cwTOF imaging systems are based on the active illumination of the target with modulated light. If two or more such systems operate in the same space, the illumination of one system can interfere with the other system. The result is wrong distance data, at least sporadically.

This paper describes a way to detect interference on a pixel basis to avoid wrong distance values. Detection and correction is implemented in software on the application controller.

by Beat De Coi, Silvio Honegger

ESPROS Photonics AG

List of contents

1. Introduction 1
2. Disturbance detection 1
3. Operation in interference situation 3
3.1 Using the previous value 3
3.2 Frequency hopping 3
3.2.1 Using the external modulation input 3
3.2.2 Integration time versus beat frequency 3
3.2.3 Using DRNU correction 4
3.3 Disturbance filtering 4
3.3.1 Memory requirements of interference filtering 5
4. Summary 5
5. Document change history 5

1. Introduction

3D TOF cameras operate with active illumination. They emit modulated light to the scenery and detect the reflected light. The light can be modulated continuously for cwTOF or pulsed for pTOF. By measuring the phase-shift with cwTOF or the time of flight with pTOF systems, the object distance can be obtained. If there are multiple cameras operating in the same area, interference of such light modulation may or will occur, called multi-camera interference (MCI). This can lead to wrong distance measurement results of one or more pixels.

In the following, we analyze cwTOF interference based on ESPROS' TOF imagers epc611, epc635 and epc660 (herein epc6xx).

In order to get a broader picture of artifacts in TOF systems, please refer also to the application notes AN10 and AN12.

2. Interference detection

TOF cameras can be disturbed while they are imaging on the same object or towards each other. Especially if the modulation signals have the same modulation frequency and wavelength. The starting point is to detect interference first, which may lead to wrong distance data.

Distance imaging over four DCS (refer to Figure 1) uses the two symmetric DCS pairs DCS0/2 and DCS1/3 which correspond to demodulation angle at 0°/180° and 90°/270°.

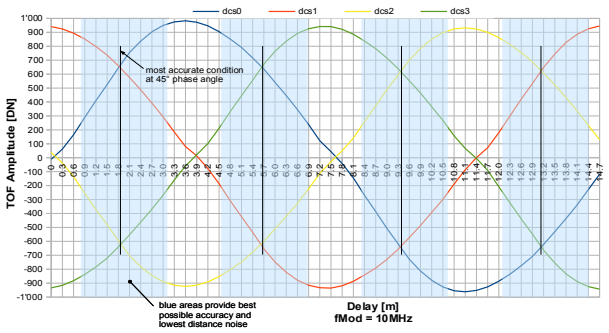

Figure 1 TOF amplitude per DCS (measurement data)

In theory, the sum of the values of such a symmetric pair should always be zero. If the sum is not zero, the distance calculation will not be correct, because the pair is not symmetrical. The source of this error can be either interference or motion blur. In this paper, we just focus on interference.

Due to imperfectness of the pixel and the readout structure, e.g. dark-signal non-uniformity (DSNU), photon-response non-uniformity (PRNU) or non-linearity, this summation in reality is typically not exactly zero, even without interference. Thus, a certain tolerance of deviation around zero has to be accepted. Only, if the sum of a DCS pair exceeds a certain threshold, the pixel distance data should be discarded. Figure 2 shows the suggested threshold which should be set by using the epc6xx imagers.

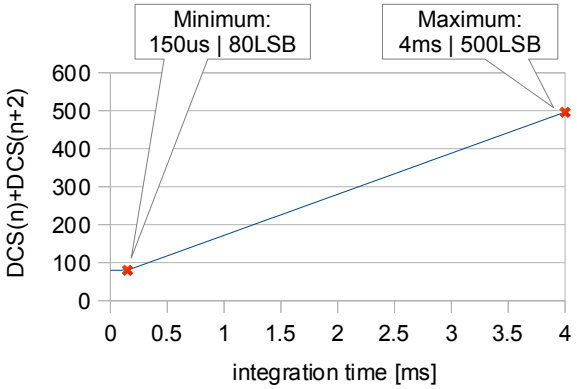

Figure 2: Detection threshold to activate the interference flag

This curve of the TOF system has been found by using the calibration setup in Figure 3 (refer to the application note AN10). The threshold needs probably to be adjusted depending on the noise of the power supply, the jitter of the illumination and process variations of the chip. Thus, it is best to obtain the threshold values during the camera calibration process.

