Tara Stereo Vision USB 3.0 Camera

SDK Linux User Manual

Version 1.6

e-con Systems

7/4/2018

e-con Systems

Your Product Development Partner

Disclaimer

e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

Contents

© Copyright e-con Systems. 2017. All rights reserved.

Introduction to Tara - See3CAM_Stereo

The Tara - See3CAM_Stereo, hereafter called as Tara, is a UVC compliant USB 3.0 SuperSpeed Stereo vision camera from e-con Systems, a leading embedded Product Design Services company which specializes in the advanced camera solutions. Tara is based on MT9V024 stereo sensor from On Semiconductor and it supports a maximum resolution of WVGA at 60fps over USB 3.0 in uncompressed format. Tara is the latest member of the Stereo Vision family of USB 3.0 SuperSpeed camera products launched by e-con Systems.

The Tara Software Development Kit (SDK) package, built on OpenCV Image Processing Library is bundled with Tara - Stereo Vision USB 3.0 Camera. Tara SDK includes the commonly used stereo camera functions such as disparity, depth measurement, and few real-time applications implemented in OpenCV. e-con Systems SDK is provided with the source code.

Tara is a monochrome camera with the S-mount (also known as M12 board lens) lens holder and pre-calibrated lens pair. The S-mount is one of the most commonly used small form-factor lens mounts for board cameras. Tara has two OnSemi's 1/3-inch MT9V024 image sensors separated by an inter-ocular distance or base line of 60 mm. With USB 3.0 interface to the host PC, Tara can stream WVGA resolution at 60 fps, VGA resolution at 60 fps and QVGA resolution at 60 fps in uncompressed Y16 format. It can also capture the still images. Tara is also backward compatible with USB 2.0 host ports and it does not require any special camera drivers to be installed in the host PC. In USB 2.0, the camera can stream WVGA resolution at 30 fps, VGA resolution at 30 fps and QVGA resolution at 60 fps in uncompressed Y16 format.

This document highlights the SDK folder structure and applications that are included in the package for Linux Operating System (OS) and describes the special demo applications that are included in the SDK.

Prerequisites

You must have basic knowledge in OpenCV, to understand and use the Tara SDK.

Hardware Requirements

The hardware requirements for Tara are a PC or Laptop with USB 3.0 or USB 2.0 port.

Software Requirements

The software requirements for Tara are OS-Linux Ubuntu 16.04 64-bit.

Description

Tara is a USB 3.0 camera capable of streaming camera frames WVGA (2 x 752 x 480) resolution at 60 fps, VGA (2 x 640 x 480) resolution at 60 fps and QVGA (2 x 320 x 240) resolution at 60 fps. Tara can support still capture of images with the required exposure level. It also supports all the features with a USB 2.0 fallback. In USB 2.0, Tara can stream all resolutions at lower frame rates.

It currently supports format Y16 in all the resolutions and in both USB 3.0 and USB 2.0. In this format, the data from both the sensors is obtained in a sequential manner that is, in WVGA (2 x 752 x 480) resolution each sensor data is of 752 x 480 resolution which is sent from the camera in a sequential manner. This is then split as left and right data in the e-CAMView using the Stereo Transform filter to be displayed as two separate images.

Tara SDK Folder Structure

The Tara_SDK_LINUX_REL_package_xxxx directory contains the folder structure as shown below.

```
|--------  Documents
|          |-- Building_SDK_Solutions.pdf
|          |-- Tara_Linux_API_Manual.pdf
|          |-- Tara_SDK_IMU_SampleApp_User_Manual.pdf
|          `-- Tara_SDK_Linux_User_Manual.pdf
|--------  Prebuilts
|           `-- Ubuntu-16.04
|           |          |-- binary_x64
|           |          `-- lib_x64
|           `-- dependencies.sh
|           `-- install.sh
|           `-- uninstall.sh
`--------  Source
           |-- common
           |-- examples
           |-- configure.sh
           |-- Makefile
           `-- README.txt
```

Documents

This folder contains User Manuals of Tara camera as listed below:

- *Building_SDK_Solutions_Linux.pdf* - This document explains the steps to be followed in building the SDK libraries and binaries. The prebuilts are given for Ubuntu-16.04(LTS). When using Ubuntu 16.04, you can directly use the prebuilts, whereas for other version you need to build the SDK on your system.

- *Tara_Linux_Extension_Unit_API_Document.pdf* - This document describes Tara namespace used in SDK where all the common functions are provided together.
- *Tara_SDK_IMU_SampleApp_User_Manual.pdf* - This document describes the usage of an Inertial Measurement Unit (IMU) unit (LSM6DS0 (or) LSM6DS33) integrated with e-con's stereo camera - Tara in application development.
- *Tara_SDK_Linux_User_Manual.pdf* - Currently reading document.

