

*Software Specification*

# **PDAF Library API Specification**

---

Ver. 1.00

Sony Corporation

# Revision History

Version	Date	Description
1.00	March 24, 2016	First version

# Table of Contents

COVER PAGE.....	1
REVISION HISTORY.....	2
TABLE OF CONTENTS .....	3
1. ABSTRACT .....	4
2. ACRONYMS AND ABBREVIATIONS.....	5
3. FUNCTION LIST .....	5
4. FUNCTION SPECIFICATION .....	6
4.1 FUNCTION PdLIBGETVERSION .....	6
4.1.1 Structure PdLibVersion_t.....	7
4.2 FUNCTION PdLIBGETDEFOCUS .....	8
4.2.1 Structure DefocusOKNGThrLine_t.....	9
4.2.2 Structure PdLibInputData_t.....	11
4.2.3 Structure PdLibOutputData_t.....	25
5. ERROR CODES LIST.....	27
6. D_PD_ERROR_VALUE.....	29

# 1. Abstract

The specification requires basic knowledge of AF such as ContrastAF.

Sony develops PDAF Library (1) in Figure 1-1.

PDAF Library (1) converts phase difference data from image sensor into defocus.

PDAF (2) gets defocus from PDAF Library (1) and notify defocus to HybridAF (4).

HybridAF(4) controls PDAF (2) and ContrastAF (3) to get fine focus in short time.

Software components of PDAF (2), ContrastAF (3) and HybridAF (4) in Figure 1-1 are assumed to be developed by user and are out of scope of the specification.

The specification describes an API of Sony PDAF Library (1).

The API is designed for Sony CMOS image sensor with PDAF.

Explanation of the API is based on system architecture in Figure 1-1.

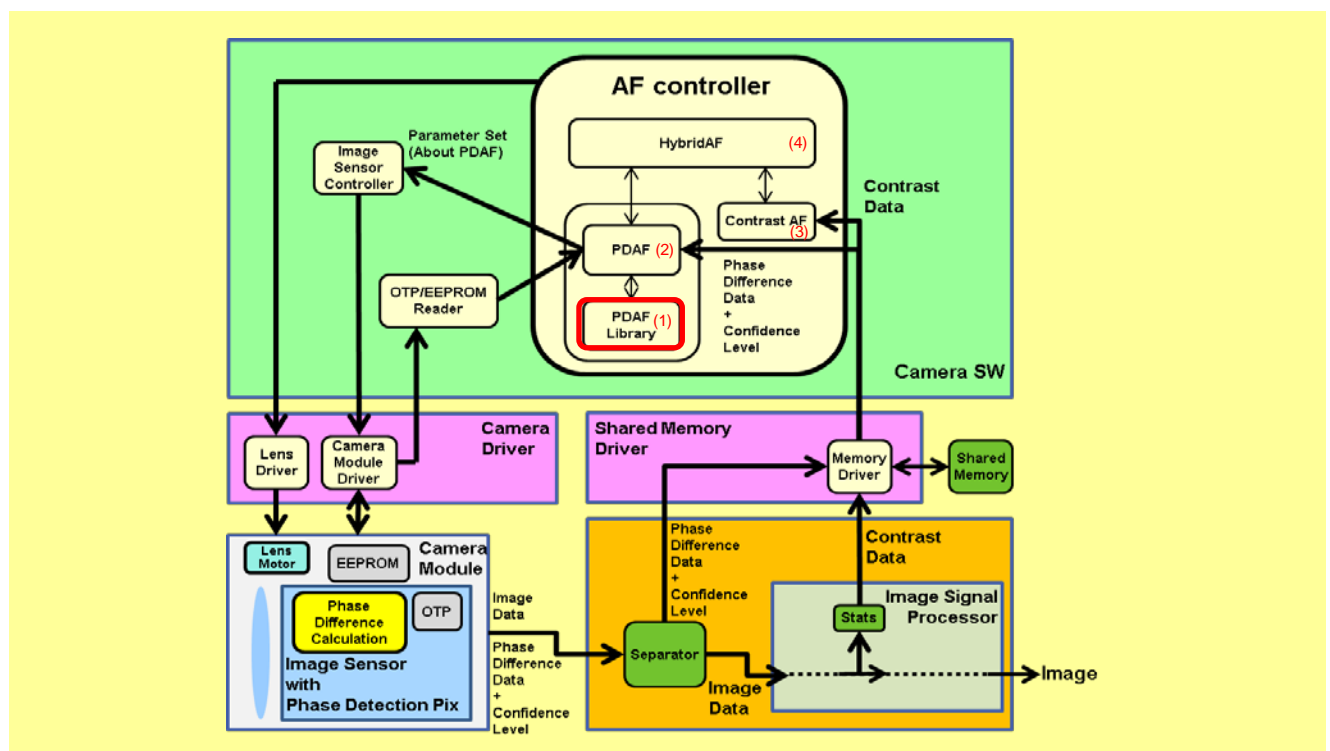


Figure 1-1 System Architecture

## 2. Acronyms and Abbreviations

The following acronyms and abbreviations are used in the specification.

**Table 2-1 Acronyms and Abbreviations**

API	Application Programming Interface
AF	Auto Focus
PD	Phase Detection
PDAF	Phase Detection Auto Focus

## 3. Function List

The API is defined as functions in Table 3-1.

**Table 3-1 Function List**

<b>Function</b>	<b>Description</b>
PdLibGetVersion	Get version information of PDAF Library.
PdLibGetDefocus	Get defocus data according to a PDAF window.

## 4. Function Specification

### 4.1 Function PdLibGetVersion

**Format:**

```
void PdLibGetVersion (  
    PdLibVersion_t *pfa_PdLibVersion  
);
```

**Description:**

Get version information of PDAF Library.

**Arguments:**

*\*pfa\_PdLibVersion*  
Version information of PDAF Library.  
Refer to [4.1.1 Structure PdLibVersion\\_t](#).  
In/Out : Out  
Type : PdLibVersion\_t  
Range : –

**Return Value:**

None.

**Constraints:**

None.

### 4.1.1 Structure PdLibVersion\_t

**Format:**

```
typedef struct {  
    unsigned long   MajorVersion;  
    unsigned long   MinorVersion;  
} PdLibVersion_t;
```

**Description:**

Version information of PDAF Library.

**Member:**

*MajorVersion*

Integer part of PDAF Library version.

Refer to Table 4-1.

In/Out : Out

Type : unsigned long

Range : 0x00000000 – 0xFFFFFFFF

*MinorVersion*

Decimal part of PDAF Library version.

Refer to Table 4-1.

In/Out : Out

Type : unsigned long

Range : 0x00000000 – 0xFFFFFFFF

**Table 4-1 Version**

PDAF Library Version	<u>MajorVersion</u> (Integer Part)	<u>MinorVersion</u> (Decimal Part)
0.01	0	1
0.02	0	2
0.10	0	10
0.20	0	20
1.00	1	0

## 4.2 Function PdLibGetDefocus

### Format:

```
signed long PdLibGetDefocus (  
    PdLibInputData_t *pfa_PdLibInputData,  
    PdLibOutputData_t *pfa_PdLibOutputData  
);
```

### Description:

Get defocus data according to a PDAF window.

When there are several PDAF windows,  
the function needs to be called the number of times of PDAF windows  
to get defocus data in each PDAF window.

Image sensor has 16x12(HxV) windows or 8x6(HxV) windows for fixed window mode  
and 8 windows for flexible window mode.

### Arguments:

*\*pfa\_PdLibInputData*

Input data needed for defocus data output.

Refer to [4.2.2 Structure PdLibInputData\\_t](#).

In/Out : In

Type : PdLibInputData\_t

Range : –

*\*pfa\_PdLibOutputData*

Defocus data.

Refer to [4.2.3 Structure PdLibOutputData\\_t](#).

In/Out : Out

Type : PdLibOutputData\_t

Range : –

### Return Value:

Notify if the function is performed successfully or not.

OK: D\_PD\_LIB\_E\_OK

NG: Refer to [5. Error Codes List](#)

### Constraints:

None.



