping point corresponding to this zone in the coordinate graph as an independent illuminant. To be specific, choose **Add light source** or **Delete sensitive color** from the shortcut menu, and enter the region radius in the **Radius** dialog box.

2.4.3.4 Viewing the Color Temperature Curve and Statistics Chart

When the HiSilicon PQ Tools displays statistics of the current picture and zones, it also displays a coordinate graph that shows the mapping between the color temperature curve and statistics, as shown in Figure 2-44.

Figure 2-44 Coordinate graph (color temperature curve and statistics)



The red curve indicates the color temperature curve, the blue points indicate the statistics of zones, and the green point indicates the global statistics.

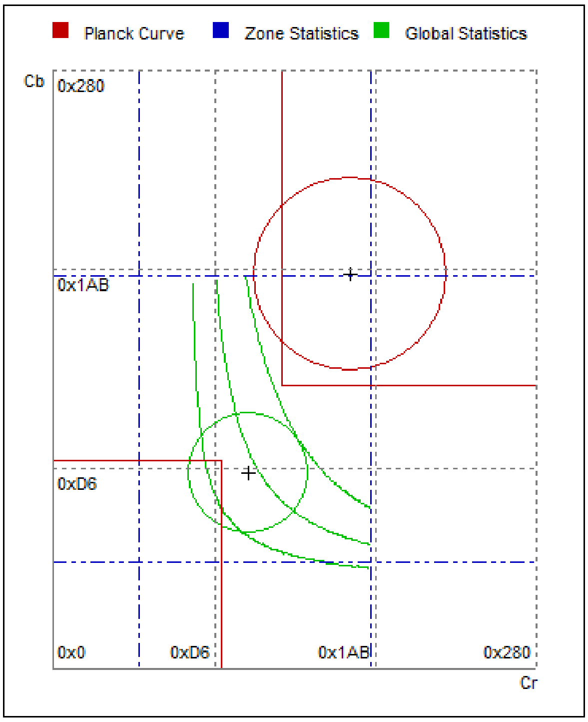
If you click to select the blue points indicating zoned statistics in Figure 2-44, the cells indicating the corresponding zones are automatically selected on the left of Figure 2-42.

You can choose to display the mapping between the color temperature curve and statistics by selecting the **Statistics** check box in the **Curve Visibility** group box.

2.4.3.5 Viewing the White Zone Chart in the Coordinate Graph

The white zone chart can also be displayed in the coordinate graph.

Figure 2-45 Coordinate graph (including the white zone chart)



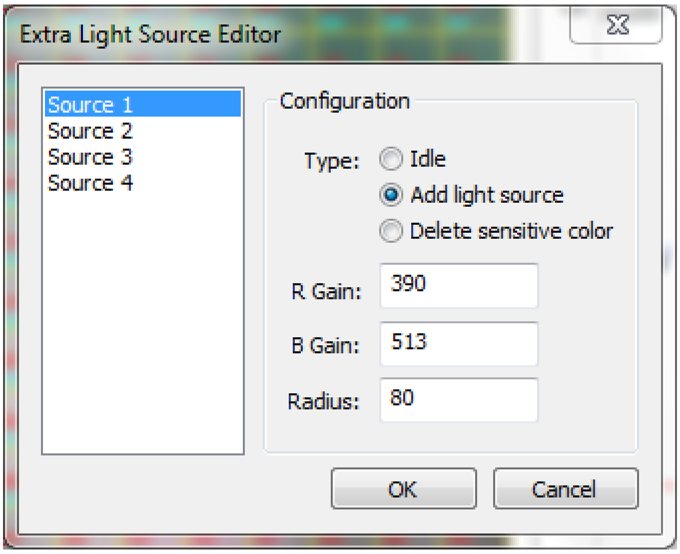
- The three green curves indicate the range of the basic white zone, and they are relevant to the color temperature curve.
- The two rectangular regions in the upper right corner and lower left corner indicate the regions to be deleted from the white zone to prevent the interference from the purple color and green color respectively.
- The four crossed blue lines indicate the value ranges of the Cr and Cb components.
- The two cross marks indicate the independent illuminants. The circles indicate the influence ranges of the independent illuminants. The green circle indicates the range of the region that is added to the white zone. The red circle indicates the range of the region corresponding to an interference color that is deleted from the white zone.

You can update the white zone chart by clicking the **Refresh** button in the **Extra White Source** group box.

2.4.3.6 Configuring Independent Illuminants

You can directly configure the information about an independent illuminant in the white balance analyzer. Click the **Manage** button in the **Extra Light Source** group box. The configuration dialog box of the independent illuminant shown in Figure 2-46 is displayed.

Figure 2-46 Configuration dialog box of the independent illuminant



At most four regions corresponding to four independent illuminants can be configured. To configure the region corresponding to an independent illuminant, perform the following steps:

Step 1 Select the ID of the illuminant to be edited from the list on the left.

Step 2 Configure the type of the independent illuminant. **Idle** indicates that a region is not configured as the region corresponding to the independent illuminant. **Add light source** indicates that a region is added as the region corresponding to an independent illuminant. **Delete sensitive color** indicates that the region corresponding to an interference color is deleted.

Step 3 Configure the gain values of the R and B components. Note that the gains values of the R and B components are the values before white balance processing.

Step 4 Set the region radius.

Step 5 Repeat step 1 to step 4 if multiple illuminants need to be configured.

Step 6 Click **OK**.

----End

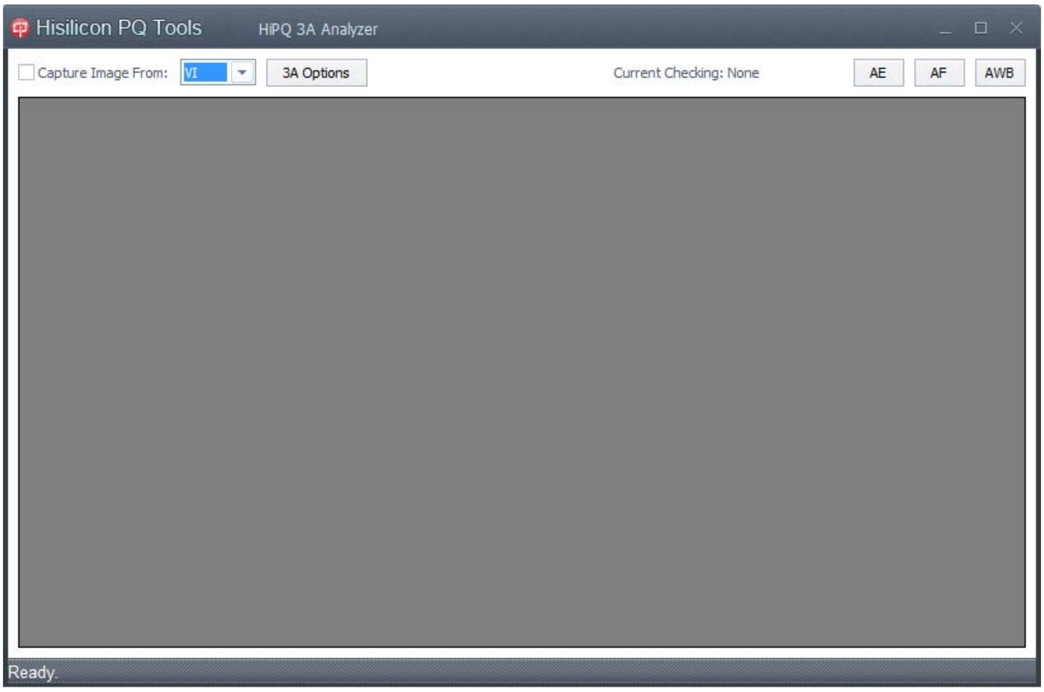
2.4.4 3A Analyser

The HiPQ 3A Analyser allows you to view 3A statistics.

2.4.4.1 GUI

Choose **HiPQ 3A Analyser** from the **Select an Extra Component** drop-down box in the main GUI of the HiSilicon PQ Tools. The **HiPQ 3A Analyser** window shown in Figure 2-47 is displayed.

Figure 2-47 HiPQ 3A Analyser



2.4.4.2 Obtaining Images and Statistics

Currently the 3A Analyser supports analysis only when the HiSilicon PQ Tools is connected to boards. After opening the GUI of the 3A Analyser, obtain images and corresponding statistical data from the board as follows:

Step 1 Connect the main program of the HiSilicon PQ Tools with the board.

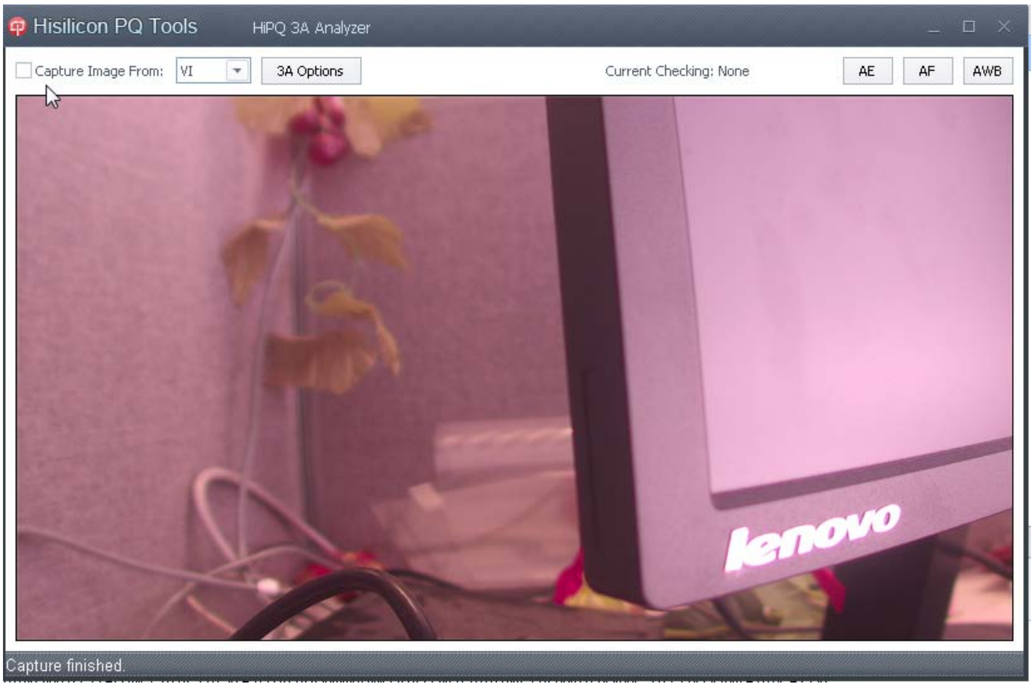
Step 2 In the main GUI of the 3A Analyser, choose the image data source from the **Capture Image from** drop-down box. Currently the 3A Analyser can obtain image data from VI, VPSS or VO.

Step 3 Select **Capture Image from**.

----End

After that, the 3A Analyzer automatically obtains data from the board until you deselect **Capture Image**. After successfully obtaining data from the board, the picture is displayed in the gray area in the lower part of the tool GUI, as shown in Figure 2-48.

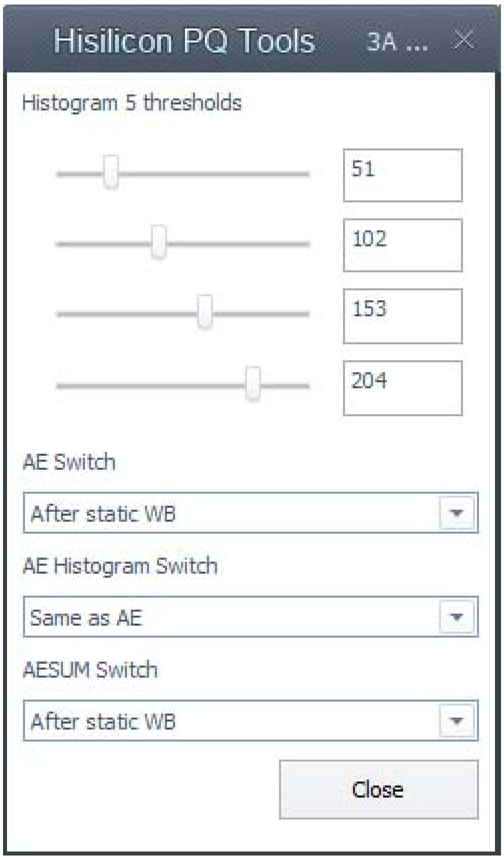
Figure 2-48 Displaying picture data in the 3A Analyzer



2.4.4.3 Configuring AE Options

Option configuration is required for the AE module before data is obtained. Click **3A Options** on the main GUI of the 3A Analyzer. Then the 3A configuration option dialog box is displayed, as shown in Figure 2-49.

