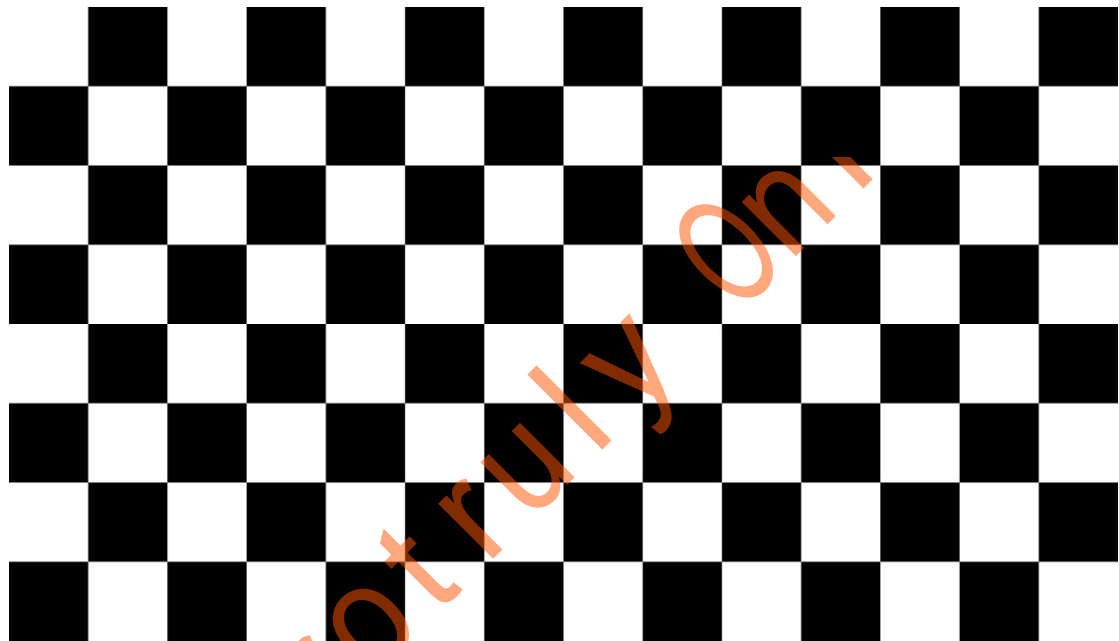

A7L&A12 warp 流程

注：A7L A12 warp 流程大部分的操作是相同的，若无特别说明，则该动作通用。

着重强调：不同型号的镜头，校正脚本和 bin 文件是不能通用的。同一个型号的镜头，不同的 sensor，需要比较 sensor 的尺寸，适用大尺寸 sensor script 的 real expect 值，可以用在小尺寸 sensor 上，反之则不可以；

需要准备的材料：

1) 畸变测试卡（推荐长宽 3.6m * 2.4m，单个方格 8cm*8cm）



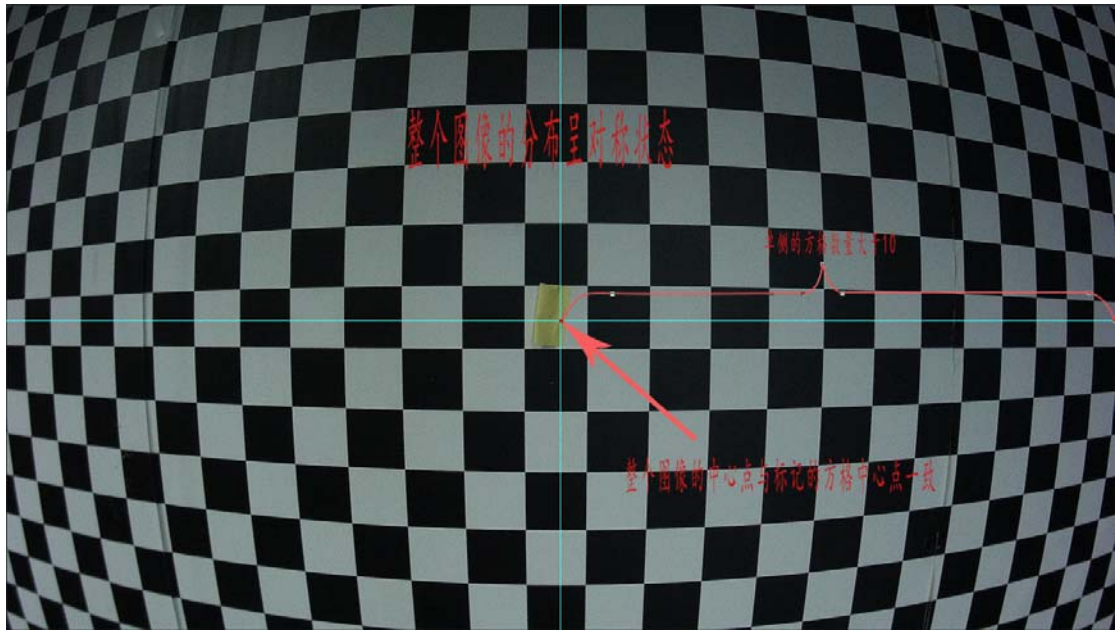
2) PC，接好串口的 DV，三脚架，SD 卡，photoshop。

Step 1 . 拍摄用于计算镜头光学畸变的照片：

拍摄要点：

- 1) 拍摄当中 DV 的焦平面要平行于畸变测试卡，镜头的光轴与中心方格（下图中贴了黄色标记的方格）连线垂直于测试卡。
- 2) 镜头成像场的中心应当与 sensor 的几何中心重合，否则畸变校正的精度会受影响。
- 3) DV 应当固定在三脚架上，有条件应当接入 HDMI 即时观察当前 DV 的机位是否正确。

