CMV50000 FPN and PRNU Correction

How to correct for FPN and PRNU
# Content Guide

<table>
<thead>
<tr>
<th></th>
<th>Introduction</th>
<th>3</th>
<th>Image Corrections</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>2.1 Stitch FPN</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.2 Column and Row FPN</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.3 Pixel FPN</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>2.4 Summary</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Revision Information</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Legal Information</td>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1 Introduction

The CMV50000 has, like any CMOS image sensor, fixed pattern noise (FPN). This FPN can be pixel-, column- or row-based.

The pixel-based FPN manifests itself as pixel-to-pixel variations when illuminated with the same amount of light. Below is an example of a 100×100 pixel region of interest (ROI). This is a dark image (no illumination) and digitally gained x16 to show the FPN more clearly.

Figure 1:
Pixel FPN

As you can see, in spite of all pixels having the same illumination (here: none), they do not all have the same grey value (in DN). Most of them are similar and some are outliers.

The column- and row-based FPN manifests itself as column-to-column and row-to-row variations when illuminated with the same amount of light. All pixels in a row/column will have the same variation (on top of the pixel FPN). Below you can see the same image as above, but at full resolution. You can see faint horizontal and vertical patterns appearing.

The sensor has on-chip correction for row FPN, this is why the row FPN is much less visible than the column FPN.

Because of the large physical size of the CMV50000, the sensor is ‘stitched’. Stitching is used to make image sensors that are larger than the field of view of the lithographic equipment used during the fabrication of the wafers. As now multiple lithographic sequences (exposures) are needed to make one image sensor, at the boundaries of each sequence, you might see a difference. For CMV50000 this boundary is at the middle column of the sensor, meaning that columns 0-3959 and columns 3960-
7919 are done in 2 different litho steps. When looking at a zoom of the center and again digitally gain x16, you can see this difference between the left and right side of the pixel array.

Figure 2:
Column and Row FPN

Figure 3:
Stitch FPN

Every FPN has an offset and a gain component. The offset is a fixed difference (calculated in a dark image) and the gain is the FPN changing with illumination or pixel value (also called photo-response non-uniformity; PRNU).
2 Image Corrections

Based on the resources you want to spend on corrections and the need for optimal images, you may decide to correct only for certain FPNs. In most cases, this would be the order of priority of needed corrections:

- Stitch FPN
- Column FPN
- Row FPN
- Pixel FPN

In general, the human eye is x10 more sensitive to pattern noise (stitch, column, and row) than random noise (pixel). Also for pixel FPN correction (also called flat-field correction; FFC), you need a lot more resources and power than the other corrections. Depending on the needs, you can also choose to do offset, gain or both corrections. Pixel flat-field correction will yield the best image quality in the end.

If you do column FPN correction, you do not need to do stitch FPN correction separately. If you do pixel FPN correction, you do not need to do any other FPN correction any more.

2.1 Stitch FPN

2.1.1 Stitch Offset

The stitch FPN offset correction consists of adding a fixed value to all pixels in the first 3960 columns as the left side is typically a bit lower. You could also subtract a fixed value from the right side, but when doing FPN corrections, subtractions should be avoided to prevent clipping pixels at 0DN.

In the example below you see an x16 gained center crop of the original image. On the left side, you have the original image where you can see the difference between the left and right part of the pixel array. In the right image, I added 3DN to the left side of the pixel array and as you can see, the stitch is not visible anymore.
The procedure to find the correct offset value looks something like this:

1. Take 1 or multiple (and average them) images without any illumination.
2. Calculate the average of the 100 columns to the left and right side of the stitch boundary. Call them avg_So_L and avg_So_R.
3. The stitch offset correction value corr_So = avg_So_R – avg_So_L
4. To correct pixels: add corr_So to all pixels in the first 3960 columns in every image you take.

