Calibration Basics

*by **pmd**technologies*

Abstract

The purpose of this document is to introduce possible calibration steps that will enhance the performance of the 3D cameras.

Table of Contents

List of Figures

1. Introduction

The purpose of this document is to introduce possible calibration steps that will enhance the performance of the 3D cameras. Each subsection describes a different adjustment step and contains easy-to-follow calibration instructions. The involved steps are

- **Intrinsic Lens calibration**: The phase and distance calculation of each pixel in a pmd sensor delivers the radial component of an object in a spherical coordinate system. In order to transform this information to the more practical Cartesian xyz- three-dimensional position for each pixel, we need to know the individual direction each pixel is looking at. This direction is given by the lens system of the camera.
- **Wiggling**: The most straight forward phase calculation equation (arctan) which should be used for 3d data calculation is based on sine wave mathematics. In real world however, both the light pulse shape as well as the correlated internal pmd signal is not an ideal sine wave. This fact leads to systematic fixed errors, which are easy to calibrate.
- **Temperature drift**: Some parts of a pmd camera for optical and electrical modulation signal generation are temperature sensitive. In first order this leads to a temperature dependent signal delay, so that the measured phase changes with temperature. One easy way to correct these errors is to measure the temperature with an internal sensor and correct the drift with a look-up-table (LUT) or in a linear approximation.
- **Offset and FPPN**: The signal shapes and the initial phase delay between emitted light and pmd pixel modulation are influenced by various system parameters. Depending on the required absolute accuracy, each pixel value needs to be corrected by a pixel and sensor individual offset level. This fixed sensor pattern of distance offsets in the correlation data needs to be calibrated.

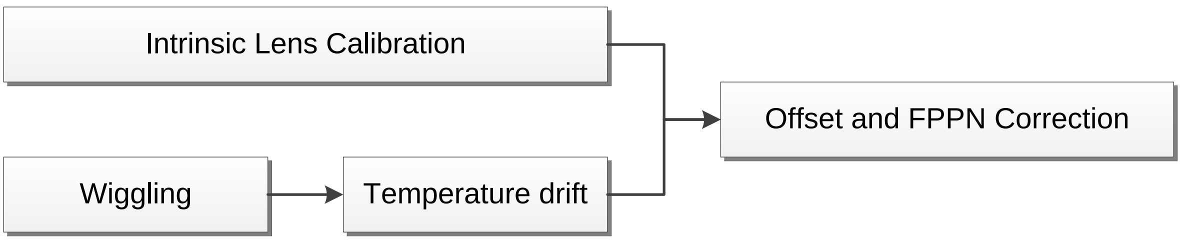

Figure 1 Calibration flow diagram

The sequence in which the calibration datasets need to be processed is not completely arbitrary. Figure 1 shows the involved steps:

- **Intrinsic Lens Calibration**: The described calibration method requires only the 2D amplitude data and is independent from phase accuracy and other calibration steps.
- **Wiggling and Temperature drift**: Wiggling has to be determined prior evaluation of the temperature drift data.
- **Offset and FPPN Correction**: The described method is based on the measurement of a flat surface. In order to gain the phase offsets from this, the correct real distance of the light rays via the acquired lens calibration has to be known and the temperature drift has to be eliminated.

2. Intrinsic Lens Calibration

Every pixel of a pmd sensor acquires data from a subset of the real world in 3D. The cross section of this subset grows with the distance of the scene to the sensor and thus we may speak of a pyramidal or conical geometry in world coordinates which every pixel observes. These observed structures are commonly very thin. This is why we will refer to the region observed by a pixel as viewing ray and model it by a straight line in 3D space.

The distances measured by the pmd sensor are given as scalar values. The distance data of the whole sensor can be processed in any common 2D image processing way, but for accurate 3D data processing (scene reconstruction, object size measurement, 3D distance measurement, computation of volumes of objects, etc.) geometric relations between the single pixels must be established, i.e. the angles and orientations of the viewing rays must be known. Two constraints are assumed:

- all viewing rays of all pixels are straight
- all viewing rays meet in one single point (the focal point)

With known geometric parameters for each pixel (i.e. viewing rays) arbitrary points within the Cartesian world coordinate system can be projected onto the 2D sensor coordinate system and vice versa. These coordinate system transformations are used for computation of 3D points from measured distances and for reprojection of world coordinate points onto the image plane. The process to determine the geometric relations of all viewing rays of a pmd sensor is called intrinsic lens calibration.

If the mapping of the system optics is not distorted, straight lines in 3D space are projected to straight lines in 2D image space. For many data processing algorithms it is absolutely necessary to obey this constraint in order for them to work properly. In pmd range imaging another constraint adds to this rule: a straight line measured in 2D image space may not necessarily represent a straight line in 3D coordinates. This means in case the applied optics show distortion the viewing rays need to be computed in such a way that the (distorted) 2D image acquired by the sensor is mapped to an (undistorted) 3D scene.

Most pmd reference designs operate with optics showing a certain distortion. These distortion effects have been investigated in detail both for common 2D imaging as well as for pmd sensors and various methods have been presented to compensate them. Correction of distortion often for 2D cameras is referred to as “intrinsic calibration” or “camera calibration”. Within this document we will use the term “intrinsic lens calibration” and the parameters determined by the calibration procedure are called “intrinsic lens (camera) parameters”.

Today’s common approach to intrinsic lens calibration is based on a projection- or camera-matrix with four or five additional distortion parameters. As projection model for the camera system the pinhole camera model is used where all viewing rays from all pixels of the sensor meet in a common point. This common point is denoted as the “focal point” of the camera.

