Customer Documentation pmd calibration SW

*by **pmd**technologies*

Abstract

This document describes the calibration software. The configuration and the execution of the program will be explained.

Table of Contents

List of Figures

List of Tables

1. Introduction

After acquiring the measurement data by the pmd3 software (see separate documentation), the calibration software processes this data and computes the compensation parameters. The first section will explain the configuration parameters.

1.1. Configuration

The following table shows the configuration parameters for the calibration software. See appendix for an example configuration.

Typically, these parameters are determined once for your specific camera module and do not need to be changed again. Only if you change the lens, for example, is it required to modify part of the parameters.

Section	Parameter	Unit	Type	Default Value	Description	Required?
Global	OutBaseDir	[]	string	%TEMP%	output directory	no
	debug	[]	boolean	false	generate debug plots (slows down the process)	no
Fiber box (wiggling)	FB.FiberLenghts	[m]	array of floats	-	calibrated fiber lengths (one-time calibration of fiber box itself using golden sample required)	yes, for fiber box based calibration
	FB.pattern_size	[]	int	[5,6]	row * colums fiber spots	no
Lens calibration	LC.pattern_center	[m]	float	-	[x,y,z] vector from the lens' focal point to the mid LED (the red one)	yes, very precise!!
	LC.SpotSize	[]	int	5	spot size for the detection algorithm	no
	LC.lens_parameter	[]	float	-	initial estimation of the lens parameters (only used for the detection and the fit's initial values) [fx,fy,cx,cy,k1,k2,p1,p2,k3]	yes
	LC.lens_model	[]	int	1	1: polynomial model, 2: fish-eye model (not yet supported during data processing, only calibration)	No
	pattern_mask	[]	int	-	LED pattern configuration (row- and column-wise); 1 means: IR LED should be there, 0 means no LED there; the red LED is not visible for the pmd camera (0) e.g. [1,0,1,0,1; ... 0,0,0,0,0,0]	yes
	pattern_spacing	[m]	float	-	The spacing of the drill holes in the aluminum plate (typically 30mm).	yes
Mask	Mask.max_pixel_count	[]	int	not active	if indicated, not more than this number of pixels will be used, i.e. pixels will be masked in the order of descending intensities up to this limit	no
	Mask.min_intensity	[ADC units]	int	not active	if indicated, pixels with lower intensity will be masked out	no
	Mask.max_inclination_angle	[degree]	float	not active	if indicated, pixels outside this maximum inclination angle will be masked out	no
Noise Parameters	NP.NoiseParameter_80320000Hz	[]	float	-	noise parameters for the specified modulation frequency	yes, except you do not want to work with global parameters (SW code change required!)
	NP.NoiseParameter_60240000Hz	[]	float	-	noise parameters for the specified modulation frequency	yes, except you do not want to work with global parameters (SW code change required!)
Temperature Drift	TD.TempCompensation_80320000Hz	[]	float	-	temperature drift parameters for the specified modulation frequency	yes, except you do not want to work with global parameters
	TD.TempCompensation_60240000Hz	[]	float	-	temperature drift parameters for the specified modulation frequency	yes, except you do not want to work with global parameters

Table 1: calibration software configuration parameters

1.2. Execution Flow

Figure 1 shows the execution flow.