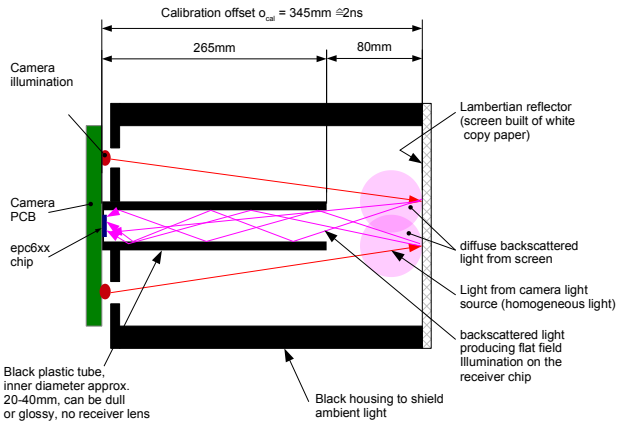

Figure 3: Proposed calibration setup

The mathematical description of the slope of the curve in Figure 2 is

$$\text{gain}_{\text{thres}} = \frac{(500-80)\text{LSB}}{(4000-150)\text{us}} \quad (1)$$

the curve description:

$$\text{sum}_{\text{thres}} = (t_{\text{int}} - 150\text{us}) * \text{gain}_{\text{thres}} + 80\text{LSB} \quad (2)$$

```
if ( sumthres < 80LSB )
   sumthres = 80LSB
endif
```

The result $\text{sum}_{\text{thres}}$ from the function can now be used to decide if there is an interfered DCS in a symmetric DCS pair. The code to take the decision is as follows:

(3)

```
sum_DCS02x, y = DCS0x,y + DCS2x,y
sum_DCS13x, y = DCS1x,y + DCS3x,y

if (sumthres > abs(sum_DCS02x, y ) )
   "DCS0/2x, y are valid"
else
   "DCS0/2x, y are not valid"
endif

if (sumthres > abs(sum_DCS13x, y ) )
   "DCS1/3x, y are valid"
else
   "DCS1/3x, y are not valid"
endif
```

If one of these pairs is not valid, the distance calculation will result in a wrong distance and thus, this result has to be discarded.

The following measurements are captured in a modified calibration setup. Camera 1 takes flat-field images through the black plastic tube against the white closure head, like Figure 3 (refer to application note AN10). Camera 2 points on the same spot as camera 1 and is acting as a disturber camera using the same modulation frequency. The diagram Figure 4 shows 50 consecutive unfiltered raw distance measurements of camera 1 which was disturbed by camera 2. During the measurement, the disturber camera was capturing images with an integration time of 2ms.

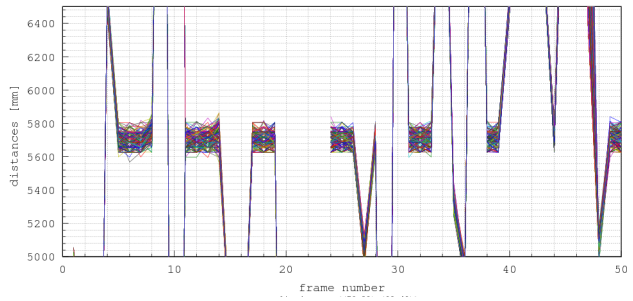

Figure 4: Multi camera without disturbance detection. Valid distances are between 5600 and 5800 mm. The spikes up and down are from interfered acquisitions.

As it can be seen in Figure 4, several measurements report a stable raw distance of approximately 5'700 mm whereas some others are completely off. This is represented by the sharp spikes up and down which is equal to longer or shorter distances.