## 4.2.1 Structure DefocusOKNGThrLine\_t

Format:

```
typedef struct {
    unsigned long    PointNum;
    unsigned long    *p_AnalogGain;
    unsigned long    *p_Confidence;
} DefocusOKNGThrLine_t;
```

Description:

Threshold line which determines Defocus OK/NG. Refer to Figure 4-2.

User decides the threshold line.

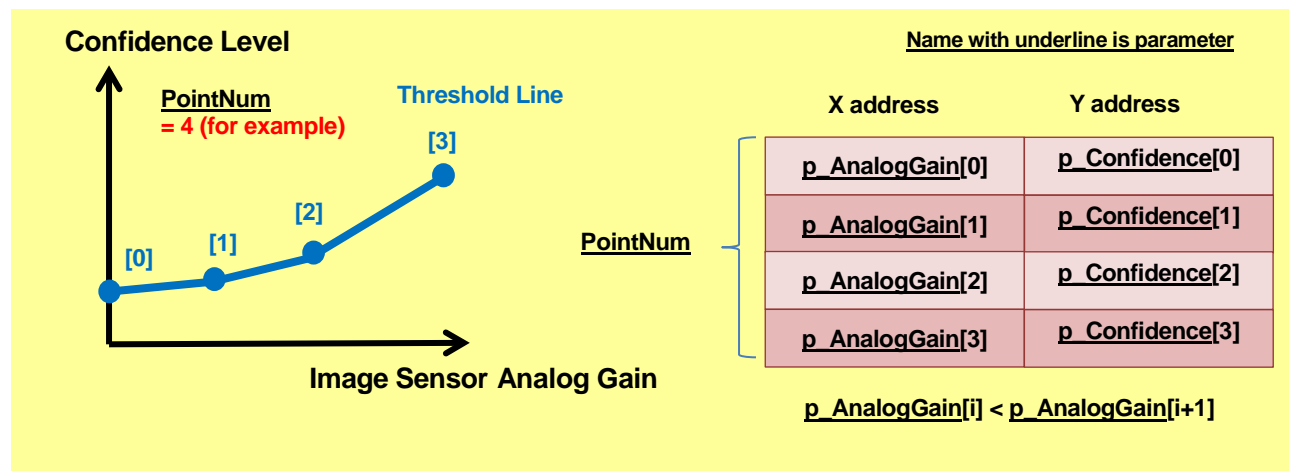


Figure 4-2 Threshold Line for Defocus OK / NG

Member:

*PointNum*

Number of points on the threshold line. Refer to Figure 4-2 and Figure 4-3.

In/Out : In

Type : unsigned long

Range : 0x00000002 – 0x00000020

*\*p\_AnalogGain*

Array of x address of points on the threshold line. Refer to Figure 4-2 and Figure 4-3.

In/Out : In

Type : unsigned long

Range : 0x00000000 – 0x07FFFFFF

*\*p\_Confidence*

Array of y address of points on the threshold line. Refer to Figure 4-2 and Figure 4-3.

In/Out : In

Type : unsigned long

Range : 0x00000000 – 0x07FFFFFF

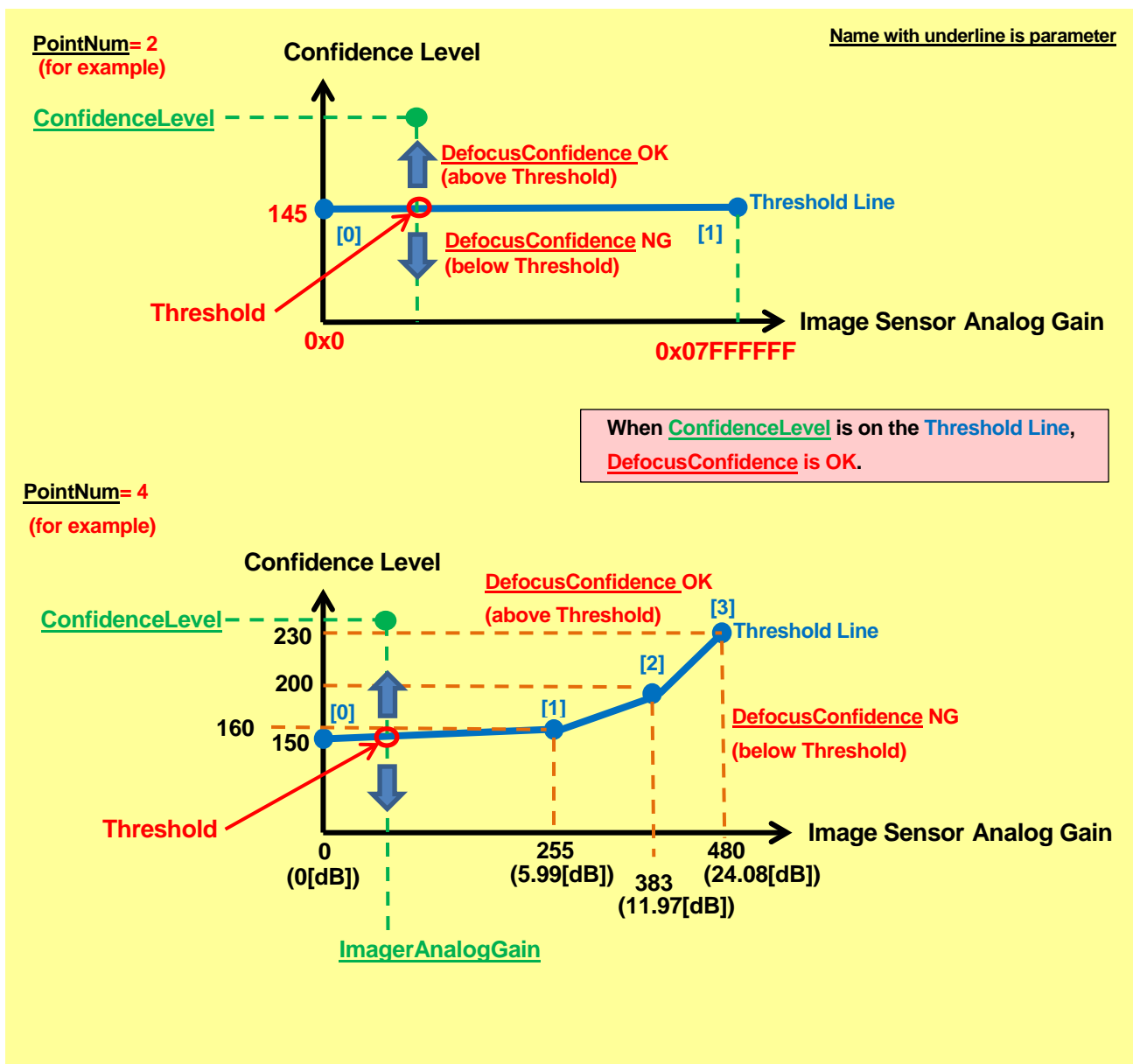


Figure 4-3 Threshold Line Settings for Defocus OK / NG

## 4.2.2 Structure PdLibInputData\_t

### Format:

```
typedef struct {  
    // Phase Difference Data and Confidence Level  
    signed long    PhaseDifference;  
    unsigned long  ConfidenceLevel;  
    // PDAF Window  
    unsigned short XSizeOfImage;  
    unsigned short YSizeOfImage;  
    unsigned short XAddressOfWindowStart;  
    unsigned short YAddressOfWindowStart;  
    unsigned short XAddressOfWindowEnd;  
    unsigned short YAddressOfWindowEnd;  
    // Slope and Offset (defocus vs phase difference)  
    unsigned short XKnotNumSlopeOffset;  
    unsigned short YKnotNumSlopeOffset;  
    signed long    *p_SlopeData;  
    signed long    *p_OffsetData;  
    unsigned short *p_XAddressKnotSlopeOffset;  
    unsigned short *p_YAddressKnotSlopeOffset;  
    signed long    AdjCoeffSlope;  
    // Defocus OK/NG  
    unsigned long  ImagerAnalogGain;  
    unsigned short XKnotNumDefocusOKNG;  
    unsigned short YKnotNumDefocusOKNG;  
    DefocusOKNGThrLine_t *p_DefocusOKNGThrLine;  
    unsigned short *p_XAddressKnotDefocusOKNG;  
    unsigned short *p_YAddressKnotDefocusOKNG;  
    // Phase Detection Pixel Density  
    unsigned long  DensityOfPhasePix;  
} PdLibInputData_t;
```

### Description:

Input data needed for defocus data output.