Prebuilts

The prebuilts contains the executable of all the applications in the SDK. It also contains the library files of Tara, Extension unit and the dependency libraries of the OpenCV. The executable and all the dependencies are built for 64-bit in Linux OS Ubuntu -16.04 respectively. The shell scripts in the folder can be used to download the necessary dependencies and can install the prebuilts binaries into **/usr/local/tara-sdk/** location.

Source

The source contains the source files for the commonly used functions and sample applications. These source files can be used to generate all the libraries and binaries to evaluate the Tara SDK.

Block Diagram of Tara SDK in Linux

The block diagram of Tara SDK folder structure is shown below.

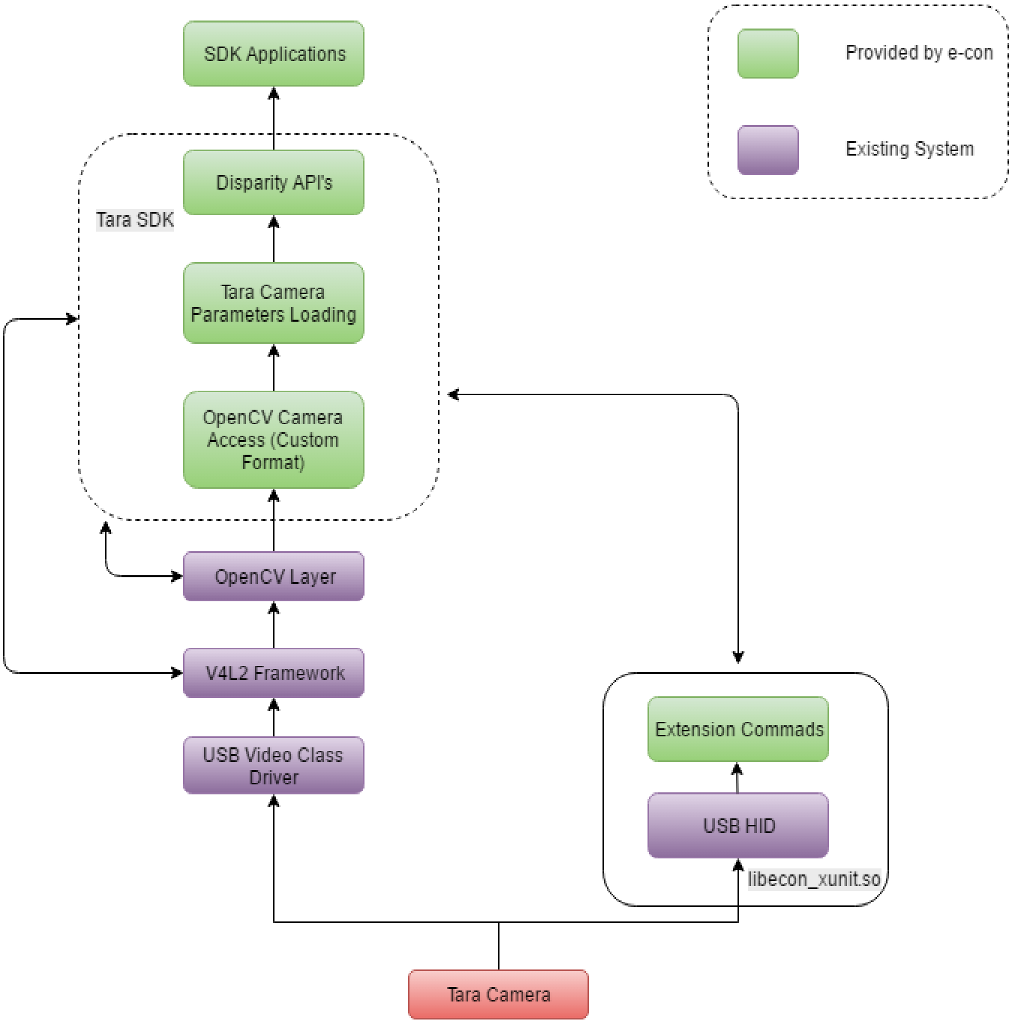


Figure 1: Block Diagram of Tara SDK in Linux

Dependencies to be Installed

For using OpenCV and Point Cloud library, you must install the dependencies as follows:

- To install some general development libraries, you can run the following command.

```
$ sudo apt-get install build-essential make cmake
cmake-qt-gui g++
```

- To install libav video input or output and glib development libraries, you can run the following command.

```
$ sudo apt-get install libavformat-dev libavutil-dev
libswscale-dev libglib2.0-dev libtbb-dev
```