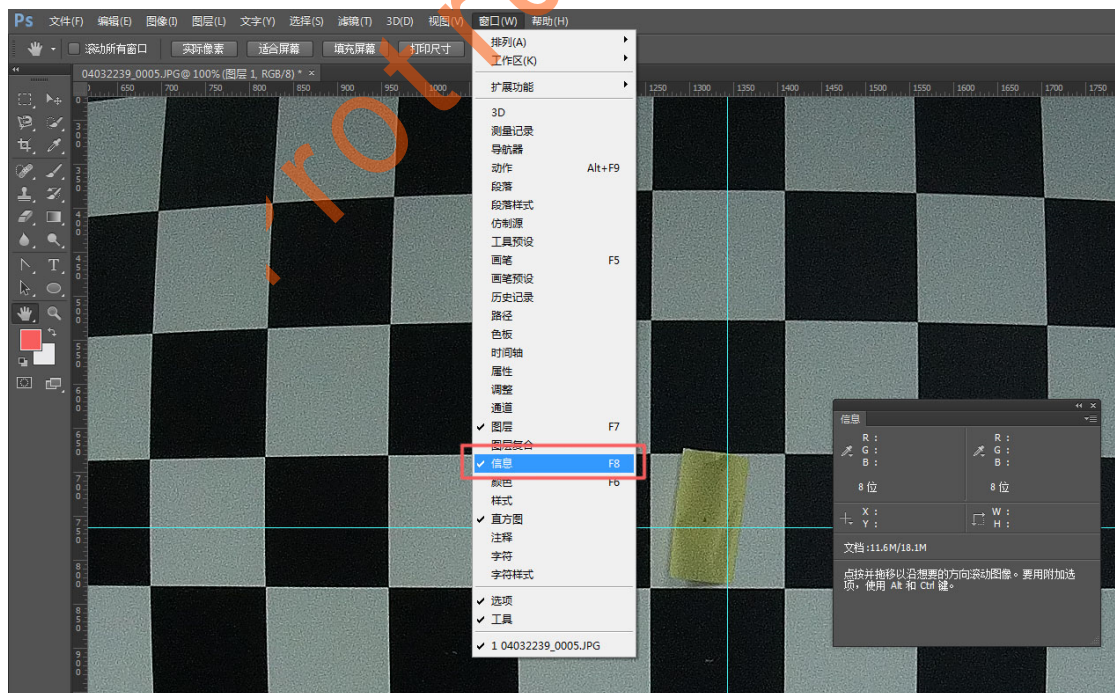
如下是一张较好的示例：



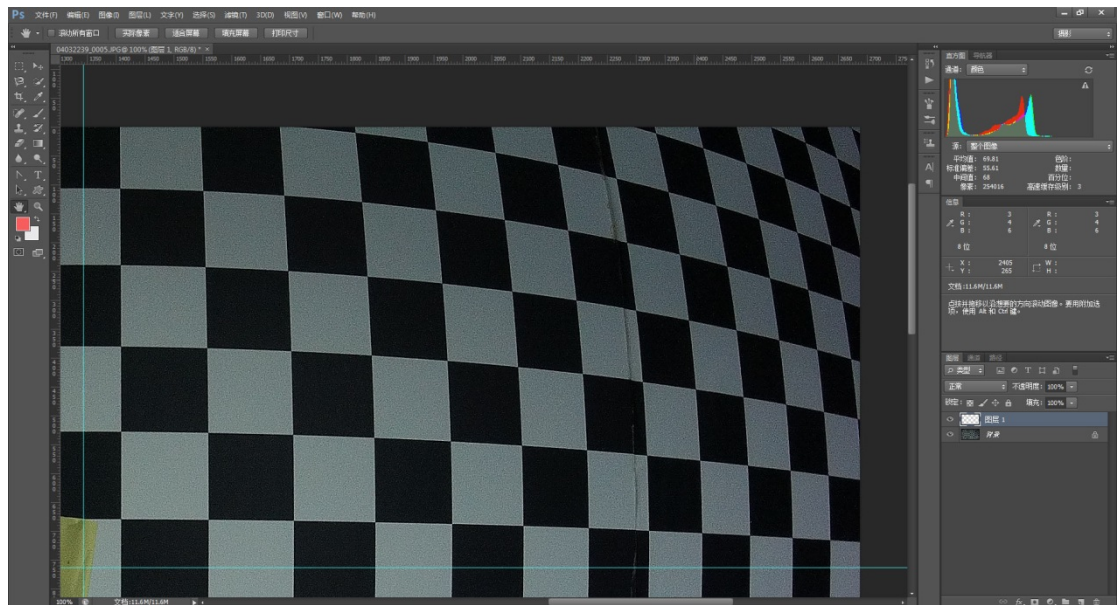
（注：若镜头供应商能提供光学畸变数据，则此步及下一步可省略。从经验来看，镜头供应商可以提供镜头畸变的理论值，但是实际应用的可靠性差一些）