## Member: About phase difference data and confidence level

### *PhaseDifference*

Phase difference data which is output data from image sensor. Unit is pixel.

Bit width of output data (s6.4) from image sensor needs to be fit to s27.4.

4bit left shifting, range from -1024 ( $-64.0000 \ll 4$ ) to 1023 ( $63.9375 \ll 4$ ).

Refer to Figure 4-4.

In/Out : In

Type : signed long

Range : 0xF8000000 – 0x07FFFFFF

### *ConfidenceLevel*

Confidence level which is output data from image sensor.

Bit width of output data (u11.0) from image sensor needs to be fit to u32.0.

Refer to Figure 4-4.

In/Out : In

Type : unsigned long

Range : 0x00000000 – 0x07FFFFFF

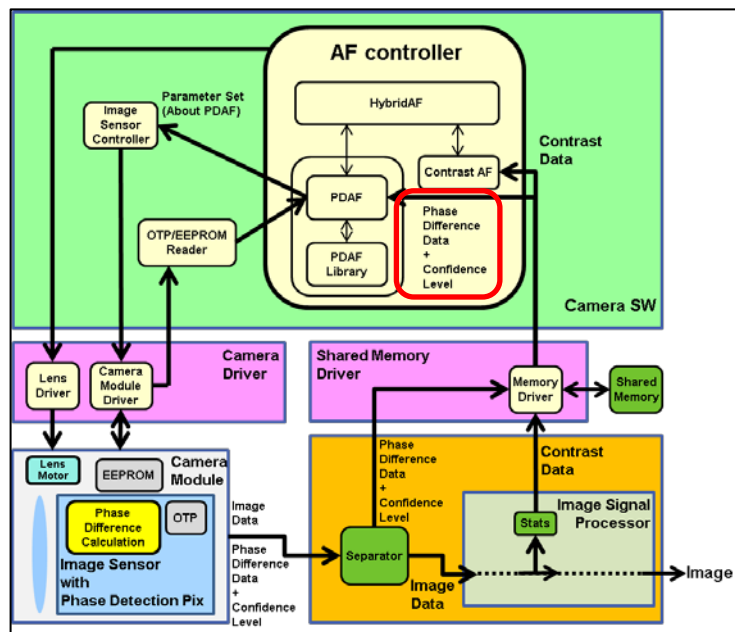


Figure 4-4 Phase Difference Data and Confidence Level

### Member: About PDAF window

PDAF window information must be in synchronization with phase difference data and confidence level.

#### *XSizeOfImage*

X size of image in all-pixel mode. Refer to Figure 4-5.

In/Out : In

Type : unsigned short

Range : 0x0002 – 0xFFFF

#### *YSizeOfImage*

Y size of image in all-pixel mode. Refer to Figure 4-5.

In/Out : In

Type : unsigned short

Range : 0x0002 – 0xFFFF

#### *XAddressOfWindowStart*

X address of PDAF window start position in all-pixel mode. Refer to Figure 4-5.

In/Out : In

Type : unsigned short

Range : 0x0000 – *XAddressOfWindowEnd-1*

#### *YAddressOfWindowStart*

Y address of PDAF window start position in all-pixel mode. Refer to Figure 4-5.

In/Out : In

Type : unsigned short

Range : 0x0000 – *YAddressOfWindowEnd-1*

#### *XAddressOfWindowEnd*

X address of PDAF window end position in all-pixel mode. Refer to Figure 4-5.

In/Out : In

Type : unsigned short

Range : *XAddressOfWindowStart+1* – *XSizeOfImage-1*

#### *YAddressOfWindowEnd*

Y address of PDAF window end position in all-pixel mode. Refer to Figure 4-5.

In/Out : In

Type : unsigned short

Range : *YAddressOfWindowStart+1* – *YSizeOfImage-1*

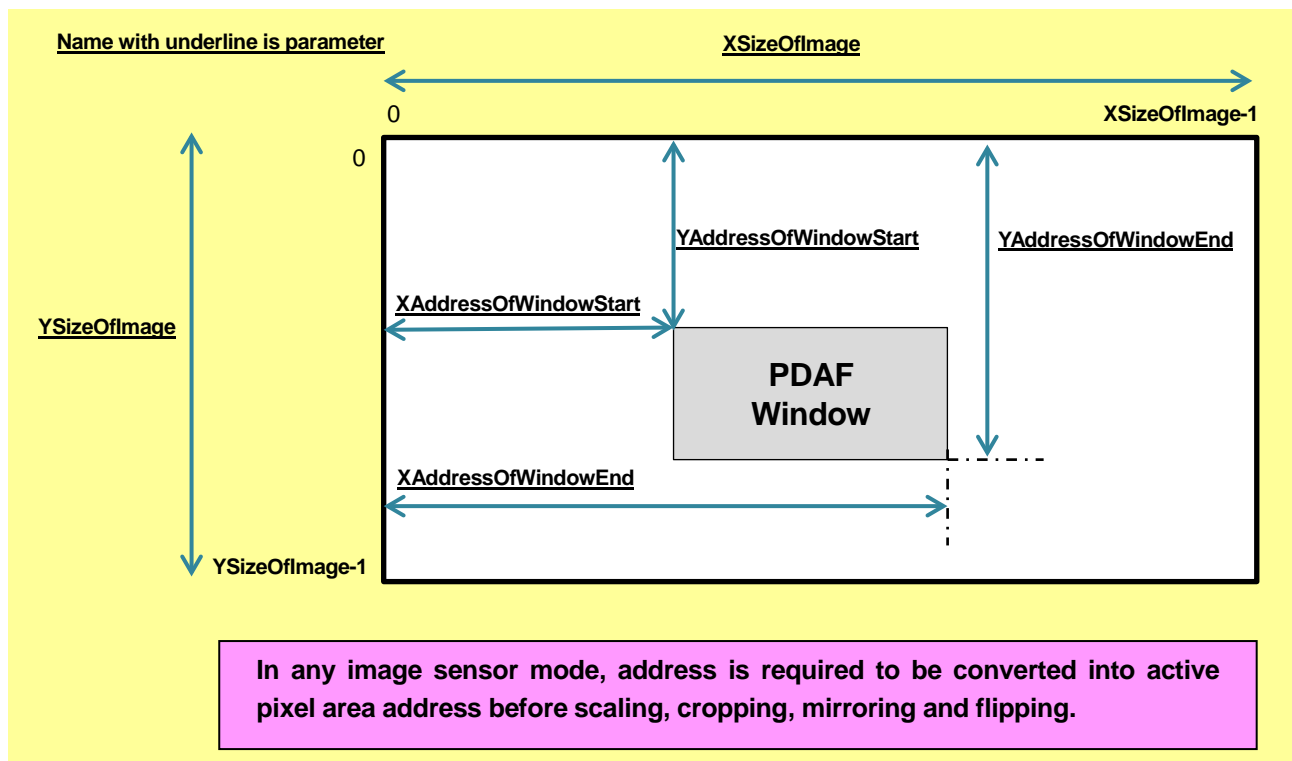


Figure 4-5 PDAF Window

**Member: About slope and offset (defocus vs phase difference)**

Setting parameters are needed to be provided from camera module integrator.

*XKnotNumSlopeOffset*

Number of knots in x-direction. Refer to Figure 4-6.

In/Out : In

Type : unsigned short

Range : 0x0002 – 0xFFFF

*YKnotNumSlopeOffset*

Number of knots in y-direction. Refer to Figure 4-6.