For more details about algorithms and principles of intrinsic camera calibration the interested reader is encouraged to have a look at the following online resources:

http://docs.opencv.org/trunk/modules/calib3D/doc/camera_calibration_and_3D_reconstruction.html

http://www.ics.uci.edu/~majumder/vispercep/cameracalib.pdf

http://en.wikipedia.org/wiki/Camera_resectioning

http://en.wikipedia.org/wiki/Pinhole_camera_model

2.1. Parameters for intrinsic lens calibration

The projection vectors for pmd sensors which are derived by intrinsic lens calibration are computed from 9 parameters:

- focal length: (f_x, f_y) these values are commonly the same
- principal point: (c_x, c_y) pixel-position of the principal point within the image
- radial distortion: (k_1, k_2, k_3)
- tangential distortion: (p_1, p_2)

The focal length describes the distance of the focal point from the image plane. The principal point is the position within the image plane where the viewing rays from the camera intersect orthogonally with the image plane. Assuming a symmetric setup of camera chip and optics and perfect manufacturing conditions the principal point should be right in the middle of the sensor. Due to mounting tolerances for real-world sensors this is not always true.

Depending on the applied distortion model other/additional distortion coefficients may be used. When working with pmd sensors the proposed ones have proven to be accurate enough.

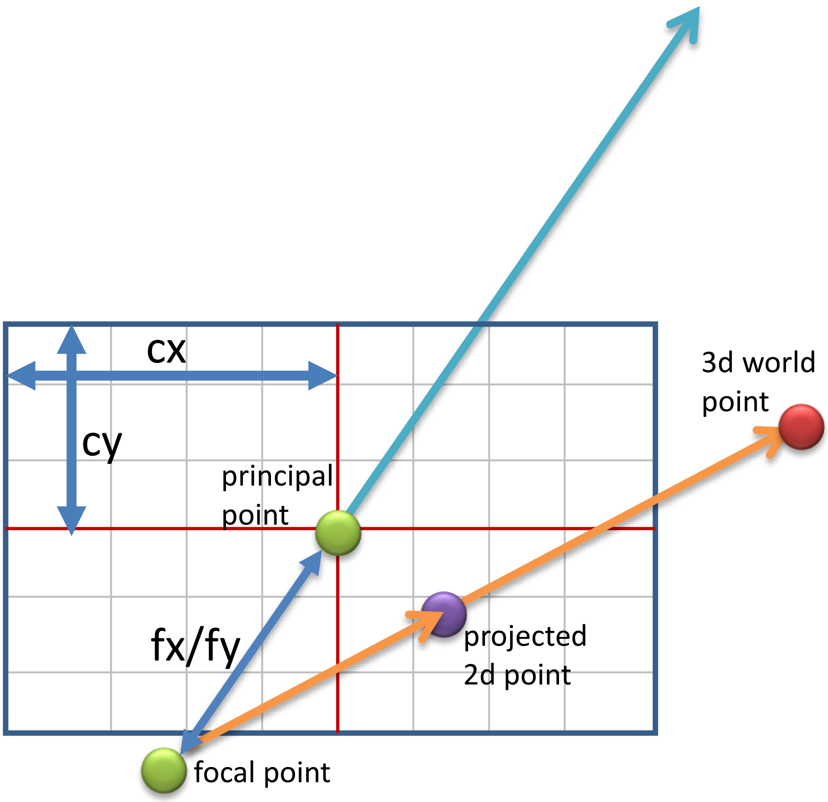

Figure 2 Focal point, principle point and projection

2.2. Projection from one coordinate system to another

As mentioned in the introduction of this section intrinsic calibration provides parameters and formulas to transform a scalar distance value from a 2D image position to a Cartesian 3D world point and vice versa.

2.2.1. From Cartesian 3D space to image plane

The camera is positioned in 3D space such that its focal point coincides with the origin of the world coordinate system and it is looking along the z-axis in positive direction with the y-axis pointing upwards. Now the pixel positions (u, v) which correspond to 3D point (x, y, z) are computed as

$$\mathrm{x}' = \frac{\mathrm{x}}{\mathrm{z}}$$

$$\mathrm{y}' = \frac{\mathrm{y}}{\mathrm{z}}$$

$$r^2 = x'^2 + y'^2$$

$$x'' = x'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x' y' + p_2(r^2 + 2x'^2)$$

$$y'' = y'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 x' y' + p_1\left(r^2 + 2y'^2\right)$$

$$u = f_x x'' + c_x$$

$$v = f_y y'' + c_y$$

2.2.2. From image plane to 3D space

The formulas given above are not analytically invertible, so projection from 2D image space to world coordinates in 3D must be done by an optimization approach. Fortunately the number of pixels per pmd sensor is finite, thus for each pixel a unit vector pointing into the direction of the corresponding viewing ray can be computed and stored in memory. This only has to be done once for every pixel and essentially this process of computing the unit vectors is the main part of intrinsic calibration. After intrinsic calibration, for each pixel i with scalar distance value d_i and unit vector (v_i^x, v_i^y, v_i^z) the corresponding Cartesian 3D coordinates (x_i, y_i, z_i) are computed as

$$x_i = d_i v_i^x$$

$$y_i = d_i v_i^y$$

$$z_i = d_i v_i^z$$

2.3. 2D distortion vs. 3D distortion

Undistortion of range imaging data can be done in two ways:

- the distances can be remapped such that the 3D projection will be rectangular
- the unit vectors can be aligned such that they mimic the actual distortion of the system optics

2.3.1. 2D image undistortion

When working with 2D images distortion/undistortion is simply defined as a re-mapping of image values from one pixel-position to another. For every pixel position p_i^u in the undistorted image (which has to be computed) another pixel position p_i in the originally acquired image is given. The values of p_i are not necessarily integer but may point to some sub-pixel position of the original image. If we are assuming bilinear interpolation when computing the undistorted value, we always use a set of four pixel values from the original image to compute one undistorted value.