Figure 2-49 3A configuration option dialog box (for Hi3516A)



For Hi3516A chip, the configuration of the segment for the 5-segment histogram is supported. Under **Histogram 5 thresholds**, drag the slide label for setting the threshold of each segment or directly input a value (0–255) in the text box on the right to configure the segments for the histogram. After the picture data is refreshed by the 3A Analyzer, the effect can be viewed in the AE statistics dialog box.

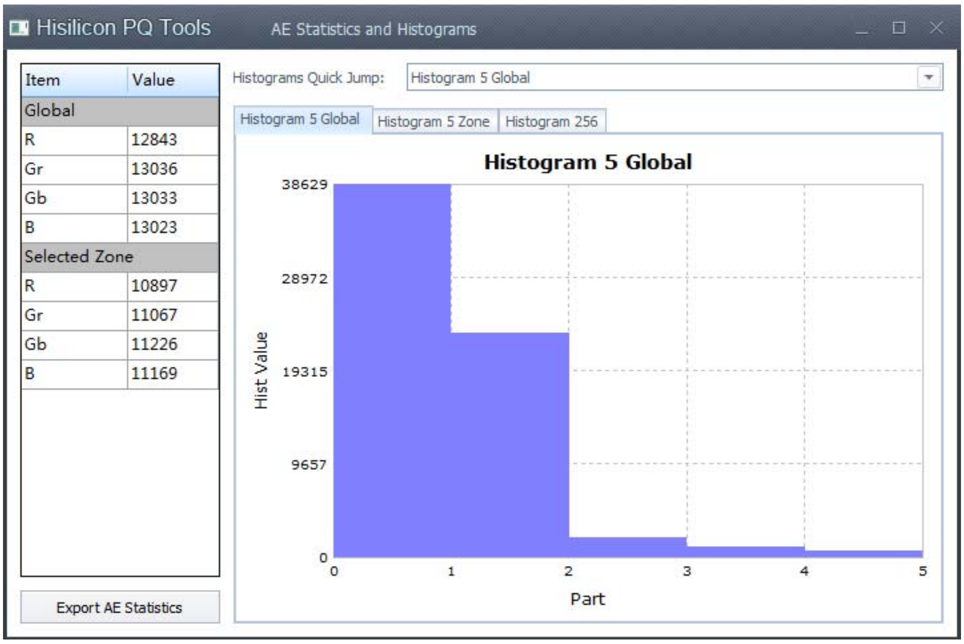
CAUTION

After all parameters are configured, a frame of data needs to be captured because the previously captured data does not apply to the parameter configuration.

2.4.4.4 Viewing AE Statistics

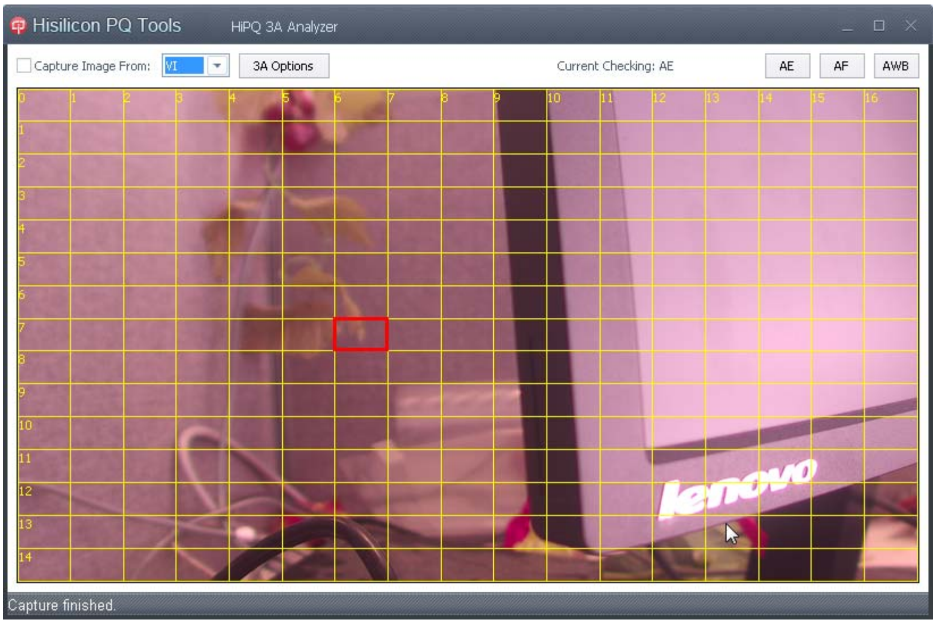
Click **AE** in the upper right corner of the 3A Analyzer. Then the AE statistics dialog box is displayed, as shown in Figure 2-50.

Figure 2-50 AE statistics dialog box



The picture area on the main GUI of the 3A Analyzer is divided into multiple zones by yellow lines. You can click to select a zone. See Figure 2-51.

Figure 2-51 Selecting an AE zone



The functions of the areas in the **AE Statistics and Histograms** dialog box (see Figure 2-50) are as follows:

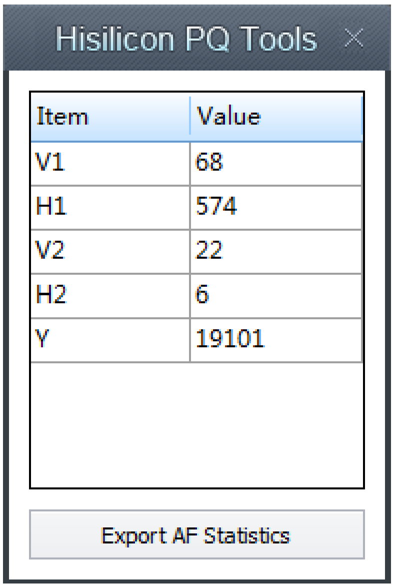
- View the average values of global and zoned components. The statistics change when the selected zone is changed on the main GUI of the 3A Analyzer.

- Save the AE statistics obtained from the board into a plaintext .txt file.
- View the AE statistics histogram. The type of the histogram (5-/256-/1024-segment histogram) that can be viewed varies according to the chip model. You can switch the histogram by using the **Histograms Quick Jump** drop-down box or clicking the corresponding tab.

2.4.4.5 Viewing AF Statistics

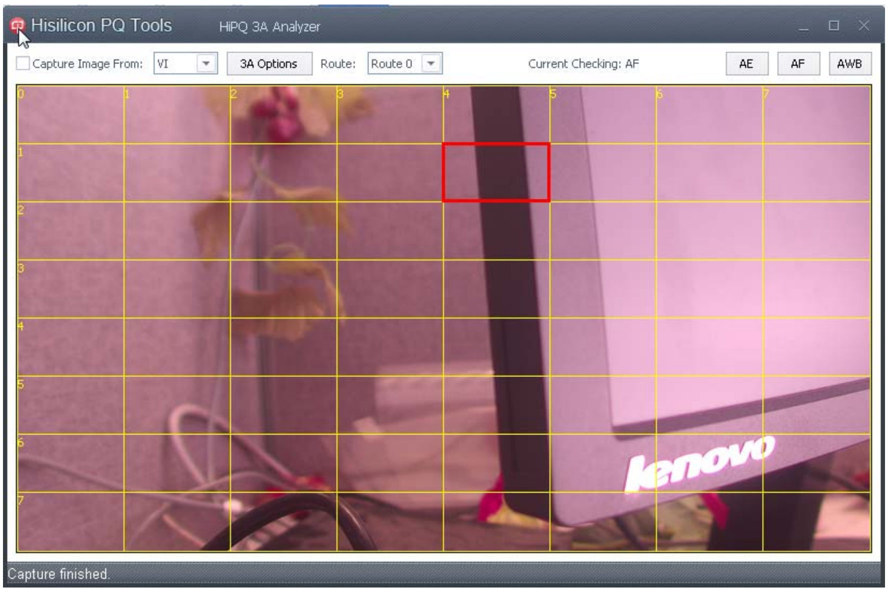
Click **AF** in the upper right corner of the 3A Analyzer. Then the AF statistics dialog box is displayed, as shown in Figure 2-52.

Figure 2-52 AF statistics dialog box



The picture area on the main GUI of the 3A Analyzer is divided into multiple zones by yellow lines. You can click to select a zone. See Figure 2-53.

Figure 2-53 Selecting an AF zone



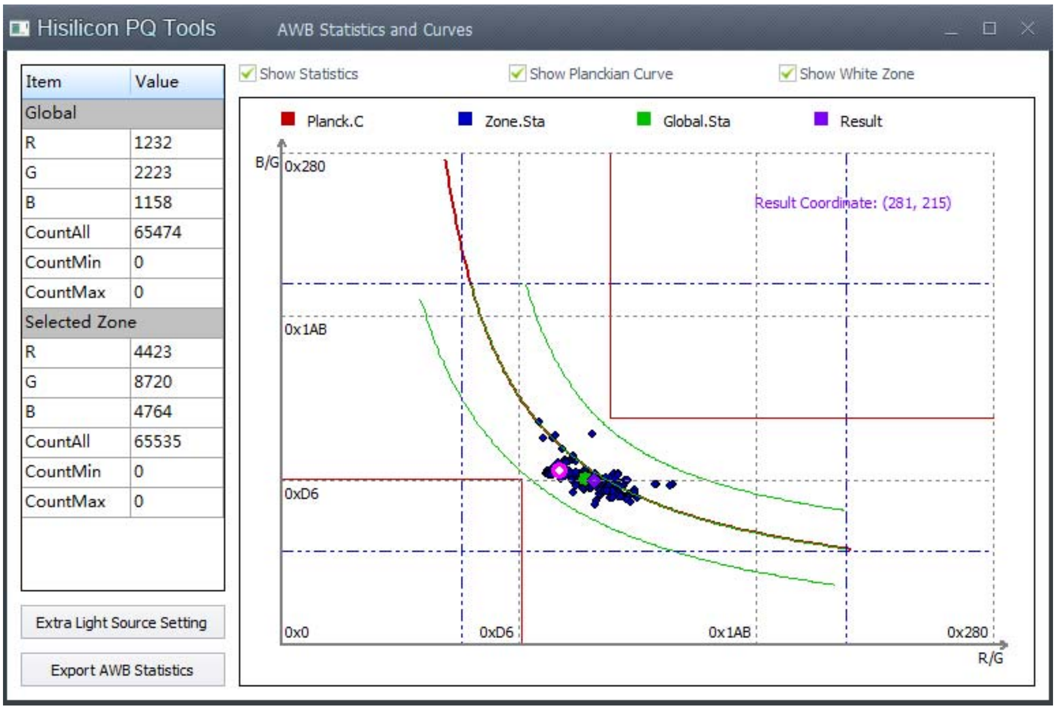
You can view the AF statistics of the selected zone in the statistics AF statistics dialog box. The statistics change when the selected zone is changed on the main GUI of the 3A Analyzer.

Click **Export AF Statistics** on the lower part of the AF statistics dialog box. Then the AF statistics obtained from the board are saved into a plaintext .txt file.

