AN14104

i.MX 8M Plus Stereo Vision

Rev. 1.0 — 8 December 2023 Application note

Document information

Information	Content
Keywords	AN14104, Basler camera, i.MX 8M Plus, stereo vision, pylon, depth map, calibration
Abstract	This document describes how to synchronize two Basler cameras on i.MX 8M Plus and compute a depth map using OpenCV and Python.

1 Introduction

Stereo vision is an extraction of 3D information from two images captured by two synchronized cameras that can be compared with the human vision.

This document provides detailed information about a proof of concept for i.MX 8M Plus stereo vision and includes the following sections:

- Hardware and software requirements
- Camera synchronization
- Build the image
- Calibration
- Depth map computation
- Results

2 Hardware and software requirements

Table 1 provides details of the hardware and software used.

Table 1. Hardware and software used

Category	Description
Hardware	Development kit: NXP i.MX 8M Plus EVK LPDDR4 Camera kit: 2 x Basler cameras daA3840-30mc Micro SD card: • SanDisk Ultra 32-GB micro SDHC • Class 10 is used for the current experiment USB: Micro-USB cable for the debug port
Software	Host PC running a recent version of Ubuntu 20.10 Linux 5.15.71_2.2.0 The source code mentioned in this document is available at: MICRSE-1784-8mplus-stereo-vision-release.

3 Build the image

For information on building Yocto, refer to *i.MX Yocto Project User's Guide* (document IMXLXYOCTOUG).

To download the i.MX Yocto Project community BSP recipe layers, perform the following steps:

1. Create a directory for the project. In the directory, use the following command:

```
$ repo init -u git://github.com/nxp-imx/imx-manifest.git -b imx-linux-kirkstone -m
 imx-5.15.71-2.2.0.xml
$ repo sync
$ DISTRO=fsl-imx-wayland MACHINE=imx8mp-lpddr4-evk source imx-setup-release.sh -b
 build
```

2. To use the Basler kit and the Pylon Viewer application, integrate the support given by Basler in the BSP provided by the i.MX Yocto Project. The two layers `meta-basler-imx8` and `meta-basler-tools` are cloned and added to the project sources, as follows:

```
$ cd ../sources
$ git clone -b kirkstone https://github.com/basler/meta-basler-tools.git
$ git clone https://github.com/basler/meta-basler-imx8.git
```

3. After cloning the two layers, read the license files.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

4. For accepting the licenses and installing the required packages, append the following command lines in `build/conf/local.conf`:

```
# BASLER

 ACCEPT_BASLER_EULA = "1"
 IMAGE_INSTALL:append = " packagegroup-dart-bcon-mipi"
 IMAGE_INSTALL:append = " gst-plugin-pylon"
```

5. Follow the format and add the two layers in `build/conf/bblayers.conf`:

```
BBLAYERS += "${BSPDIR}/sources/meta-basler-imx8"
BBLAYERS += "${BSPDIR}/sources/meta-basler-tools"
```

6. To build the image, run the following command:
Note: *The full image is required.*

```
$ bitbake imx-image-full
```

4 Camera synchronization

Ensure that the captured frames are synchronous. In the BSP, it is not a default functionality, which means it must be added as explained in the following sections.

Note: *For the final application, the GPIO hardware trigger has been used.*

Two different ways to trigger the synchronous capture are as follows:

- PWM trigger
- GPIO hardware trigger

4.1 PWM trigger

The output of the PWM is a toggling signal whose frequency and duty cycle can be modulated by programming the appropriate registers, as follows:

1. Rebuild the `imx8mp-evk-dual-basler.dts` after adding the following command lines:

```
&pinctrl_pwm4 {
fsl,pins = <
MX8MP_IOMUXC_GPIO1_IO15__PWM4_OUT 0x116
>;
```

2. Deploy the image and `dtb` on the SD card.
3. Stop in the U-Boot and change `dtb`.

```
u-boot=> edit fdtfile
edit: imx8mp-evk-dual-basler.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

4. After booting, open a wayland terminal using a mouse and a keyboard, which are connected to the platform. Run the pylon application as follows:

```
$ pylon
```


All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Turn on both the cameras that appear in the device window at "Basler BCON GenTL Producer > Basler BCON GenTL Interface Module".

5. Select the following configurations on each of the video cameras:
   a. In the features window, go to Acquisition Control and select the following parameters:
      - Trigger Sector = Frame start
      - Trigger Mode = On
      - Trigger Activation = Raising Edge
   b. Click "Camera > Save Features" and generate the pylon feature stream (`pfs`) for each camera. These two `pfs` files are used by the GStreamer command. Close the pylon application.
6. On the i.MX 8M Plus terminal, perform the following steps:
   a. Disable the clock as follows:

```
$ echo 0 > /sys/class/pwm/pwmchip2/export
```

   b. Set a period of 33 ms and a duty cycle of 50 %.