By applying algorithm (3), wrong data points are detected and can be removed (shown in Figure 5):

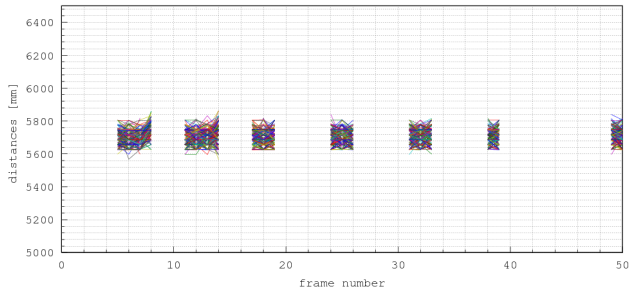

Figure 5: Multi camera with disturbance detection

3. Operation in interference situation

There are various options to deal with interfered measurements, which typically are likely just on few pixels. In the following, some concepts are described.

3.1 Discard corrupt measurements

The most simple way is to discard corrupt pixel data as shown in Figure 5. Since there are typically not many pixels affected, this is an easy and effective way. However, a loss of distance information in the image occurs.

3.2 Using the previous value

Another way is to use the last distance value of an interfered pixel. This will create a certain lag on such interfered pixels. But, typically not all pixels in an image are affected because they are not looking to the same location at the same time, so the lag is quite limited.

3.3 Frequency hopping

Due to the fact that the epc6xx chips do a synchronous demodulation of the received light, the chip operates in a very narrow modulation frequency band. This is like an FM radio which is sensitive in the selected frequency band only. Frequencies, which are a certain value higher or lower than the selected frequency, do not interfere with the main frequency. For the epc6xx chips, a frequency separation of 50 kHz does not interfere anymore. Thus, interference can be avoided by using different modulation frequencies of the TOF sensors which operate in the same space. It is based on the technique called frequency-division multiple access (FDMA).

However, are there for example hundreds of automated guided vehicles (AGV) in a fully automated warehouse, too many different frequencies have to be used. In addition, the frequency management can become quite challenging. As a conclusion, frequency selection should be automatically and on the fly in runtime. A possible algorithm to select another frequency is, first to detect interference. If interference takes place too many times, then choose another modulation frequency. Typically, e.g. for a modulation frequency at 10 MHz, 17 frequency bands separated by 50 kHz from the center frequency are enough to operate many systems in the same space. So, if interference occurs, a different frequency should be chosen randomly. This process can be done dynamically.

3.3.1 Using the external modulation input

Frequency hopping can be done only by using the external modulation input of the epc6xx imager chip in combination with e.g. an "I²C-programmable - any frequency clock generator" (e.g. from Silicon Labs: Si5351A/B/C) because it needs a finer change step size for the frequency than it is possible to do it with the chip internal clock divider. Refer for more details to the corresponding imager and clock module data sheets.

3.3.2 Integration time versus beat frequency

The interference by the modulation frequencies of 2 cameras synthesizes a so called beat frequency. The changing amplitude of this beat frequency creates and acts for the epc6xx chip like changing ambient-light. Thereof, it can be eliminated only by the on-chip ambient-light suppression. Due to this fact, the length of the integration time, respectively the frequency difference between the 2 modulation frequencies have to be selected the way to achieve a good averaging of this beat signal for getting, over the integration time period, the DC part only.

Based on this assumption, integration times < beat period will interfere all the time because the average value is changing heavily depending of the interference time between beat period and integration time (overlap resp. cross-over time).

To prevent of this, a good approach will be having the integration time in minimum > 10x the beat period. This reduces the average variations to an acceptable value and limits the distortions. For the example in the case of Figure 4, this will be a factor of

40 = 800 µs / 20 µs for a 50 kHz jump.