2.4.4.6 Viewing AWB Statistics

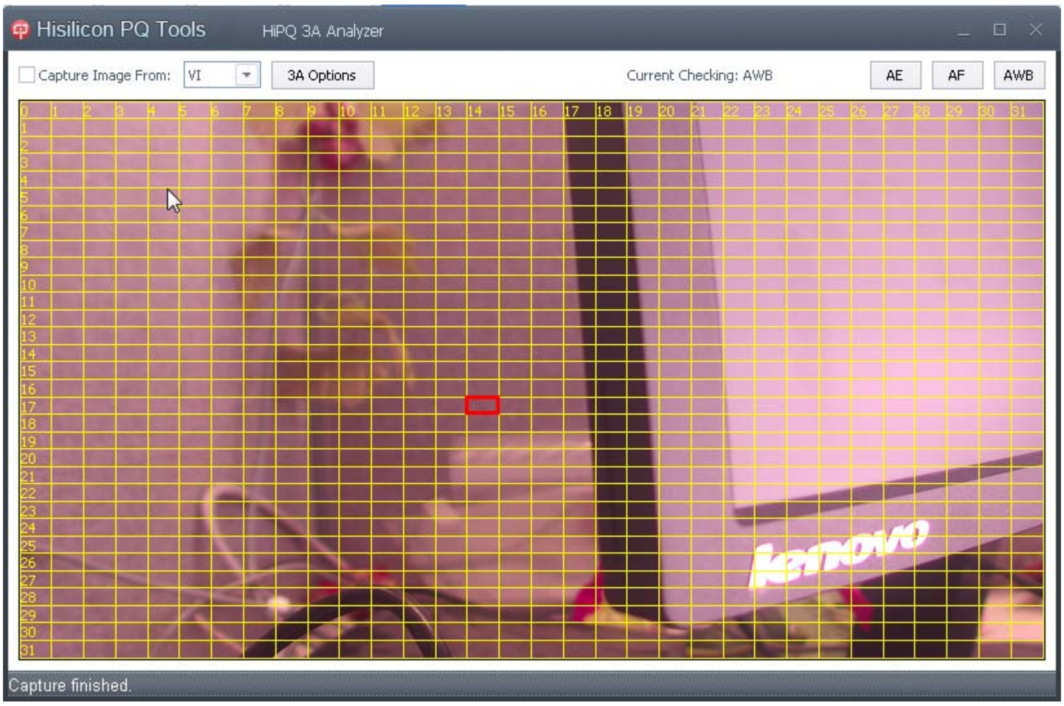
Click **AWB** in the upper right corner of the 3A Analyzer. Then the AWB statistics dialog box is displayed, as shown in Figure 2-54.

Figure 2-54 AWB statistics dialog box (taking Hi3516C V300 as an example)



The picture area on the main GUI of the 3A Analyzer is divided into multiple zones by yellow lines. You can click to select a zone, as shown in Figure 2-55.

Figure 2-55 Selecting an AWB zone



The contents displayed in the **AWB Statistics and Curves** dialog box are described as follows:

- Red curve: Planckian curve
- Green points: corresponding points of the global statistics in the Planckian coordinate system
- Purple points: corresponding points of the AWB calculation results in the Planckian coordinate system
- Blue points: corresponding points of the zoned statistics in the Planckian coordinate system
- Green curves: basic white zones. The green curve in the middle is basically equivalent to the Planckian curve, and is related to the shift.
- Blue dotted lines: value ranges of **Rgain** and **Bgain**. The two values are fixed in manual mode, and are related to the configured high and low color temperatures on the board in automatic mode.
- Two regions formed by red lines in the upper right corner and lower left corner: zones deleted from the white zones to eliminate interference from the purple and green colors. The two regions are related to the AWB parameters **CurveLLimit** and **CurveRLimit**.
- Crosses: independent illuminants. The surrounding red or green circle indicates the influence range of the independent illuminant. The green circle indicates the range of the region that is added to the white zone, and the red circle indicates the range of the region corresponding to an interference color that is deleted from the white zone.
- **Result Coordinate** in purple: coordinate value of the AWB calculation result on the Planckian curve
- Select or deselect **Show Statistics**, **Show CT Curves**, and **Show Light Sources** to display or hide the statistics on the coordinate graph, color temperature curve, and white zone (including the independent illuminant) respectively.

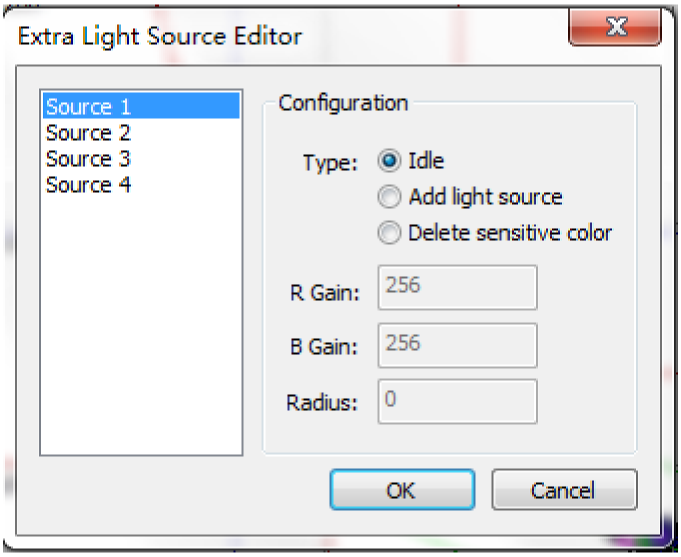
In Figure 2-55, if a zone is selected, the points corresponding to the zoned statistics on the right coordinate graph in Figure 2-54 are highlighted. If you click to select the points indicating zoned statistics in Figure 2-54, the corresponding zones are automatically selected in Figure 2-55.

Click **Export AWB Statistics**. Then the AWB statistics obtained from the board are saved into a plaintext .txt file.

2.4.4.7 Editing AWB Independent Illuminants

You can edit the independent illuminants on the AWB page of the 3A analyzer. Click the **Extra Light Source Setting** button. Then the the editing dialog box of the independent illuminant is opened, as shown in Figure 2-56.

Figure 2-56 Editing diagram of the independent illuminant



At most four regions corresponding to four independent illuminants can be configured. Perform the following steps:

Step 1 Select the independent illuminant region (from source 1 to source 4) that is to be edited.

Step 2 Select the type of the independent illuminant: **Idle** indicates that a region is not configured as the region corresponding to the independent illuminant. **Add light source** indicates that a region is added as the region corresponding to an independent illuminant. **Delete sensitive color** indicates that the region corresponding to an interference color is deleted.

Step 3 Specify the red and blue gains of the illuminant and also the affected radius.

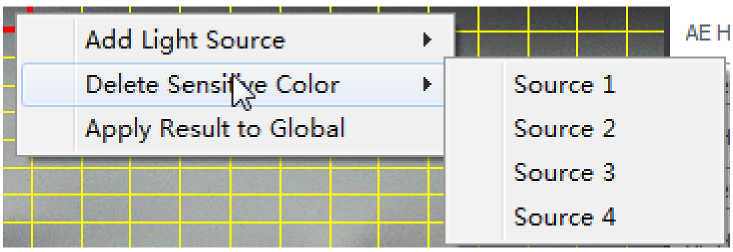
Step 4 Repeat step 1, step 2, and step 3 as required, and then click the **OK** button.

----End

After the setting is complete, you can capture the 3A analysis information once again. The Planckian coordinate graph is refreshed immediately and displays the independent illuminant and the corresponding region.

In addition, the user can add the independent illuminants quickly by selecting the region on the image. Right-click the region. Then the option menu shown in Figure 2-57 is displayed.

Figure 2-57 Option menu of adding independent illuminants



The user can select an independent illuminant region to add an independent illuminant in the **Add Light Source** menu item, and select an independent illuminant region to delete the interference region in the **Delete Sensitive Color** menu item. After that, you can enter the region radius in the dialog box to complete the adding operation.

CAUTION

If the independent illuminant region has been set before, it will be covered.

2.4.4.8 Applying Local AWB Result to Global AWB

The HiSilicon PQ Tools supports applying a local AWB result to the global AWB. This operation can be performed by right-clicking the region and select **Apply Result to Global**, as shown in Figure 2-57.

CAUTION

After this function is enabled, the AWB mode is forced into manual mode. If the automatic mode is required, you need to adjust related register parameters on the main page of the HiSilicon PQ Tools.

2.4.5 Auto Focus Simulator

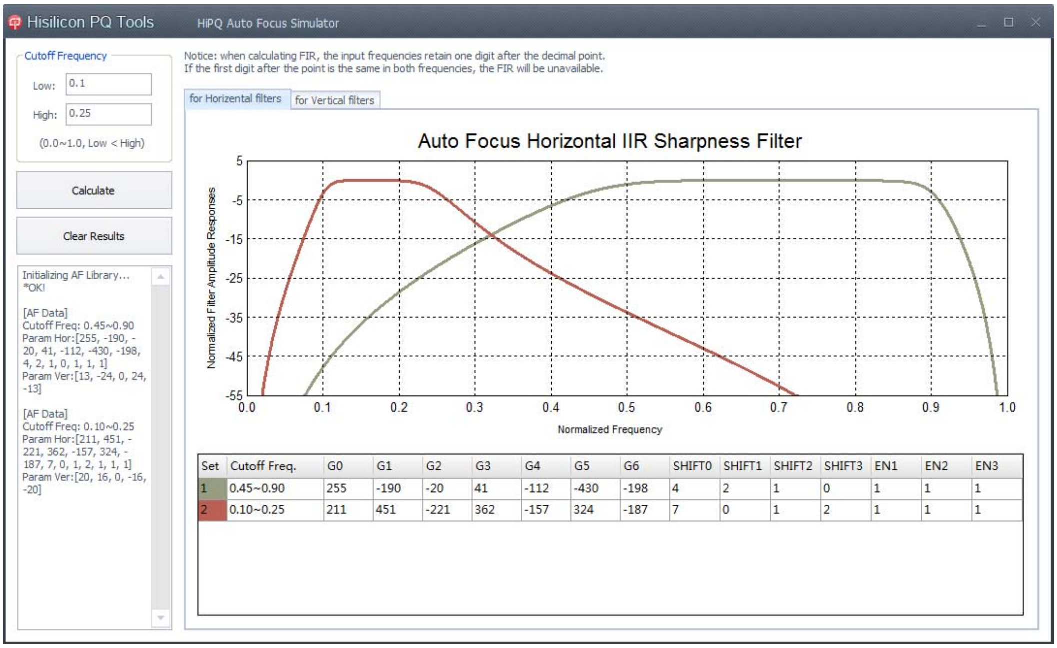
The HiPQ Auto Focus Simulator allows the picture quality debugging personnel to set and obtain appropriate AF parameters based on the cutoff frequency.

Currently the HiPQ Auto Focus Simulator is used only to adjust the picture quality of the Hi3516A. Before using the HiPQ Auto Focus Simulator, you need to install the Matlab 2012a (32-bit) version.

2.4.5.1 GUI

Choose **HiPQ Auto Focus Simulator** from the **Select an Extra Component** drop-down box in the main GUI of the HiSilicon PQ Tools. The **HiPQ Auto Focus Simulator** window shown in Figure 2-58 is displayed.

Figure 2-58 HiPQ Auto Focus Simulator



2.4.5.2 Obtaining the AF Parameters

Enter the high cutoff frequency and low cutoff frequency in the **High** text box and **Low** text box of the **Cutoff Frequency** group box respectively, and click **Calculate**. The infinite impulse response (IIR) and finite impulse response (FIR) parameter groups are automatically listed on the right. In addition, the frequency response curve of this group of data is drawn. You can click the tabs to view the corresponding parameters as required.

For the calculated results, the corresponding cells in the **Set** column are marked with different background colors. The background colors of the cells are the same as those of the frequency response curves.

Currently the **HiPQ Auto Focus Simulator** can list at most 20 groups of parameters. If 20 groups of parameters are listed, no more value is displayed. In this case, you can click **Clear Results** to clear the calculation results in the list.

