MigMate Statistics Engine

Micron Technology, Inc.
9/26/2007

1 Introduction

The purpose of this document is to describe the MigMate software's capabilities for calculating image statistics. These capabilities are encapsulated in a common library or engine which can be accessed through a variety of interfaces in the MigMate GUI, as well as the object model that the MigMate exposes to the integrated Visual Basic IDE and to other applications via COM. Here we will describe the programmatic interface to the engine, as well as give definitions of the various statistic types currently supported by the engine.

2 Overview

As of this writing, the engine supports the following statistic types:

Type	Subtype	Statistic	Symbol	Section
Overall		Mean	`StatMean`	4.1.1
		Maximum	`StatMaximum`	4.1.2
		Minimum	`StatMinimum`	4.1.3
		Histogram	`StatHistogram`	4.1.4.1
		Cumulative Histogram	`StatCumulativeHistogram`	4.1.4.2
		Normalized Cumulative Histogram	`StatNormalizedCumulativeHistogram`	4.1.4.3
	Total Noise	Total Noise	`StatTotalNoise`	4.1.5.1
		Total Noise (RMS)	`StatRMSTotalNoise`	4.1.5.2
		Standard Deviation	`StatStandardDeviation`	4.1.5.3
	FPN	Fixed Pattern Noise	`StatFixedPatternNoise`	4.1.6.1
		Fixed Pattern Noise (RMS)	`StatRMSFixedPatternNoise`	4.1.6.2
	Temporal Noise	Temporal Noise	`StatTemporalNoise`	4.1.7.1
		Temporal Noise (RMS)	`StatRMSTemporalNoise`	4.1.7.2
Frame	Frame Flicker	Frame Flicker Noise	`StatFrameFlickerNoise`	4.2.1.1
		Frame Flicker Noise (RMS)	`StatRMSFrameFlickerNoise`	4.2.1.2
Row		Row Means	`StatRowMeans`	4.3.1
	Total Noise	Row Total Noise	`StatRowTotalNoise`	4.3.2.1
		Row Total Noise (RMS)	`StatRMSRowTotalNoise`	4.3.2.2
		Row Standard Deviation	`StatRowStdDev`	4.3.2.3
	FPN	Row FPN	`StatRowFPN`	4.3.3.1
		Row FPN (RMS)	`StatRMSRowFPN`	4.3.3.2
		Horizontal FPN Level (SMIA)	`StatSMIAHorizontalFPNLevel`	4.3.3.3
		Horizontal FPN Max (SMIA)	`StatSMIAHorizontalFPNMax`	4.3.3.4
	Temporal Noise	Row Temporal Noise	`StatRowTemporalNoise`	4.3.4.1
		Row Temporal Noise (RMS)	`StatRMSRowTemporalNoise`	4.3.4.2
		Row Noise Max (SMIA)	`StatSMIARowNoiseMax`	4.3.4.3
Column		Column Means	`StatColumnMeans`	4.4.1
	Total Noise	Column Total Noise	`StatColumnTotalNoise`	4.4.2.1
		Column Total Noise (RMS)	`StatRMSColumnTotalNoise`	4.4.2.2

		Column Standard Deviation	`StatColumnStdDev`	4.4.2.3
	FPN	Column FPN	`StatColumnFPN`	4.4.3.1
		Column FPN (RMS)	`StatRMSColumnFPN`	4.4.3.2
		Vertical FPN Level (SMIA)	`StatSMIAVerticalFPNLevel`	4.4.3.3
		Vertical FPN Max (SMIA)	`StatSMIAVerticalFPNMax`	4.4.3.4
	Temporal Noise	Column Temporal Noise	`StatColumnTemporalNoise`	4.4.4.1
		Column Temporal Noise (RMS)	`StatRMSColumnTemporalNoise`	4.4.4.2
		Column Noise Max (SMIA)	`StatSMIAColumnNoiseMax`	4.4.4.3
Pixel		Pixel Means	`StatPixelMeans`	4.5.1
	Temporal Noise	Pixel Temporal Noise	`StatPixelTemporalNoise`	4.5.2.1
		Pixel Temporal Noise (RMS)	`StatRMSPixelTemporalNoise`	4.5.2.2
Local		Local Standard Deviation (SMIA)	`StatSMIALocalStdDev`	4.6.1

Table 1. Supported statistic types.