2.1.2 Stitch Gain

The stitch gain (PRNU) correction uses a similar method. Here we multiply the right side with a certain gain value (1.xxx or 0.xxx). In the example below, we multiplied the right side by 1.025 (so adding 2.5%) and the stitch is gone.
The procedure to find the correct offset value looks something like this:

1. Take 1 or multiple (and average them) images without any illumination. Add corr_So to the left pixels and calculate the overall average of the full image. This is your dark level (avg_dark) and should be close to the EOB_TARGET register value.

2. Take 1 or multiple (and average them) images with a very uniform illumination across the pixel array so that the average grey value is somewhere between 50% and 70% of the maximum (around 2000 to 3000DN). This is your grey level avg_grey.

3. Calculate the average of the 100 columns to the left and right side of the stitch boundary. Call them avg_Sg_L1 and avg_Sg_R1.

4. Subtract the offset values, so only the illumination part is left:
   \[\text{avg\_Sg\_L2} = \text{avg\_Sg\_L1} - \text{avg\_So\_L}\]
   \[\text{avg\_Sg\_R2} = \text{avg\_Sg\_R1} - \text{avg\_So\_R}\]

5. The stitch gain correction factor \(\text{corr\_Sg} = \frac{\text{avg\_Sg\_R2}}{\text{avg\_Sg\_L2}}\)

6. To correct pixels: for every pixel value in the first 3960 column, first subtract avg_dark and then multiply the result with \(\text{corr\_Sg}\). Then add avg_dark again to the result. Be careful for negative values.

These correction values should be calculated once (per sensor/camera) and saved (preferably during testing/production, in a stable and known environment), so they can be used during the real-time corrections later.

An example:

After taking the dark and illuminated images, we have measured following known values:
Figure 8:
Example Data

<table>
<thead>
<tr>
<th>Measured Data</th>
<th>Calculated Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg_dark = 128DN</td>
<td></td>
</tr>
<tr>
<td>avg_So_L = 126DN</td>
<td>avg_Sg_L2 = 2349DN</td>
</tr>
<tr>
<td>avg_So_L = 129DN</td>
<td>avg_Sg_R2 = 2396DN</td>
</tr>
<tr>
<td>avg_grey = 2500DN</td>
<td>corr_Sg = 1.020</td>
</tr>
<tr>
<td>avg_Sg_L1 = 2475DN</td>
<td>corr_So = 3DN</td>
</tr>
<tr>
<td>avg_Sg_R2 = 2525DN</td>
<td></td>
</tr>
</tbody>
</table>

If we now read out a pixel in the right side of the pixel array, we do not need to correct and leave the pixel value as is. Only pixels on the left side should be corrected for stitch FPN. For every pixel value (val_pix_old) we can calculate its corrected value (pix_val_new) like this:

\[ \text{pix_val_new} = \left( \text{val_pix_old} + \text{corr}_\text{So} - \text{avg_dark} \right) \times \text{corr}_\text{Sg} + \text{avg_dark} \]

For example if a pixel reads out 3800DN, its corrected value would be 3877DN. If we just multiply the old pixel value with corr_Sg, the pix_val_new is 3876DN, which is very close to what we want. So you can decide to simplify the correction algorithm to this.

### 2.2 Column and Row FPN

The column and row FPN correction methods are identical.

#### 2.2.1 Column/Row Offset

Due to the design and the on-chip row correction, row and column offset is quite low.

Below is an ROI of a dark image, where I stretched the histogram (equalize) to make the FPN more visible. Row offset is not visible, while column offset is very faint, so the pixel FPN is the dominant factor here. It is therefore viable to choose not to do column/row offset correction.
2.2.2 Column/Row Gain

The column PRNU is more pronounced than the offset. Below is a digitally enhanced grey ROI before and after column FPN correction. Row PRNU is very low on the sensor.

For correcting the column PRNU, a method like below can be followed, where we find the gain factor per column and correct each column with that factor.
1. Take 1 or multiple (and average them) images without any illumination. Apply the stitch offset correction.

2. Take the average of each column to give you a 1D array of 7920 values: avg_Co_arr. These values should be close EOB_TARGET.