Step 2 计算光学畸变数据

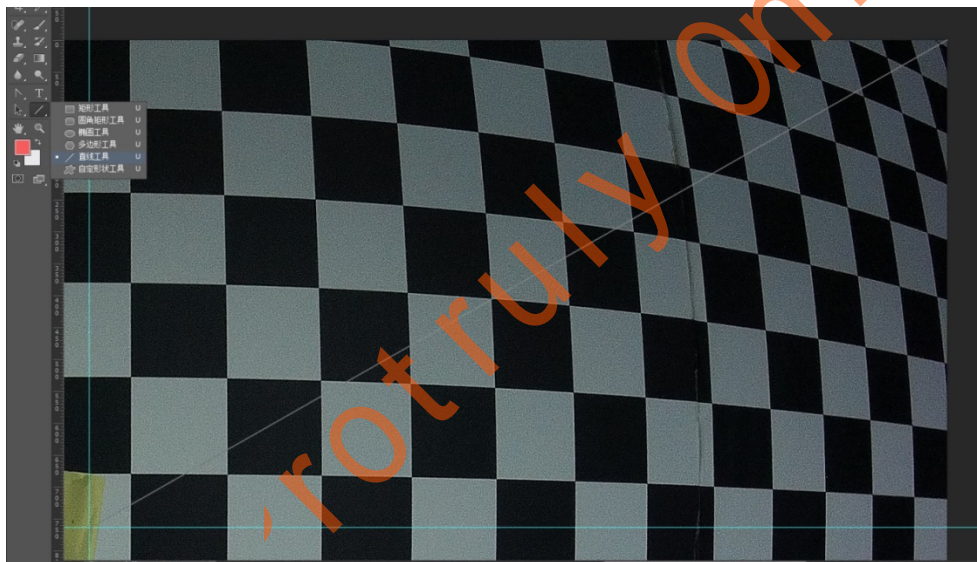
1) 把拍摄的照片导入 Photoshop，放大，中心方格居中，调出“信息”面板：



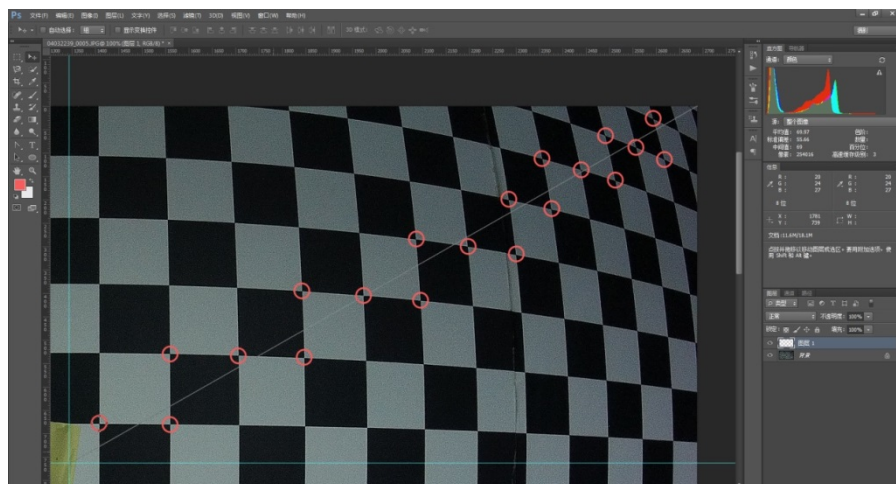
2) 至如下状态：



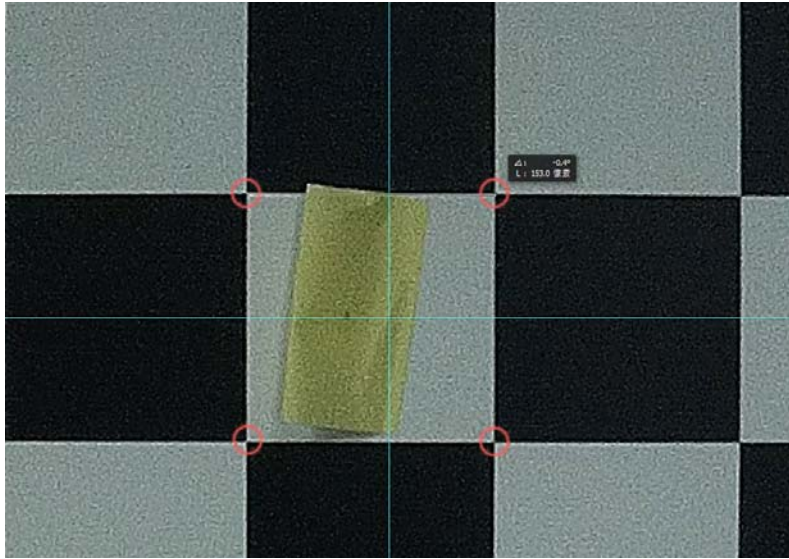
3) 使用直线工具连接照片的中点与右上角:



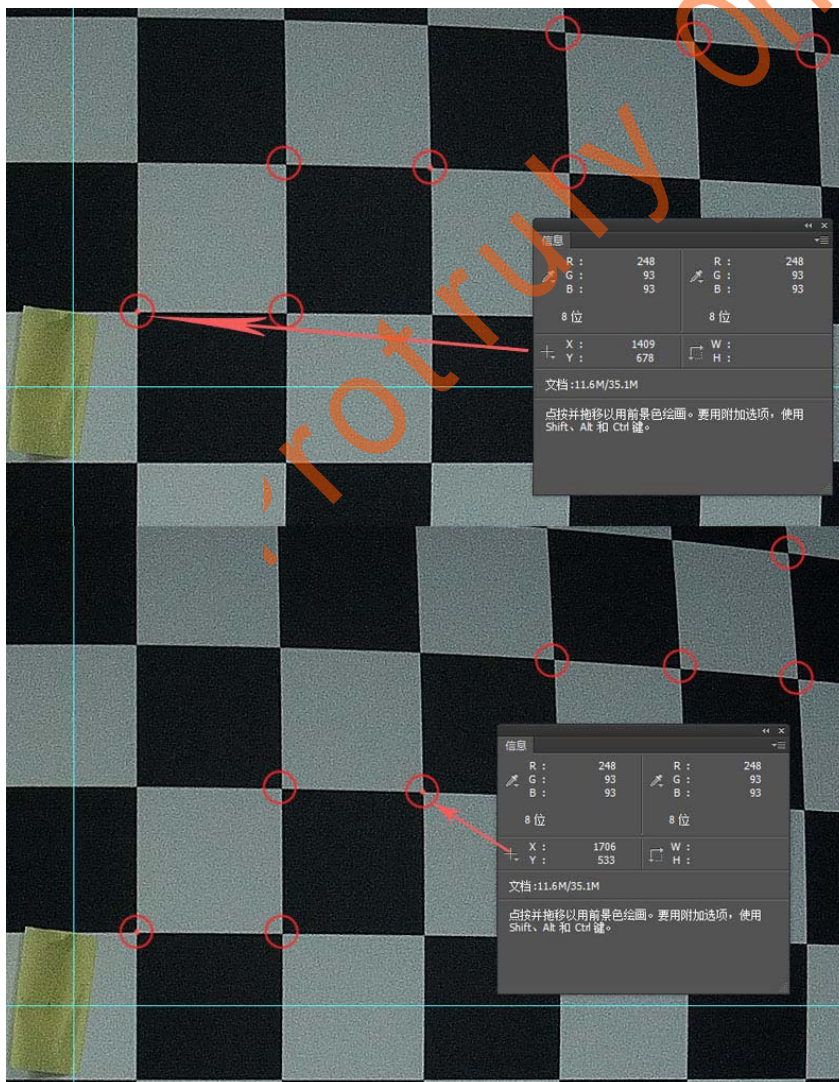
4) 标注离 step2 所画直线最近的点:



- 5) 测量红圈标注的点到整个图像的中心点的实际距离，并根据黑白格数量、方格单位长度来推算理论距离。



先测算中心方格的边长，方法有多种，我采用直线工具直接拉，拉四次得到平均值 153pixel。镜头的畸变总是中心部分是最小的，这里近似认为中心方格无畸变。



测得离中心最近的第 1 个标注点坐标是 (1409, 678)，第 4 个点坐标是 (1706, 533)，整幅图片的长宽是 2688*1512，因此中心点坐标是 (1344, 756)，那么：

第 1 个点距离中心点的实际距离 = $\sqrt{(1409 - 1344)^2 + (678 - 756)^2} = 101.53 \text{ pixel}$

因为已经认为第一个点是近似无畸变的，所以理论距离=实际距离=101.5pixel=0.20306mm。

(注：这张图片是 OV4689 拍的，sensor pixel size=2 μm)

再来看第 4 个点，在 photoshop 中实测的距离

= $\sqrt{(1706 - 1344)^2 + (533 - 756)^2} = 425.17 \text{ pixel} = 0.85034 \text{ mm}$ ；

第四个点到第一个点，X 方向，相差了 2 个格子，Y 方向相差 1 个格子，推算这个点到中心点的理论距离

$$\begin{aligned} &= \sqrt{(2 * \text{中心方格长} + |x_1 - x_{\text{中心点}}|)^2 + (1 * \text{中心方格长} + |y_1 - y_{\text{中心点}}|)^2} \\ &= \sqrt{(2 * 153 + |1409 - 1344|)^2 + (1 * 153 + |678 - 756|)^2} = 437.04 \text{ pixel} \\ &= 0.87408 \text{ mm} \end{aligned}$$

依此类推，算出所有的标记点的实际距离：

```
real  0  0.20306 0.45796 0.62524 0.85034 1.09642 1.23016 1.44236 1.65141 1.75773
1.93447 2.10494 2.18425 2.32579 2.39907 2.51767 2.62524 2.68294 2.77051 2.85324
2.89510 2.96367
```