2.5 Auxiliary Correction Tools

2.5.1 Calibration Importer

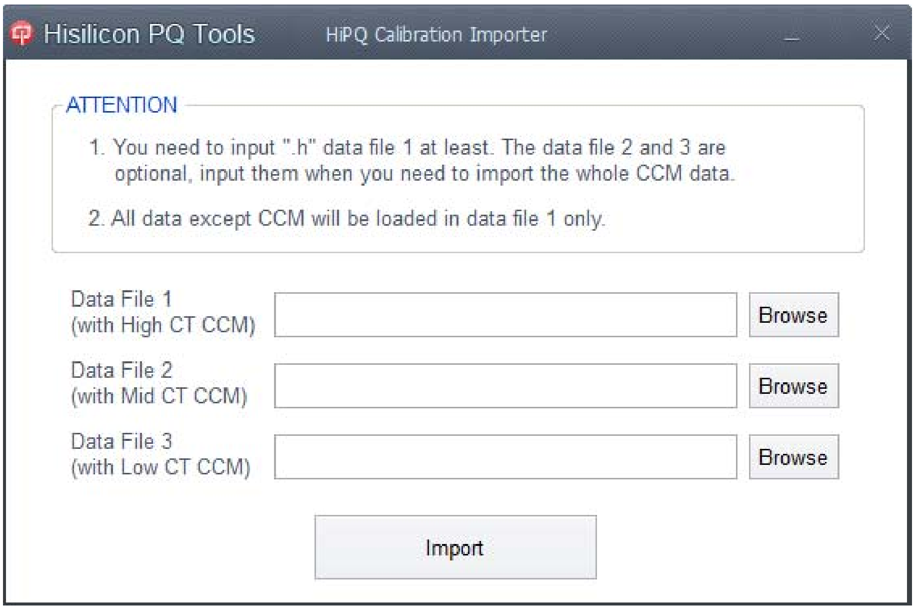
The HiSilicon PQ Tools provides the function of importing the .h data files generated by the Apical ISP Calibration tool, which allows the picture quality adjustment personnel to quickly apply the output data of calibration importer to the HiSilicon PQ Tools or the board.

Currently the calibration importer applies only to the picture quality adjustment of the Hi3518 series chips.

2.5.1.1 GUI

Choose **HiPQ Calibration Importer** from the **Select an Extra Component** drop-down box in the main GUI of the HiSilicon PQ Tools. The **HiPQ Calibration Importer** window shown in Figure 2-59 is displayed.

Figure 2-59 HiSilicon PQ Calibration Importer GUI



The Calibration Importer allows you to import the .h data files generated by the ISP Calibration tool to the opened PQ table.

2.5.1.2 Importing Data

Before importing data files, you need to obtain three .h files that are exported at high, middle, and low color temperatures respectively. The ISP Calibration tool can export the data at only one color temperature at a time. However, the data of the CCM (an important calibration parameter) are based on three color temperatures.

As shown in Figure 2-59, click **Browse** to select three .h files, and click **Import**. Then data is imported. You can find that the values of the calibration fields on the pages of the main GUI change.

CAUTION

The calibration data items except CCM are obtained from the data files 1 at the corresponding color temperature.

2.6 Other Auxiliary Tools

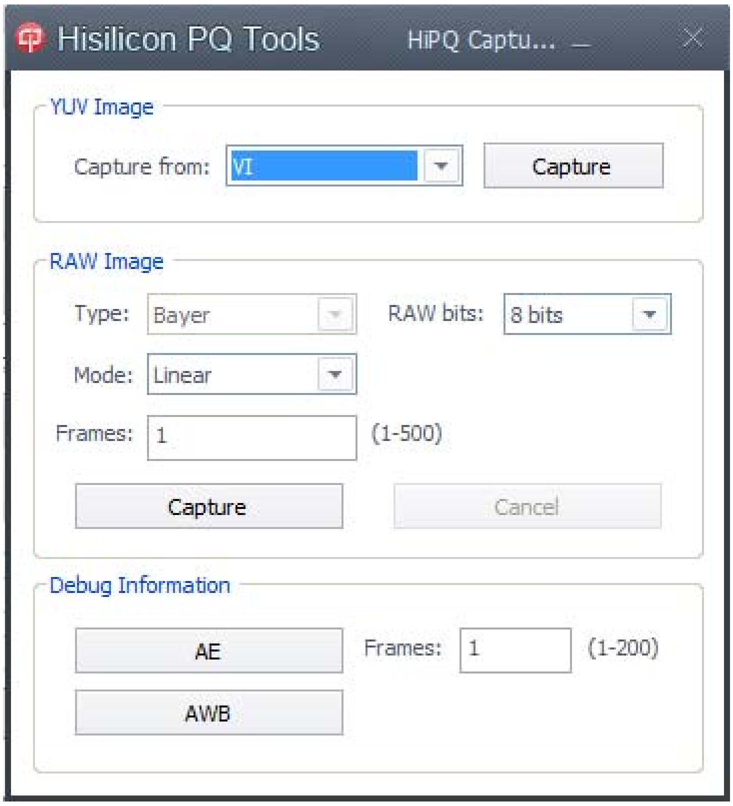
2.6.1 HiPQ Capture Tool

The HiPQ Capture Tool is used to capture images in RAW or YUV format and AE/AWB debugging information from the board.

2.6.1.1 GUI

Choose **HiPQ Capture Tool** from the **Select an Extra Component** drop-down box in the main GUI of the HiSilicon PQ Tools. The **HiPQ Capture Tool** window shown in Figure 2-60 is displayed.

Figure 2-60 HiPQ Capture Tool



2.6.1.2 Capturing YUV Image Data

Perform the following steps:

Step 1 In the **YUV Image Capture** group box, choose a position (VI, VPSS or VO) to capture the YUV data from **Capture from**.

Step 2 Click **Capture** in the **YUV Image Capture** group box.

Step 3 Select the save path in the displayed **Save As** dialog box after the data is captured successfully.

----End

2.6.1.3 Capturing RAW Image Data

Perform the following steps:

Step 1 In the **RAW Image Capture** group box, choose the bit width for the RAW data to be captured from **RAW bits**.

Step 2 Choose the mode of capturing the RAW data from the **Mode** drop-down list, which can be **Linear**, **WDR (VCNum0)**, or **WDR (VCNum1)**.

Step 3 In the **Frames** text box, enter the number of frame s to be captured (1–500).

Step 4 Click **Capture** in the **RAW Image Capture** group box.

Step 5 Select the save path in the displayed **Save As** dialog box after the data is captured successfully.

----End

2.6.1.4 Capturing AE/AWB Debugging Information

Perform the following steps:

Step 1 Enter the number of frames for capturing debugging information in the **Frames** text box of the **Debug Information** group box.

Step 2 Click **AE** or **AWB** as required (in the **Frames** text box on the right, enter the number of frames to be captured for capturing the AE debugging information)

Step 3 Select the save path in the displayed **Save As** dialog box after the data is captured successfully.

----End

CAUTION

- The captured RAW and YUV data is pure data without any header data.
- The captured AE and AWB debugging information is converted plaintext data.

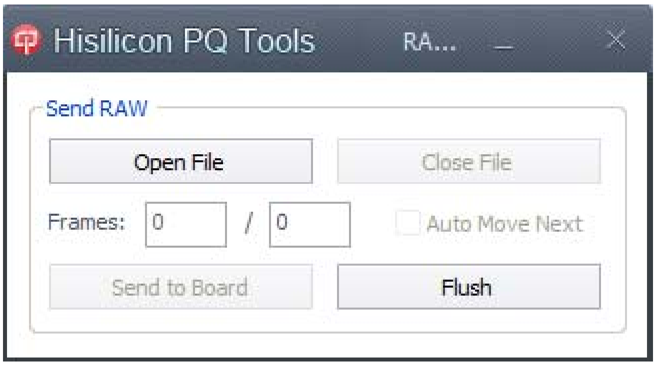
2.6.2 RAW Utilities

The RAW Utilities is used to export the captured RAW data to the board and capture RAW data of multiple frames.

2.6.2.1 GUI

Choose **HiPQ RAW Utilities** from the **Select an Extra Component** drop-down box in the main GUI of the HiSilicon PQ Tools. The **RAW Utilities** window shown in Figure 2-61 is displayed.

Figure 2-61 RAW Utilities



2.6.2.2 Exporting the Captured RAW Data to the Board

Perform the following steps:

Step 1 Click **Open File** in the **Send RAW** group box, and then select an RAW file.

Step 2 Specify the parameters of the RAW file in the displayed dialog box. Skip this step if the RAW file is obtained using a capture tool or HiPQ Capture Tool.

Step 3 Enter the sequence number (starting from 1) of RAW picture to be transmitted to the board in the **Frames** text box.

Step 4 Click **Send to Board** to transmit the RAW data to the board.

Step 5 Repeat step 3 and step 4 if multiple frames of data need to be transmitted. If the transmission is complete, click **Flush** to notify the board to stop transmitting data.

----End

If **Auto Move Next** is selected, the number of frames to be transmitted increases by 1 after a frame of RAW data is successfully transmitted. Multiple frames of data can be consecutively transmitted to the board using this function.

CAUTION

The opened RAW file is continuously occupied, and cannot be used by the plug-ins of the HiSilicon PQ Tools or external programs. Click **Close File** to close the file. Then the file can be used by other tools without closing the HiSilicon PQ Tools.

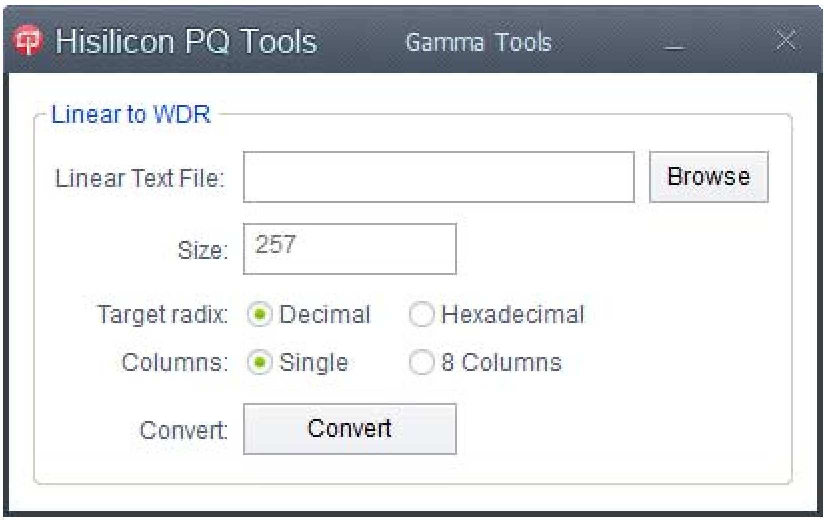
2.6.3 Gamma Tools

The Gamma Tools is used to convert the linear gamma data into data in WDR mode.

2.6.3.1 GUI

Choose **HiPQ Gamma Tools** from the **Select an Extra Component** drop-down box in the main GUI of the HiSilicon PQ Tools. The **Gamma Tools** window shown in Figure 2-62 is displayed.

Figure 2-62 Gamma Tools



2.6.3.2 Converting the Linear Gamma Data into Data in WDR Mode

Perform the following steps:

Step 1 Select a text file that contains linear Gamma data. Ensure that there is one data segment in a row in the text file, and the file format is supported by the Gamma Tools.

Step 2 Select the display format of the output data, which can be decimal or hexadecimal.

Step 3 Select the number of columns for displaying the output data (one or eight columns). If the output data is displayed in eight columns, the output values are separated in a format similar to the code array.

Step 4 Click **Convert** to start conversion.

Step 5 Select the save path in the displayed **Save As** dialog box after the conversion is complete.

----End

2.7 Parameter Descriptions

This section describes the APIs in the SDK corresponding to module parameters.

NOTE

For details about the APIs, see the *HiISP Development Reference* (Hi3516A) and the *3A HiISP Development Reference* (chips excluding Hi3516A).