In/Out : In

Type : unsigned short

Range : 0x0002 – 0xFFFF

#### *\*p\_SlopeData*

Array of slope data. Refer to Figure 4-6.  
Slope data is fixed-point number of s21.10.  
In/Out : In  
Type : signed long  
Range : 0xF8000000 – 0x07FFFFFFF

Array of offset data. Refer to Figure 4-6.  
In/Out : In  
Type : signed long  
Range : 0xF8000000 – 0x07FFFFFFF

#### *\*p\_XAddressKnotSlopeOffset*

Array of x address of knots. Refer to Figure 4-7.  
In/Out : in  
Type : unsigned short  
Range : 0x0000 – 0xFFFF

#### *\*p\_YAddressKnotSlopeOffset*

Array of y address of knots. Refer to Figure 4-8.  
In/Out : In  
Type : unsigned short  
Range : 0x0000 – 0xFFFF

#### *AdjCoeffSlope*

Adjustment coefficient of slope which changes according to image sensor mode.  
Refer to Figure 4-6.  
In/Out : In  
Type : signed long  
Range : Select the following definitions according to image sensor mode.

D\_PD\_LIB\_SLOPE\_ADJ\_COEFF\_SENS\_MODE0 : Normal Full-pixel mode  
D\_PD\_LIB\_SLOPE\_ADJ\_COEFF\_SENS\_MODE1 : HDR(Full-pixel) mode  
D\_PD\_LIB\_SLOPE\_ADJ\_COEFF\_SENS\_MODE2 : Normal V2 Binning mode  
D\_PD\_LIB\_SLOPE\_ADJ\_COEFF\_SENS\_MODE3 : HDR(V2 Binning) mode  
D\_PD\_LIB\_SLOPE\_ADJ\_COEFF\_SENS\_MODE4 : Normal V4 Binning mode

Note: PDAF in these modes are not supported.

Normal V1/2 Sub-sampling mode  
Normal V1/4 Sub-sampling mode  
Normal V2 Binning + V1/2 Sub-sampling mode  
Normal V4 Binning mode  
Normal V4 Binning + V1/2 Sub-sampling mode

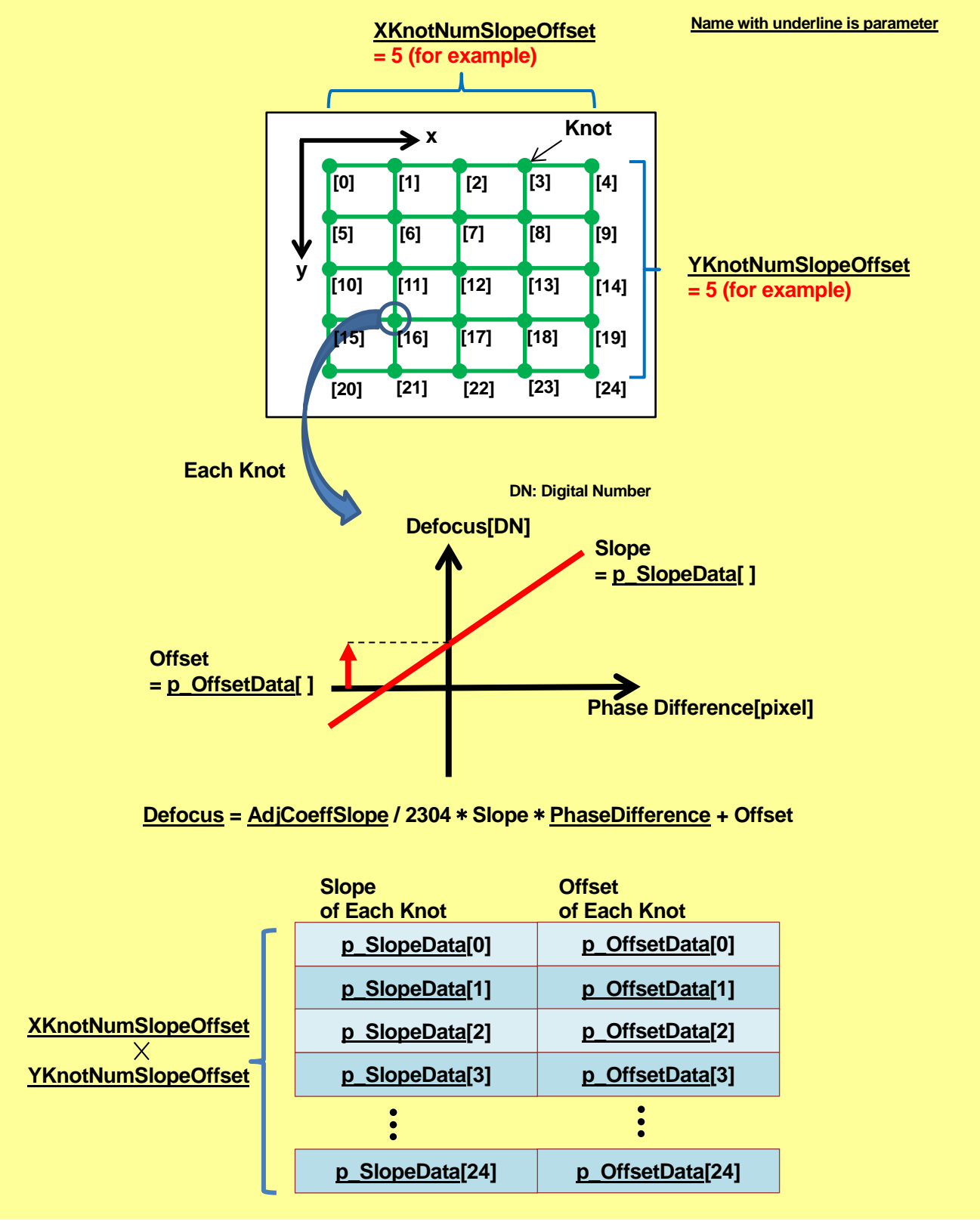


Figure 4-6 Slope and Offset (defocus vs phase difference)