- To install video4Linux camera development libraries, you can run the following command.

```
$ sudo apt-get install libv4l-dev
```

- To install eigen3 math development libraries, you can run the following command.

```
$ sudo apt-get install libeigen3-dev
```

- To install openGL development libraries (to allow creating graphical windows), you can run the following command.

```
$ sudo apt-get install libglew-dev
```

- To install GTK development libraries (to allow creating graphical windows), you can run the following command.

```
$ sudo apt-get install libgtk2.0-dev
```

- To install Udev development libraries (to allow access to device information), you can run the following command.

```
$ sudo apt-get install libudev-dev
```

- To install Point Cloud library in Ubuntu 14.04, you can run the following command.

```
$ sudo add-apt-repository ppa:v-launchpad-jochen-
sprickerhof-de/pcl
$ sudo apt-get update
$ sudo apt-get install libpcl-all
```

- To install Point Cloud library in Ubuntu 16.04, you can run the following command.

```
$ sudo apt-get install libpcl-dev
```

Note:

- This command will suggest additional dependency packages to be installed. Press **Y** to continue downloading the additional packages.
- For easier convenience, the dependencies to be installed are written as a shell script. Use the (dependencies.sh) script in the Prebuilts folder to install the dependencies at ease. Please refer to Installing Tara SDK Package section to know its usage.

Installing Tara SDK Package

The steps to install the Tara SDK package are as follows:

1. To unzip the Tara_SDK_LINUX_REL_package_xxxx.zip**,** you must run the following command.

```
$ unzip Tara_SDK_LINUX_REL_package_xxxx.zip
```

While unzipping, it contains a tar archive and a TaraSDKInstallation_Linux.txt text file. You can refer to the TaraSDKInstallation_Linux.txt for instructions.

2. Extract the tar archive by running the following command.

```
$ tar -xhzvf Tara_SDK_LINUX_REL_package_xxxx.tar.gz
```

After running the command, a directory named Tara_SDK_LINUX_REL_PACKAGE_xxxx is extracted.

3. To change the location to the Prebuilts folder in the extracted package, you must run the following command.

```
$ cd <Extracted Package
Location>/Tara_SDK_LINUX_REL_PACKAGE_xxxx/Prebuilts
```

4. To install all the necessary dependencies, you must run the following command.

```
$ ./dependencies.sh
```

5. To run the install.sh shell script with superuser privilege, you must run the following command.

```
$ sudo ./install.sh
```

The prebuilts binaries and libraries are installed for the respective LTS version and architecture into **/usr/local/tara-sdk** path and append the path variables to bashrc and /etc/profile.

6. Reload the shell to use the modified environment variables by running the following command.

```
$ source ~/.bashrc
```

Note: In Tara_SDK_LINUX_REL_package_xxxx, xxxx refers to the release version.

Currently the prebuilts are given only for Ubuntu-16.04, and for other versions, you can skip this section and follow the steps in *Building_SDK_Solutions.pdf* document to install the Tara SDK.

Executing the Samples

To run the binaries, you must install the dependencies and the Tara SDK package. After installing the dependencies and the Tara SDK package, you can run the applications with the following command.

```
$ sudo -i <BinaryName>
```

For example,

```
$ sudo -i TaraCamViewer
$ sudo -i FaceDetection
```

The Binaries are FaceDetection, HeightCalibration, HeightEstimation, ImuApplication, PointCloud, TaraCamViewer, TaraDepthViewer, TaraDisparityViewer, VolumeEstimation.

The steps to execute the samples are as follows:

1. Once the application is launched, the list of enumerated camera devices will be displayed and then select the Device ID with the name See3CAM_Stereo as shown below.

```
Device List:
------------------
Device ID: 0, Device Name: See3CAM_Stereo
Enter the Device ID to Process: 0
```

2. If the stereo Device ID is selected, the resolutions supported by the device will be displayed, and then select the Resolution ID to start streaming the camera with that resolution as shown below.

```
Resolutions Supported:
-----------------------
ID: 0, Resolution: 752x480
ID: 1, Resolution: 640x480
ID: 2, Resolution: 320x240
Enter the Resolution ID to Stream: 0
```