Due to the fact that both, the original (distorted) and the computed (undistorted) image will have rectangular shape, in the undistorted image some information will be lost or some pixels will be introduced for which no valid values can be interpolated. The amount of pixels which will be lost or invalid depends on the degree of distortion and the size of the undistorted image.

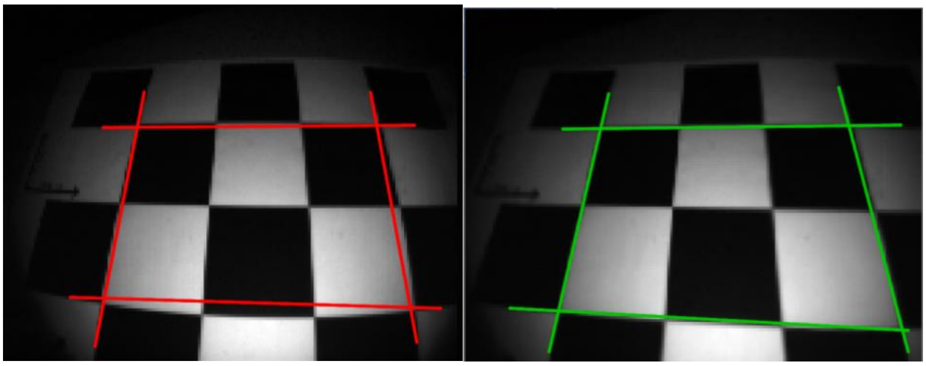

Figure 3 Distorted (left) and undistorted (right) 2D images

2.3.2. 3D range image undistortion

In range imaging undistortion is usually applied different to common 2D image undistortion. While the original 2D amplitude and distance image datasets are left unchanged and no bilinear interpolation is done to compute rectified images, the unit vectors are aligned with the viewing rays and thus an undistorted 3D representation of the 2D data is computed.

In detail this means:

- The 2D data of the sensor is left unchanged, distorted amplitude images remain distorted
- The 3D coordinates computed from these distorted 2D images deliver a 3D scene which will be undistorted (i.e. straight lines are straight again)

On the other hand, undistortion may as well be done analogously to common 2D image undistortion. In this case first the original data has to be undistorted using interpolation techniques as mentioned and afterwards the interpolated/undistorted distances are projected in 3D space. The corresponding unit vectors are different from the ones computed for the first undistortion approach.

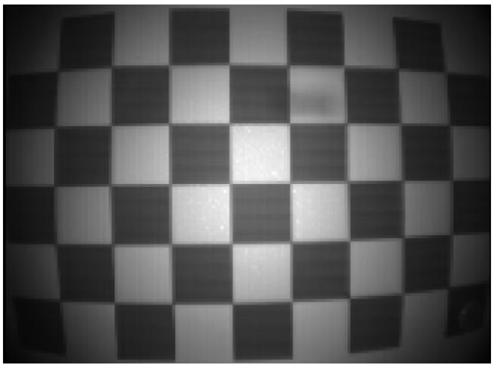

Figure 4 Distorted 2D amplitude image.