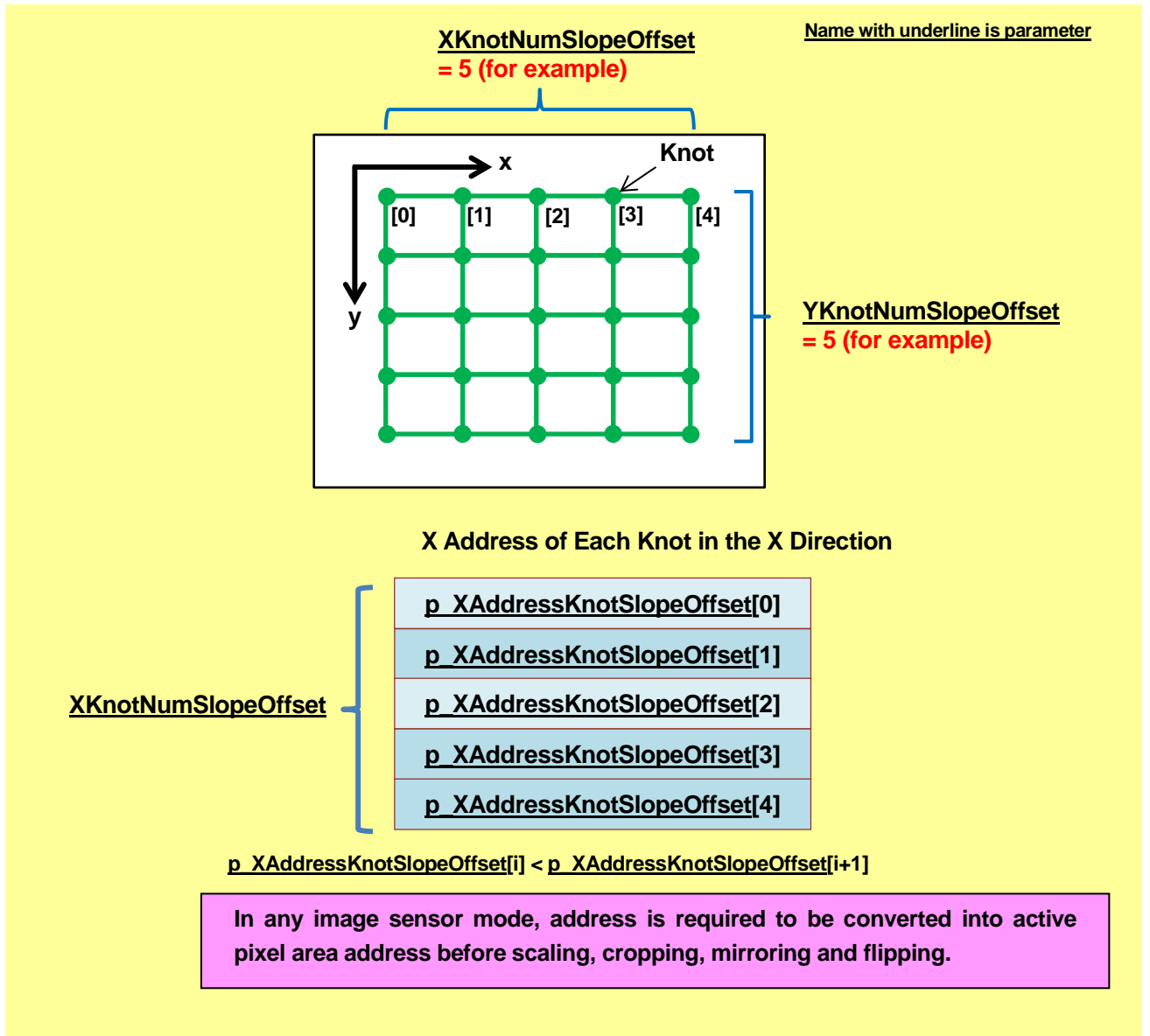


Figure 4-7 X Address of Knots for Slope and Offset

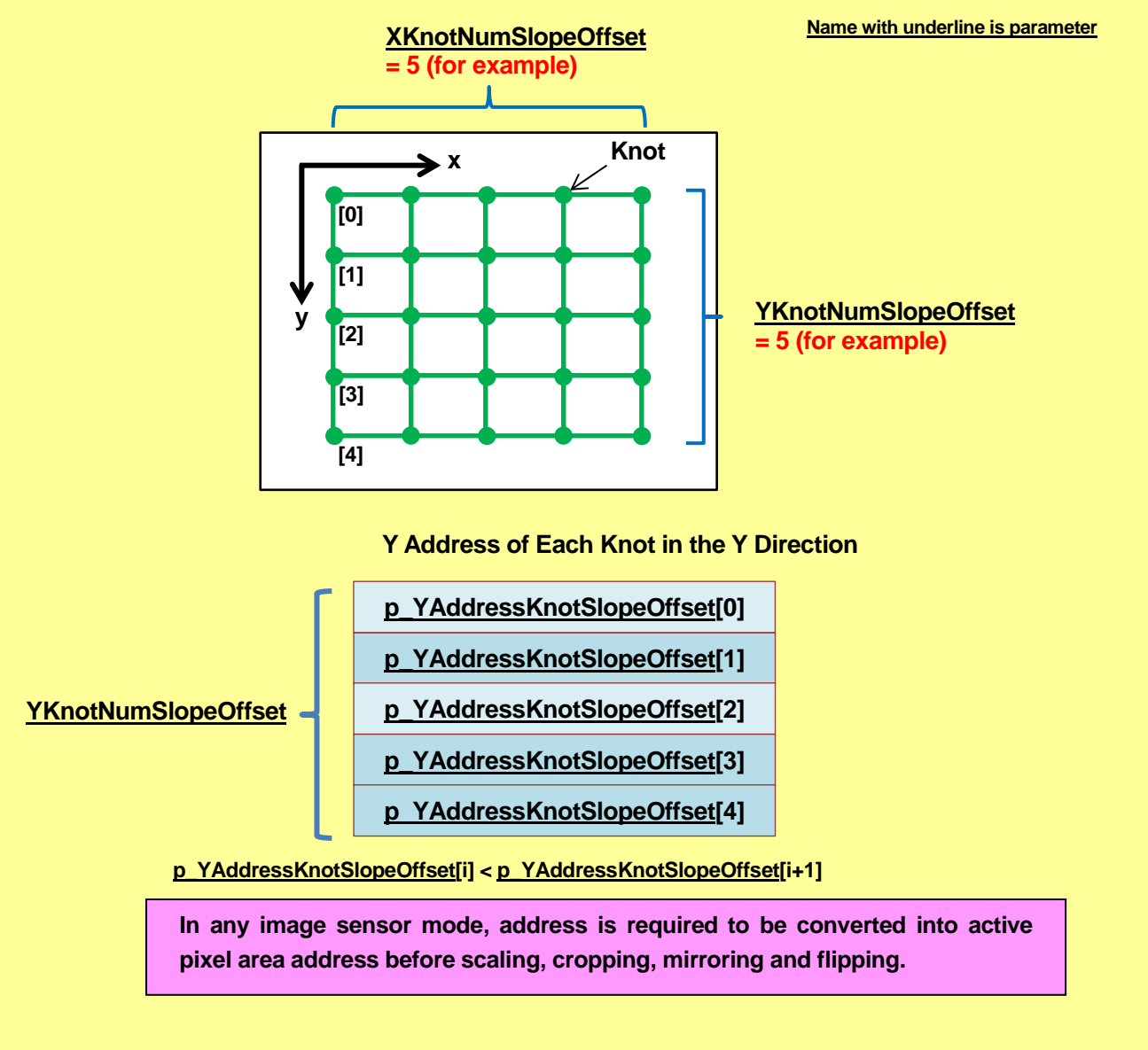


Figure 4-8 Y Address of Knots for Slope and Offset

## Member: About defocus OK/NG

User decides parameters to determine defocus OK/NG.

### *ImagerAnalogGain*

Image sensor analog gain which must be in synchronization with phase difference data and confidence level.

In/Out : In  
Type : unsigned long  
Range : 0x00000000 – 0x07FFFFFF

### *XKnotNumDefocusOKNG*

Number of knots in x-direction. Refer to Figure 4-9.

In/Out : In  
Type : unsigned short  
Range : 0x0000 – 0xFFFF

When 0 is set to both *XKnotNumDefocusOKNG* and *YKnotNumDefocusOKNG*, functionality of determining “Defocus confidence OK/NG” is disabled.

When 1 is set to both *XKnotNumDefocusOKNG* and *YKnotNumDefocusOKNG*, one knot point is available. Refer to Figure 4-12.

The threshold for Defocus OK/NG which corresponds to the knot point is applied any area on the image.

### *YKnotNumDefocusOKNG*

Number of knots in y-direction. Refer to Figure 4-9.

In/Out : In  
Type : unsigned short  
Range : 0x0000 – 0xFFFF

### *\*p\_DefocusOKNGThrLine*

Array of threshold line data which determines Defocus OK/NG. Refer to Figure 4-9.

Refer to [4.2.1 Structure DefocusOKNGThrLine\\_t](#).

In/Out : In  
Type : DefocusOKNGThrLine\_t  
Range : –

### *\*p\_XAddressKnotDefocusOKNG*

Array of x address of knots. Refer to Figure 4-9.

In/Out : In  
Type : unsigned short  
Range : 0x0000 – 0xFFFF

### *\*p\_YAddressKnotDefocusOKNG*

Array of y address of knots. Refer to Figure 4-10.