3. The image windows streaming the left and right frames or disparity will be displayed.

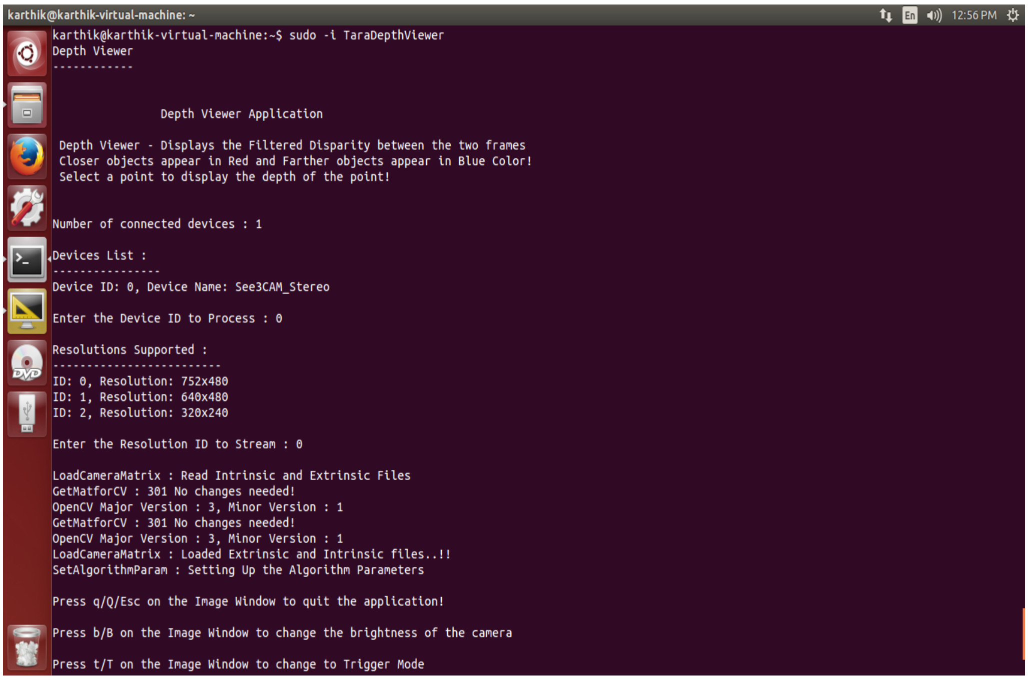
   a. Press **q/Q/Esc** on the image window to quit the application.
   b. Press **r/R** on the image window to view the right image of the camera in the samples where the right frame is not displayed.
   c. Press **d/D** on the image window to view the gray scale disparity map in the Depth Viewer and Disparity Viewer application.
   d. Press **b/B** on the image window to alter the brightness of the camera. The range is from 1 to 7.
   e. Press **a/A** on the image window to change the camera to Auto Exposure.
   f. Press **e/E** on the image window to alter the exposure of the camera. The range is from 10 to 1000000 micro seconds.

The additional supports in Depth Viewer and Tara Camera Viewer applications are as follows:

1. Press **t/T** on the image window to change the camera to Trigger mode.

   a. Trigger Mode is supported only in Manual Exposure so when the Trigger mode is selected, the camera is switched to Manual Exposure from Auto Exposure.
   b. When **a/A** is pressed in Trigger Mode, a message will appear on the console window as **Switch to Master Mode to change to Auto Exposure!!**.

2. Press **m/M** on the image window to change the camera to Master mode**.**



```
karthik@karthik-virtual-machine:~$ sudo -i TaraDepthViewer
Depth Viewer
------------


                Depth Viewer Application

 Depth Viewer - Displays the Filtered Disparity between the two frames
 Closer objects appear in Red and Farther objects appear in Blue Color!
 Select a point to display the depth of the point!


Number of connected devices : 1

Devices List :
----------------
Device ID: 0, Device Name: See3CAM_Stereo

Enter the Device ID to Process : 0

Resolutions Supported :
-------------------------
ID: 0, Resolution: 752x480
ID: 1, Resolution: 640x480
ID: 2, Resolution: 320x240

Enter the Resolution ID to Stream : 0

LoadCameraMatrix : Read Intrinsic and Extrinsic Files
GetMatforCV : 301 No changes needed!
OpenCV Major Version : 3, Minor Version : 1
GetMatforCV : 301 No changes needed!
OpenCV Major Version : 3, Minor Version : 1
LoadCameraMatrix : Loaded Extrinsic and Intrinsic files..!!
SetAlgorithmParam : Setting Up the Algorithm Parameters

Press q/Q/Esc on the Image Window to quit the application!

Press b/B on the Image Window to change the brightness of the camera

Press t/T on the Image Window to change to Trigger Mode
```

Figure 2: Sample Console

Note: By default, all applications will be in Manual Exposure with the default exposure value (8000 microseconds).

Please refer to Applications in SDK section to know about the applications included in SDK.