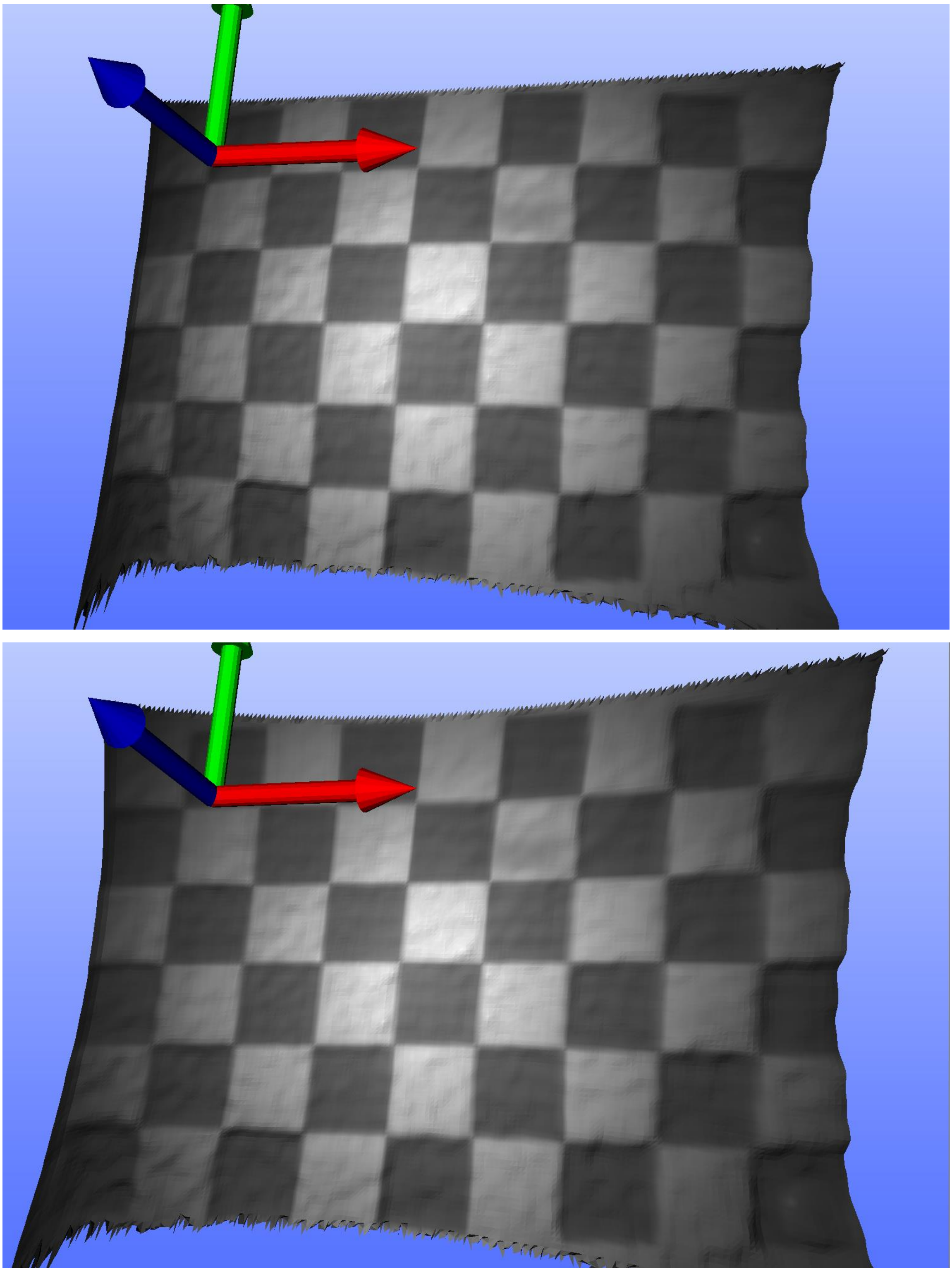

Figure 5 Distorted (above) and undistorted (below) 3D data. The effect is mainly seen in the lines towards the edges of the plain. Especially on the top and the left edge the concave shape from the undistortion can be seen in the lower image.

While the first approach directly uses the distance data as delivered from the sensor, the second method requires an additional interpolation step. The first approach (which is the only one currently supported by pmdtechnologies) has several advantages over the latter one:

- If a pixel has invalid distance values, it is flagged. When interpolating pixels which may have different flag values, results can be unreliable
- When interpolating different flag values, the resulting flag value can't be determined accurately. New (interpolated) flags would be needed
- Interpolation directly involves some kind of lateral filtering which might not be wanted by the operator of the sensor
- Interpolation takes additional computational time/resources

It is important to keep in mind, that correctly calibrated and undistorted range imaging systems will still deliver distorted 2D amplitude and phase images (see section 2.3.1).

2.4. Computation of intrinsic calibration

In order to compute the 3D coordinates for a 2D distance value only the corresponding unit vector is needed. As these unit vectors for a pmd reference design cannot be directly computed given the system characteristics, the intrinsic parameters need to be estimated from which then the vectors can be derived.

For computation of the intrinsic parameters, feature constraints between feature points in 2D image space and the corresponding points in 3D world coordinates are needed. These constraint sets are commonly acquired with the sensor system using planar checkerboard patterns which show a very high contrast between the black and the white tiles), but a more suitable one-shot method is shown in Cal-12-1-AN-SeriesCalibration..

Computation of intrinsic camera parameters is a highly non-linear optimization task which may produce significantly different results for one single sensor system, depending on the feature constraints passed to the algorithm. This means that a simple method to validate the intrinsic calibration parameters is necessary to check if the computed unit vectors may be reliable or not.

For every feature constraint which is used for the optimization procedure, a 2D pixel position as well as a 3D world coordinate is given. Like mentioned before, each point from one coordinate system may be transformed to the other. In fact this is continuously done during the calibration procedure.

After calibration every 3D coordinate point can be reprojected onto the image plane using the formulas given earlier. If the calibration is perfect, the reprojected 2D point would exactly be the feature which was initially detected and used as input for the optimization procedure. In this case the pinhole camera model with the applied distortion formulas would model the physical setup of the camera perfectly. Yet this is never the case. On one hand the mathematical camera model applied does only approximate the real physical system and on the other hand the initial feature detection of the 2D features may be imperfect such that the artificially set 3D positions may have slight errors. The shift for a 2D feature point between its initially detected 2D position and the reprojection from the corresponding 3D coordinate is called reprojection error. This error is usually given in pixel. Considering the fact that sensor chips may have different pixel sizes, reprojection errors cannot necessarily be directly compared over several different camera modules if they use different pixel sizes.

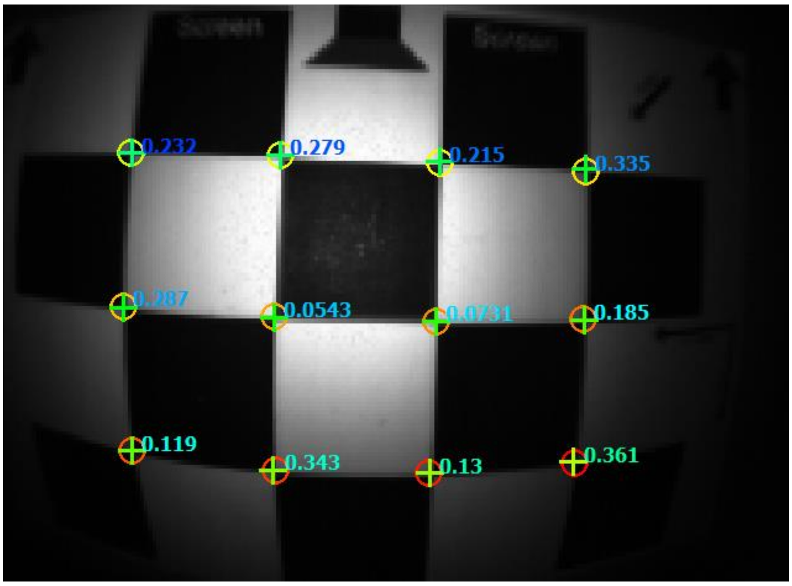

Figure 6 Example of reprojection error.