In addition, the following symbolic aliases are included to make it easier to choose the right statistic type:

Alias	**Symbol**	**Meaning**
Row Noise Level (SMIA)	`StatSMIARowNoiseLevel`	`StatRowTemporalNoise`
Column Noise Level (SMIA)	`StatSMIAColumnNoiseLevel`	`StatColumnTemporalNoise`

Table 2. Statistic type aliases.

The exact definition of each of these statistics is given in the subsection of this document specified in Table 1. Each statistic type may be calculated for a single frame, a collection of frames held in memory (called a Frame Set), or for a large number of frames as they are acquired from a device. Statistics can be calculated on individual color *planes* (e.g. GreenR, Red, Blue, and GreenB), for devices with non-uniform (e.g. Bayer) CFAs, as well as on individual color *components* (e.g. Red, Green, Blue or Y, Cb, Cr) for images that contain such information (e.g. RGB or YCbCr images).[1][2] The engine can also act across color plane and/or component boundaries, to obtain, for example, the mean value of all pixels in all color planes in an image. In addition, regions of interest within an image or set of images

---

[1] The reader should note the crucial distinction the engine makes between a color plane and a color component. The engine regards a *plane* as a specification of where a pixel lies within a two-dimensional pattern repeated throughout the array, as in the familiar Bayer pattern. In contrast, separating an image into color *components* is a three-dimensional operation, where every pixel, regardless of its location in the array, has a fixed number of components that together specify its color.

By separating these two notions, the engine allows greater flexibility in its usage. For example, consider an RGB image created by interpolating the color planes of a grayscale Bayer image. If the interpolation algorithm is one-to-one, the MigMate will retain CFA information for the resultant image, so that the image can be separated into both components and planes. This allows statistics to be calculated separately for, say, the Blue components of GreenB pixels, the Green components of Red pixels, and so forth. One application of this capability would be to collect statistics that give the designer of an interpolation algorithm the ability to quantify the robustness of the algorithm in terms of its ability to maintain similar noise characteristics for a given color component of a pixel regardless of the pixel's underlying color plane.

[2] Keep in mind that, as with other frame operations in the MigMate, calculating statistics on a multi-component frame automatically and transparently upsamples (normalizes) the components, if necessary. For example, an RGB image will automatically be normalized to at least 8 bits per component, and a subsampled YCbCr image will be upsampled to the 4:4:4 ratio. This transformation is always reversible, however; there is never a loss of precision.

can be defined and statistics can be calculated for only the pixel components lying within a given region.

3 Usage

The design goal for the programmatic interface for the engine is to allow simple statistics to be acquired as easily as possible, while still permitting more complex and time-consuming statistical tasks to be performed. This has led to a division of the interface into the three subinterfaces described in this section.

3.1 Statistics on a Single Frame

To obtain a statistic for a single frame, the MigMate's object model has a single access function:[3]

```
Public Function Frame.Statistic(
        statistic As StatisticType,
        [plane As PlaneType = PlaneAll],
        [component As PixelComponent = ComponentAll],
        [rgn As Region]) As Variant
```

The square brackets here indicate optional parameters. The possible values for `statistic` are given by Tables 1 and 2. The possible values for `plane` and `component` are given by Tables 3 and 4, respectively. If no region is specified, the entire frame is taken as the region. The return type of the function is declared as `Variant` to accommodate the three types of data that statistics return: whole frames (pixel-wise statistics), arrays of values (`StatHistogram`, `StatRowMeans`, and `StatColumnMeans`), and single values (all other statistics).