Channel	Center frequency [MHz]	Beat frequency [kHz]	Beat period [µs]
0	10.00	Reference	---
1	9.60	400	2.50
2	9.65	350	2.86
3	9.70	300	3.33
4	9.75	250	4.00
5	9.80	200	5.00
6	9.85	150	6.67
7	9.90	100	10.00
8	9.95	50	20.00
9	10.05	50	20.00
10	10.10	100	10.00
11	10.15	150	6.67
12	10.20	200	5.00
13	10.25	250	4.00
14	10.30	300	3.33
15	10.35	350	2.86
16	10.40	400	2.50

Table 1: Beat frequencies and periods for interference with 10 MHz modulation frequency and 50 kHz channel grid (see as reference TOF>cam 635)

Based on Table 1, row “beat period”, it is clear that short integration times will have a greater risk of interference and a higher cross-talk level thereof, compared to long integration times. Conclusion: A jump to a modulation frequency causing a high beat frequency is suggested for short integration times.

3.3.3 Using DRNU correction

If frequency hopping applies, although the handling of the DRNU tables according application note AN10 has to be done carefully. Due to the fact of only “small” frequency deviations from the calibration frequency, it will not be necessary in most cases to hold for all frequency steps (9.6 … 10.4 MHz) own sets of calibration tables. Only large changes of modulation frequencies, e.g. 20 MHz down to 10 MHz, will affect these tables strongly because the lowpass filter influences of the limited bandwidth of the imager and illumination electronic will take place and causes large, noticeable phase-shifts and signal attenuations in the signal path.

A suggested procedure for DRNU compensation with frequency hopping is

1. Acquire a distance measurement with the hopping frequency f_hop. e.g. 9.95 MHz (1 step of 50 kHz).
2. Calculate and compensate (completely, including ambient-light and temperature) the raw distance D_comp, the usual way according AN10 based on and using frequency f_calib e.g. 10 MHz. It is the reference / base frequency of the frequency hopping. Although, f_calib is the frequency where the DRNU tables are calibrated and apply. This result is valid and so far “correct” for the calibration unambiguity range UA_calib based on the measured phase-angle.
3. Rescale D_comp to the hopping unambiguity range UA_hop of the hopping frequency.
   D_hop = D_comp (UA_hop / UA_calib).
   D_hop shows the correct distance except the distance offset.
4. Calculate the final distance D by adjusting D_hop to the correct distance offset
   D = D_hop – Dpllcalib + Dpllhop
   whereas
   - D_pllcalib is the calibration PLL frequency dependent total distance shift caused by chip setting of register 0x8B e.g. for a PLL frequency of 80 MHz -
   1 step will be -1.875 m (1/8 unambiguity UA_calib).
   - D_pllhop is the hopping PLL frequency dependent total distance shift caused by chip setting of register 0x8B e.g. for a PLL (external modulation) frequency of 8 x 9.95 MHz = 79.6 MHz - 1 step will be around -1.866 m (1/8 unambiguity UA_hop).

Note: Take care to use all the time data formats fulfilling the necessary distance resolution needs. Otherwise distance noise increases and accuracy gets lost.

3.4 Disturbance filtering

Another approach to avoid a big distance error caused by a multi camera situation is to filter out these distance shifts. A basic filter algorithm, which allows to remove big distance shifts, looks like as follows:

(4)

// calculating current to previous distance difference:

$$diff_{x,y}[i] = dx, y - dx, y [i-1]$$

// calculating k:

$$k_{x,y} = \frac{k_{diff}}{|diff_{x,y}|}$$

```
if ( diffx,y [i] !=0)          // avoid zero division
else
    kx,y = kmax
endif

if ( kx,y > kmax )
    kx,y = kmax
endif
```

// filtering:

```
dfilter x, y [i] = kx,y *dx, y + (1–kx,y) * dfilter x, y [i-1]
```

The basic idea of this filter is that with $k_{x,y}$ = 0, the output of this filter is the last value and a $k_{x,y}$ = 1 let only passing through the actual input values.

dx, y [i]:	current measured distance on pixel x,y
dx, y [i-1]:	previous unfiltered distance on pixel x,y
dfilter x, y [i] :	current filtered distance on pixel x,y
dfilter x, y [i-1] :	previous filtered distance on pixel x,y

$k_{x,y}$:	filter gain, which changes filter sensitivity and can vary between 0 and k_{max}
k_{max}:	maximum filter gain, which is allowed.
$d_{x,y}$:	current distance value of a corrected pixel
k_{diff}:	factor to scale down the $diff_{x,y}$ [I] and vary the $k_{x,y}$ sensitivity to the this differential part

With this filter it is possible to filter big distance shifts. But, if there is a shift because of a distance change, the filter will respond quite slow. Just at that moment, the filter should not filter and clears his history. The algorithm below checks, how many times the difference $d_{diff\,x,y}$ [i] is smaller than the threshold t_{noise}. This is used to recognize, if the distance measurement is stable.