2.7.1 Hi3518 Parameters

Table 2-3 APIs corresponding to the parameter related to SystemControl

Functional Module	SDK API
ModuleControl	HI_MPI_ISP_SetModuleControl HI_MPI_ISP_GetModuleControl
ISP FreezeFmw	HI_MPI_ISP_FreezeFmw
Color bar	None

Table 2-4 APIs corresponding to the parameter related to Gamma table

Functional Module	SDK API
Gamma	HI_MPI_ISP_GetGammaTable HI_MPI_ISP_SetGammaTable

Table 2-5 APIs corresponding to the parameter related to ExposureAttr

Functional Module	SDK API
Exposure_type	HI_MPI_ISP_SetExposureType HI_MPI_ISP_GetExposureType
Manual_AE	HI_MPI_ISP_SetMEAttr HI_MPI_ISP_GetMEAttr
Exposure_EX	HI_MPI_ISP_SetAEAttrEx HI_MPI_ISP_GetAEAttrEx
Slow FrameRate	HI_MPI_ISP_SetSlowFrameRate HI_MPI_ISP_GetSlowFrameRate
AntiFlicker	HI_MPI_ISP_SetAntiFlickerAttr HI_MPI_ISP_GetAntiFlickerAttr
AE_Route	HI_MPI_ISP_SetAERouteAttr HI_MPI_ISP_GetAERouteAttr
AE_DelayAttr	HI_MPI_ISP_SetAEDelayAttr HI_MPI_ISP_GetAEDelayAttr
AE_SubFlickerAttr	HI_MPI_ISP_SetAEDelayAttr HI_MPI_ISP_GetAEDelayAttr
AE_DcirisAttr	HI_MPI_ISP_SetDcirisAttr HI_MPI_ISP_GetDcirisAttr

Table 2-6 APIs corresponding to the parameter related to ExposureInfo

Functional Module	SDK API
ExposureInfo	HI_MPI_ISP_SetExpStaInfo HI_MPI_ISP_GetExpStaInfo
QueryInnerStateInfoEx	HI_MPI_ISP_QueryInnerStateInfoEx

Table 2-7 APIs corresponding to the parameter related to AI

Functional Module	SDK API
AI	HI_MPI_ISP_SetAIAttr HI_MPI_ISP_GetAIAttr

Table 2-8 APIs corresponding to the parameter related to WBAttr

Functional Module	SDK API
WB type	HI_MPI_ISP_SetWBType HI_MPI_ISP_GetWBType
AWB Alg type	HI_MPI_ISP_SetAWBAlgType HI_MPI_ISP_GetAWBAlgType
AdvAWBAttr	HI_MPI_ISP_SetAdvAWBAttr HI_MPI_ISP_GetAdvAWBAttr
AWBAttr	HI_MPI_ISP_SetAWBAttr HI_MPI_ISP_GetAWBAttr
MWBAttr	HI_MPI_ISP_SetMWBAttr HI_MPI_ISP_GetMWBAttr
BlackLevel	HI_MPI_ISP_SetBlackLevelAttr HI_MPI_ISP_GetBlackLevelAttr

Table 2-9 APIs corresponding to the parameter related to WBInfo

Functional Module	SDK API
ColorTemp	HI_MPI_ISP_GetColorTemp
WBInfo	HI_MPI_ISP_SetWBStaInfo HI_MPI_ISP_GetWBStaInfo

Table 2-10 APIs corresponding to the parameter related to CCM

Functional Module	SDK API
SaturationAttr	HI_MPI_ISP_SetSaturationAttr HI_MPI_ISP_GetSaturationAttr
CCM	HI_MPI_ISP_SetCCM HI_MPI_ISP_GetCCM
ColorTone	HI_MPI_ISP_SetColorTone HI_MPI_ISP_GetColorTone

Table 2-11 APIs corresponding to the parameter related to Timing and Image

Functional Module	SDK API
Timing	HI_MPI_ISP_GetInputTiming
Image	HI_MPI_ISP_SetImageAttr HI_MPI_ISP_GetImageAttr

Table 2-12 APIs corresponding to the parameter related to DRC

Functional Module	SDK API
DRC	HI_MPI_ISP_SetDRCAttr HI_MPI_ISP_GetDRCAttr

Table 2-13 APIs corresponding to the parameter related to AntiFalse

Functional Module	SDK API
AntiFalse	HI_MPI_ISP_SetAntiFalseColorAttr HI_MPI_ISP_GetAntiFalseColorAttr

Table 2-14 APIs corresponding to the parameter related to AntiFog

Functional Module	SDK API
AntiFog	HI_MPI_ISP_SetAntiFogAttr HI_MPI_ISP_GetAntiFogAttr

Table 2-15 APIs corresponding to the parameter related to Defect Pixel

Functional Module	SDK API
Defect Pixel	HI_MPI_ISP_SetDefectPixelAttr HI_MPI_ISP_GetDefectPixelAttr

Table 2-16 APIs corresponding to the parameter related to Shading

Functional Module	SDK API
ShadingAttr	HI_MPI_ISP_SetShadingAttr HI_MPI_ISP_GetShadingAttr
ShadingTable	HI_MPI_ISP_SetShadingTable HI_MPI_ISP_GetShadingTable

Table 2-17 APIs corresponding to the parameter related to Denoise

Functional Module	SDK API
Denoise	HI_MPI_ISP_SetDenoiseAttr HI_MPI_ISP_GetDenoiseAttr

Table 2-18 APIs corresponding to the parameter related to SharpenAttr

Functional Module	SDK API
SharpenAttr	HI_MPI_ISP_SetSharpenAttr HI_MPI_ISP_GetSharpenAttr

Table 2-19 APIs corresponding to the parameter related to DemosaicAttr

Functional Module	SDK API
DemosaicAttr	HI_MPI_ISP_SetDemosaicAttr HI_MPI_ISP_GetDemosaicAttr
NoiseProfile	HI_MPI_ISP_SetNoiseProfileTable HI_MPI_ISP_GetNoiseProfileTable

Table 2-20 APIs corresponding to the parameter related to CrosstalkAttr

Functional Module	SDK API
CrosstalkAttr	HI_MPI_ISP_SetCrosstalkAttr HI_MPI_ISP_GetCrosstalkAttr

Table 2-21 APIs corresponding to the parameter related to DIS

Functional Module	SDK API
DISAttr	HI_MPI_ISP_SetDISAttr HI_MPI_ISP_GetDISAttr
DISInfo	HI_MPI_ISP_GetDISInfo

Table 2-22 APIs corresponding to the parameter related to FPN

Functional Module	SDK API
FPN	HI_MPI_ISP_QueryInnerStateInfoEx

Table 2-23 APIs corresponding to the parameter related to VPSS

Functional Module	SDK API
GRP_PARAM	HI_MPI_VPSS_SetGrpParam HI_MPI_VPSS_GetGrpParam

Table 2-24 APIs corresponding to the parameter related to H264

Functional Module	SDK API
-	HI_MPI_VENC_SetChnAttr HI_MPI_VPSS_GetChnAttr HI_MPI_VENC_SetRcPara HI_MPI_VENC_GetRcPara

Table 2-25 APIs corresponding to the parameter related to VENC_Param

Functional Module	SDK API
LostFrameStrategy	HI_MPI_VENC_GetLostFrameStrategy HI_MPI_VENC_SetLostFrameStrategy

2.7.2 Hi3516A Parameters

Table 2-26 APIs corresponding to the parameter related to Top

Functional Module	SDK API
ISP_DEV	None
Venc Chn	None
FmwState	HI_MPI_ISP_SetFMWState HI_MPI_ISP_GetFMWState
Color bar	None
Bypass Setting	HI_MPI_ISP_SetModuleControl HI_MPI_ISP_GetModuleControl
ModParam	HI_HAL_ISP_SetModeParam HI_HAL_ISP_GetModeParam

Table 2-27 APIs corresponding to the parameter related to ExposureAttr

Functional Module	SDK API
ExposureAttr	HI_MPI_ISP_SetExposureAttr HI_MPI_ISP_GetExposureAttr

Table 2-28 APIs corresponding to the parameter related to WDRExposureAttr

Functional Module	SDK API
WDRExposureAttr	HI_MPI_ISP_SetWDRExposureAttr HI_MPI_ISP_GetWDRExposureAttr

Table 2-29 APIs corresponding to the parameter related to ExposureInfo

Functional Module	SDK API
ExposureInfo	HI_MPI_ISP_QueryExposureInfo

Table 2-30 APIs corresponding to the parameter related to AE Route

Functional Module	SDK API
AE Route	HI_MPI_ISP_SetAERouteAttr HI_MPI_ISP_GetAERouteAttr

Table 2-31 APIs corresponding to the parameter related to AI

Functional Module	SDK API
AIAttr	HI_MPI_ISP_SetIrisAttr HI_MPI_ISP_SetIrisAttr
DcirisAttr	HI_MPI_ISP_SetDcirisAttr HI_MPI_ISP_GetDcirisAttr
PirisAttr	HI_MPI_ISP_SetPirisAttr HI_MPI_ISP_GetPirisAttr

Table 2-32 APIs corresponding to the parameter related to WBAttr

Functional Module	SDK API
White Balance	HI_MPI_ISP_SetWBAttr HI_MPI_ISP_GetWBAttr
MWBATTR	HI_MPI_ISP_SetWBAttr HI_MPI_ISP_GetWBAttr
WBAttr	HI_MPI_ISP_SetWBAttr HI_MPI_ISP_GetWBAttr
AWBAttrEx	HI_MPI_ISP_SetAWBAttrEx HI_MPI_ISP_GetAWBAttrEx
BlackLevel	HI_MPI_ISP_SetBlackLevelAttr HI_MPI_ISP_GetBlackLevelAttr

Table 2-33 APIs corresponding to the parameter related to WBInfo

Functional Module	SDK API
WBInfo	HI_MPI_ISP_QueryWBInfo

Table 2-34 APIs corresponding to the parameter related to PubAttr

Functional Module	SDK API
PubAttr	HI_MPI_ISP_SetPubAttr HI_MPI_ISP_GetPubAttr

Table 2-35 APIs corresponding to the parameter related to CCM

Functional Module	SDK API
CCM	HI_MPI_ISP_SetCCMAttr HI_MPI_ISP_GetCCMAttr
ColorToneAttr	HI_MPI_ISP_SetColorToneAttr HI_MPI_ISP_GetColorToneAttr

Table 2-36 APIs corresponding to the parameter related to ACM_Coef

Functional Module	SDK API
ACM__Coef	HI_MPI_ISP_SetAcmCoeff

Table 2-37 APIs corresponding to the parameter related to ACM_Ctrl

Functional Module	SDK API
ACM__Ctrl	HI_MPI_ISP_SetAcmAttr HI_MPI_ISP_GetAcmAttr

Table 2-38 APIs corresponding to the parameter related to Saturation

Functional Module	SDK API
SaturationAttr	HI_MPI_ISP_SetSaturationAttr HI_MPI_ISP_GetSaturationAttr

Table 2-39 APIs corresponding to the parameter related to Demosaic

Functional Module	SDK API
Demosaic	HI_MPI_ISP_SetDemosaicAttr HI_MPI_ISP_GetDemosaicAttr

Table 2-40 APIs corresponding to the parameter related to Sharpen

Functional Module	SDK API
Sharpen	HI_MPI_ISP_SetSharpenAttr HI_MPI_ISP_GetSharpenAttr

Table 2-41 APIs corresponding to the parameter related to NR

Functional Module	SDK API
Denoise	HI_MPI_ISP_SetNRAttr HI_MPI_ISP_GetNRAttr
3D NR	HI_MPI_VPSS_SetGrpParam HI_MPI_VPSS_GetGrpParam
NoiseProfile	HI_MPI_ISP_SetNPTable HI_MPI_ISP_GetNPTable

Table 2-42 APIs corresponding to the parameter related to 3DNR

Functional Module	SDK API
VppNRbEx_Group	HI_MPI_VPSS_SetNRV3Param HI_MPI_VPSS_GetNRV3Param

Table 2-43 APIs corresponding to the parameter related to FSWDR

Functional Module	SDK API
FSWDR	HI_MPI_ISP_SetFSWDRAttr HI_MPI_ISP_GetFSWDRAttr

Table 2-44 APIs corresponding to the parameter related to DRC

Functional Module	SDK API
DRC	HI_MPI_ISP_SetDRCAttr HI_MPI_ISP_GetDRCAttr

Table 2-45 APIs corresponding to the parameter related to Gamma

Functional Module	SDK API
Gamma	HI_MPI_ISP_SetGammaAttr HI_MPI_ISP_GetGammaAttr

Table 2-46 APIs corresponding to the parameter related to DCI

Functional Module	SDK API
DCI	HI_MPI_VI_SetDCIParam HI_MPI_VI_GetDCIParam

Table 2-47 APIs corresponding to the parameter related to CSC

Functional Module	SDK API
CSC	HI_MPI_VI_SetCSCAttr HI_MPI_VI_GetCSCAttr

Table 2-48 APIs corresponding to the parameter related to AntiFalseColor

Functional Module	SDK API
AntiFalse	HI_MPI_ISP_SetAntiFalseColorAttr HI_MPI_ISP_GetAntiFalseColorAttr