3. Take 1 or multiple (and average them) images with a very uniform illumination across the pixel array so that the average grey value is somewhere between 50% and 70% of the maximum (around 2000 to 3000DN). You do not have to apply the stitch gain correction here, as the column gain correction will correct for it.

4. Take the average of each column and subtract the corresponding offset value from avg_Co_arr. This gives a 1D array of 7920 values: avg_Cg_arr.

5. Take the average of this array: avg_Cg

6. For every value in avg_Cg_arr, divide avg_Cg by that value. This gives you a 1D array with 7920 gain correction factors, one for every column: corr_Cg_arr

7. To correct the columns in an image: for every pixel in a column, subtract the avg_Co_arr value for that column, multiply the result with the value in corr_Cg_arr for the same column. Add the avg_Co_arr value again to that result to have the corrected pixel value.

These avg_Co_arr and corr_Cg_arr arrays should be calculated once (per sensor/camera) and saved (preferably during testing/production, in a stable and known environment), so they can be used during the real-time corrections later.

An example:

After taking the dark and illuminated images, we have measured following known values (for the first 3 columns):

<table>
<thead>
<tr>
<th>Measured Data</th>
<th>Calculated Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg_Co_arr = [128.5, 126.3, 129.8, …]</td>
<td>avg_Cg = 2399.033</td>
</tr>
<tr>
<td>avg_Cg_arr = [2387.2, 2395.1, 2414.8, …]</td>
<td>corr_Cg_arr = [1.00496, 1.00164, 0.99347, …]</td>
</tr>
</tbody>
</table>

Now, for every pixel value (pix_val_old) we read out we can calculate its corrected value (pix_val_new) like this:

\[ pix\_val\_new = \left(\left(pix\_val\_old - avg\_Co\_arr[n]\right) \times corr\_Cg[n]\right) + avg\_Co\_arr[n] \]

Where n is the column that pixel belongs to.

For example if a pixel in column 1 reads out 3800DN, its corrected value would be 3818DN. If we just multiply the old pixel value with corr_Cg[n], the pix_val_new is 3819DN, which is very close to what we want. Therefore, you can decide to simplify the correction algorithm and not use the offset values.

For row PRNU correction, you can use the same method, but measured and applied on the rows.
2.3 Pixel FPN

Doing individual correction for every pixel demands a lot of resources and computing power, but it will yield the best image quality in the end. It is similar to the column correction, only now the correction arrays will be 7920x6004 big, so you basically have to store 2 full resolution 47.5Mp (16b or even 32b) images on camera.

2.3.1 Pixel Offset

To correct the pixel offset, you can follow the following method:

1. Take multiple images (at least 16 (more is better) to average out temporal noise) without any illumination and average them. This gives a 2D array of 7920x6004 values: avg_Po_arr. Take the overall average value of avg_Po_arr = avg_Po and subtract that from every pixel value in avg_Po_arr (watch out for negative numbers!). The result is 2D array with 7920x6004 offset correction values: corr_Po_arr.

2. To correct pixels for offset, subtract the corresponding pixel correction value in corr_Po_arr from the pixel value you read out.

Save the corr_Po_arr 2D array on the camera.

2.3.2 Pixel Gain

To correct the pixel offset, you can follow the following method:

1. Take multiple (at least 4 (more is better)) images with a very uniform illumination across the pixel array so that the average grey value is somewhere between 50% and 70% of the maximum (around 2000 to 3000DN). Average the images and subtract avg_Po_arr to give you a 2D array only contain pixel PRNU: avg_Pg_arr.

2. Take the average of avg_Pg_arr and for every value in avg_Pg_arr divide that value by this overall average. This gives you a 7920x6004 2D array of pixel gain correction values: corr_Pg_arr.

3. To correct pixel for gain, first do pixel offset correction and then multiply every pixel with its corresponding pixel gain correction value in corr_Pg_arr.

Save the corr_Pg_arr 2D array on the camera.