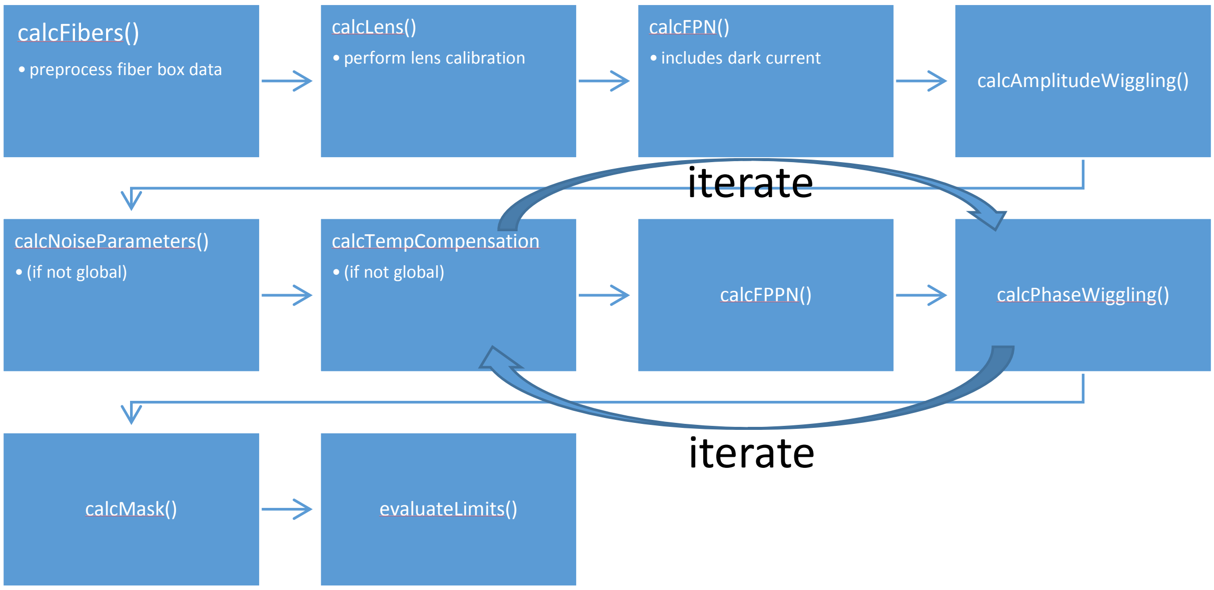

Figure 1: Program flow

1.2.1. Pre-process fibre data

The first step is to pre-process the fibre box data (except for LTS-based calibrations of course). This pre-processing can happen on a different station, if required, i.e. if you use two computers to acquire the data from the two calibration boxes, this step can compact the data and e.g. store it on the DUT.

1.2.2. Lens Calibration

Next, the lens calibration is calculated using the difference of the acquired data with the LEDs turned on and turned off. In this difference image, only the spots are present, so that the computation is easy. The reference world positions of the LED points is built from the pattern mask and the pattern spacing. Due to the manufacturing process of the aluminium plate, it is known that all these points reside within a plane and that the spacing is equal in horizontal and vertical direction. Thus, the pattern mask is a simple on-off description of the rows and columns (corresponding to the drill holes in the plate), where a value of 1 means that there should be an IR LED. In combination with the spacing, this results in the X-Y-coordinates of the LEDs, and in combination with the *pattern_center* parameter, the complete plane is known precisely relative to the entrance pupil of the time-of-flight lens.

1.2.3. Wiggling and FPPN

The noise parameters are usually supplied globally. Then an iterative process starts to compute temperature drift and phase wiggling parameters as well as the pixel offsets. The iterative process is required since the measurement data is always affected by all these effect simultaneously. This is different from the LTS-based calibration.

1.2.4. Pixel Mask

The last calibration step is to determine the pixel mask. The pixel mask serves to label pixels which are either defective or which deliver a bad depth measurement performance due to a variety of reasons, e.g. bad lighting when the field of illumination does not perfectly match the field of view. Masked pixels will be removed in the pmd processing chain, i.e. when the calibration is applied. They will not be part of the resulting point cloud.

The most straightforward and recommended method is to mask out all pixels beyond a certain inclination angle (see parameter *Mask.max_inclination_angle*). During the lens calibration, the inclination angles are calculated based on the given world coordinates of the LEDs. Now, typical lenses show a reduction in image brightness in the image periphery compared to the image centre (this effect is called vignetting).

This is one reason why it makes sense to allow only a certain maximum angle (see Figure 2) up to which this effect is tolerable. Another argument in favour of the maximum inclination angle is that it is known that the image periphery is more susceptible to influence from stray light or from multiple path propagation. Naturally, the maximum inclination angle must be within a range in which LEDs are available for the lens calibration as otherwise the lens model might even not be valid up to this angle.