理论距离：

```
expect 0  0.20306 0.46240 0.63379 0.87408 1.14451 1.29810 1.55566 1.82821 1.97605
2.23927 2.51220 2.65723 2.92318 3.07928 3.33966 3.60721 3.75981 4.02270 4.29131
4.44149 4.70609
```

step 3 填写 warp 脚本

脚本是一个文本文件 (*.txt)，命名自定。此处 A7L A12 的内容稍有差异：

A7L:

```
sensor 2688 1512 5.376 1344 756
real  0  0.20153 0.45796 0.62524 0.85035 1.09642 1.23016 1.44236 1.65141 1.75773
1.93447 2.10494 2.18425 2.32579 2.39907 2.51767 2.62524 2.68294 2.77051 2.85324
2.89510 2.96367
expect 0  0.20153 0.46240 0.63379 0.87302 1.14451 1.29810 1.55566 1.82821 1.97605
2.23927 2.51220 2.65723 2.92318 3.07928 3.33966 3.60721 3.75981 4.02270 4.29131
4.44149 4.70609
Compensate_ratio 100
Compensate_ratio_swa_x 100
Compensate_ratio_swa_y 0
r_ratio 100
end
```

解释如下:

sensor 2688 1512 5.376 1344 756

分别是 sensor 的长 (pixel), 高 (pixel), sensor 长度 (mm), sensor 中心点 X 坐标, Y 坐标, sensor 中心点坐标可以不指定, 若不指定, 则默认=sensor 长/2 sensor 高/2。指定中心点是为了应对出现镜头成像圆的中心点不在 sensor 中心点的情况。Sensor 的长宽请参照驱动, 找到最大的 Vout。

real 0 0.20153 0.45796 0.62524 0.85035 1.09642 1.23016 1.44236 1.65141 1.75773
1.93447 2.10494 2.18425 2.32579 2.39907 2.51767 2.62524 2.68294 2.77051 2.85324
2.89510 2.96367

expect 0 0.20153 0.46240 0.63379 0.87302 1.14451 1.29810 1.55566 1.82821 1.97605
2.23927 2.51220 2.65723 2.92318 3.07928 3.33966 3.60721 3.75981 4.02270 4.29131
4.44149 4.70609

就是我们测得的标记点到中心点的实际距离与期望(理论)距离, 按照由近到远的顺序排列。

Compensate_ratio 100

整体畸变的补偿强度

Compensate_ratio_swa_x 100

水平方向的补偿强度

Compensate_ratio_swa_y 0

竖直方向的补偿强度, 竖直方向的校正能力有一个极限值, 对于 TV 畸变 20% 以内的镜头, 可以做到 100%。同时竖直方向的校正会使水印也发生竖直拉伸。请格外注意。

r_ratio 100

比例尺, 上面计算的畸变值的单位都是 mm, 比例尺为 1:1。

A12:

TileWidth 64

TileHeight 64

sensor 1928 1080 5.784 932 540

real 0 0.40120 0.78924 1.03034 1.35024 1.58725 1.85195 2.10648 2.26145 2.46243
2.59646 2.75619 2.90257 2.99052 3.10663 3.18395 3.27849 3.36401 3.41963 3.49293
3.54026

expect 0 0.40120 0.80080 1.06316 1.43138 1.72514 2.08156 2.47546 2.73741 3.11876
3.39564 3.76854 4.16072 4.42194 4.80641 5.07757 5.45600 5.84743 6.10826 6.49411
6.76241

Compensate_ratio 100

Compensate_ratio_swa_x 100

Compensate_ratio_swa_y 0

r_ratio 100

end

TileWidth 64

TileHeight 64

允许用户指定划分 sensor tile 的单位长度，只能取 16, 32, 64, 128, 256, 512, A7L 已经默认为 64, A12 如果不指定，默认为 128。取值越小，校正精度越高，同时 ISP 负担越重。

其他的和 A7L 是一致的。

Step 4 运行脚本

A7L:

- 1) 把脚本文件存储在 TF 卡上，插入 DV，开机，确保停止一切录像拍照；
- 2) 连接串口，运行 `t sensor -detail`

```

a:\> t sensor -detail

=====a:\> == Sensor Device Info =====
Full_view_width:      2688
Full_view_height:     1520
sensor_cell_width:    2.00 um
sensor_cell_height:   2.00 um

===== General Info =====
Sensor mode:          20
Sbrg config:          0
Readout mode:         0x000000
data bits:            10 bits
Bayer pattern:        BG
Main frame rate:      29.970030 fps
Sampling Factor(hori): 1:1
Sampling Factor(vert): 1:1

===== Sensor Output Status =====
Line_length_pck:      3607 (HB = 919)
Frame_length_lines:   3108 (VB = 1588)
Source start_x:        16
Source start_y:         8
Source end_x:          2703
Source end_y:          1527
Source width:          2688
Source height:         1520

===== Vin Status =====
Capture start_x:       0
Capture start_y:       4
Capture width:         2688
Capture height:        1512

===== Vin Window in Pixel Array ===
start point:          (16, 12)
end point:             (2703, 1523)
center point:          (1360, 768)

===== Actual Frame Timing =====
Pixel/Bit clock:      4200000000 Hz
Pixel rate (I/O):     3360000000 pixel/sec
Frame rate:           29.971752 frame/sec
Row time:             10.735000 us

```

确认一下当前的 Vin(OV4689 最大 Vin 是 2688*1512) 是否和脚本的 sensor 设置是一致的。若不是，请切换一个分辨率模式。（这样做的目的是确保脚本里的 sensor 2688 1512 5.376 能覆盖到驱动里所有的 VIN 窗口，如果没有完全覆盖到当前 VIN，则运行脚本的时候会提示一个错误，如果已经确认脚本没有错误，此步可以跳过）