2.4 Summary

Below you will find an overview of the correction data needed for each correction and the size it takes when saving it as 16b or 32b data.
Figure 10:  
Correction Overview

<table>
<thead>
<tr>
<th>Correction</th>
<th>Correction Data</th>
<th>Size (16b)</th>
<th>Size (32b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stitch offset</td>
<td>Corr_So: 1 value</td>
<td>16b</td>
<td>32b</td>
</tr>
<tr>
<td>Stitch gain</td>
<td>Corr_Sg: 1 value</td>
<td>16b</td>
<td>32b</td>
</tr>
<tr>
<td>Column offset</td>
<td>Corr_Co_arr: 1D array of 7920 values</td>
<td>123.75kb</td>
<td>247.5kb</td>
</tr>
<tr>
<td>Column gain</td>
<td>Corr_Cg_arr: 1D array of 7920 values</td>
<td>123.75kb</td>
<td>247.5kb</td>
</tr>
<tr>
<td>Row offset</td>
<td>Corr_Ro_arr: 1D array of 7920 values</td>
<td>123.75kb</td>
<td>247.5kb</td>
</tr>
<tr>
<td>Row gain</td>
<td>Corr_Rg_arr: 1D array of 7920 values</td>
<td>123.75kb</td>
<td>247.5kb</td>
</tr>
<tr>
<td>Pixel offset</td>
<td>Corr_Po_arr: 2D array of 7920x6004 values</td>
<td>725.6Mb</td>
<td>1.417Gb</td>
</tr>
<tr>
<td>Pixel gain</td>
<td>Corr_Pg_arr: 2D array of 7920x6004 values</td>
<td>725.6Mb</td>
<td>1.417Gb</td>
</tr>
</tbody>
</table>
## 3 Revision Information

<table>
<thead>
<tr>
<th>Changes from previous version to current revision v1-00</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial version 1-00</td>
<td></td>
</tr>
</tbody>
</table>

- Page and figure numbers for the previous version may differ from page and figure numbers in the current revision.
- Correction of typographical errors is not explicitly mentioned.
4 Legal Information

Copyrights & Disclaimer
Copyright ams AG, Tobelbader Strasse 30, 8141 Premstaetten, Austria-Europe. Trademarks Registered. All rights reserved. The material herein may not be reproduced, adapted, merged, translated, stored, or used without the prior written consent of the copyright owner.

Information in this document is believed to be accurate and reliable. However, ams AG does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Applications that are described herein are for illustrative purposes only. ams AG makes no representation or warranty that such applications will be appropriate for the specified use without further testing or modification. ams AG takes no responsibility for the design, operation and testing of the applications and end-products as well as assistance with the applications or end-product designs when using ams AG products. ams AG is not liable for the suitability and fit of ams AG products in applications and end-products planned.

ams AG shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business or indirect, special, incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data or applications described herein. No obligation or liability to recipient or any third party shall arise or flow out of ams AG rendering of technical or other services.

ams AG reserves the right to change information in this document at any time and without notice.

RoHS Compliant & ams Green Statement
RoHS Compliant: The term RoHS compliant means that ams AG products fully comply with current RoHS directives. Our semiconductor products do not contain any chemicals for all 6 substance categories, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, RoHS compliant products are suitable for use in specified lead-free processes.

ams Green (RoHS compliant and no Sb/Br): ams Green defines that in addition to RoHS compliance, our products are free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material).

Important Information: The information provided in this statement represents ams AG knowledge and belief as of the date that it is provided. ams AG bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties, ams AG has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. ams AG and ams AG suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

Headquarters
ams AG
Tobelbader Strasse 30
8141 Premstaetten
Austria, Europe
Tel: +43 (0) 3136 500 0

Please visit our website at www.ams.com
Buy our products or get free samples online at www.ams.com/ICdirect
Technical Support is available at www.ams.com/Technical-Support
Provide feedback about this document at www.ams.com/Document-Feedback
For sales offices, distributors and representatives go to www.ams.com/contact
For further information and requests, e-mail us at ams_sales@ams.com