In/Out : In  
Type : unsigned short  
Range : 0x0000 – 0xFFFF

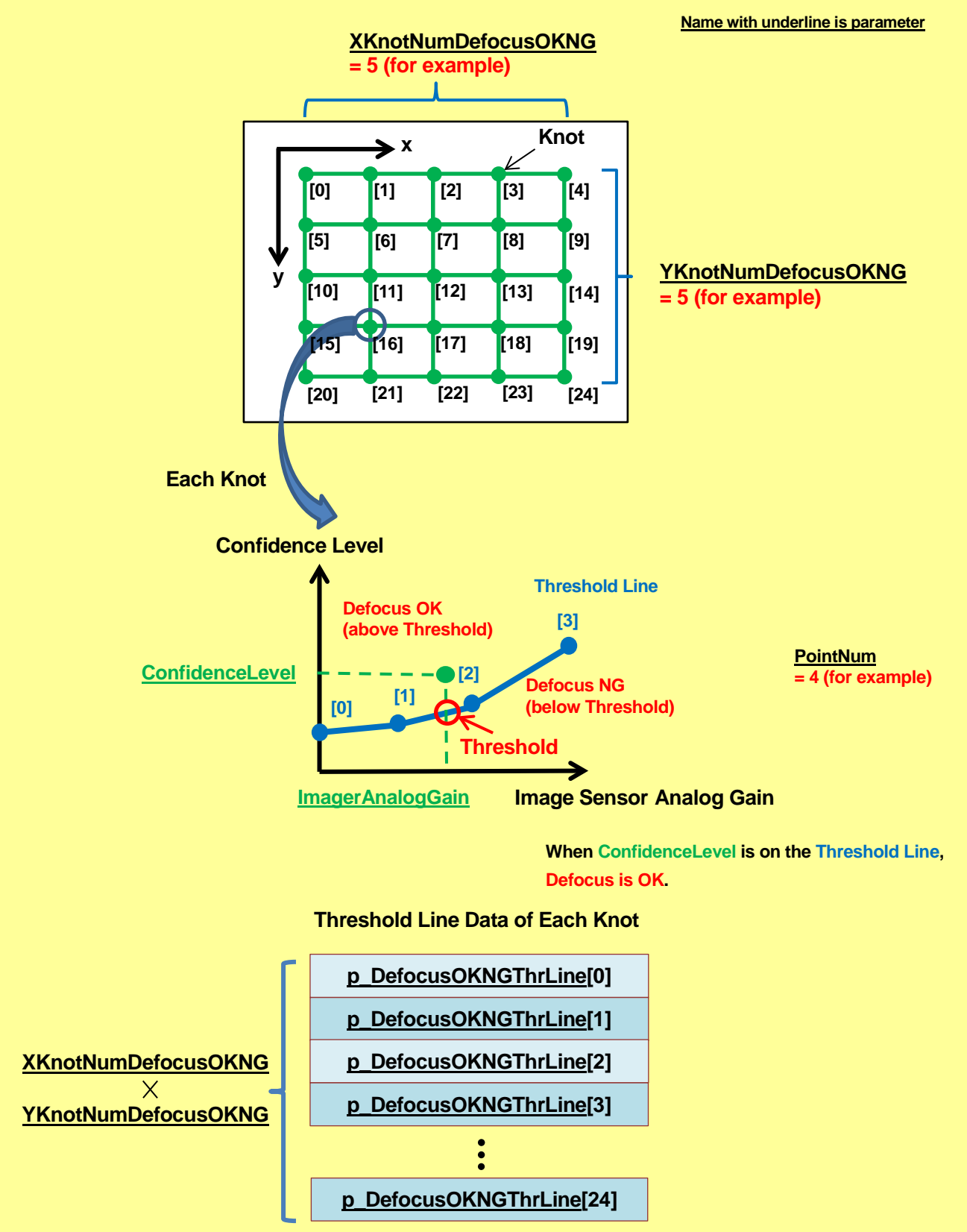


Figure 4-9 Defocus OK / NG and Threshold Line

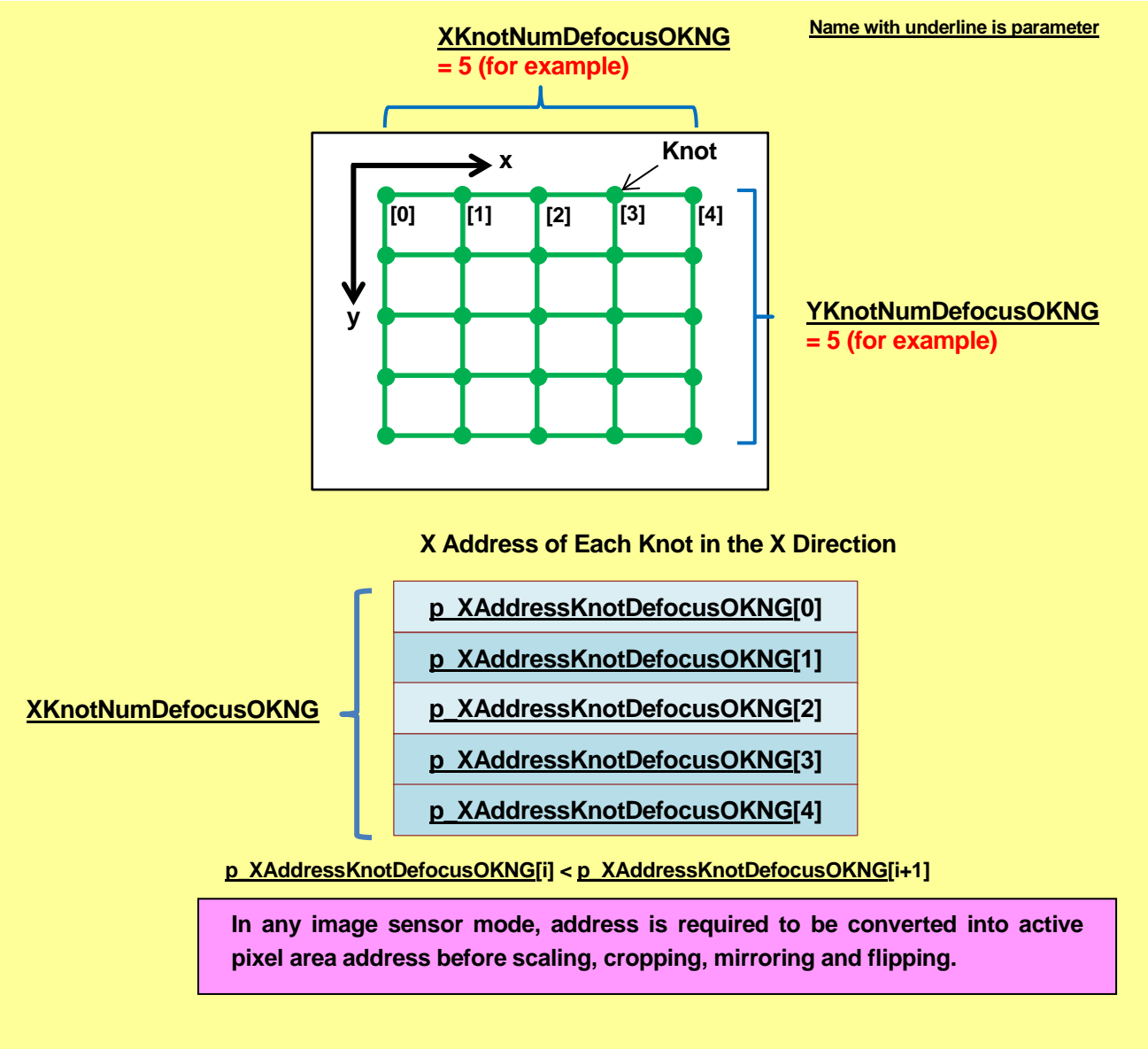


Figure 4-10 X Address of Knots for Threshold Line Data

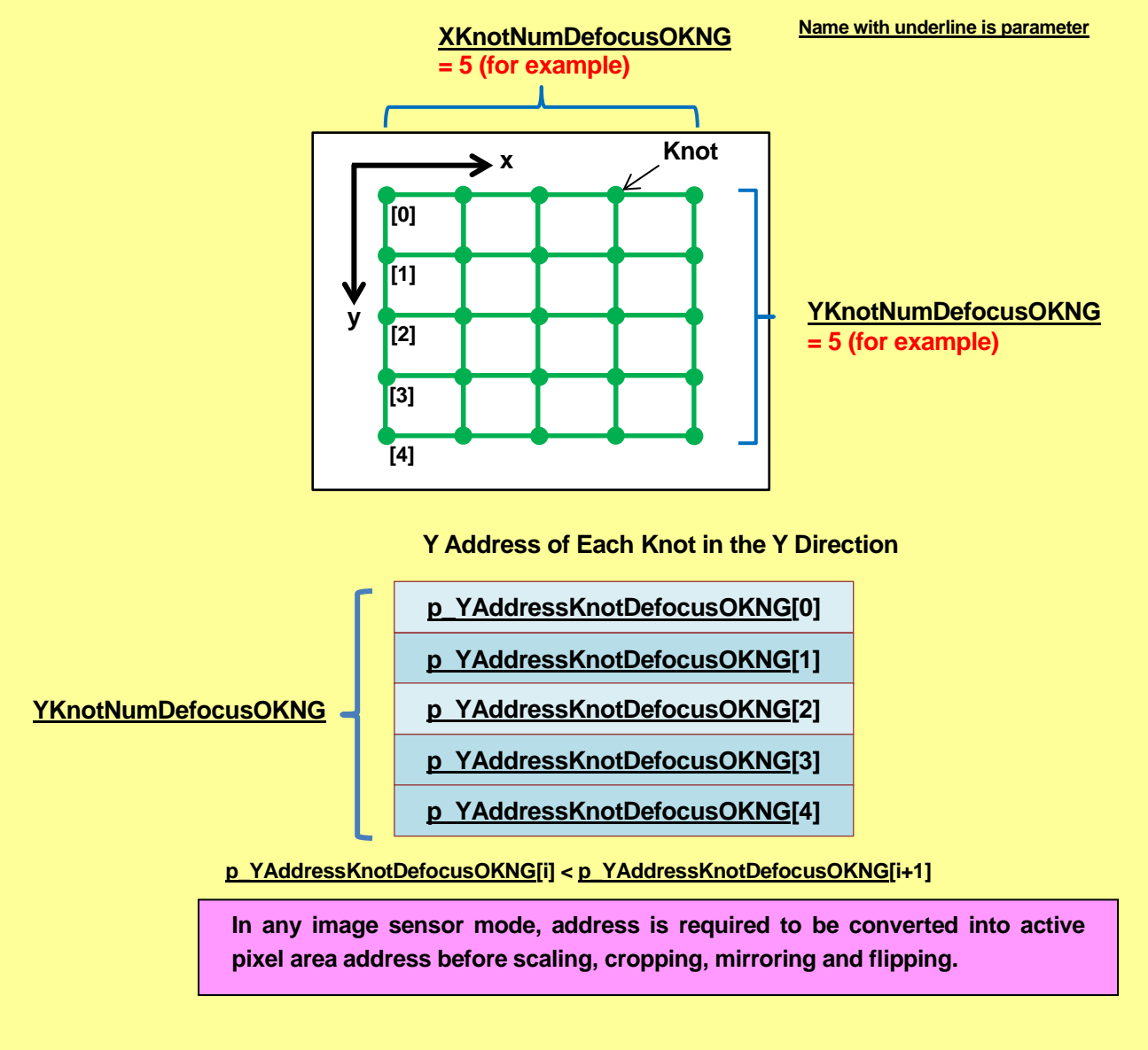


Figure 4-11 Y Address of Knots for Threshold Line Data

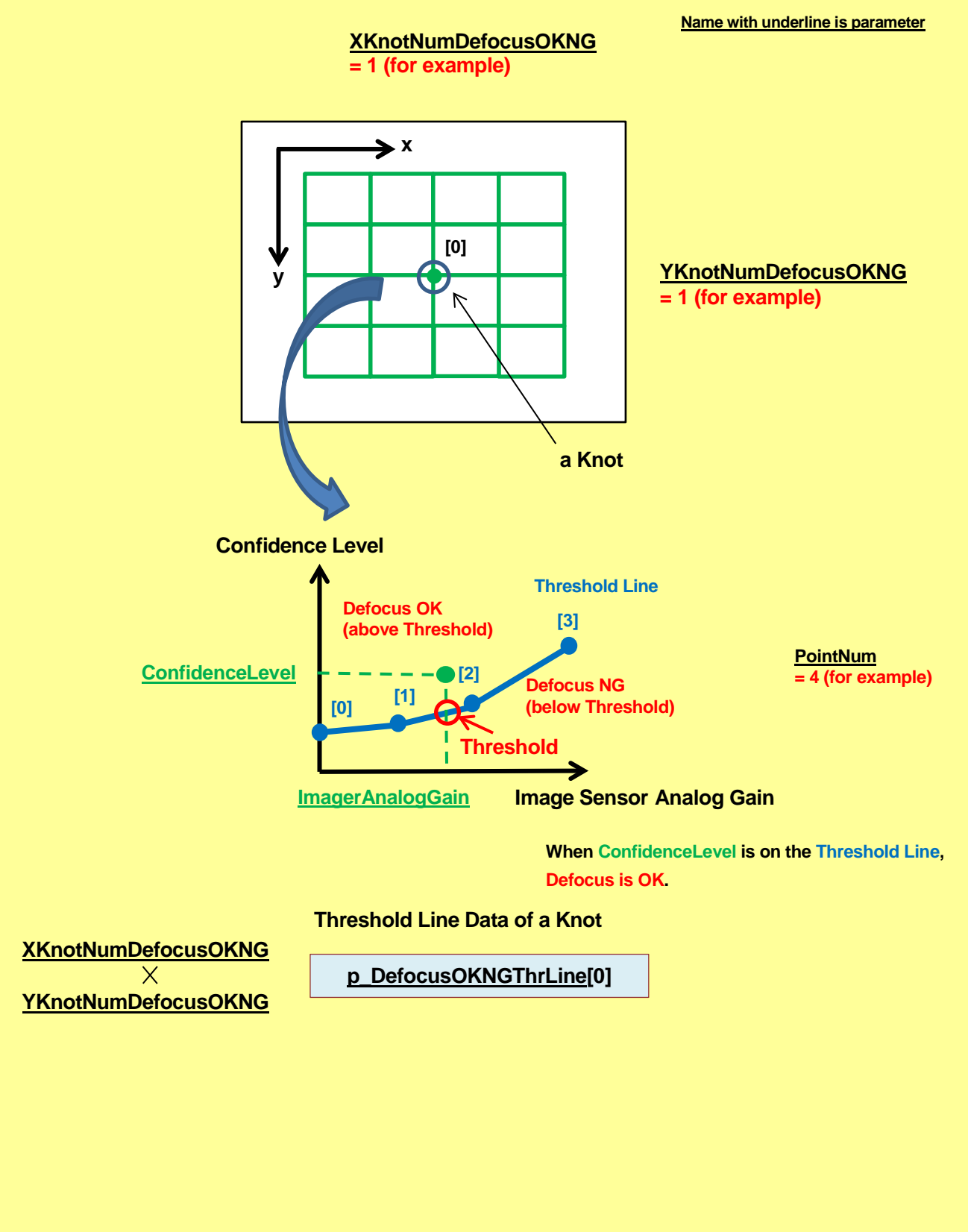


Figure 4-12 Set to both XKnotNumDefocusOKNG = 1 and YKnotNumDefocusOKNG = 1

**Member: About phase detection pixel density**

*DensityOfPhasePix*

Density of phase detection pixel which changes according to image sensor mode.

In/Out : In

Type : unsigned long

Range : Select the following definitions according to image sensor mode.

D\_PD\_LIB\_DENSITY\_SENS\_MODE0 : Normal Full-pixel mode

HDR(Full-pixel) mode

Normal V2 Binning mode

HDR(V2 Binning) mode

Normal V4 Binning mode

Please input "D\_PD\_LIB\_DENSITY\_SENS\_MODE0" as a fixed value regardless of image sensor mode.

Note: Note: PDAF in these modes are not supported.

Normal V1/2 Sub-sampling mode

Normal V1/4 Sub-sampling mode