CFA	Plane	Symbol
Mono	N/A	`PlaneAll`
Bayer	All (treat as Mono)	`PlaneAll`
	GreenR	`PlaneBayerGreenR`
	Red	`PlaneBayerRed`
	Blue	`PlaneBayerBlue`
	GreenB	`PlaneBayerGreenB`
Paired Bayer	All (treat as Mono)	`PlaneAll`
	GreenR1	`PlanePairedBayerGreenR1`
	Red1	`PlanePairedBayerRed1`
	GreenR2	`PlanePairedBayerGreenR2`
	Red2	`PlanePairedBayerRed2`
	Blue1	`PlanePairedBayerBlue1`
	GreenB1	`PlanePairedBayerGreenB1`
	Blue2	`PlanePairedBayerBlue2`
	GreenB2	`PlanePairedBayerGreenB2`

Table 3. Color planes.

[3] Unless otherwise indicated, code samples in this document are given in the syntax of Visual Basic for Applications, the scripting environment included with the MigMate application. Other languages' syntax for accessing members of external COM objects can vary significantly.

Image Type	Component	Symbol
Grayscale	N/A	`ComponentAll`
RGB	All	`ComponentAll`
	Red	`ComponentRed`
	Green	`ComponentGreen`
	Blue	`ComponentBlue`
YCbCr	All	`ComponentAll`
	Y	`ComponentY`
	Cb	`ComponentCb`
	Cr	`ComponentCr`

Table 4. Color components.

Of course, the given plane, component, and/or region must match the CFA, type, and dimensions of the input frame. Note that the pixel components included in the calculation will be within the intersection of the given plane, component, and region, if any, so that it is an error to pass in `PlaneBayerBlue` and a planar region containing only pixels in the Red plane, for example.

3.1.1 Example

The following code sample prints out the mean value of all the GreenR pixels in the active frame in MigMate, bounded by the active region:

```
With Windows.ActiveFrames
        Output.Write .ActiveFrame.Statistic( _
                StatMean, PlaneBayerGreenR, , .Regions.ActiveRegion)
End With
```

3.2 Statistics on a Collection of Frames

The interface for obtaining statistics on a collection of frames is identical to that for a single frame:

```
Public Function Frames.Statistic(
        statistic As StatisticType,
        [plane As PlaneType = PlaneAll],
        [component As PixelComponent = ComponentAll],
        [rgn As Region]) As Variant
```

The difference is that the statistic will be compiled using the pixels in corresponding positions in every frame in the collection. Every frame must have the same CFA, type, and dimensions, and these must match the passed-in parameters.

3.2.1 Examples

The following code sample prints out the mean value of all the GreenR pixels in every frame in the active frame set in MigMate, bounded by the active region:

```
With Windows.ActiveFrames
        Output.Write .Statistic( _
                StatMean, PlaneBayerGreenR, , .Regions.ActiveRegion)
End With
```

This next code sample acquires thirty YCbCr frames from a device and prints out the total, fixed pattern, and temporal noise for the Y component:

```
Dim frms As New Frames
frms.Resize 30
```

```
FrameGrabber.StartSync frms, 30
Output.Write "Total Noise: " & frms.Statistic(StatTotalNoise, , ComponentY)
Output.Write "FPN: " & frms.Statistic(StatFixedPatternNoise, , ComponentY)
Output.Write "Temporal Noise: " & frms.Statistic(StatTemporalNoise, , ComponentY)
```

Finally, this code sample appends to the active frame set in MigMate a frame whose pixel components are the averages of the corresponding pixel components in every frame in the set:

```
Windows.ActiveFrames.Append Windows.ActiveFrames.Statistic(StatPixelMeans)
```

3.3 Statistics on Large Numbers of Frames

When statistics are required for collections of frames numbering more than a few dozen, it becomes infeasible to store every frame in memory, and the above interfaces will not suffice. There is need, therefore, for a third interface to the statistics engine, this time a slightly more complicated one: the `CumulativeStats` object. The interface for this object is described in detail in the MigMate's online object model documentation, so we will not duplicate it here. Instead, we will describe its basic usage and refer the reader to that documentation for further information.