(5) // calculating number of valid threshold checks

```
if ( | diffx,y [i] | < tnoise)
    thresCheckx,y += 1   // incrementing
else
    thresCheckx,y = 0
endif
```

The bigger the number of $thresCheck_{x,y}$, the more stable is the distance measurement on a target. The threshold should be set higher than the expected distance noise. Otherwise every noise peak clears the counter ($thresCheck_{x,y}$) or can cause an unwanted filter reset. Because the next step (6) clears the filter for this pixel which has for example two valid threshold checks passed (numCheck = 2) and the distance changed bigger than the threshold t_{noise}. In other words, the target has moved since 2 frames and the filter has to be reset.

(6) // clear history caused by target movement

```
if ( thresCheckx,y >= numCheck)
    if ( |dx, y – validx,y | > tnoise )
        validx,y = dx, y  [i]      // save last valid distance
        dfilt x, y [i] = dx, y  [i]  // clear filter
    else
        validx,y = dx, y  [i]      // save last valid distance
    endif
endif
```

numCheck:	number of valid threshold checks, which has to be reached for a filter reset
$valid_{x,y}$:	$valid_{x,y}$ distance is needed to detect a target movement

3.4.1 Memory requirements of interference filtering

The arrays dx, y [i-1], dfilt x, y [i-1], validx,y, thresCheckx,y have to be saved during the runtime. The distance values are saved with uint16 and the thresCheckx,y with uint8.

Thus, the memory requirement is as follows for the following chips:

epc611	448 Byte (in the 8x8 TOF imaging mode)
epc635	67.2 kByte
epc660	537.6 kByte

4. Summary

As reported herein, there are different ways to react on cwTOF camera interferences. Which solution is best, depends of the application and its situation – or a mix thereof. It is demonstrated that it will be possible to react to such phenomena in an adequate way to minimize measurement disturbance. In a lot of application cases it would probably be necessary to use a combination is these possibilities to get the expected result, special in autonomous application respectively automatically and randomly frequency hopping.

5. Document change history

V1.00: Released

IMPORTANT NOTICE

ESPROS Photonics AG and its subsidiaries (ESPROS) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to ESPROS' terms and conditions of sale supplied at the time of order acknowledgment.

ESPROS warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with ESPROS' standard warranty. Testing and other quality control techniques are used to the extent ESPROS deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

ESPROS assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using ESPROS components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

ESPROS does not warrant or represent that any license, either express or implied, is granted under any ESPROS patent right, copyright, mask work right, or other ESPROS intellectual property right relating to any combination, machine, or process in which ESPROS products or services are used. Information published by ESPROS regarding third-party products or services does not constitute a license from ESPROS to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from ESPROS under the patents or other intellectual property of ESPROS.

Resale of ESPROS products or services with statements different from or beyond the parameters stated by ESPROS for that product or service voids all express and any implied warranties for the associated ESPROS product or service. ESPROS is not responsible or liable for any such statements.

ESPROS products are not authorized for use in safety-critical applications (such as life support) where a failure of the ESPROS product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of ESPROS products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by ESPROS. Further, Buyers must fully indemnify ESPROS and its representatives against any damages arising out of the use of ESPROS products in such safety-critical applications.

ESPROS products are neither designed nor intended for use in military/aerospace applications or environments unless the ESPROS products are specifically designated by ESPROS as military-grade. Only products designated by ESPROS as military-grade meet military specifications. Buyers acknowledge and agree that any such use of ESPROS products which ESPROS has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

ESPROS products are neither designed nor intended for use in automotive applications or environments unless the specific ESPROS products are designated by ESPROS as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, ESPROS will not be responsible for any failure to meet such requirements.