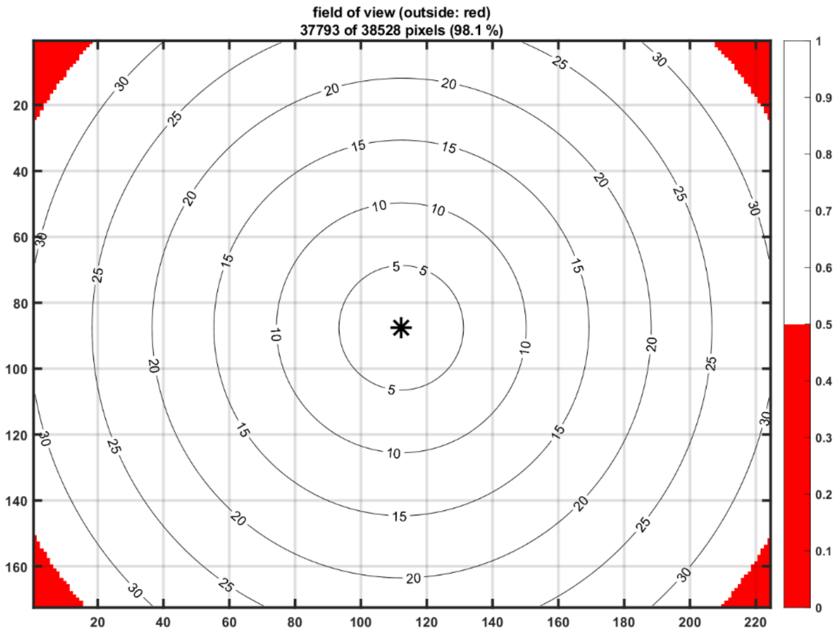

Figure 2: red areas show pixels masked due to the maximum allowed inclination angle

During all further processing steps, defective pixels will be included in the mask. For instance, pixels showing too less contrast or showing always-on or always-off behaviour (white / dark pixels), will be marked as defective.

A further option for the mask is to set a maximum number of pixels. This option simply sorts out a given number of pixels with lowest amplitude values. However, this option is deprecated and kept only for backward compatibility.

As the mask is computed as the last step, it does not influence the other calibration parameters. However, it is taken into consideration for the test limit evaluation (cf. next section) and for the visualization of the calibration results (e.g. colour bars are adjusted to the range of valid pixels).

2. Test Points

There are a lot of test points which are checked during the calibration. The following table lists them and explains their meaning.

Section	Parameter	Unit	Type	Hint	Description
Global	TP.valid_pixel_count	[]	int	ROI-specific	after mask application, this number of pixels should remain
	TP.FPN_mean	[ADC units]	int	[2000..2100]	mean of FPN must be within this range
	TP.DarkCurrent_mean	[]	float	[-0.1..0.1]	mean of dark current
FPPN	TP.PhaseSTD1_mean	[m]		[1-e4..2e-2]	mean / max of the phase of the FPPN acquistion
	TP.PhaseSTD2_mean	[m]		[1-e4..2e-2]	mean / max of the phase of the FPPN acquistion
	TP.PhaseSTD1_max	[m]		[0..0.1]	mean / max of the phase of the FPPN acquistion
	TP.PhaseSTD2_max	[m]		[0..0.1]	mean / max of the phase of the FPPN acquistion
	TP.Amplitude1_mean	[]		[400 .. 1400]	mean of the amplitude of the FPPN acquisition
	TP.Amplitude2_mean	[]		[400 .. 1400]	mean of the amplitude of the FPPN acquisition
	TP.Amplitude1_max	[]		[400 .. 1400]	max amplitude of the FPPN acquisition
	TP.Amplitude2_max	[]		[400 .. 1400]	max amplitude of the FPPN acquisition
Noise parameters	TP.PhaseNoiseRatio1_mean	[]	float	[0.8..1.2]	verifiy global noise parameters
	TP.PhaseNoiseRatio2_mean	[]	float	[0.8..1.2]	
Wiggling	TP.AmplitudeWigglingOffset1	[]	float	[0.6..1.2]	
	TP.AmplitudeWigglingOffset2	[]	float	[0.6..1.2]	
	TP.AmplitudeWigglingAmplitude1_max	[]	float	[0.01..0.1]	
	TP.AmplitudeWigglingAmplitude2_max	[]	float	[0.01..0.1]	
	TP.PhaseWigglingAmplitude1_max	[]	float	[0.01..0.1]	
	TP.PhaseWigglingAmplitude2_max	[]	float	[0.01..0.1]	
	FB.phase_noise_limits	[rad]	float	[1-e3..1e-2]	std of phase for the fiber spots
	FB.intensity_limits	[ADC units]	int	[20 1400]	spot intensity limits (pixels with different values are discarded)
	FB.amplitude_limits	[ADC units]	int	[20 1400]	spot amplitude limits (pixels with different values are discarded)
	FB.fiber_count_limits	-	int	[5,30]	minimum and maximum number of spots
Lens calibration	LC.fx_limits	[]	float	<+-5%	limits for fx
	LC.fy_limits	[]	float	<+-5%	limits for fy
	LC.cx_limits	[]	float	<+-5%	limits for cx
	LC.cy_limits	[]	float	<+-5%	limits for cy
	LC.projection_error_limits	[pixel]	float	<0.3	limits of the fit quality (the projection error)
	LC.rvec_x_limits	[rad]	float	<0.1	which degree of rotation should be tolerated?
	LC.rvec_y_limits	[rad]	float	<0.1	which degree of rotation should be tolerated?
	LC.rvec_z_limits	[rad]	float	<0.1	which degree of rotation should be tolerated?
Temperature drift	TP.TempCompensation1	[]	float	<0.001	lower / upper limit of the temp. compensation slope
	TP.TempCompensation2	[]	float	<0.001	lower / upper limit of the temp. compensation slope