Table 2-49 APIs corresponding to the parameter related to CrosstalkRemoval

Functional Module	SDK API
CrosstalkRemoval	HI_MPI_ISP_SetCrosstalkAttr HI_MPI_ISP_GetCrosstalkAttr

Table 2-50 APIs corresponding to the parameter related to DeFog

Functional Module	SDK API
DeFog	HI_MPI_ISP_SetDeFogAttr HI_MPI_ISP_GetDeFogAttr

Table 2-51 APIs corresponding to the parameter related to Shading

Functional Module	SDK API
ShadingAttr	HI_MPI_ISP_SetShadingAttr HI_MPI_ISP_GetShadingAttr

Table 2-52 APIs corresponding to the parameter related to DIS

Functional Module	SDK API
DIS	HI_MPI_ISP_SetDISAttr HI_MPI_ISP_GetDISAttr

Table 2-53 APIs corresponding to the parameter related to Defect Pixel

Functional Module	SDK API
DPCalibrate	HI_MPI_ISP_SetDPCalibrate HI_MPI_ISP_GetDPCalibrate
DPAttr	HI_MPI_ISP_SetDPAttr HI_MPI_ISP_GetDPAttr

Table 2-54 APIs corresponding to the parameter related to FPN

Functional Module	SDK API
FPNCalibrate	HI_MPI_ISP_FPNCalibrate
FPNCorrection	HI_MPI_ISP_SetFPNAttr HI_MPI_ISP_GetFPNAttr

Table 2-55 APIs corresponding to the parameter related to H264

Functional Module	SDK API
	HI_MPI_VENC_GetRcParam HI_MPI_VENC_SetRcParam HI_MPI_VENC_GetChnAttr HI_MPI_VENC_SetChnAttr

Table 2-56 APIs corresponding to the parameter related to H265

Functional Module	SDK API
	HI_MPI_VENC_GetRcParam HI_MPI_VENC_SetRcParam HI_MPI_VENC_GetChnAttr HI_MPI_VENC_SetChnAttr

Table 2-57 APIs corresponding to the parameter related to RCParam

Functional Module	SDK API
SuperFrameCfg	HI_MPI_VENC_SetSuperFrameCfg HI_MPI_VENC_GetSuperFrameCfg
LostStrategy	HI_MPI_VENC_SetFrameLostStrategy HI_MPI_VENC_GetFrameLostStrategy

Table 2-58 APIs corresponding to the parameter related to StatisticsConfig

Functional Module	SDK API
AWBConfig	HI_MPI_ISP_SetStatisticsConfig HI_MPI_ISP_GetStatisticsConfig
AFConfig	HI_MPI_ISP_SetStatisticsConfig HI_MPI_ISP_GetStatisticsConfig

Table 2-59 APIs corresponding to the parameter related to ISPInfo

Functional Module	SDK API
ISPInfo	HI_MPI_ISP_QueryInnerStateInfo

Table 2-60 APIs corresponding to the parameter related to Venc_ex

Functional Module	SDK API
Intra Refresh	HI_MPI_VENC_SetIntraRefresh HI_MPI_VENC_GetIntraRefresh
ROIConfig	HI_MPI_VENC_SetRoiCfg HI_MPI_VENC_GetRoiCfg
ROIBG_Frame_Rate	HI_MPI_VENC_SetRoiBgFrameRate

Functional Module	SDK API
	HI_MPI_VENC_GetRoiBgFrameRate
Color2Grey	HI_MPI_VENC_SetColor2Grey HI_MPI_VENC_GetColor2Grey
REF_EX	HI_MPI_VENC_SetRefParamEx HI_MPI_VENC_GetRefParamEx

3 Application Reference

3.1 Importing and Exporting HiSilicon PQ Tools Parameters

3.1.1 Importing and Exporting Parameters Using the HiSilicon PQ Tools

A configuration file can be imported to or exported from the HiSilicon PQ Tools, and HiSilicon PQ Tools parameters can be hardened on the board.

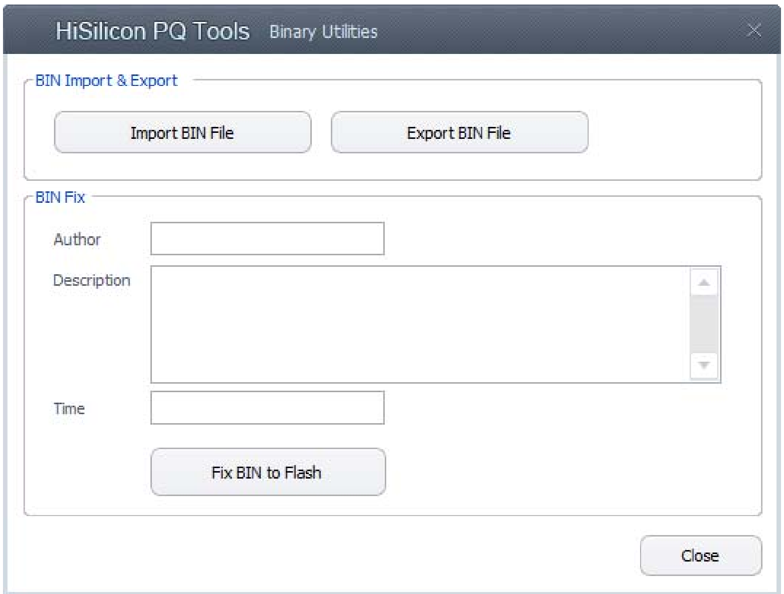
To import a parameter file to the HiSilicon PQ Tools or export the parameters used by the HiSilicon PQ Tools, save or open a .sav file by following section 1.4.3.2 "Saving a PQ Data File" and section 1.4.3.3 "Opening a PQ Data File."

To import parameters to the board or export parameters from the board for backup, use the binary data processing function as follows:

Step 1 Import a .bin file to the board or export parameters from the board as a .bin file.

After the board is connected and a PQ table is opened, click  on the toolbar to open the **Binary Utilities** dialog box shown in Figure 3-1.

Figure 3-1 Binary Utilities dialog box



The **Binary Utilities** dialog box allows you to perform the following operations:

1. Export parameters from the board as a .bin file: Click **Export BIN File**, set a file name and save path, and click **Save** in the displayed **Save As** dialog box. Then the current parameters of the board are saved in the specified path.
2. Import a .bin parameter file to the board: Click **Import BIN File**, select a .bin parameter file, and click **Open** in the displayed **Open** dialog box. After the HiSilicon PQ Tools sends the imported file to the board, the parameters take effect immediately.
3. Harden parameters: Enter information in **Author**, **Description**, and **Time** in the **BIN Fix** group box, and click **Fix BIN to Flash**. Then the HiSilicon PQ Tools sends a hardening command to the board to write the current PQ parameters of the board to the flash memory.

Step 2 Harden configurations to the flash memory.

Harden the .bin file: Click  on the toolbar to open the **Binary Utilities** dialog box. Enter information in **Author**, **Description**, and **Time** in the **BIN Fix** group box, and click **Fix BIN to Flash**. Then the board generates a .bin file in the specified save path based on the current parameters of the board. You can configure the .bin file. There is a **config.cfg** file in the directory where the board program is located. The **[Fix]** and **[Export]** items in **config.cfg** indicate the path of the .bin file to be hardened and the path of the .bin file exported from the board respectively. You can enter paths and file names in **[Fix]** and **[Export]**. For example, if **fixfile.bin** is entered in **[Fix]**, the current path is the path of the .bin file to be hardened, and the file to be hardened is **fixfile.bin**.

CAUTION

- The exported .bin file is dedicated to the board and can be opened in the main program of the HiSilicon PQ Tools only by importing it in the **Binary Utilities** dialog box.
- The .bin file must be hardened before it is imported or exported. Otherwise, the import or export operation fails.
- When the .bin file is hardened to the flash memory, the path specified in **[Fix]** must be the same as the path specified in **[Export]**. Otherwise, the .bin file that is successfully hardened cannot be exported.
- The data irrelevant to the statistics of the ISP module is stored in the .bin file.

----End

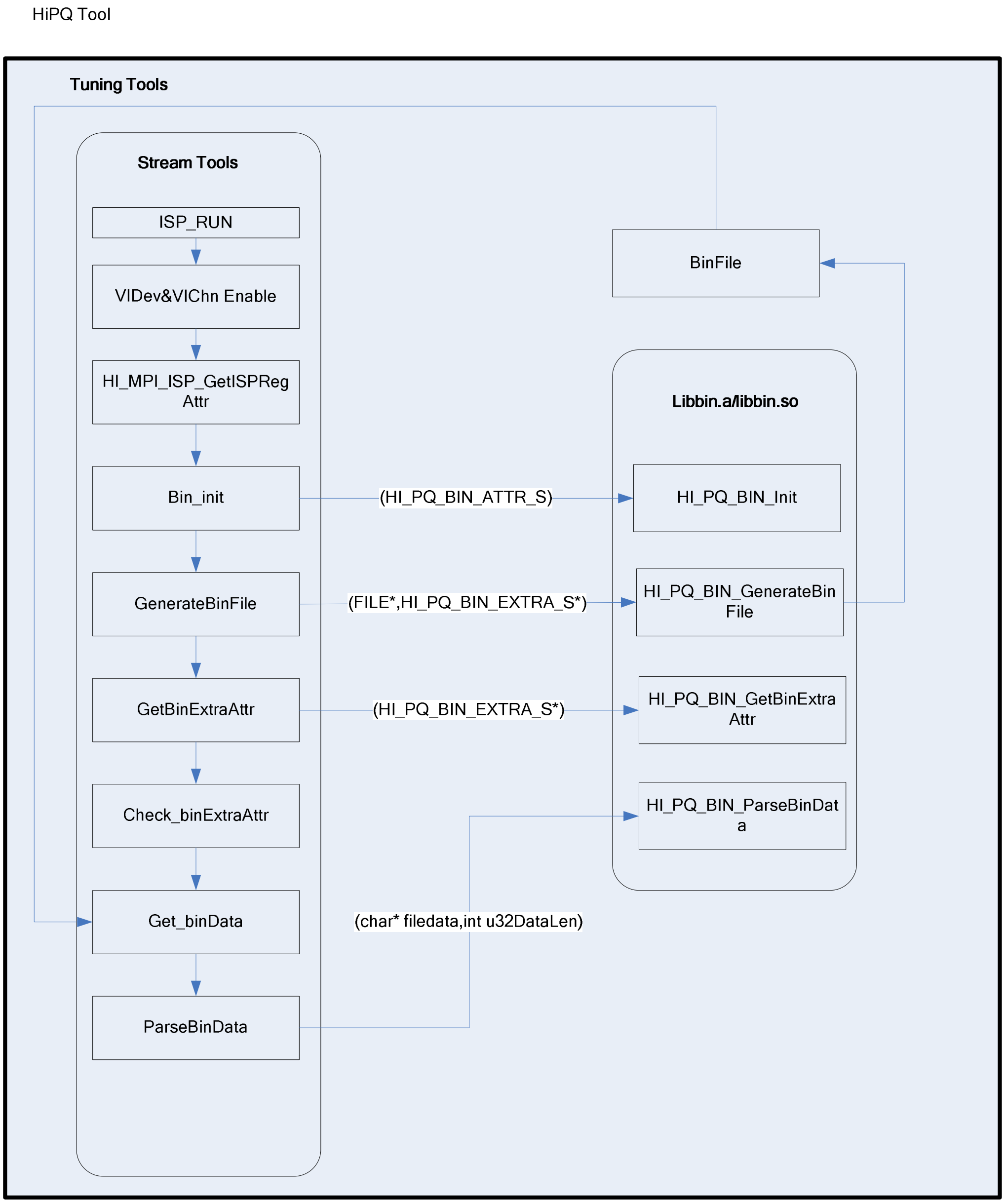
3.1.2 Importing and Exporting Picture Quality Parameters Using Libraries

The HiSilicon PQ Tools provides library files for importing and exporting parameters. You can choose functions according to your own needs.