Uninstalling Tara SDK Package

The steps to uninstall the Tara SDK package are as follows:

1. Change the location to the Prebuilts folder in the extracted package using the following command.

```
$ cd <Extracted Package
Location>/Tara_SDK_LINUX_REL_PACKAGE_xxxx/Prebuilts
```

2. Run the uninstall.sh shell script with superuser privilege using the following command.

```
$ sudo ./uninstall.sh
```

This will remove the binaries from the installed location and delete the path variables added to bashrc and /etc/profile.

Developing Applications with Tara using OpenCV

You can develop the applications with Tara using OpenCV.

Note: To run the prebuilt binaries included in the SDK package, you must ignore the below steps.

The steps to develop applications using OpenCV are as follows:

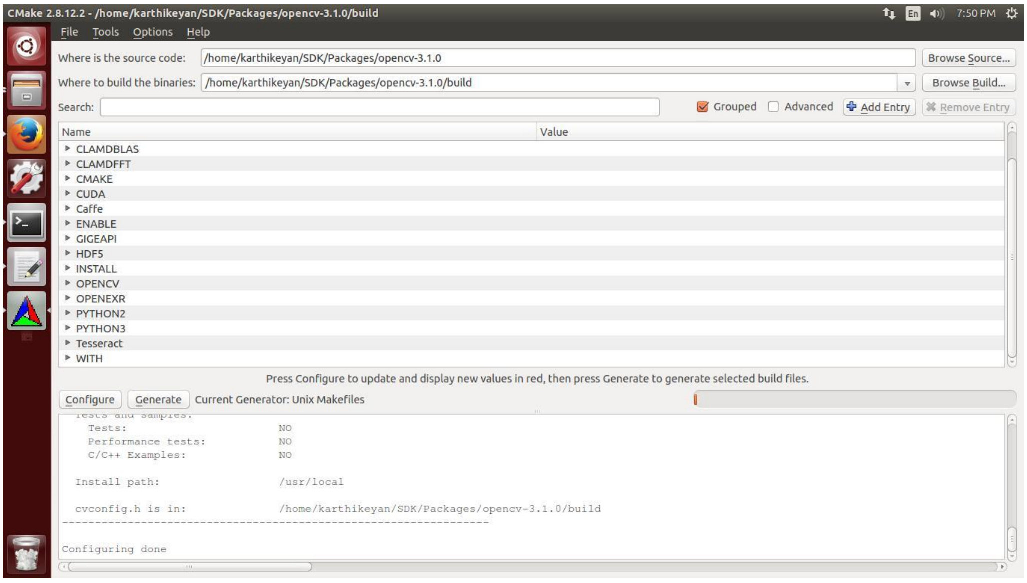
1. Download the latest version of opencv (Version 3.4.1) for Linux using https://github.com/opencv/opencv/tree/3.4.1 link.
2. Download the contrib modules additionally from the github repository using https://github.com/opencv/opencv_contrib/tree/3.4.1 link and by running the following command.

```
$ unzip opencv_contrib-3.4.1.zip
$ unzip opencv-3.4.1.zip
$ cd opencv-3.4.1
$ mkdir build && cd build
$ cmake-gui
```

   a. Browse the source code path - <Dir>/opencv-3.4.1.
   b. Browse the build path - <Dir>/opencv-3.4.1/build.
   c. Press the **Configure** button, with default native compilers set (Unix Makefiles).

Select all the packages and press the **Configure** button. For an easier overview of Build options, enable the **Grouped** option in the binary directory.

For some packages, CMake may not find all the required files or directories. In this case, CMake will throw an error in its output window (located at the bottom of the Graphical User Interface (GUI)) and set its field values, to not found constants. Therefore, you need to manually set the queried directories or files path. Then press the **Configure** button to view if the value is accepted or not. Follow the above steps, until all entries are good.



CMake 2.8.12.2 - /home/karthikeyan/SDK/Packages/opencv-3.1.0/build

Where is the source code: /home/karthikeyan/SDK/Packages/opencv-3.1.0

Where to build the binaries: /home/karthikeyan/SDK/Packages/opencv-3.1.0/build

Search:

☑ Grouped ☐ Advanced

Name	Value
CLAMDBLAS	
CLAMDFFT	
CMAKE	
CUDA	
Caffe	
ENABLE	
GIGEAPI	
HDF5	
INSTALL	
OPENCV	
OPENEXR	
PYTHON2	
PYTHON3	
Tesseract	
WITH	

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Unix Makefiles

```
  Tests and samples:
    Tests:                     NO
    Performance tests:         NO
    C/C++ Examples:            NO

  Install path:                /usr/local

  cvconfig.h is in:            /home/karthikeyan/SDK/Packages/opencv-3.1.0/build
-----------------------------------------------------------------

Configuring done
```

Figure 3: Developing Application with Tara using OpenCV

The configurations to be followed are as follows:

   a. Provide the path for extra modules as, OPENCV_EXTRA_MODULES_PATH: <Dir>/opencv_contrib-3.1.0/modules.
   b. Uncheck WITH_LIBV4L option.
   c. Check WITH_TBB & BUILD_TBB option.
   d. After configuring, press the **Generate** button if there are no errors. The Makefiles will be generated and close the GUI.