Normal V2 Binning + V1/2 Sub-sampling mode

Normal V4 Binning mode

Normal V4 Binning + V1/2 Sub-sampling mode



### 4.2.3 Structure PdLibOutputData\_t

**Format:**

```
typedef struct {  
    signed long    Defocus;  
    signed char    DefocusConfidence;  
    unsigned long  DefocusConfidenceLevel;  
    signed long    PhaseDifference;  
} PdLibOutputData_t;
```

**Description:**

Defocus data.

**Members:**

*Defocus*

Defocus. Unit is DN (Digital Number). Refer to Figure 4-6.

Defocus is fixed-point number of s17.14.

Note: When user uses Defocus from PDAF Library, Defocus needs to be multiplied by -1.

Refer to Figure 4-14.

In/Out : Out

Type : signed long

Range : 0x80000000 – 0x7FFFFFFF

*DefocusConfidence*

Defocus OK/NG. Refer to Figure 4-13.

In/Out : Out

Type : signed char

Range : Return the following definitions.

D\_PD\_LIB\_E\_OK : OK

-ENCWDDON Determining defocus OK or NG is not compiled with by disable this  
functionality

-ELDCL Low defocus confidence level

-EPDVALERR Input error value of phase difference data

*DefocusConfidenceLevel*

Defocus OK/NG level. Refer to Figure 4-13.