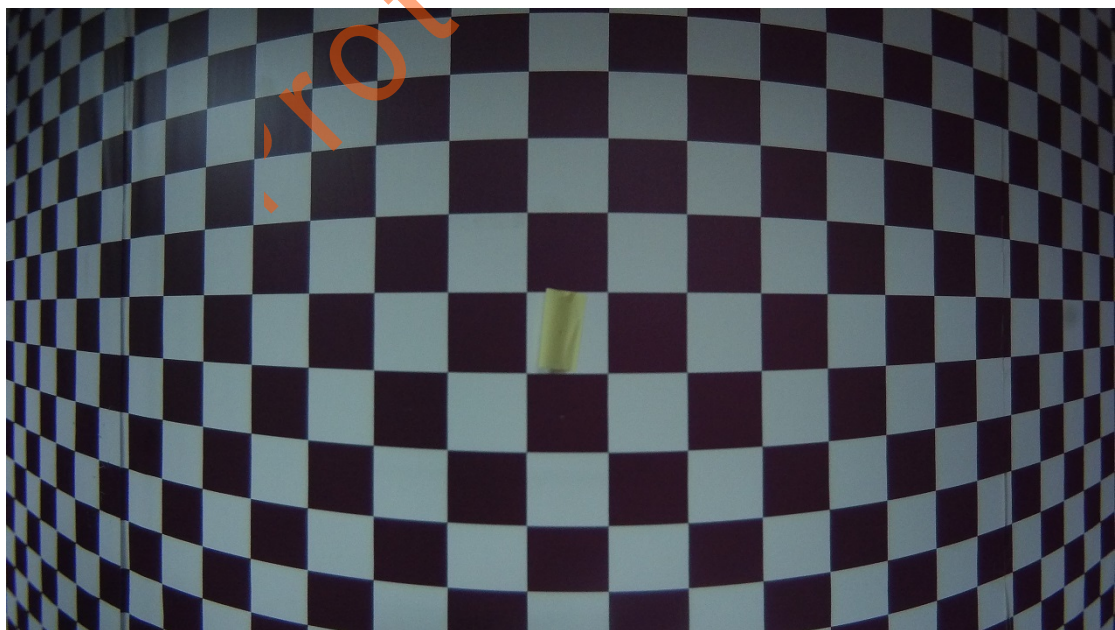
3 运行 `t cal -warp_spec 20 d:*.txt` (d:*.txt 是脚本文件的路径及文件名)


```

a:\> cd d:\
d:\> ls
d---- Sep 28 2015 14:06:44      0 [DCIM]
f-a-- Jan  3 2008 01:52:50      416 bpc.txt.log
f-a-- Nul   1 2015 16:15:22      475 script.txt
f-a-- Nul   1 2015 16:16:22     65492 warp_by_spec.warp
f-a-- Nul  23 2015 16:37:42     1467 bpc.txt
d:\> t cal -warp_spec 20 d:\script.txt
[00663398] Compensate_ratio_swa_x 100
[00663399] Compensate_ratio_swa_y 0
[00664085] -----> still cap mode -1 1
[0066d:\> 4087] -----> still cap mode -1 1
[00664095] IMG_GEO_STILL mode = 3/ 3
[00664095] width = 2688 height = 1512
[00664095] [set_zf]-----> still cap mode -1 1
[00664095] -----> still cap mode -1 1
[00664095] still main 2688, 1512
[00664096] The 0th vert output height = 0
[00664098] IMG_GEO_STILL mode = 3/ 3
[00664098] width = 2688 height = 1512
[00664098] [set_zf]-----> still cap mode -1 1
[00664098] still main 1920, 1080
[00664098] -----> still cap mode -1 1
[00664098] Warp_by_spec succeed.
[00664098] Warp_by_spec End.

```

此时表示生成校正 bin 文件成功，同时校正效果已经生效，不要关机，直接去验证效果，达到如下效果为宜：



A12:

1) 检查 Vin 状态

t sensor detail 0 1

```
a:\> t sensor detail 0 1
[00057045][CA9] SensorUT_TestCmd cmd: detail
[00057045][CA9]
===== Sensor Device Info =====
[00057045][CA9] Full_view_width:      2688
[00057045][CA9] Full_view_height:     1520
[00057045][CA9] sensor_cell_width:    2.00 um
[00057045][CA9] sensor_cell_height:   2.00 um
[00057045][CA9]
===== General Info =====
[00057045][CA9] Sensor mode:          3
[00057045][CA9] Main Frame rate:      29.970030 fps
[00057045][CA9] Sampling Factor(hori): 1:1
[00057045][CA9] Sampling Factor(vert): 1:1
[00057045][CA9]
===== Sensor Output Status =====
[00057045][CA9] Line_length_pck:      3360 (HB=672)
[00057045][CA9] Frame_length_lines:   1668 (VB=156)
[00057045][CA9] data bits:            10 bits
[00057045][CA9] DSP Phase Shift:      0
[00057045][CA9] Source start_x:       0
[00057045][CA9] Source start_y:       4
[00057045][CA9] Source end_x:         2687
[00057045][CA9] Source end_y:         1515
[00057045][CA9] Source width:         2688
[00057045][CA9] Source height:        1512
[00057045][CA9]
===== Vin Window in Pixel Array ==
[00057045][CA9] start point:          (16, 8)
[00057045][CA9] end point:            (2703, 1527)
[00057045][CA9] center point:         (1360, 768)
[00057045][CA9]
===== Actual Frame Timing =====
[00057045][CA9] Pixel/Bit clock:      23995205 Hz
[00057045][CA9] Pixel rate (I/O):     168000000.000000 pixel/sec
[00057045][CA9] Frame rate:           29.976019 frame/sec
[00057045][CA9] Row time:             19.990099 us
[00057045][CA9] Vin HB time:          4.000000 us
[00057045][CA9] Vin VB time:          2960.000000 us
[00057045][CA9]
```