3. Build and install the OpenCV libraries into the file system using the following command.

```
$ sudo make -j4 install
```

The OpenCV libraries is build and installed in **/usr/local/lib/** location.

Commonly used Functions in SDK

The commonly used functions are provided as a single class for modularity, to reuse them in examples. The common functions used in SDK are loading the calibrated files from the camera flash, computing disparity map, rectification of frames, camera enumeration and initialization, and so on. The Extension unit commands are built as separate library for modularity. For further details, please refer to *Tara_Linux_API_Manual.pdf* document.

Applications in SDK

In this section, the applications in SDK are explained in detail.

Face Detection

The Face Detection application is explained in detail as below.

Use Case

This application is used to measure the distance of the person standing in front of the camera. This application detects multiple faces using the Haar Cascade classifier of OpenCV. If there are any faces in the scene, the detected faces are marked with a rectangle and the distance of the detected face from the camera is displayed on the input image.

Note: This application uses the haarcascade_frontalface_alt2.xml file in the Face folder inside the installed directory for detecting the faces.

Environment

An environment with normal lighting conditions is sufficient.

Running the Application

Please refer to Executing the Samples section and replace FaceDetection in BinaryName.

Test Cases

The full face must be visible in both the frames. The tilt of face to certain extent is detected.

Result

The result of Face Detection application is shown below.

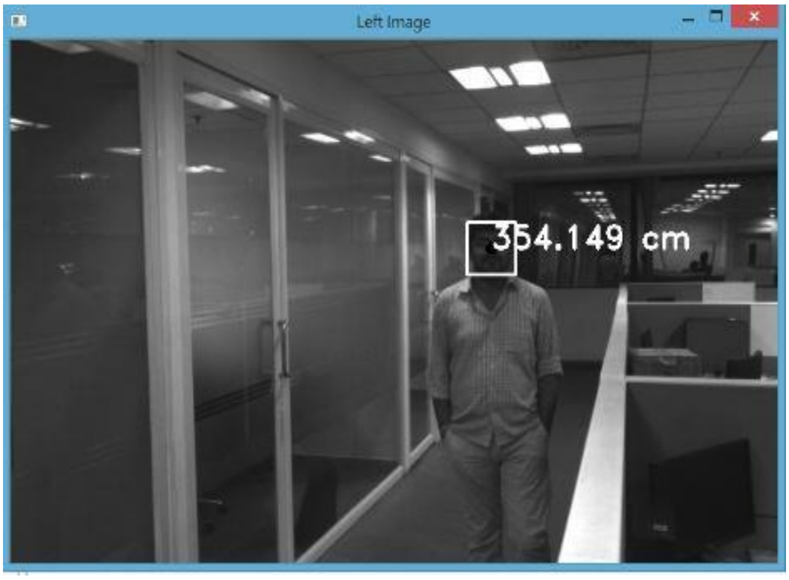


Figure 4: Face Detection

Height Calibration

The Height Calibration application is explained in detail as below.

Use Case

To measure the height of a person from the camera, there must be some reference value (Base Height - Distance of the base from the camera). The Height Calibration application measures the height of a person from the camera. You can select a base point, and the application calculates the Base Height value and save it in a BaseHeight.txt text file inside the Height directory's folder. This text file is later used by the Height Estimation application.

Environment

To obtain a perfect depth measurement of the base, you must use some textured element in the base and select those points to measure the depth. The depth deviation directly depends on the texture of the point selected.

Running the Application

Please refer to Executing the Samples section and replace HeightCalibration in BinaryName.

Result

The result of Height Calibration application is shown below.