In order to accumulate statistics on large numbers of frames, without having each frame stored in memory, we must store only those intermediate terms that are necessary to calculate the final result. To do this, we must have *a priori* knowledge of which statistic types will be requested at the end of the collection cycle. For efficiency, we also require that we know beforehand what planes, components, and/or regions are of interest to the user. The `CumulativeStats` object therefore uses the following usage pattern:

1. Create a unique `CumulativeStats` object for each region of interest, or just one if the entire frame is needed.
2. If using regions, call `SetRegion` on each object to associate it with a region.
3. For each color plane and/or component that is required, set the corresponding `EnablePlane` and/or `EnableComponent` properties on each object to `True`.[4]
4. For each statistic type that will be requested, set the corresponding `EnableStatistic` property on each object to `True`.
5. Collect as many frames as necessary, using the built-in framegrabber or any other acquisition method, and accumulate the required intermediate terms by calling `AddFrame` and/or `AddFrames` on each object.
6. Call `Result` for each combination of statistic, plane, and component (which must have been enabled) on each object to obtain the results.
7. Optionally, return to step 5 to accumulate more frames and obtain the results.
8. Optionally, call `Reset` on each object and return to step 5 to clear all intermediate terms and start over.

3.3.1 Example

An example should help clarify the usage of `CumulativeStats`. Here we acquire 10,000 frames and print out the total, fixed pattern, and temporal noise for the Blue plane:

```
' Step 1
Dim cstats as New CumulativeStats
```

[4] If no color planes or components are enabled, the `CumulativeStats` object will attempt to guess which planes and components will most likely be required as soon as the first frame is added. For example, if the first frame is a grayscale image with a Bayer CFA, then all four Bayer planes and `ComponentAll` will automatically be enabled; if it is an RGB frame, then `PlaneAll` and `ComponentRed`, `ComponentGreen`, and `ComponentBlue` will be enabled.

```
' Step 3
cstats.EnablePlane(PlaneBayerBlue) = True

' Step 4
cstats.EnableStatistic(StatTotalNoise) = True
cstats.EnableStatistic(StatFixedPatternNoise) = True
cstats.EnableStatistic(StatTemporalNoise) = True


' Step 5
Dim frms As New Frames
frms.Resize 50
For i = 1 To 10000 Step 50
        FrameGrabber.StartSync frms, 50
        cstats.AddFrames frms, 50
Next i

' Step 6
Output.Write "Total Noise: " & cstats.Result(StatTotalNoise, PlaneBayerBlue)
Output.Write "FPN: " & cstats.Result(StatFixedPatternNoise, PlaneBayerBlue)
Output.Write "Temporal Noise: " & cstats.Result(StatTemporalNoise, PlaneBayerBlue)
```

4 Statistic Definitions

In this section we give mathematical definitions of each of the statistic types currently supported by the MigMate statistics engine. These definitions have been derived from discussions with the Characterization group and from the documents listed in the References section; they are by no means the only possible ones. The MigMate group is open to adding new statistic types with slightly or very different definitions from these at any time; however, for backwards compatibility, the definitions given here will be fixed indefinitely.

First, some preliminary definitions:

- Let N denote the number of frames used in the calculation.
- Let $\mathbf{L}$ denote the set of pixel locations given by the combination of plane, component, and/or region used in the calculation.
- Let $p_n(l)$ denote the value of the pixel at location l in the nth frame.
- Let $\overline{p}_n$ denote the mean value of all pixels in the nth frame:

$$\overline{p}_n \equiv \frac{1}{|\mathbf{L}|}\sum_{l\in\mathbf{L}} p_n(l).$$

- Let $\overline{p}(l)$ denote the mean value of the pixel at location l across all N frames:

$$\overline{p}(l) \equiv \frac{1}{N}\sum_{n=1}^{N} p_n(l).$$

- When the area used in the calculation is rectangular:
  - Let Y and X denote the number of rows and columns in the area, respectively.
  - Let $p_n(x,y)$ denote the value of the pixel in the xth column and yth row of the area in the nth frame.
  - Let $h_n(y)$ and $v_n(x)$ denote the mean value of the pixels in the yth row and xth column of the area, respectively, in the nth frame. That is,

$$h_n(y) \equiv \frac{1}{X}\sum_{x=0}^{X-1} p_n(x,y) \text{ and } v_n(x) \equiv \frac{1}{Y}\sum_{y=0}^{Y-1} p_n(x,y).$$