To come up with an easy and reliable estimate of the quality of the intrinsic calibration, the mean reprojection error over all feature correspondences which were used to compute intrinsic lens calibration is computed.

Acceptable reprojection errors of pmd sensor systems are in the range of 0.1 to 0.5 pixels

3. Wiggling

The so-called Wiggling results in systematic distance offsets between the true distance and the calculated distance using the atan Formula.

$$\text{Wiggling} = \text{measured Distance} - \text{real Distance} \tag{1}$$

This wiggling is a periodic function of the distance and oscillates with a wavelength that is one fourth of the unambiguous range. This effect is explained in more detail in G-1-1-AN-ToF-Working-Basics.

The wiggling can be measured by using a translation stage, where the distance between camera and the object (typically a flat white wall) can be varied. Mass-production capable setups are described in Cal-12-1-AN-SeriesCalibration. Figure 7 shows the obtained wiggling for one camera over a certain distance range. The data was acquired using a modulation frequency of 30 MHz and by using a fixed exposure time of 2000 µs. With increasing distance the signal amplitude is reduced which results in larger distance errors as indicated by the blue error bars.

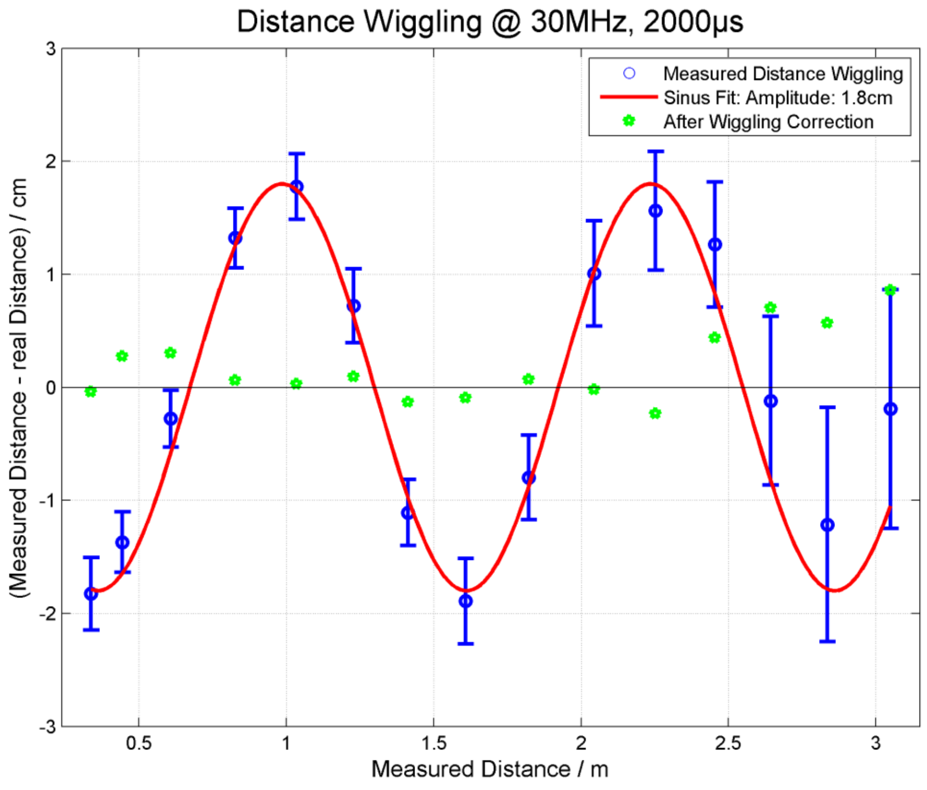

Figure 7 Distance Wiggling with Correction

At 30 MHz modulation frequency the unambiguous range is 5 m and hence the wiggling wavelength should be 1.25 m in accordance with the measurement. By using the measured wiggling which is a function of the measured distance the true distance can be calculated for each pixel as follows:

$$\text{real Distance} = \text{measured Distance} + \text{wiggling(measured Distance)}. \tag{2}$$

Depending on the required accuracy the wiggling dependence can be either stored in a LUT or alternatively expressed by an analytical function. The second approach is shown by the red line in Figure 7, which is a simple sinusoidal fit of the wiggling. The remaining distance error for this correction is shown by the green data points. By this simple fit which is fully characterized by only two parameters (amplitude and phase) the obtained accuracy is better than 5 mm.

The calibration process can be summarized by the following steps:

1. Mount camera on a translation stage measuring towards at a flat white wall (or vice versa)
2. shield the setup from external influences (background light),
3. select ROI on the optical / geometrical axis,
4. vary the camera to wall distance and collect the ROI- and time-averaged distance data (make sure the camera is at thermal equilibrium by operating at a fixed frame rate and constant integration times and turning it on for some minutes before taking the data (see also section 4),
5. from the relation of wiggling vs. measured distance create a LUT or obtain the analytical fit parameters.

4. Temperature Drift

Any temperature change of the illumination unit might lead to a different global offset in the distance image. This is due to the fact that the internal capacitance of the illuminator depends slightly on its temperature. This leads to a shift of the signal phase resulting in a global distance offset for the complete image. To correct for this drift a temperature sensor is positioned close to the illuminator to measure its temperature.