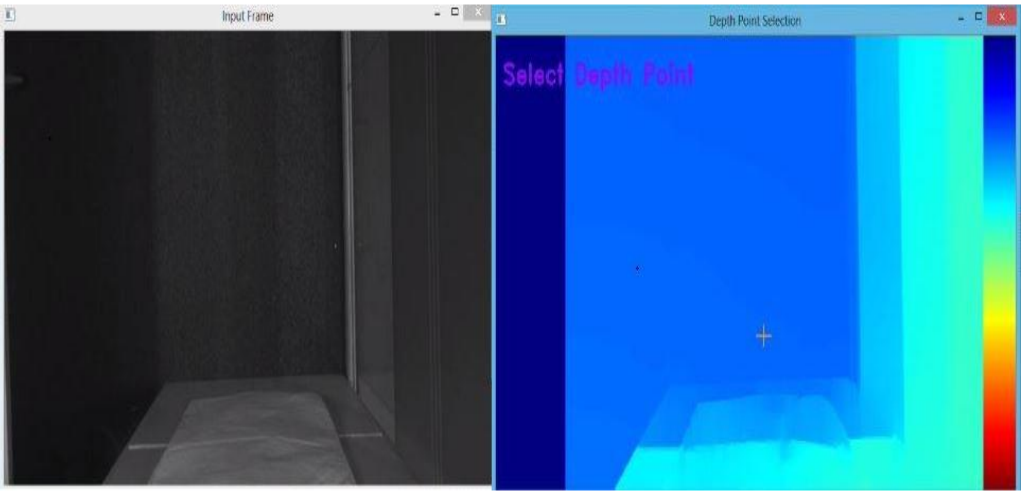


Figure 5: Input Frame and Depth Point Selection Screen

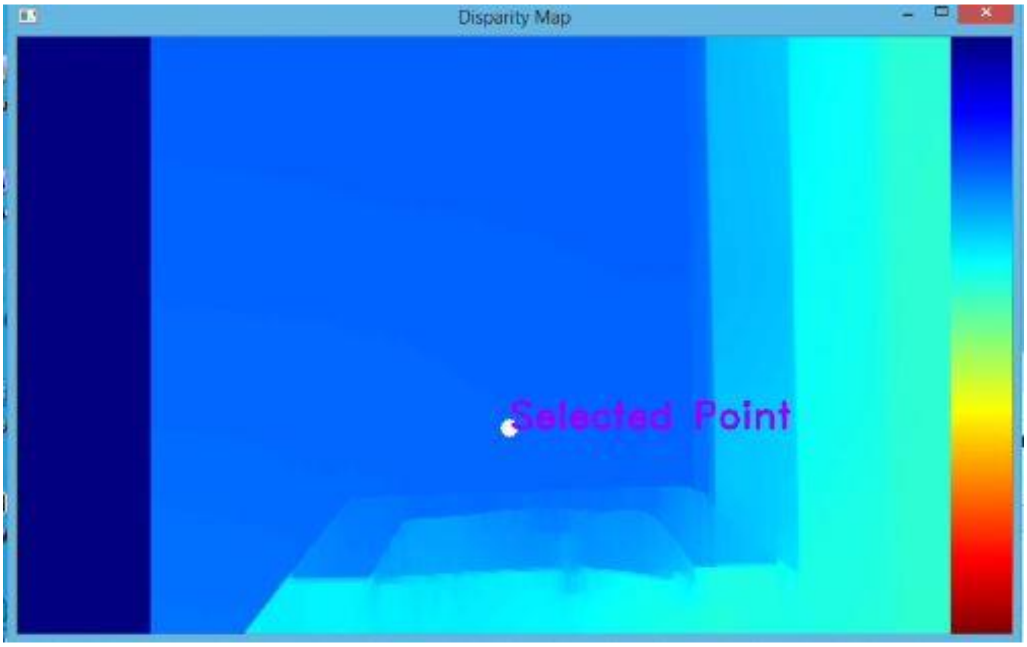


Figure 6: Disparity Map of Height Calibration

Height Estimation

The Height Estimation application is explained in detail as below.

Use Case

This application estimates the height of a person standing under the camera. The head height is estimated from the disparity map and the person height is given with the reference taken from the base using the Height Calibration application.

Note: This application uses the BaseHeight.txt file in the Height folder inside the installed directory for estimating the height.

Environment

The depth is estimated from the disparity map of the person standing under the camera. The point which is used as the head height is selected by scanning the lowest depth in the 1/3rd of the image. The head of the person must be in the 1/3rd of the image, to avoid scanning of full image. To have a clear disparity map, the camera must be 0.5m above the head.