- Let $\bar{h}(y)$ and $\bar{v}(x)$ denote the mean value of the pixels in the yth row and xth column of the area, respectively, across all N frames. That is,
$$\bar{h}(y) \equiv \frac{1}{N}\sum_{n=1}^{N} h_n(y) \text{ and } \bar{v}(x) \equiv \frac{1}{N}\sum_{n=1}^{N} v_n(x).$$

- Let M denote the maximum possible code for a pixel component, that is, $M \equiv 2^b - 1$, where b is the number of bits of precision per component.

4.1 Overall Statistics

4.1.1 Mean

$$\langle \bar{p} \rangle \equiv \frac{1}{|\mathbf{L}| N} \sum_{l \in \mathbf{L}} \sum_{n=1}^{N} p_n(l)$$

4.1.4 Maximum

$$\max \equiv \max_{l \in \mathbf{L}} \left(\max_{n=1}^{N} \left(p_n(l) \right) \right)$$

4.1.3 Minimum

$$\min \equiv \min_{l \in \mathbf{L}} \left(\min_{n=1}^{N} \left(p_n(l) \right) \right)$$

4.1.4 Histograms

4.1.4.1 Histogram

The vector
$$\langle HIST \rangle \equiv \langle \|0\|, \|1\|, \ldots, \|M\| \rangle,$$
where $\|i\|$ is the number of pixel components with value i in the area $\mathbf{L}$ in all frames.

4.1.4.2 Cumulative Histogram[5]

The vector
$$\langle HIST \rangle \equiv \langle \|0\|, \|1\|, \ldots, \|M\| \rangle,$$
where $\|i\|$ is the number of pixel components with values up to (and including) i in the area $\mathbf{L}$ in all frames.

4.1.4.3 Normalized Cumulative Histogram

The vector
$$\langle HIST \rangle \equiv \langle \|0\|, \|1\|, \ldots, \|M\| \rangle,$$

[5] Note that the term 'cumulative' is used here in a different sense than cumulative statistics as discussed in section 3.3. This statistic can be used as an instantaneous statistic on a frame or frames, or as a cumulative statistic, just like any of the other statistics.

where $\|i\|$ is the number of pixel components with values up to (and including) i in the area $\mathbf{L}$ in all frames, multiplied by $\frac{100}{|\mathbf{L}|N}$

4.1.5 Total Noise

4.1.5.1 Standard Method

$$\sigma_{Total} \equiv \begin{cases} \sqrt{\frac{1}{|\mathbf{L}|}\sum_{l\in\mathbf{L}}\left(p_1(l)-\langle\overline{p}\rangle\right)^2}, & N=1 \\ \sqrt{\frac{1}{|\mathbf{L}|(N-1)}\sum_{l\in\mathbf{L}}\sum_{n=1}^{N}\left(p_n(l)-\langle\overline{p}\rangle\right)^2}, & N>1 \end{cases}$$

4.1.5.2 Root-Mean-Square Method

$$\sigma_{RMS} \equiv \sqrt{\frac{1}{|\mathbf{L}|N}\sum_{l\in\mathbf{L}}\sum_{n=1}^{N}\left(p_n(l)-\langle\overline{p}\rangle\right)^2}$$

4.1.5.3 Standard Deviation Method

$$\sigma \equiv \begin{cases} 0, & |\mathbf{L}|=N=1 \\ \sqrt{\frac{1}{|\mathbf{L}|N-1}\sum_{l\in\mathbf{L}}\sum_{n=1}^{N}\left(p_n(l)-\langle\overline{p}\rangle\right)^2}, & |\mathbf{L}|N>1 \end{cases}$$