- HI_PQ_BIN_Init: initializes parameters that bin needs
- HI_PQ_BIN_GenerateBinFile: generates bin files
- HI_PQ_BIN_GetBinExtraAttr: obtains information about bin header file
- HI_PQ_BIN_ParseBinData: parses bin files and validates recorded parameters

You can refer to the recommended usage process, as shown in Figure 3-2.

Figure 3-2 Recommended usage process of Bin



3.1.2.2 API Reference

HI_PQ_BIN_Init

[Description]

Initializes parameters that bin needs.

[Syntax]

```
void HI_PQ_BIN_Init(HI_PQ_BIN_ATTR_S stBinAttr);
```

[Parameter]

Parameter Name	Definition	Input/Output
stBinAttr	Bin attribute structure	Input

[Return Value]

Return Value	Definition
None	None

[Error Code]

Error Code	Description
None	None

[Requirement]

Header file: hi_pq_bin.h

Library file: libbin.a

[Note]

This function must be called before other interfaces of bin are called.

Before this function is called, you need to call HI_MPI_ISP_GetISPRegAttr in the SDK to obtain the address and length of each module, and the information obtained needs to be entered in HI_PQ_BIN_ATTR_S structure.

The number of times that this function is called is not limited.

[Example]

```
int s32Ret = -1;
HI_PQ_BIN_ATTR_S stBinAttr;
ISP_REG_ATTR_S  stIspRegAttr;
s32Ret = HI_MPI_ISP_GetISPRegAttr(&stIspRegAttr);
if(0 != s32Ret)
{
printf("HI_MPI_ISP_GetISPRegAttr err 0x%x\n",s32Ret);
return -1;
}
stBinAttr.stIspRegAttr.u32AeExtRegAddr = stIspRegAttr.u32AeExtRegAddr;
stBinAttr.stIspRegAttr.u32AeExtRegSize = stIspRegAttr.u32AeExtRegSize;
stBinAttr.stIspRegAttr.u32AwbExtRegAddr = stIspRegAttr.u32AwbExtRegAddr;
stBinAttr.stIspRegAttr.u32AwbExtRegSize = stIspRegAttr.u32AwbExtRegSize;
stBinAttr.stIspRegAttr.u32IspExtRegAddr = stIspRegAttr.u32IspExtRegAddr;
stBinAttr.stIspRegAttr.u32IspExtRegSize = stIspRegAttr.u32IspExtRegSize;
```

```
stBinAttr.stIspRegAttr.u32IspRegAddr = stIspRegAttr.u32IspRegAddr;
stBinAttr.stIspRegAttr.u32IspRegSize = stIspRegAttr.u32IspRegSize;
stBinAttr.u32ChipId = 0x3516a;
HI_PQ_BIN_Init(stBinAttr);
```

HI_PQ_BIN_GenerateBinFile

[Description]

Generates bin file.

[Syntax]

```
int HI_PQ_BIN_GenerateBinFile(FILE* fp,HI_PQ_BIN_EXTRA_S* pstBinExtra);
```

[Parameter]

Parameter Name	Definition	Input/Output
Fp	File handle	Input
pstBinExtra	Pointer to the structure of bin configuration information	Input

[Return Value]

Return Value	Definition
0	Success
Other values	Failure. The return value is an error code.

[Error Code]

Error Code	Description
0xCB000001	The input argument pointer is null.
0xCB000002	There are errors in the structure of initial configuration.
0xCB000003	No memory space is allocated.
0xCB000004	The model of initially configured chip does not match that of the current chip.

[Requirement]

Header file: hi_pq_bin.h

Library file: libbin.a

[Note]

HI_PQ_BIN_Init must be called before this function is called.

Before this function is called, the file handle needs to be accessed. When the calling is ended, the handle needs to be recycled manually.

[Example]

```
int s32Ret = -1;
FILE* pFile = NULL;
HI_PQ_BIN_EXTRA_S stBinExtra;

pFile = fopen("Sample.bin","wb+");
if (NULL == pFile)
{
return -1;
}
memset(stBinExtra.au8Author,0,sizeof(stBinExtra.au8Author));
memset(stBinExtra.au8Desc,0,sizeof(stBinExtra.au8Desc));
memset(stBinExtra.au8Time,0,sizeof(stBinExtra.au8Time));
s32Ret = HI_PQ_BIN_GenerateBinFile(pFile,&stBinExtra);
if(0 != s32Ret)
{
printf("HI_PQ_BIN_GenerateBinFile error! errno(0x%x)\n",s32Ret);
fclose(pFile);
return -1;
}
fclose(pFile);
```

HI_PQ_BIN_GetBinExtraAttr

[Description]

Obtains information about bin header file.

[Syntax]

```
int HI_PQ_BIN_GetBinExtraAttr(unsigned char*
pu8FileData,HI_PQ_BIN_EXTRA_S* pstBinExtraAttr);
```

[Parameter]

Parameter Name	Definition	Input/Output
pu8FileData	Pointer to the content of bin file	Input
pstBinExtraAttr	Pointer to the structure of bin header file configuration information	Output

[Return Value]

Return Value	Definition
0	Success
Other values	Failure. The return value is an error code.

[Error Code]

Error Code	Description
0xCB000001	The input argument pointer is null.

[Requirement]

Header file: hi_pq_bin.h

Library file: libbin.a

[Note]

HI_PQ_BIN_Init must be called before this function is called.

Before this function is called, the bin file information must be read, and the pointer to the start address information must be transferred to the interface.

[Example]

```
int s32Ret = -1;
FILE* pFile = NULL;
long size = 0;
int len = 0;
HI_PQ_BIN_EXTRA_S stBinExtraAttr;
pFile = fopen("Sample.bin","r");
if (NULL == pFile)
{
return -1;
}
fseek(pFile, 0, SEEK_END);
size = ftell(pFile);
fseek(pFile, 0, SEEK_SET);
pu8FileData = (unsigned char*)malloc(size);
if(NULL == pu8FileData)
{
printf("malloc error\n");
fclose(pFile);
break;
}
memset(pu8FileData, 0, size);
len = fread(pu8FileData, sizeof(unsigned char), size, pFile);
```

```
if(len<=0)
{
printf("read erro\n");
free(pu8FileData);
pu8FileData = NULL;
fclose(pFile);
break;
}
s32Ret = HI_PQ_BIN_GetBinExtraAttr(pu8FileData,&stBinExtraAttr);
if(0 != s32Ret)
{
printf("HI_PQ_BIN_GetBinExtraAttr error! errno(0x%x)\n",s32Ret);
}
free(pu8FileData);
pu8FileData = NULL;
fclose(pFile);
```

HI_PQ_BIN_ParseBinData

[Description]

Parses bin files and validates recorded parameters.

[Syntax]

```
int HI_PQ_BIN_ParseBinData(unsigned char* pu8FileData,unsigned int
u32DataLen);
```

[Parameter]

Parameter Name	Definition	Input/Output
pu8FileData	Pointer to the content of bin file	Input
u32DataLen	Data length	Input

[Return Value]

Return Value	Definition
0	Success
Other values	Failure. The return value is an error code.

[Error Code]

Error Code	Description
0xCB000001	The input argument pointer is null.

Error Code	Description
0xCB000003	No memory space is allocated.
0xCB000005	The data is incomplete.
0xCB000006	Data lengths do not match.
0xCB000007	The information in the header file is damaged.
0xCB000008	The data is damaged.

[Requirement]

Header file: hi_pq_bin.h

Library file: libbin.a

[Note]

HI_PQ_BIN_Init must be called before this function is called.

Before this function is called, the bin file information must be read, and the pointer to the start address information must be transferred to the interface.

[Example]

```
int s32Ret = -1;
FILE* pFile = NULL;
long size = 0;
int len = 0;
pFile = fopen("Sample.bin","r");
if (NULL == pFile)
{
break;
}
fseek(pFile, 0, SEEK_END);
size = ftell(pFile);
fseek(pFile, 0, SEEK_SET);
pu8FileData = (unsigned char*)malloc(size);
if(NULL == pu8FileData)
{
printf("malloc error\n");
fclose(pFile);
break;
}
memset(pu8FileData, 0, size);
len = fread(pu8FileData, sizeof(unsigned char), size, pFile);
if(len<=0)
{
printf("read erro\n");
```

```
free(pu8FileData);
pu8FileData = NULL;
fclose(pFile);
break;
}
s32Ret = HI_PQ_BIN_ParseBinData(pu8FileData,size);
if(0 != s32Ret)
{
printf("HI_PQ_BIN_ParseBinData error! errno(0x%x)\n",s32Ret);
}
free(pu8FileData);
pu8FileData = NULL;
```

3.1.2.3 Data Structures

HI_PQ_BIN_ATTR_S

[Description]

Initializes bin parameters.

[Syntax]

```
typedef struct hiPQ_BIN_ATTR_S
{
  unsigned int  u32ChipId;
  HI_PQ_REG_ATTR_S  stIspRegAttr;
}HI_PQ_BIN_ATTR_S;
```

[Member]

Member name	Definition
u32ChipId	Methods to show the chip type: 0x3518a\0x3516a
stIspRegAttr	Information about the register, which can be obtained by using **HI_MPI_ISP_GetISPRegAttr** SDK interface

[See Also]

HI_PQ_REG_ATTR_S

HI_PQ_REG_ATTR_S

[Description]

Register information structure

[Syntax]

```
typedef struct hiPQ_REG_ATTR_S
```

```
{
    unsigned int u32IspRegAddr;
    unsigned int u32IspRegSize;
    unsigned int u32IspExtRegAddr;
    unsigned int u32IspExtRegSize;
    unsigned int u32AeExtRegAddr;
    unsigned int u32AeExtRegSize;
    unsigned int u32AwbExtRegAddr;
    unsigned int u32AwbExtRegSize;
}HI_PQ_REG_ATTR_S;
```

[Member]

Member name	Definition
u32IspRegAddr	Start address for ISP physical registers
u32IspRegSize	Total length of ISP physical registers
u32IspExtRegAddr	Start address for ISP external registers
u32IspExtRegSize	Total length of ISP external registers
u32AeExtRegAddr	Start address for ISP AE module registers
u32AeExtRegSize	Total length of ISP AE module registers
u32AwbExtRegAddr	Start address for ISP AWB module registers
u32AwbExtRegSize	Total length of ISP AWB module registers

[See Also]

HI_PQ_BIN_ATTR_S

HI_PQ_BIN_EXTRA_S

[Description]

Configuration information about the header of Bin file

[Syntax]

```
typedef struct hiPQ_BIN_EXTRA_S
{
    unsigned char  au8Author[32];
    unsigned char  au8Desc[1024];
    unsigned char  au8Time[32];
}HI_PQ_BIN_EXTRA_S;
```

[Member]

Member name	Definition
au8Author	Author
au8Desc	Describe the version
au8Time	Generation time

[See Also]

None

3.2 Replacing the 3A Algorithm

You can change the value corresponding to an address by configuring the .xml file. To replace the 3A algorithm, update the address and format options of the 3A algorithm in the .xml file (for details, see section 3.3 "Adding Adjustment Options for Physical Registers"), and import the configured .xml file by using the HiSilicon PQ Tools.

NOTE

The Huawei LiteOS system does not support this function currently.

3.3 Adding Adjustment Options for Physical Registers

The following 15 parameter adjustment options are provided in the released .xml file. If you want to add options, enter them in the format of the existing options in the .xml file.