The distance drift can be caused by an ambient temperature change or by the current induced temperature rise inside the illuminator. An example of the second cause is shown in Figure 8 where the measured distance as well as the temperature of the illuminator is shown over time for a camera operating with 55 fps and 1000 µs exposure time.

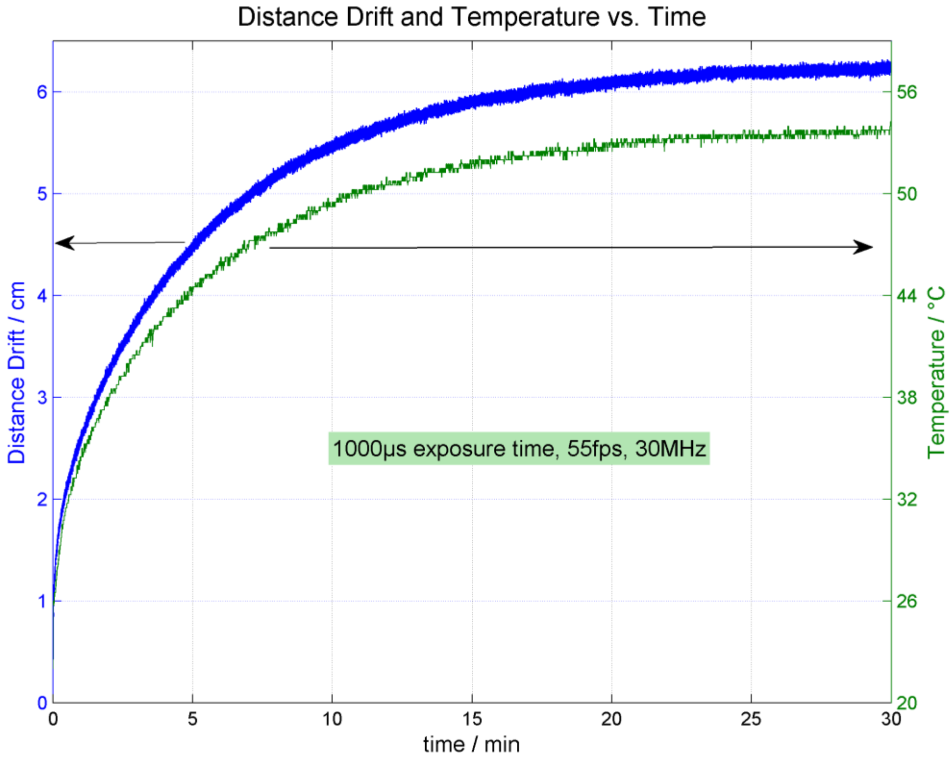

Figure 8 Distance Drift and Temperature increase over Time

Since the distance drift scales linearly with the temperature over the relevant temperature range, it can be easily compensated as is shown in Figure 9 with a correction coefficient of 0.19 cm/K. By this compensation the drift can be reduced to well below 1 cm.

The advantage of directly measuring the temperature is that ambient temperature changes as well as current induced self-heating of the illuminator can be compensated.

After changes that affect the current induced heating of the illuminator, i.e. a change of the integration time or the frame rate, small drifts might still occur within short timeframes of a few seconds, since the measured temperature lags behind the actual illuminator temperature.

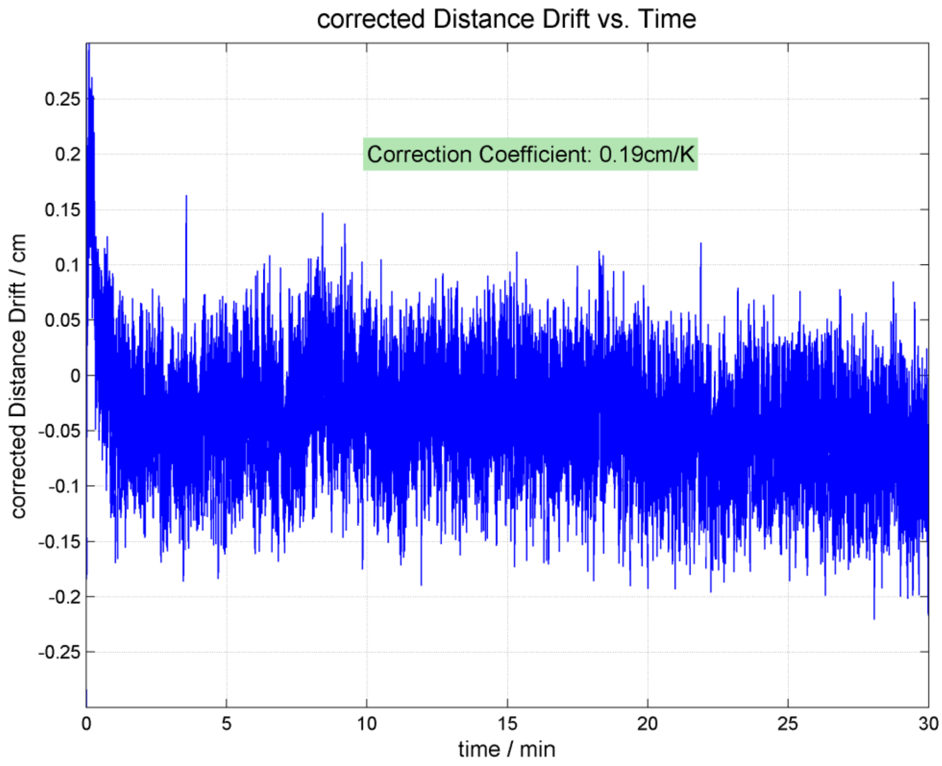

Figure 9 Temperature corrected Distance

It is important to note that the temperature drift correction works best if the calibration setup resembles the intended use case as close as possible. If for instance the calibration was done with the uncovered stand-alone camera module whereas in the application the camera is integrated into an existing setup, the thermal properties will be changed and hence the temperature correction might not be ideal.