4.1.6 Fixed Pattern Noise

4.1.6.1 Standard Method[6]

$$\sigma_{FPN} \equiv \begin{cases} \sigma_{Total}, & N=1 \\ \sqrt{\frac{1}{|\mathbf{L}|N(N-1)}\left(\sum_{l\in\mathbf{L}}\left(\sum_{n=1}^{N}p_n(l)\right)^2-\frac{1}{|\mathbf{L}|}\left(\sum_{l\in\mathbf{L}}\sum_{n=1}^{N}p_n(l)\right)^2\right)}, & N>1 \end{cases}$$

[6] This somewhat daunting definition was chosen in order to maintain the invariant relationship $\sigma_{Total}{}^2=\sigma_{FPN}{}^2+\sigma_{Temp}{}^2$ between the standard methods of calculating Total, Fixed Pattern, and Temporal Noise. The reader is encouraged to verify that this definition can be made equivalent to the definition of σ_{FPNRMS} by replacing the $(N-1)$ term in the denominator here with N.

4.1.6.2 Root-Mean-Square Method

$$\sigma_{FPNRMS} \equiv \sqrt{\frac{1}{|\mathbf{L}|}\sum_{l\in\mathbf{L}}\left(\overline{p}(l)-\langle\overline{p}\rangle\right)^2}$$

4.1.7 Temporal Noise

4.1.7.1 Standard Method

$$\sigma_{Temp} \equiv \begin{cases} 0, & N=1 \\ \sqrt{\frac{1}{|\mathbf{L}|(N-1)}\sum_{l\in\mathbf{L}}\sum_{n=1}^{N}\left(p_n(l)-\overline{p}(l)\right)^2}, & N>1 \end{cases}$$

4.1.7.2 Root-Mean-Square Method

$$\sigma_{TempRMS} \equiv \sqrt{\frac{1}{|\mathbf{L}|N}\sum_{l\in\mathbf{L}}\sum_{n=1}^{N}\left(p_n(l)-\overline{p}(l)\right)^2}$$

4.2 Frame-wise Statistics

4.1.4 Frame Flicker Noise

4.2.1.1 Standard Method

$$\sigma_{Frame} \equiv \begin{cases} 0, & N=1 \\ \sqrt{\frac{1}{(N-1)}\sum_{n=1}^{N}\left(\overline{p}_n-\langle\overline{p}\rangle\right)^2}, & N>1 \end{cases}$$

4.2.1.2 Root-Mean-Square Method

$$\sigma_{FrameRMS} \equiv \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\overline{p}_n-\langle\overline{p}\rangle\right)^2}$$

4.3 Row-wise Statistics

4.3.1 Row Means

The vector

$$\langle\overline{h}\rangle \equiv \langle\overline{h}(0),\overline{h}(1),\ldots,\overline{h}(Y-1)\rangle.$$

4.3.2 Row Total Noise

4.3.2.1 Standard Method

$$\sigma_{RowTotal} \equiv \begin{cases} \sqrt{\dfrac{1}{Y}\displaystyle\sum_{y=0}^{Y-1}\left(h_1(y)-\langle\overline{p}\rangle\right)^2}, & N=1 \\ \sqrt{\dfrac{1}{Y(N-1)}\displaystyle\sum_{y=0}^{Y-1}\sum_{n=1}^{N}\left(h_n(y)-\langle\overline{p}\rangle\right)^2}, & N>1 \end{cases}$$

4.3.2.2 Root-Mean-Square Method

$$\sigma_{RowTotalRMS} \equiv \sqrt{\frac{1}{YN}\sum_{y=0}^{Y-1}\sum_{n=1}^{N}\left(h_n(y)-\langle\overline{p}\rangle\right)^2}$$

4.3.2.3 Standard Deviation Method

$$\sigma_{Row} \equiv \begin{cases} 0, & Y=N=1 \\ \sqrt{\dfrac{1}{YN-1}\displaystyle\sum_{y=0}^{Y-1}\sum_{n=1}^{N}\left(h_n(y)-\langle\overline{p}\rangle\right)^2}, & YN>1 \end{cases}$$