Table 2: Test Points

The test point evaluation is not intended to be switched off! You must ensure to set appropriate limits!

3. Resulting Parameters

The following table shows the resulting calibration parameters.

Parameter	Unit	Type	Description	Further Documentation
CalibrationROI	[pixels]	int	This is the ROI which was calibrated. The offset [0,0] is relative to the ROI that Royale reads out.	SW-4-1-PR-Processing.pdf
date	[]	string	The date when the calibration was performed (not related to the date of the data acquisition)	
user	[]	string	login name of the user who performed the calibration	
serial	[]	string	module serial number (format 0000_0000)	
Sensor Serial	[]	string	Mira imager serial number (format 0000_0000_0000_0000)	
FPN	[ADC units]	int	fixed pattern noise; should be centered around 2^11	Cal-5-1-PR-Calibration.pdf
FPPN	[m]	float	array with FPPN for each modulation frequency	Cal-5-1-PR-Calibration.pdf
frequency	[Hz]	int	array with the modulation frequencies for the phase sequences (not for the gray-scale acquisitions)	G-1-1-AN-ToF-Working-Basics.pdf
lens_parameter	[]	float	[fx fy cx cy k1 k2 p1 p2 k3]	Cal-5-1-PR-Calibration.pdf
Mask	[]	logical	Logical true means that the pixel is flagged out e.g. due to the lens FoV or due to too low illumination.	Cal-5-1-PR-Calibration.pdf
NoiseParameter	[]	float	array with the noise parameters for each modulation frequency; for each frequency, there is a sub-array of four values (offset, slope, ??, reserved)	SW-4-1-PR-Processing.pdf
TempCompensation	[]	float		Cal-5-1-PR-Calibration.pdf
AmplitudeWiggling	[]	float	array with the amplitude wiggling parameters per modulation frequency; for each frequency, there is a set containing the offset, the harmonics, the amplitudes and the phases	Cal-5-1-PR-Calibration.pdf
PhaseWiggling	[]	float	array with the phase wiggling parameters per modulation frequency; for each frequency, there is a set containing the offset, the harmonics, the amplitudes and the phases	Cal-5-1-PR-Calibration.pdf