The temperature analysis is usually done for a series of cameras. Typically, the coefficients for different cameras are quite similar (difference smaller than 10 %). Thus the resulting correction coefficients can be averaged to yield a global correction value for all cameras.

The usual procedure is as follows:

1. Mount camera at a fixed distance to a flat white wall,
2. shield the setup from external influences (background light),
3. select ROI on the optical axis,
4. vary the temperature of the active illuminator by current induced heating (high frame rate and integration time) and/or by changing the ambient temperature,
5. simultaneously measure the ROI-averaged distance and the temperature over a certain time range,
6. obtain the thermal correction coefficient from the relation distance drift vs. temperature,
7. repeat previous steps for a set of cameras,
8. use averaged coefficients for global correction.

5. Offset and FPPN Correction

The raw phase value computed with the atan formula needs to be scaled by the maximum measurement range to derive the raw distance value. Even after other systematic calibration steps like wiggling and temperature compensation the resulting metric distance value will have a fixed offset towards the actual distance of the sensor to the object. The computed distances depend on several properties of the system. Signal propagation times, individual sensitivities of pixels and illumination waveforms from the active light source contribute to the measured phases and thus to the computed distance values.

Generally speaking every single pixel will have a pixel individual distance offset which must be subtracted from the raw distance value. The mean value of all pixel individual offsets is one global scalar distance offset which is simply referred to as “offset”. The difference between the offset and the pixel individual offset can be different for every pixel. It can be seen analogously to the fixed pattern noise (fpn) of common 2D cameras. Consequently this difference is called fixed-pattern-phase-noise (fppn).

5.1. Application of offset correction

For a pmd sensor system with a global offset o, raw distance value d_i^r for pixel i and a fppn-value f_i for the same pixel, the offset-corrected distance d_i is computed as

$$d_i = d_i^r + o + f_i$$

This is done for all pixel positions to obtain an offset- and fppn-corrected distance image which will lead to a correct 3D representation of the observed scene.

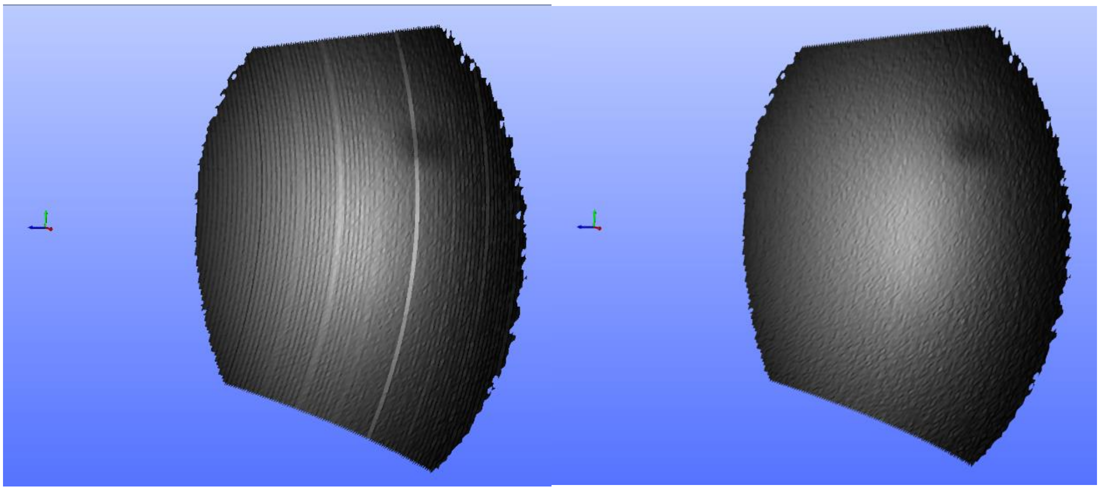

Figure 10 Raw datasets from a wall without offset applied. Left: no fppn correction is done, right fppn correction is applied. The mean distance is around 2.5m which is far too large and thus a bowl like shape results.