4.3.3 Row Fixed Pattern Noise

4.3.3.1 Standard Method[7]

$$\sigma_{RowFPN} \equiv \begin{cases} \sigma_{RowTotal}, & N=1 \\ \sqrt{\dfrac{1}{X^2YN(N-1)}\left(\displaystyle\sum_{y=0}^{Y-1}\left(\sum_{x=0}^{X-1}\sum_{n=1}^{N}p_n(x,y)\right)^2-\frac{1}{Y}\left(\sum_{y=0}^{Y-1}\sum_{x=0}^{X-1}\sum_{n=1}^{N}p_n(x,y)\right)^2\right)}, & N>1 \end{cases}$$

4.3.3.2 Root-Mean-Square Method

$$\sigma_{RowFPNRMS} \equiv \sqrt{\frac{1}{Y}\sum_{y=0}^{Y-1}\left(\overline{h}(y)-\langle\overline{p}\rangle\right)^2}$$

4.3.3.3 SMIA Horizontal FPN Level

As defined in the SMIA Specification, Version 1.0, Part 5, Section 5.3.1, except that the value returned has not yet been normalized by the full-scale deflection (FSD), as this quantity is unknown to the engine.

---

[7] See the note for Total FPN, Standard Method.

4.3.3.4 SMIA Horizontal FPN Max

As defined in the SMIA Specification, Version 1.0, Part 5, Section 5.3.1, except that the value returned has not yet been normalized by FSD.[8]

4.3.4 Row Temporal Noise

4.3.4.1 Standard Method (a.k.a. SMIA Row Noise Level)

$$\sigma_{RowTemp} \equiv \begin{cases} 0, & N=1 \\ \sqrt{\dfrac{1}{Y(N-1)}\sum_{y=0}^{Y-1}\sum_{n=1}^{N}\left(h_n(y)-\bar{h}(y)\right)^2}, & N>1 \end{cases}$$

Note that in order to obtain the Row Noise Level as defined in the SMIA Specification, Version 1.0, Part 5, Section 5.6.1, this value must be normalized by FSD and converted to decibels.

4.3.4.2 Root-Mean-Square Method

$$\sigma_{RowTempRMS} \equiv \sqrt{\frac{1}{YN}\sum_{y=0}^{Y-1}\sum_{n=1}^{N}\left(h_n(y)-\bar{h}(y)\right)^2}$$

4.3.4.3 SMIA Row Noise Max

As defined in the SMIA Specification, Version 1.0, Part 5, Section 5.6.1, except that the value returned has not yet been normalized by FSD and converted to decibels.

4.4 Column-wise Statistics

4.4.1 Column Means

The vector

$$\langle \bar{v} \rangle \equiv \langle \bar{v}(0), \bar{v}(1), \ldots, \bar{v}(X-1) \rangle .$$

4.4.2 Column Total Noise

4.4.2.1 Standard Method

$$\sigma_{ColTotal} \equiv \begin{cases} \sqrt{\dfrac{1}{X}\sum_{x=0}^{X-1}\left(v_1(x)-\langle \bar{p} \rangle\right)^2}, & N=1 \\ \sqrt{\dfrac{1}{X(N-1)}\sum_{x=0}^{X-1}\sum_{n=1}^{N}\left(v_n(x)-\langle \bar{p} \rangle\right)^2}, & N>1 \end{cases}$$

4.4.2.2 Root-Mean-Square Method

$$\sigma_{ColTotalRMS} \equiv \sqrt{\frac{1}{XN}\sum_{x=0}^{X-1}\sum_{n=1}^{N}\left(v_n(x)-\langle \bar{p} \rangle\right)^2}$$

---

[8] We have interpreted the term “valid row” in the specification to mean a row that is inside the frame and is also within 5 rows of row *j*.