- **Page Name**: current page name
- **Group Name**: module name
- **IsEntirety**: Its value is **TURE** or **FALSE**. **TURE** indicates that the entire group of a module is read and written, which applies only to the scenario of calling APIs. Select **FALSE** when adding physical registers and virtual registers.
- **Field Name**: register names
- **Type**: For the common value, set **Type** to **virtual** or **physical** (not indicating the physical or virtual register); for two values (such as enable and disable), set **Type** to **bool**; for the enumeration, set **Type** to **list**; for the matrix, set **Type** to **matrix**.
- **Parameter**: Set it only when **Type** is **list**.
- **Radix**: **10** (decimal value on the GUI) or **16** (hexadecimal value on the GUI)
- **Signed**: signed or unsigned data
- **Address**: register address. It is the most important option, and a correct register address is required.
- **AddressType**: register type (physical or virtual register). Enter **Physical** here.
- **BitRange**: register bit range. For example, 0–3 indicate bits 0–3 of the start address.
- **ValueRange**: value range of register parameters
- **Options**: register attribute (read/write or read-only)

- **DisplayCalc**: formula for the conversion calculations for some parameters. Left this option blank if no conversion calculation is required.
- **Description**: parameter description that is displayed on the GUI

After you add register adjustment options and set the preceding 15 options in sequence, the new options are displayed on the GUI when the HiSilicon PQ Tools runs.

Figure 3-3 shows the DRC parameter adjustment GUI, and Figure 3-4 is the corresponding .xml file.

Figure 3-3 DRC parameter adjustment GUI

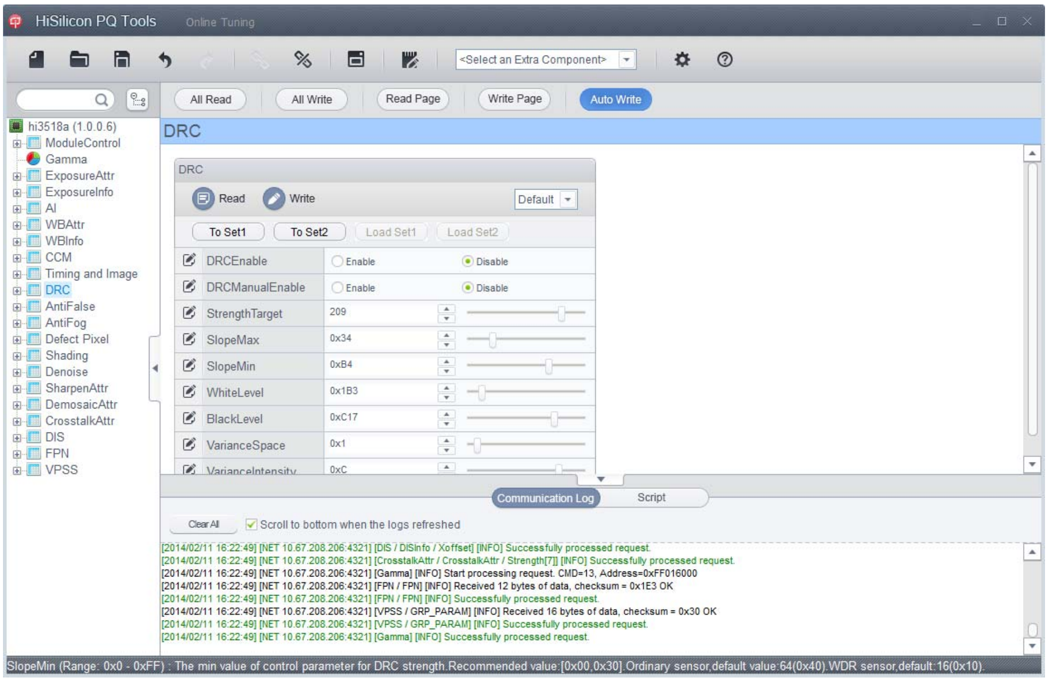


Figure 3-4 .xml configuration file of DRC parameters

3.4 Reducing the Application Size at the Board End

3.4.1 Calling Library Functions

The HiSilicon PQ Tools can run on the board by calling the library function in the Linux system. The pre-compilation macros in Table 3-1 need to be added accordingly for different chips. In addition, only the chips listed in Table 3-1 support this function.

Table 3-1 Precompilation macros corresponding to different chips

Chip	Precompilation Macro
Hi3516A V100R001	-DITTB_HICHIP= 0x3516a
Hi3516D V100R001	-DITTB_HICHIP= 0x3516a

3.4.1.2 API Reference

HI_SERVER_INIT

[Description]

Initializes the HiSilicon PQ Tools.

[Syntax]

```
HI_S32 HI_SERVER_INIT();
```

[Parameter]

Parameter	Description	Input/Output
None	None	None

[Return Value]

Return Value	Description
None	None

[Error Code]

Return Value	Description
0	Success
–1	Failure

[Requirement]

- Library file: **lib_pqcontrol.a**
- Header file: **hi_server.h**

[Note]

This function must be called before other interfaces are called.

HI_SERVER_START

[Description]

Starts the HiSilicon PQ Tools.

[Syntax]

```
HI_S32 HI_SERVER_START ();
```

[Parameter]

Parameter	Description	Input/Output
None	None	None

[Return Value]

Return Value	Description
None	None

[Error Code]

Return Value	Description
0	Success
–1	Failure

[Requirement]

- Library file: **lib_pqcontrol.a**
- Header file: **hi_server.h**

[Note]

- This function can be called only after HI_SERVER_INIT is called.
- This function is a block interface.

HI_SERVER_DINIT

[Description]

Exits the HiSilicon PQ Tools.

[Syntax]

```
HI_S32 HI_SERVER_DINIT ();
```

[Parameter]

Parameter	Description	Input/Output
None	None	None

[Return Value]

Return Value	Description
None	None

[Error Code]

Return Value	Description
0	Success
–1	Failure

[Requirement]

- Library file: **lib_pqcontrol.a**
- Header file: **hi_server.h**

[Note]

None

4 FAQs

4.1 What Do I Do If the HiSilicon PQ Tools Version and .xml File Version Mismatch?

[Symptom]

The HiSilicon PQ Tools on the board end can be properly started. However, when the board is connected after an .xml file is opened, a message is displayed, indicating that the HiSilicon PQ Tools version on the board end and the .xml file version mismatch.

[Cause Analysis]

The version of the HiSilicon PQ Tools on the board end is later than the .xml file version.

[Solution]

Check the SDK version and update the .xml file based on the HiSilicon PQ Tools corresponding to the current SDK version.

4.2 How Do I Set Associated Parameters?

[Symptom]

Some parameters are associated with others. For example, the maximum value of parameter A may depend on the current value of parameter B in the same group. However, the maximum value of parameter A displayed on the GUI is always the maximum value of parameter B currently. You need to ensure that the adjusted value of parameter A is less than or equal to the current value of parameter B. Otherwise, the picture quality is affected. For details about the value range and description of variables, see their tips on the GUI.

[Cause Analysis]

Table 4-1 lists all associated parameters.

Table 4-1 Associated parameters of Hi3518

Corresponding Functional Module	Corresponding SDK API	Associated Parameter
Exposure_EX	HI_MPI_ISP_SetAEAttrEx	**ExpTimeMin** and

Corresponding Functional Module	Corresponding SDK API	Associated Parameter
	HI_MPI_ISP_GetAEAttrEx	**ExpTimeMax** **AgainMin** and **AGainMax** **DgainMin** and **DGainMax**
AE_DcirisAttr	HI_MPI_ISP_SetDcirisAttr HI_MPI_ISP_GetDcirisAttr	**MinPwmDuty**<=**OpenPwmDuty**<= **MaxPwmDuty**
ExposureInfo	HI_MPI_ISP_SetExpStaInfo HI_MPI_ISP_GetExpStaInfo	**ExpHistThresh[0]** **ExpHistThresh[1]** **ExpHistThresh[2]** **ExpHistThresh[3]** The value of each parameter should be greater than that of the previous parameter.
WBInfo	HI_MPI_ISP_SetWBStaInfo HI_MPI_ISP_GetWBStaInfo	**WhiteLevel** and **BlackLevel** **CbMin** and **CbMax** **CrMin** and **CrMax**
CCM	HI_MPI_ISP_SetCCM HI_MPI_ISP_GetCCM	**HighColorTemp** **MidColorTemp** **LowColorTemp** The maximum value of **MidColorTemp** is **HighColorTemp** minus 400, and the maximum value of **LowColorTemp** is **MidColorTemp** minus 400.
Denoise	HI_MPI_ISP_SetDenoiseAttr HI_MPI_ISP_GetDenoiseAttr	**ThreshTarget** and **ThreshMax**
SharpenAttr	HI_MPI_ISP_SetSharpenAttr HI_MPI_ISP_GetSharpenAttr	**StrengthMin** and **StrengthTarget**

Table 4-2 Associated parameters of Hi3516A

Corresponding Functional Module	Corresponding SDK API	Associated Parameter
ExposureAttr	HI_MPI_ISP_SetExposureAttrHI_MPI_ISP_GetExposureAttr	**ExpTimeMin** and **ExpTimeMax** **AgainMin** and **AGainMax** **DgainMin** and **DGainMax** **ISPDGainMin** and **ISPDGainMax**

Corresponding Functional Module	Corresponding SDK API	Associated Parameter
AE Route	HI_MPI_ISP_SetAERouteAttr HI_MPI_ISP_GetAERouteAttr	The **IntTime** value of each node should be greater than that of the previous node.
DcirisAttr	HI_MPI_ISP_SetDcirisAttr HI_MPI_ISP_GetDcirisAttr	MinPwmDuty<=OpenPwmDuty<= MaxPwmDuty
PirisAttr	HI_MPI_ISP_SetPirisAttr HI_MPI_ISP_GetPirisAttr	MinIrisFNOTarget <=MaxIrisFNOTarget StepCount<= TotalStep
CCM	HI_MPI_ISP_SetCCMAttr HI_MPI_ISP_GetCCMAttr	**HighColorTemp** **MidColorTemp** **LowColorTemp** The maximum value of **MidColorTemp** is **HighColorTemp** minus 400, and the maximum value of **LowColorTemp** is **MidColorTemp** minus 400.
DRC	HI_MPI_ISP_SetDRCAttr HI_MPI_ISP_GetDRCAttr	**SlopeMax** and **SlopeMin**
DPCalibrate	HI_MPI_ISP_SetDPCalibrate HI_MPI_ISP_GetDPCalibrate	**CountMax** and **CountMin**
FSWDR	HI_MPI_ISP_SetFSWDRAttr HI_MPI_ISP_GetFSWDRAttr	**ShortThresh** and **LongThresh**

StatisticsConfig	HI_MPI_ISP_SetStatisticsConfig
	HI_MPI_ISP_GetStatisticsConfig
		au8HistThresh[0]< au8HistThresh[1]< au8HistThresh[2]<
		au8HistThresh[3];
		u16BlackLevel<= u16WhiteLevel;
		u16CrMin<= u16CrMax;
		u16CbMin<= u16CbMax;
		u16CrMin<= u16CrHigh<= u16CrMax;
		u16CrMin<= u16CrLow<= u16CrMax;
		u16CbMin<= u16CbHigh<= u16CbMax;
		u16CbMin<= u16CbLow<= u16CbMax;

For details about associated parameters, see the *HiISP Development Reference* (Hi3516A) or *3A HiISP Development Reference* (chips excluding Hi3516A).

[Solution]

Set associated parameters by following Table 4-1 and the *HiISP Development Reference* (Hi3516A) or *3A HiISP Development Reference* (chips excluding Hi3516A).

4.3 What Do I Do If the Out-of-Memory Issue Occurs on the Board?

[Symptom]

When the HiPQ Color Checker, HiPQ RAW Analyzer, and HiPQ White Balance Analyzer are running and data is read and written by clicking the **All Read** and **All Write** buttons, the out-of-memory issue occurs on the board.

[Cause Analysis]

The DDR memory is insufficient.

When a RAW snapshot is taken, the memory for storing at least one RAW snapshot is used.

When a YUV snapshot is taken, the memory for storing at least one YUV snapshot is used.

When a 3A data snapshot is taken, the memory for storing at least one YUV snapshot and the statistics data structure is used.

[Solution]

Allocate a larger space for the OS. For details about how to allocate the DDR space for the Hi3516A/Hi3518 series chips, see the *Description of the Installation and Upgrade of the Hi3516A/Hi3518 SDK.txt* respectively.