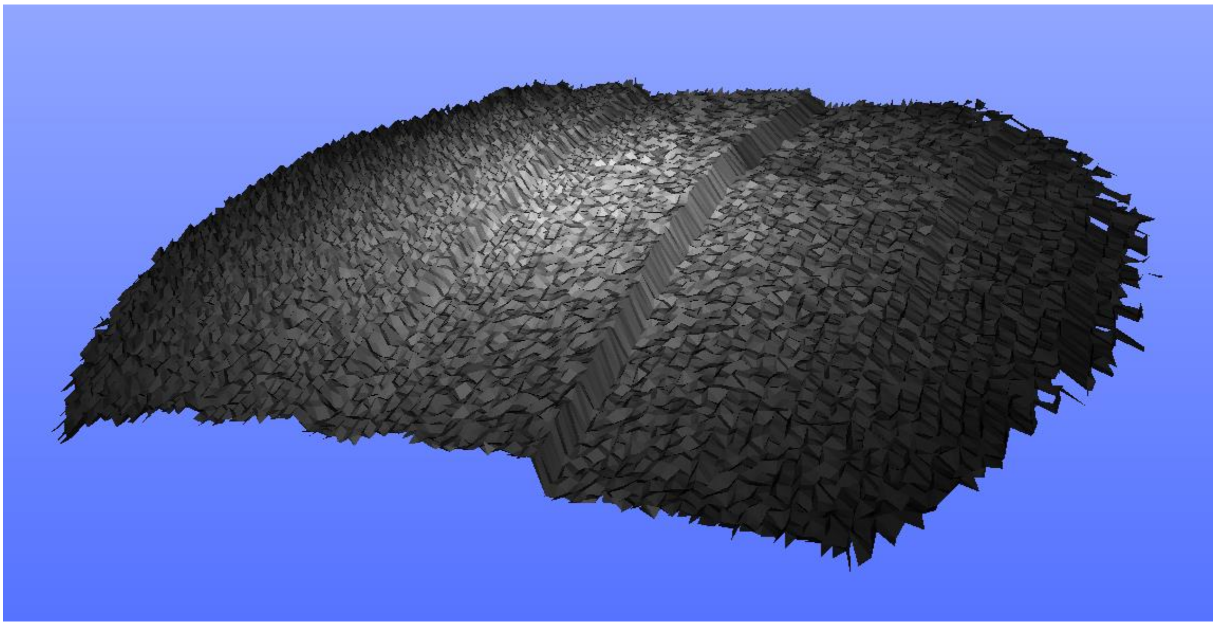

Figure 11 The same scene like above with global offset *o* applied but not pixel-individual fppn correction. The mean distance is arount 0.29m which is the actual distance of the wall to the sensor.

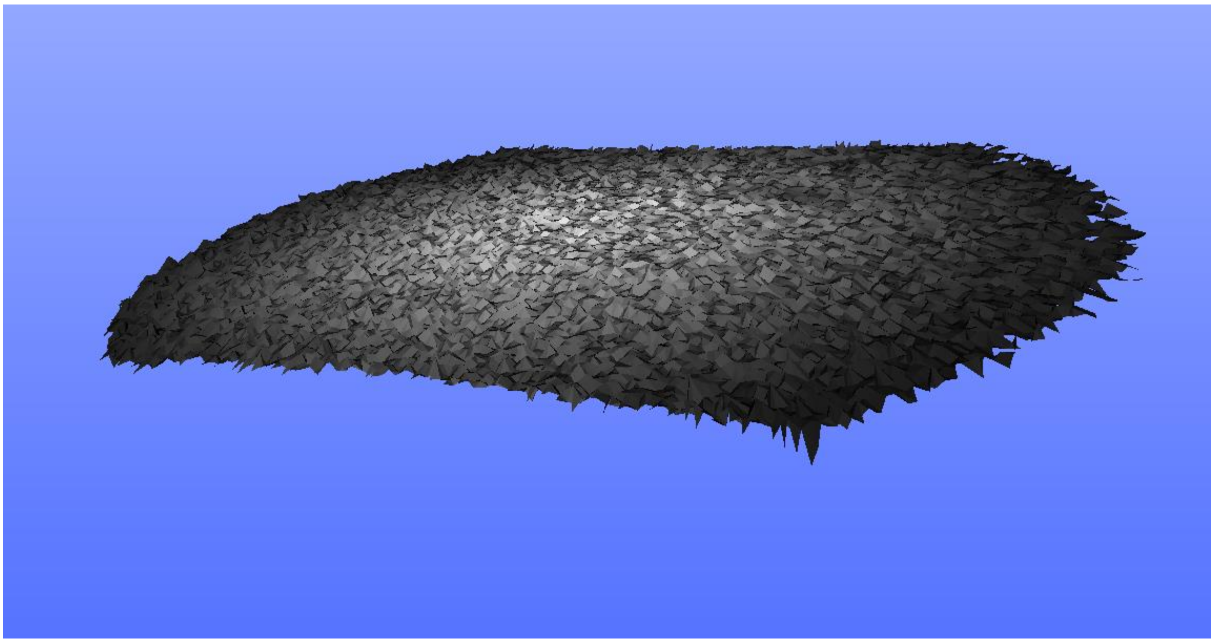

Figure 12 The above dataset with offset and fppn correction applied. The steps from the image above are removed.

5.2. Computation of offset correction

Before o and f_i can be applied to the raw distance data their values must be computed. This is done by subtraction of ground truth distance values from measured values which have been acquired with the sample that is going to be calibrated. The ground truth distance value d_i^g for pixel i is obtained by placing the sensor in a known distance from a homogenously reflecting wall. Given the unit vectors (which are explained in the intrinsic calibration section), the distance d_o of the sensor to the wall and assuming the system is looking orthogonally at the surface, d_i^g can be computed as

$$d_i^g = \sqrt{\left(d_o * v_i^x\right)^2 + \left(d_o * v_i^y\right)^2 + \left(d_o * v_i^z\right)^2}$$

The pixel-individual offset o_i (which incorporates the global offset o) is computed as the difference between the measured raw distance d_i^r and the ground truth d_i^g and is equal to the sum of the offset and the fppn of the pixel

$$o_i = d_i^r - d_i^g = o + f_i$$

With o being the mean value of all pixel-individual offsets o_i

$$o = \frac{1}{n}\sum_{i=1}^{n} o_i$$

The fppn f_i for pixel i is computed as

$$f_i = o_i - o$$

The data sequence which is used for getting the d_i^r values should be recorded with a reasonable integration time and averaged over a sufficient number of frames in order to minimize the temporal noise of the system such that the offset data becomes most reliable. Both integration time and the number of frames needed for the offset sequence may differ depending on the reference design which is being calibrated.

Document History

Document title: Calibration Basics – Cal-5-1-AN

Revision	Origin of Change	Submission Date	Description of Change
0	BAI	2016-06-27	New Application Note