```
$ echo 33333333 > /sys/class/pwm/pwmchip2/pwm0/period
$ echo 15000000 > /sys/class/pwm/pwmchip2/pwm0/duty_cycle
```

   c. Place a display with a timer in front of each camera, run the GStreamer pipeline, and wait for the cameras to configure.

```
$ gst-launch-1.0 \
pylonsrc device-index=0 num-buffers=1 \
pfs-location=daA3840-30mc_40097241.pfs ! \
video/x-raw,width=640,height=480 ! jpegenc ! \
fpsdisplaysink video-sink="filesink location=left.jpeg" \
pylonsrc device-index=1 num-buffers=1 \
pfs-location=daA3840-30mc_40292521.pfs ! \
video/x-raw,width=640,height=480 ! jpegenc ! \
fpsdisplaysink video-sink="filesink location=right.jpeg" &
```

   d. After the configuration is completed, enable the clock to trigger the cameras.

```
$ echo 1 > /sys/class/pwm/pwmchip2/pwm0/enable
```

   e. Once the cameras are triggered, two synchronous frames are saved. The format shown in Figure 1 is `h:m:s:ms`.

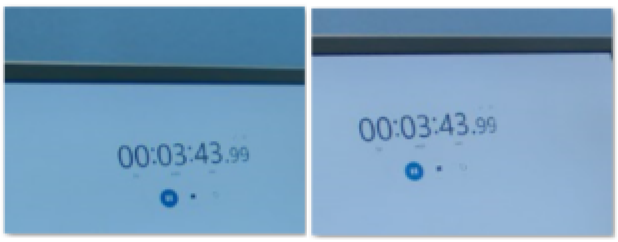

Figure 1. Synchronous frames

4.2 GPIO hardware trigger

General-purpose input/output (GPIO) is a module that controls the IOMUX on the chip. Therefore, configure the GPIO before using it.

Figure 2 shows the two BCON to MIPI adapters along with pin 5 and pin 6, which are connected to a jumper. A connection is established between the SYNC_I(C) signal and the MCLK signal on the adapter.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

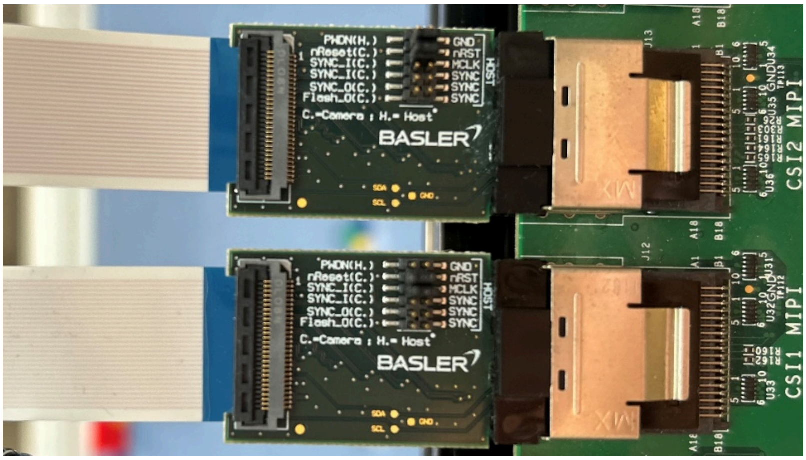

Figure 2. BCON for MIPI to mini-SAS adapter

The SYNC_I(C) signal is connected to the camera input trigger and the MCLK signal is connected to the GPIO on SoC GPIO1_IO15 as described in Table 2.

Table 2. Signals accessible on the Basler adapter

Platform	Function	Signal name	GPIO SoC	Signal name on the adapter
i.MX 8M Plus EVK	Trigger CAM0	CSI_MCLK CSI1_SYNC	GPIO1_IO15 GPIO1_IO5	CSI_MCLK CSI_SYNC
	Trigger CAM1	CSI_MCLK CSI2_SYNC	GPIO1_IO15 GPIO1_IO7	CSI_MCLK CSI_SYNC

GPIO1_IO15 is a pin dedicated for general use that can be configured using GPIO peripherals as follows:

```
--- a/arch/arm64/boot/dts/freescale/imx8mp-evk.dts
+++ b/arch/arm64/boot/dts/freescale/imx8mp-evk.dts
@@ -924,6 +924,7 @@ MX8MP_IOMUXC_HDMI_DDC_SCL__HDMIMIX_HDMI_SCL 0x400001c2
         MX8MP_IOMUXC_HDMI_DDC_SDA__HDMIMIX_HDMI_SDA 0x400001c2
         MX8MP_IOMUXC_HDMI_HPD__HDMIMIX_HDMI_HPD 0x40000010
         MX8MP_IOMUXC_HDMI_CEC__HDMIMIX_HDMI_CEC 0x40000010
+        MX8MP_IOMUXC_GPIO1_IO15__GPIO1_IO15            0x110
    /*
```

To configure GPIO1_IO15, modify the device tree, as shown above, considering the following facts:

- `printctrl_hog` is used for GPIO pinmuxes that are not explicitly used in other modules in the device tree.
- It is set by default when testing the `IOMUXC` module.

Then rebuild `imx8mp-evk-dual-basler.dts` and deploy it on the SD card.

To toggle the configured GPIO, use an application that uses `libgpiod`, which is a Linux library that allows the interfacing with GPIO. One way to add the library in the image is to build and install it manually or add it to the Yocto Project as follows:

```
$ echo IMAGE_INSTALL:append = \"libgpiod\" >> conf/local.conf
$ bitbake imx-image-full
```

The application is built with the help of the following command:

```
$ aarch64-linux-gnu-gcc gpio_toggle_example.c \
  -I <workspace>/libgpiod/libgpiod_install/include \
```


All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

```
    -L <workspace>/libgpiod/libgpiod_install/lib \
    -lgpiod -o gpio_toggle_example
```

Here is an example of how to toggle the GPIO with the help of the application built previously:

```
$ ./gpio_toggle_example /dev/gpiochip0 15
```

To manage the frames captured by the two synchronized Basler cameras, use the libraries found in the downloaded packages. The two libraries, `genicam` and `pylon`, facilitate communication with the cameras so that the `ImageEventHandler` class is used to capture the frames where the `OnImageGrabbed()` function is found. When an image is captured, each camera calls this function as shown in Figure 3.

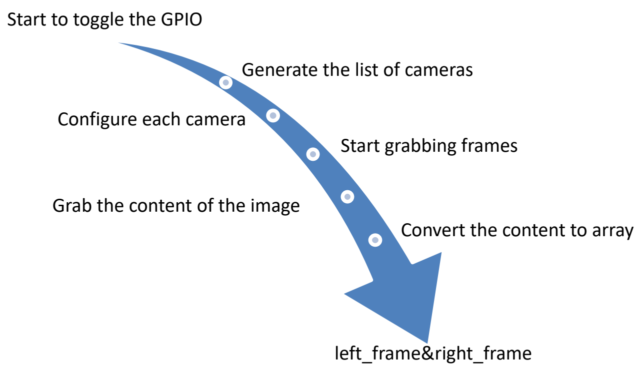

Figure 3. Steps for capturing frames

At the end, each camera generates two frames that can be used further.

5 Calibration

This section describes how to calibrate the two cameras using a chessboard pattern shown in Figure 4. The chessboard used consists of 11 columns, 8 rows, and the side of the square is 2.5 cm. This information is important to compute the parameters.

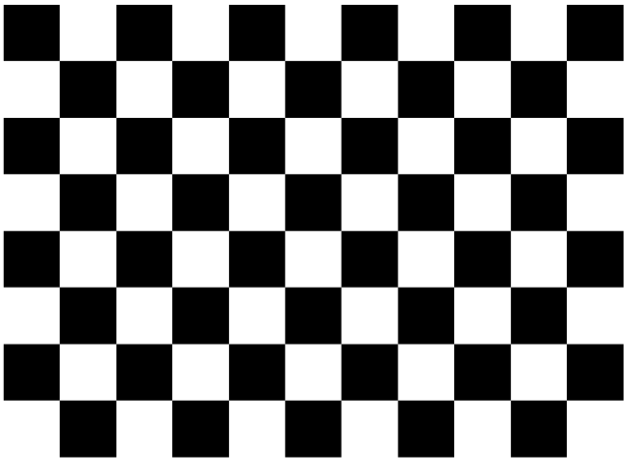

Figure 4. Chessboard A4 25 mm 11x8

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Using the pattern for each camera, take several synchronous photographs as shown in Figure 5. For the results presented in this document, at least 130 photographs have been taken. These photographs are calibrated with the help of the OpenCV function `cv2.cornerSubPix()`.

If the chessboard is detected, this function finds corners. The corners are used to compute camera calibration parameters using a `cv2.calibrateCamera()` function.

The parameters obtain for each camera are used for stereo calibration made with the help of the `cv2.stereoCalibrate()` and `cv2.stereoRectify()`.

The `cv2.stereoCalibrate()` returns the root mean square (RMS) implying that it has to be as small as possible.

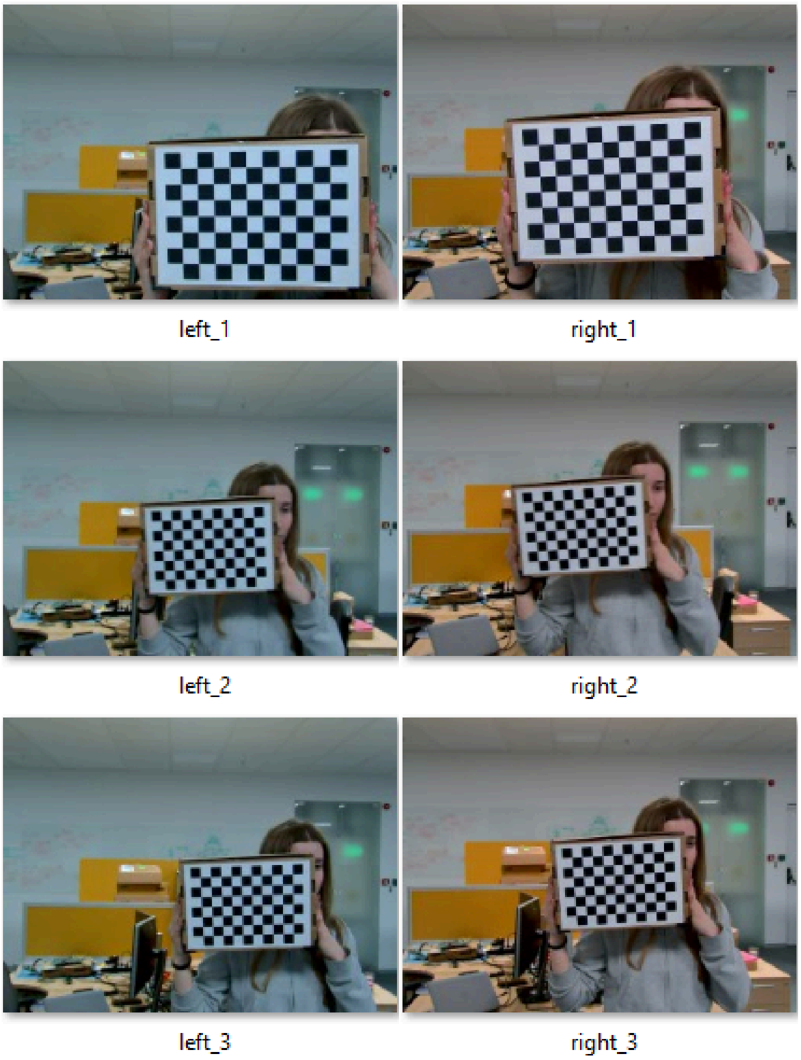

Figure 5. Frames taken for calibration

The description provided above can be performed with the help of the `calibration.py` script. This script uses the photographs for both cameras and saves the parameters in the directory. The script is as follows:

```
$ ./calibration.py
help
usage: ./calibration [-h] [--left_dir LEFT_DIR] [--right_dir RIGHT_DIR] [--width WIDTH]
 [--height HEIGHT] [--square_size SQUARE_SIZE] [--save_dir SAVE_DIR]
Stereo calibration!
```


© 2023 NXP B.V. All rights reserved.

```
--------------------------------
Use the command specifying the names of the directories and the dimensions of the chess
 board. Default values can be used if appropriate.
Take as many photos as you like and let the program decide which of them are the best 15
 photos.
At the end, the parameters that had the lowest ret value will be saved. (it should be
 less than 0.2)
optional arguments:
  -h, --help            show this help message and exit
  --left_dir LEFT_DIR   left images directory path, the default is "left"
  --right_dir RIGHT_DIR      right images directory path, the default is "right"
  --width WIDTH         chessboard width size, default is 11
  --height HEIGHT       chessboard height size, default is 8
  --square_size SQUARE_SIZE  chessboard square size, the default is 2.5
  --save_dir SAVE_DIR   the directory where to save the parameters (please create it in
 the work directory before start), the default is "parameters"
The default parameters will be set as in the following command:
  ./calibration.py --left_dir left --right_dir right --square_size 2.5 --save_dir
 parameters --height 8 --width 11.
```

For the results obtained using this Python script, 127 photographs have been captured for each camera. 15 photographs with the best value for RMS are used for computing the parameters in the calibration step, as shown in Figure 6.