Vin 是 2688*1512（我这里的例子是 OV4689，最大 Vin 就是 2688*1512），没有错误继续；

2) 脚本复制到 TF 卡，插入 DV，开机后确保没有录像，运行 t cal warp warp_spec 20 e:*.txt，若没有错误则如下所示：

```
e:\> t cal warp warp_spec 20 script.txt
[00031561][CA9] AppLib_CalibTest cmd: warp
[00031561][CA9] START_X = 16, START_Y = 8
[00031561][CA9] [AppLibCalibWarp]RawBufferAddress (0xBCF054A0) (512)?
[00031561][CA9] [AppLibCalibWarp]RawBufferAddress (0xBCF2CA20) (512)?
[00031561][CA9] Argc = 5
[00032292][CA9] cell size = 0.002000 HorGridNum = 43 VerGridNum = 25
[00032295][CA9] Compensate_ratio_swa_x 100
[00032296][CA9] Compensate_ratio_swa_y 0
[00032303][CA9] ToolInfo.Expect[IdxNum-1] 2767.365000
[00032303][CA9] [AppLibCalibWarp]RawBufferAddress (0xBCF2CC80) (4300)?
[00032303][CA9] AddTableGridNumW 36 AddTableGridNumH 20
[00032303][CA9] HorGridNum 115, VerGridNum 65
[00032303][CA9] [AppLibCalibWarp]RawBufferAddress (0xBCF2DD80) (59800)?
[00032303][CA9] [AppLibCalibWarp]RawBufferAddress (0xBCF3C760) (59800)?
[00032780][CA9] HSubSampleFactorDen 1
[00032780][CA9] HSubSampleFactorNum 1
[00032780][CA9] HSubSampleFactorDen 1
[00032780][CA9] VSubSampleFactorNum 1
[00032781][CA9] AmbaDSP_CmdSetWarpCtrlTag
e:\> [00032782][CA9] Warp_by_spec succeed.
```

此时校正效果立即生效，并在 TF 卡上生成一个 bin 文件，不要关机，此时可以直接验证效果。

Step5 编译 bin 文件至固件

A7L

- 1) 当运行脚本成功并确认达到预期效果后，把 TF 卡上的的 warp_cal.bin 文件复制出来，更名为 calib6.bin;
- 2) 打开 SDK 里的 config.h，找到如下两个定义：

```
#define CONFIG_BSP_A7LBUB_2GB 1
#define ROMFS_CONFIG_FILE "../gui/sport_dv/romfs/a7l_ft.romfs"
转到 system/bsp/目录下，根据#define CONFIG_BSP_A7LBUB_2GB 1 找到对应的目录，
打开 bsp.h
```

```
#if (CONFIG_BSP_A7LEVK)
    #define CAL_WARP_LOAD_FORMAT (CALIB_SOURCE_ROMFS)
    #define CAL_CA_LOAD_FORMAT (CALIB_SOURCE_ROMFS)
#else
    #define CAL_WARP_LOAD_FORMAT (CALIB_SOURCE_ROMFS)
    #define CAL_CA_LOAD_FORMAT (CALIB_SOURCE_NAND)
#endif
```

把 CAL_WARP_LOAD_FORMAT 定义成 CALIB_SOURCE_ROMFS enable warp 校正；

打开../gui/sport_dv/romfs/a7l_ft.romfs（这是一个文本文件），添加如下：

```
<romfs>
.....
<!-- calib table -->
<file>../src/app3/applib/calib/calib6.bin<alias>calib6.bin</alias></file>
>
.....
</romfs>
```

- 3) 把 calib6.bin 复制到 system/src/app3/applib/calib/，make clean 后编译。

A12:

- 1) 把 TF 卡上的 calib6.bin 复制出来。
- 2) 打开 applibcalibmgr.h, 打开校正功能。

/*Calibration load data format*/

```
#define CAL_STATUS_LOAD_FORMAT      (CALIB_SOURCE_NAND)
#define CAL_AF_LOAD_FORMAT          (CALIB_SOURCE_NAND)
#define CAL_GYRO_LOAD_FORMAT        (CALIB_SOURCE_NAND)
#define CAL_MSHUTTER_LOAD_FORMAT    (CALIB_SOURCE_NAND)
#define CAL_IRIS_LOAD_FORMAT        (CALIB_SOURCE_NAND)
#define CAL_VIGNETTE_LOAD_FORMAT    (CALIB_SOURCE_NAND)
#define CAL_FPN_LOAD_FORMAT         (CALIB_SOURCE_NAND)
#define CAL_WB_LOAD_FORMAT          (CALIB_SOURCE_NAND)
#define CAL_ISO_LOAD_FORMAT         (CALIB_SOURCE_NAND)
#define CAL_BLC_LOAD_FORMAT         (CALIB_SOURCE_NAND)
#define CAL_FLASH_LOAD_FORMAT       (CALIB_SOURCE_NAND)
#define CAL_AUDIO_LOAD_FORMAT       (CALIB_SOURCE_NAND)
#define CAL_WARP_LOAD_FORMAT        (CALIB_SOURCE_ROMFS)
#define CAL_CA_LOAD_FORMAT          (CALIB_SOURCE_NAND)
#define CAL_LENSSHIFT_LOAD_FORMAT   (CALIB_SOURCE_NAND)
```