Table 3: Resulting calibration parameters

4. Appendix A – Example Configuration

Here you can see an example configuration:

```
debug = 1;
# OutBaseDir = 'F:' % output directory

##########################################################
#        FiberBox configuration                          #
##########################################################
FB.FiberLengths = [NaN, 3.1265, 3.0459, 2.1049, 3.7436, 3.5844, 2.7059, 2.0271, 3.2169, 2.4590, 3.3746,
3.3017, 2.2763, 3.6614, 2.5266, 1.8569, 2.6097, 1.3494, 1.5997, 1.4263, 1.7742, 2.7794, 2.9816, 1.6848,
2.3782, 3.8210, 3.4770, 1.9460, 2.1932, 1.5098];
FB.SD.noise_threshold = 15
FB.spacing_tolerance = 0.3
FB.pattern_size = [5,6];
FB.intensiy_limits = [20,1400]         % spot intensity limits
FB.amplitude_limits = [20,1400]        % spot amplitude limits
FB.phase_noise_limits = [0.001,0.07]   % fiber phase noise limits / rad
FB.fiber_count_limits = [5,30]        % valid fiber count limits

##########################################################
#        Lens calibration                                #
##########################################################

# LED pattern center position
LC.pattern_center=[0.21,-0.15,0.406];
LC.SpotSize = 5
LC.SpotDetector_type = 2
LC.SD.noise_threshold = 15
# polynomial lens model, K6
LC.lens_model = 1
LC.lens_parameter=[215.5,215.5,107.4,87.1,0.056471,-2.559501,0,0,4.753219];
LC.pixel_mapping_limits = [0,10] % limits for pixel mapping (in pixels)
LC.fx_limits = [200,225]
LC.fy_limits = [200,225]
LC.cx_limits = [110,130]
LC.cy_limits = [84,90]
LC.rvec_x_limits = [-0.1,0.1] % rotation vector limits
LC.rvec_y_limits = [-0.1,0.1] % rotation vector limits
LC.rvec_z_limits = [-0.1,0.1] % rotation vector limits
LC.projection_error_limits = [0,0.4] % limits of projection error (fit quality)

##########################################################
#        Mask generation parameters                      #
##########################################################
# Mask.max_pixel_count = 500
# Mask.min_intensity = 10
Mask.max_inclination_angle = 33

##########################################################
#        Noise parameters                                #
##########################################################
NP.NoiseParameter_80320000Hz = [0.0176, 4.22, 0.0122, 23.2]
NP.NoiseParameter_60240000Hz = [0.0169, 4.16, 0.0124, 24.1]

##########################################################
#        Temperature drift compensation parameters       #
##########################################################
TD.TempCompensation_80320000Hz = [0, 0.00113, 0, 0]
TD.TempCompensation_60240000Hz = [0, 0.00119, 0, 0]

##########################################################
#        Test Points (general test points)               #
##########################################################
TP.valid_pixel_count = [37000, 200000]
TP.DarkCurrent_mean  = [-0.1, 0.1]        % in DN/s
TP.FPN_mean          = [2000, 2100]       % in DN
```

```
TP.PhaseSTD1_mean    = [0.0001, 0.02]      % in m
TP.PhaseSTD2_mean    = [0.0001, 0.02]      % in m
TP.PhaseSTD1_max     = [0, 0.075]           % in m
TP.PhaseSTD2_max     = [0, 0.100]           % in m
TP.Amplitude1_mean   = [400, 1400]         % in DN
TP.Amplitude2_mean   = [400, 1400]         % in DN
TP.Amplitude1_max    = [400, 1400]         % in DN
TP.Amplitude2_max    = [400, 1400]         % in DN
TP.FPPN1_std         = [0, 0.1]            % in m
TP.FPPN2_std         = [0, 0.1]            % in m
TP.AmplitudeWigglingOffset1        = [0.60, 1.2]
TP.AmplitudeWigglingOffset2        = [0.80, 1.2]
TP.AmplitudeWigglingAmplitude1_max = [0.01,0.1]
TP.AmplitudeWigglingAmplitude2_max = [0.01,0.1]
TP.PhaseWigglingAmplitude1_max     = [0.01,0.1]
TP.PhaseWigglingAmplitude2_max     = [0.01,0.1]
TP.TempCompensation1 = [0 0.001];
TP.TempCompensation2 = [0 0.001];
TP.PhaseNoiseRatio1_mean = [0.85, 1.15]    % verify global noise parameters
TP.PhaseNoiseRatio2_mean = [0.85, 1.15]    % verify global noise parameters

# test_points.sharpness = [ 0 5 ]
# test_points.pmd_beam_profile_fname = <Config_Dir>/beam_profile_PicoU6.mat
# test_points.ueye_beam_profile_fname = <Config_Dir>/ueye_beam_profile_PicoU6.mat
# test_points.defect_pixel_threshold = 0.3
# test_points.defect_pixel_maxno = 3
# test_points.min_cdd_1per_distance = 2
```

Literature

1. R. Lange, “3D Time-of-Flight distance measurement with custom solid-state image Sensors in CMOS/CCD-technology“ Siegen, 2000.

Document History

Document title: pmd calibration SW – Cal-2-1-AN

Revision	Origin of Change	Submission Date	Description of Change
0	OLo	2016-02-19	New Documentation
1	OLo	2016-05-03	New section summarizing the resulting output parameters.
2	OLo	2016-05-18	More details on the LED pattern
3	OLo	2016-05-20	More details on the pixel mask