```
Computing parameters for 15 photos starting with  30
100%|                                                | 15/15 [00:00<00:00, 102.99it/s]
100%|                                                | 15/15 [00:00<00:00, 104.52it/s]
Left calibration rms:  0.14620938850189036
Right calibration rms:  0.1472455579105701
Stereo calibration rms:  0.15647824302793037
Saved parameters for images  30 45
```

Figure 6. Calibration results

6 Depth map computation

After establishing the synchronous capture, calibrate the two cameras. The parameters obtained are used for rectification. After rectifying the two images, the depth map is calculated.

Figure 7. Depth map computation

To compute the depth map in real time, a `stero.py` script has been created, which can be used in three different modes described as follows:

```
$ ./stereo.py -help
usage: stereo.py [-h] --left_dir LEFT_DIR --right_dir RIGHT_DIR --disparity_map
 DISPARITY_MAP --parameters PARAMETERS --mode MODE
Depth map computation.
options:
  -h, --help            show this help message and exit
  --left_dir LEFT_DIR   left images directory path
  --right_dir RIGHT_DIR   right images directory path
  --disparity_map DISPARITY_MAP    directory used to save depth_map
  --parameters PARAMETERS    directory containing parameters obtained at calibration step
  --mode MODE   The mode the way the application will run:
                  1 - only capture frames with saving for use in calibration
                  2 - calculating the depth map without saving frames and the map
```


All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

```
3 - calculating the depth map with saving frames and the map
```

The same script can be used for capturing the frames for the calibration by setting the property mode.

For more efficient use of the application, a shell script has been created. This script creates the directories as required and uses them by running the base script `stereo.py`. For the shell script, the mode must be specified. For example, for the first mode, run the below command line:

```
$ ./start_stereo.sh 1
```

In the `stereo.py` script, the steps described in the second synchronization mode are performed, obtaining the left and right frames. Then, the frames are rectified using the parameters obtained in the calibration step. The rectified images are used by the `StereoSGBM_create` object. This object calculates the disparity by setting the required parameters according to the desired results.

For example, the `max_disp` parameter refers to the disparate search range. The algorithm searches for all disparities between 0 and the set value. The `blockSize` parameter sets the size of the blocks that the algorithm compares. A small value provides maps with more finely calculated disparities and if the value is higher, the algorithm can result in wrong comparisons.

To obtain better results, filter these results using a weighted least squares (WLS) filter. This filter is used for the following options:

- Maps based on WLS
- Optional use of previously found left-right consistent parameters to refine the results in a uniform area

7 Results

The image processing and algorithm runs on the CPU implying that it is a CPU intensive example. This example can result in 1 to 2 disparity maps per second.

For the first example, a white box has been used and three pictures have been captured. The box is brought closer to the camera lens. Here, the closer the box is to the camera, the darker the pixels get, as shown in Figure 8.

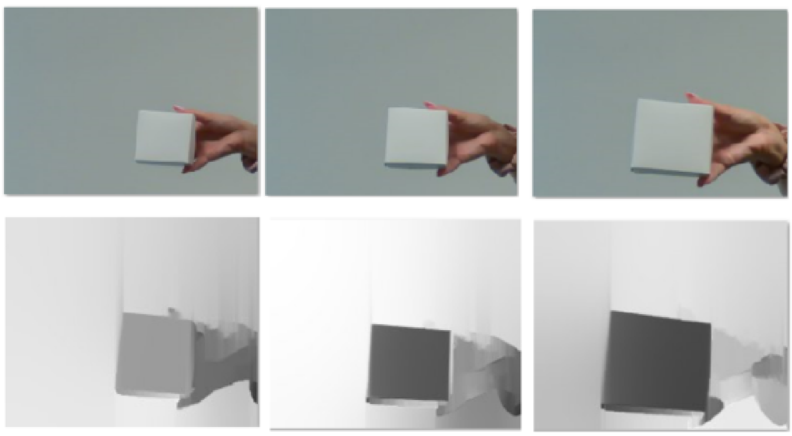

Figure 8. Results obtained with a white object

For the second example, two objects with different colors are placed at the same distance from the cameras. Here, the pixel color intensity is the same for both the objects, as shown in Figure 9.

© 2023 NXP B.V. All rights reserved.

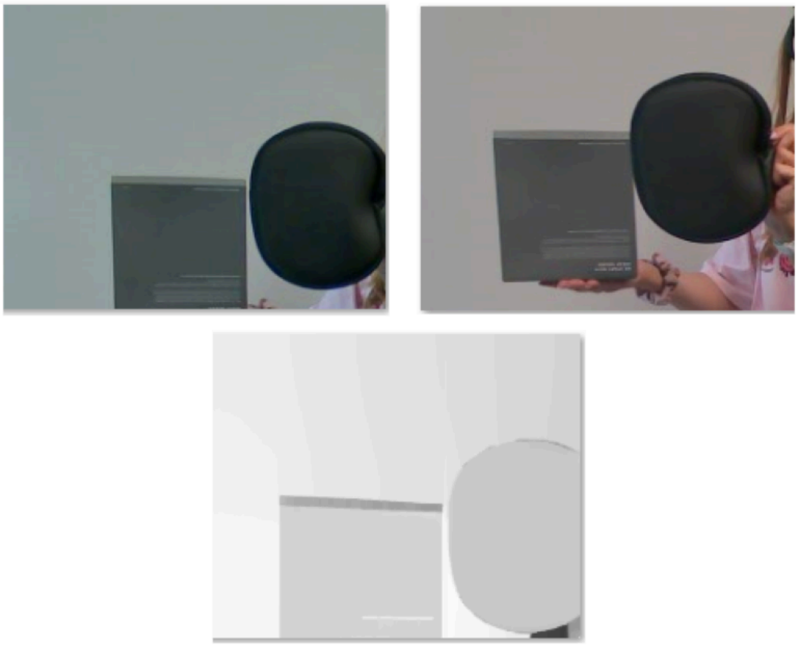

Figure 9. Results obtained with two objects aligned

For the third example, one object is placed in front of the other object. Here, the area associated with the front object is darker than the one associated with the object placed behind, as shown in Figure 10.

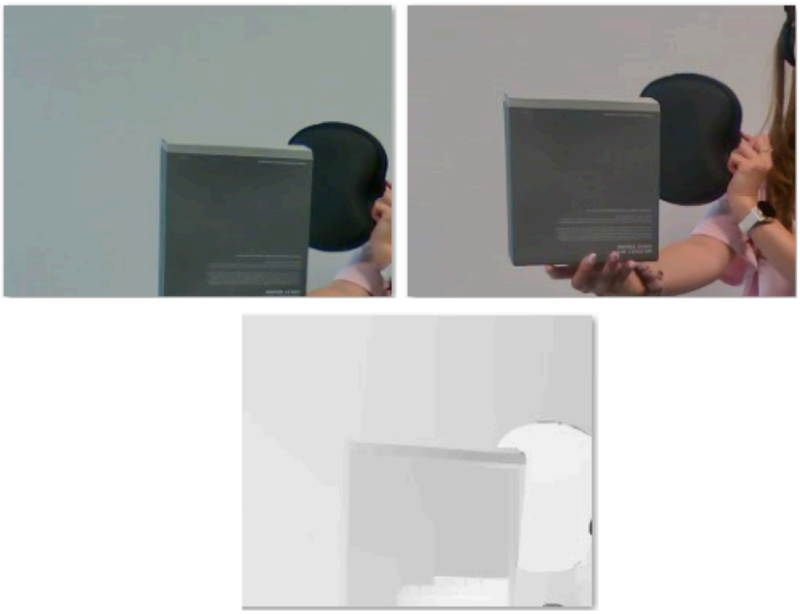

Figure 10. Results obtained with two differently positioned objects

If the objects have several types of colors, writing, or relief, mismatches can be observed in the depth map. However, the presented solution is only a proof of concept. To obtain better results for the calibration and the calculation of the depth map, improvements must be added.

8 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

© 2023 NXP B.V. All rights reserved.

Copyright 2023 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

9 Revision history

Table 3 summarizes the revisions made to this document.

Table 3. Revision history

Revision number	Release date	Description
1	8 December 2023	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

i.MX — is a trademark of NXP B.V.

© 2023 NXP B.V. All rights reserved.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2023 NXP B.V. **All rights reserved.**

For more information, please visit: https://www.nxp.com

Date of release: 8 December 2023
Document identifier: AN14104