- 3) 把 calib6.bin 复制到 app\cardv\app\rom\image\calibration\ov4689\warp\
Cardv :请自行选择对应的 app 目录;
Ov4689: 自行选择对应的 sensor;
- 4) make clean 后编译;

A12 一个偶发错误（尤其是新 sensor）解决办法

如果在 A12 运行过程中碰到（以 OV4689 为例）：

```
e:\> t cal warp warp_spec 20 script.txt
[00571512][CA9] AppLib_CalibTest cmd: warp
[00571512][CA9] START_X = 17, START_Y = 8
[00571512][CA9] [AppLibCalibWarp]RawBufferAddress (0xBCF054A0) (512)?
[00571512][CA9] [AppLibCalibWarp]RawBufferAddress (0xBCF2CAC0) (512)?
[00571512][CA9] Argc = 5
[00571517][CA9] cell size = 0.002000 HorGridNum = 43 VerGridNum = 25
[00571520][CA9] Compensate_ratio_swa_x 100
[00571520][CA9] Compensate_ratio_swa_y 0
[00571525][CA9] ToolInfo.Expect[IdxNum-1] 2767.365000
[00571525][CA9] [AppLibCalibWarp]RawBufferAddress (0xBCF2CD00) (4300)?
[00571525][CA9] AddTableGridNumW 36 AddTableGridNumH 20
[00571525][CA9] HorGridNum 115, VerGridNum 65
[00571525][CA9] [AppLibCalibWarp]RawBufferAddress (0xBCF2DE00) (59800)?
[00571525][CA9] [AppLibCalibWarp]RawBufferAddress (0xBCF3C7E0) (59800)?
[00572441][CA9] HSubSampleFactorDen 1
[00572441][CA9] HSubSampleFactorNum 1
[00572441][CA9] HSubSampleFactorDen 1
[00572441][CA9] USubSampleFactorNum 1
[00572441][CA9] Error: Calib vin geo x 17, y 8, w 2688, h 1512, HorSmp 1/1, VerSmp 1/1 cannot cover
[00572441][CA9] current vin geo x 16, y 8, w 2688, h 1512, HorSmp 1/1, VerSmp 1/1
[00572441][CA9] AmbaDSP_CmdSetWarpCtrlTag
[00572441][CA9] Warp_by_spec succeed.
```

说明当前指定的校正范围没有完全覆盖当前的 Vin 窗口。这个时候需要改一下驱动：

首先从 AppLibCalibWarp_GenWarpFromSpec(int Argc, char *Argv[])看：

```
SensorMode.Bits.Mode = pAmbaCalibInfoObj[0]->AmbaVignetteMode;
AmbaSensor_GetModeInfo(AppEncChannel, SensorMode, &VinInfo);
    StartX = VinInfo.InputInfo.PhotodiodeArray.StartX;
    StartY = VinInfo.InputInfo.PhotodiodeArray.StartY;
```

以上代码从驱动得到一个 Vin 窗口信息，并把校正的起点设置成驱动里 vin 的起点：

```
AMBA_CALIB_INFO_OBJ_s AmbaCalibInfo_OV4689Obj = {
    .AmbaVignetteMode          = AMBA_SENSOR_OV4689_V1_10_2688_1512_60P,
    .AmbaNumBadPixelMode       = GetArraySize(OV4689_AmbaBadPixelModeTable),
    .AmbaBadPixelModeTable     = OV4689_AmbaBadPixelModeTable,
    .AmbaBadPixelModeAssignment = OV4689_AmbaBadPixelModeAssignment,
    .AmbaCalibWidth            = 2688,
    .AmbaCalibHeight           = 1512,
};
```

实际上调用的是 **AMBA_SENSOR_OV4689_V1_10_2688_1512_60P**，在 AmbaSensor_OV4689Table.c 里定义了：

```
/* AMBA_SENSOR_OV4689_V1_10_2688_1512_60P */
{ OV4689_S1_10_2688_1520_60P, {24023976, 1200 * 4, 1670, { .Interlace =
0, .TimeScale = 60000, .NumUnitsInTick = 1001}}},
```

最终的 Input info 如下:

```
const AMBA_SENSOR_INPUT_INFO_s OV4689InputInfo[OV4689_NUM_READOUT_MODE_REG] = {  
/* OV4689_S1_10_2688_1520_60P      *//{{ 16, 8, 2688, 1520},  
{ AMBA_SENSOR_SUBSAMPLE_TYPE_NORMAL, 1, 1}, { AMBA_SENSOR_SUBSAMPLE_TYPE_NORMAL, 1,  
1}},
```

如上修改 startx=16, starty=8 后就可以了。