In/Out : Out

Type : unsigned long

Range : 0x00000000 – 0xFFFFFFFF

*PhaseDifference*

Phase difference which is the same information as input data.

In/Out : Out

Type : signed long

Range : 0xF8000000 – 0x07FFFFFF

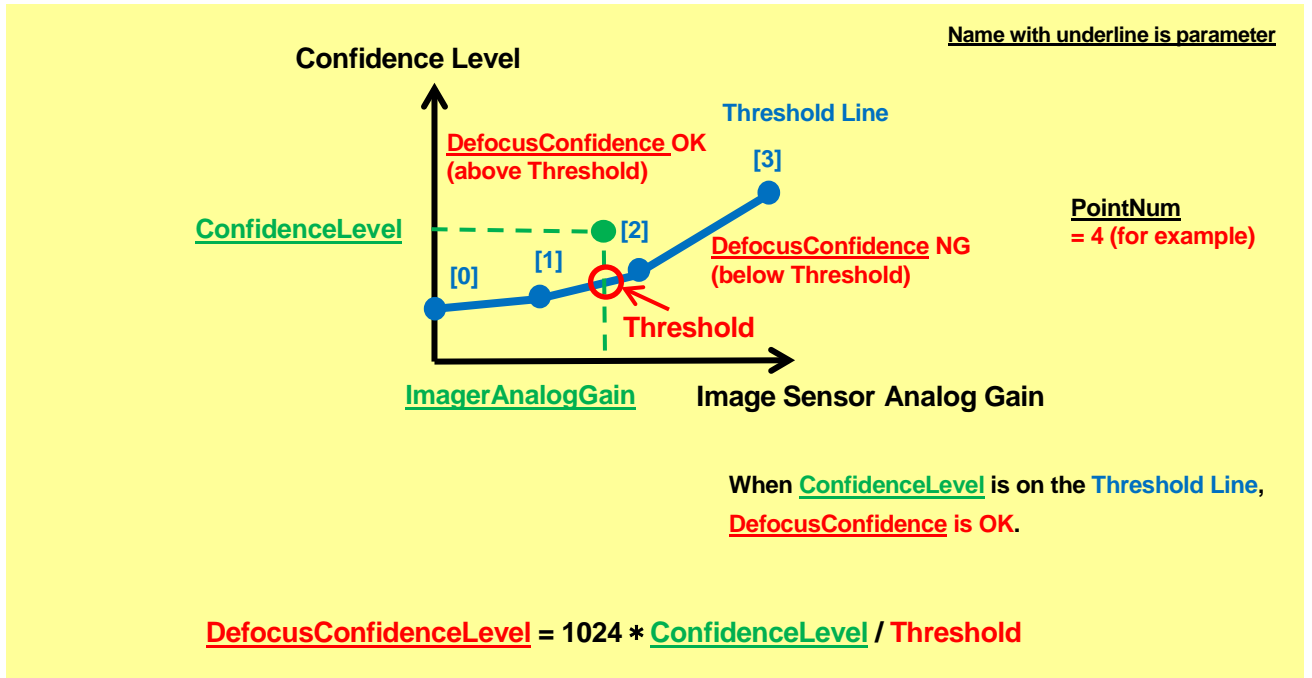


Figure 4-13 Defocus Confidence and Defocus Confidence Level

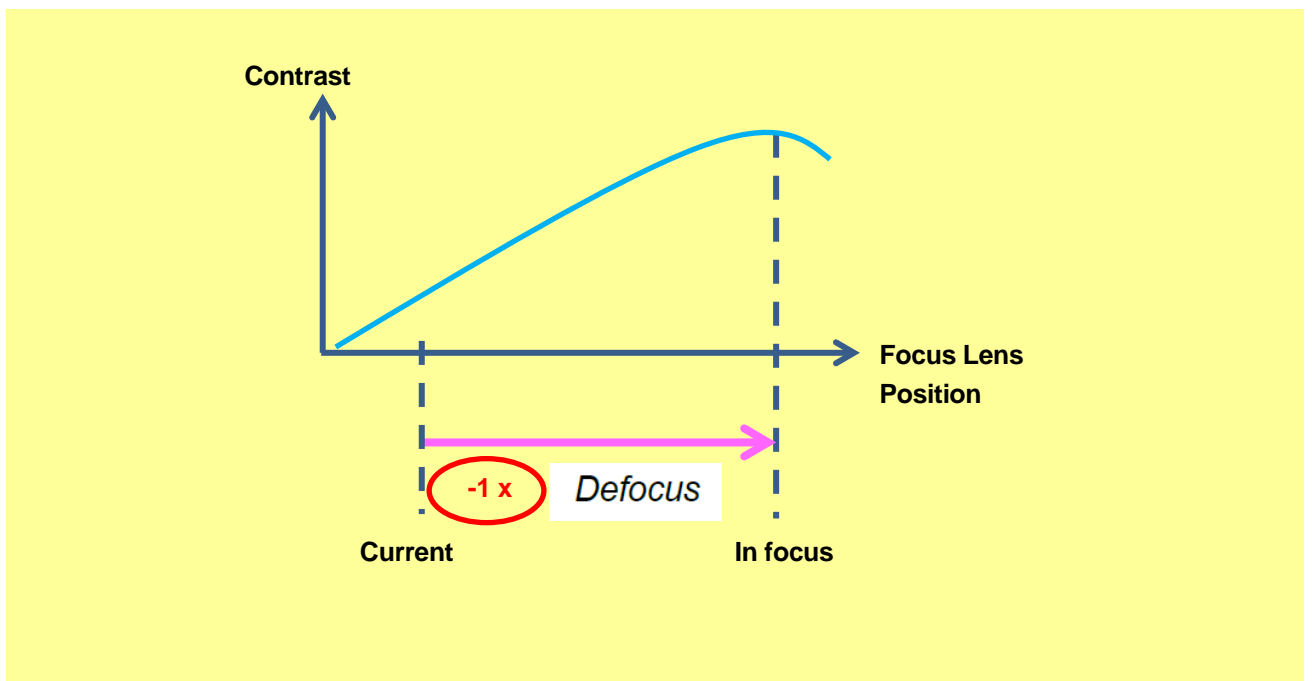


Figure 4-14 Note about Defocus

## 5. Error Codes List

Below is a list of the symbolic error names that are defined on the [PdLibGetDefocus](#) API.

The negative value for error code is significant only when the return value of the call indicated an error.

**Table 5-1 Error codes list**

Error names	Description
ENCWDDON	Determining defocus OK or NG is not compiled with by disable this functionality
EPDVALERR	Input error value of phase difference data
EINXSOI	XSizeOfImage Input out of range
EINYSOI	YSizeOfImage Input out of range
EINPDAFWX	PDAFWindowsX Input out of range
EINPDAFWY	PDAFWindowsY Input out of range
EINSO	SlopeOffset Input out of range
EINACS	AdjCoeffSlope Input out of range
EINSOXAK	SlopeOffsetXAddressKnot Input out of range
EINSOYAK	SlopeOffsetYAddressKnot Input out of range
EINVALIDISCONFJ	Invalid of Disable Confidence Judgement
EINVALIDISIHG	Invalid of Disable compensation relation with image height
EINDONTPN	DefocusOKNGThrPointNum Input out of range
EINDONXAK	DefocusOKNGXAddressKnot Input out of range
EINDONYAK	DefocusOKNGYAddressKnot Input out of range
EINDOP	DensityOfPhasePix Input out of range
ELDCL	Low DefocusConfidenceLevel



## 6. D\_PD\_ERROR\_VALUE

D\_PD\_ERROR\_VALUE is defined in PDAFLibrary.h.  
This value is different by each sensor type as below.

```
#define D_PD_ERROR_VALUE          (-64)
                                /* IMX230, IMX298, IMX330, IMX338 : -32 */
                                /* Other sensor type: -64 */
```

Please set this value according to your environment.

# SONY

SONY is a registered trademark of Sony Corporation.

Names of Sony products and services are the registered trademarks and/or trademarks of Sony Corporation or its Group companies.

Other company names and product names are registered trademarks and/or trademarks of the respective companies.