4.4.2.3 Standard Deviation Method

$$\sigma_{Col} \equiv \begin{cases} 0, & X = N = 1 \\ \sqrt{\dfrac{1}{XN-1}\displaystyle\sum_{x=0}^{X-1}\sum_{n=1}^{N}\left(v_n(x) - \langle \overline{p} \rangle\right)^2}, & XN > 1 \end{cases}$$

4.4.3 Column Fixed Pattern Noise

4.4.3.1 Standard Method[9]

$$\sigma_{ColFPN} \equiv \begin{cases} \sigma_{ColTotal}, & N = 1 \\ \sqrt{\dfrac{1}{XY^2N(N-1)}\left(\displaystyle\sum_{x=0}^{X-1}\left(\sum_{y=0}^{Y-1}\sum_{n=1}^{N} p_n(x,y)\right)^2 - \frac{1}{X}\left(\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}\sum_{n=1}^{N} p_n(x,y)\right)^2\right)}, & N > 1 \end{cases}$$

4.4.3.2 Root-Mean-Square Method

$$\sigma_{ColFPNRMS} \equiv \sqrt{\frac{1}{X}\sum_{x=0}^{X-1}\left(\overline{v}(x) - \langle \overline{p} \rangle\right)^2}$$

4.4.3.3 SMIA Vertical FPN Level

As defined in the SMIA Specification, Version 1.0, Part 5, Section 5.2.1, except that the value returned has not yet been normalized by FSD.

4.4.3.4 SMIA Vertical FPN Max

As defined in the SMIA Specification, Version 1.0, Part 5, Section 5.2.1, except that the value returned has not yet been normalized by FSD.[10]

4.4.4 Column Temporal Noise

4.4.4.1 Standard Method (a.k.a. SMIA Column Noise Level)

$$\sigma_{ColTemp} \equiv \begin{cases} 0, & N = 1 \\ \sqrt{\dfrac{1}{X(N-1)}\displaystyle\sum_{x=0}^{X-1}\sum_{n=1}^{N}\left(v_n(x) - \overline{v}(x)\right)^2}, & N > 1 \end{cases}$$

Note that in order to obtain the SMIA Column Noise Level as defined in the SMIA Specification, Version 1.0, Part 5, Section 5.5.1, this value must be normalized by FSD and converted to decibels.

---

[9] See the note for Total FPN, Standard Method.
[10] We have interpreted the term "valid column" in the specification to mean a row that is inside the frame and is also within 5 columns of column *i*.

4.4.4.2 Root-Mean-Square Method

$$\sigma_{ColTempRMS} \equiv \sqrt{\frac{1}{XN}\sum_{x=0}^{X-1}\sum_{n=1}^{N}\left(v_n\left(x\right)-\overline{v}\left(x\right)\right)^2}$$

4.4.4.3 SMIA Column Noise Max

As defined in the SMIA Specification, Version 1.0, Part 5, Section 5.5.1, except that the value returned has not yet been normalized by FSD and converted to decibels.

4.5 Pixel-wise Statistics

4.5.1 Pixel Means

The frame $\overline{p}$ consisting of locations $l \in \mathbf{L}$ with values given by $\overline{p}\left(l\right)$.

4.5.2 Pixel Temporal Noise

4.5.2.1 Standard Method

The frame σ_{Pixel} consisting of locations $l \in \mathbf{L}$ with values given by

$$\sigma_{Pixel}\left(l\right) \equiv \begin{cases} 0, & N=1 \\ \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}\left(p_n\left(l\right)-\overline{p}\left(l\right)\right)^2}, & N>1 \end{cases}.$$

4.5.2.2 Root-Mean-Square Method

The frame $\sigma_{PixelRMS}$ consisting of locations $l \in \mathbf{L}$ with values given by

$$\sigma_{PixelRMS}\left(l\right) \equiv \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(p_n\left(l\right)-\overline{p}\left(l\right)\right)^2}.$$

4.6 Local (Kernel-based) Statistics

4.6.1 SMIA Local Standard Deviation

As defined in the SMIA Specification, Version 1.0, Part 5, Section 1.3.6, with the locality parameter $K=5$.

5 References

Noise Calculation Approach and Definitions. Micron Technology, Inc.
H:\mti\SECURE\MicronImaging\Pasadena_Char\CharSoft\Noise_calculation.pdf

SMIA Specification, Version 1.0, Part 5: Camera Characterisation Specification (Rev A). Nokia & ST Microelectronics.
http://edm.micron.com/cgi-bin/mtgetdoc.exe?itemID=09005aef81afd967