Running the Application

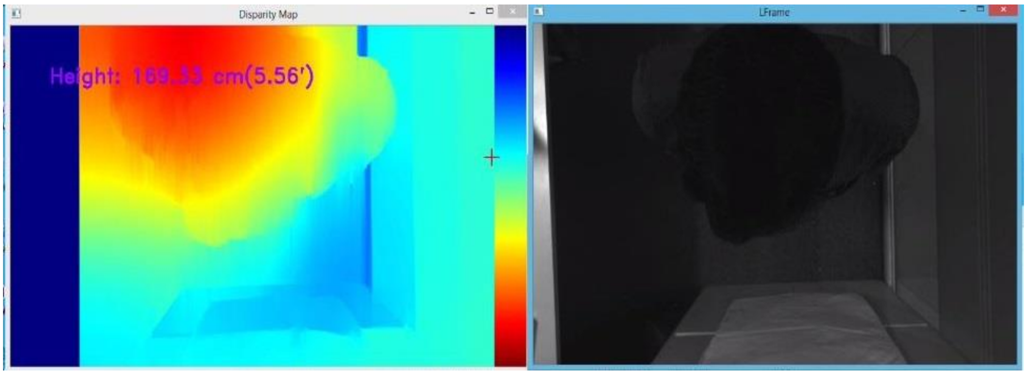
Please refer to Executing the Samples section and replace HeightEstimation in BinaryName.

Test Cases

Validation such that the scene has a person standing below the camera is not done. The head height is subtracted from the base height to know the height of the person. Hence Base Height file in the Height folder is mandatory, otherwise the application will produce an error. The height of the person depends on the base height measured.

Result

The result of Height Estimation application is shown below.



Height: 169.33 cm(5.56')

Figure 7: Height Estimation

Volume Estimation

The Volume Estimation application is explained in detail as below.

Use Case

This application estimates the volume of the box placed in the scene using the Simple Edge Detection technique. Initially, the distance of the base is estimated by averaging the depth taken at a point in few frames. To have an accurate depth measurement, the base must have a proper texture.

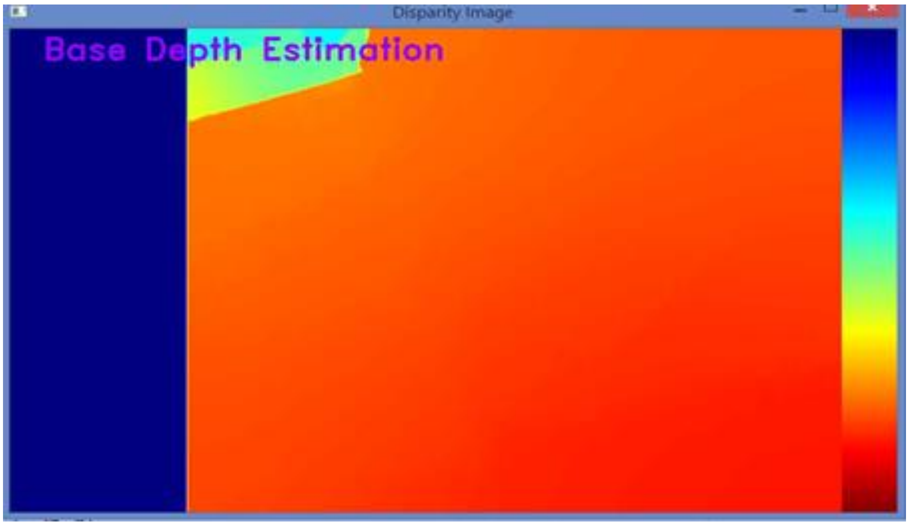


Figure 8: Disparity Map of Volume Estimation

With the reference to the base height estimated in the previous stage, the height is found.

Environment

To have an accurate depth measurement, you must use some textured element placed in the base for the base depth estimation stage. The depth deviation directly depends on the texture of scene, since the height of the box is estimated using the base height measured.

Running the Application

Please refer to Executing the Samples section and replace VolumeEstimation in BinaryName.

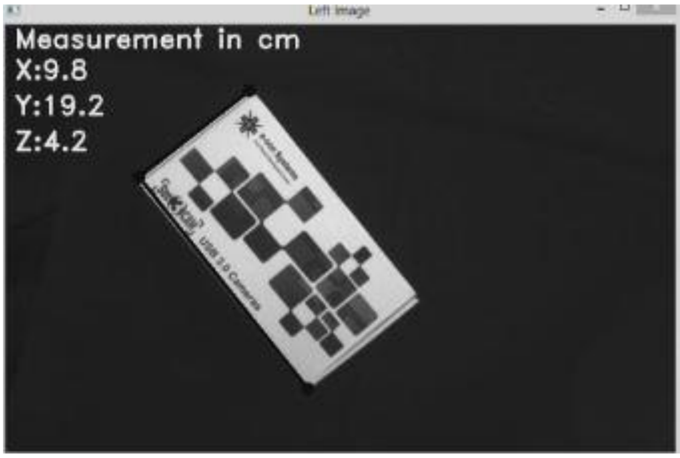
Test Cases

The scene must have only the box to be measured, since the application does not validate for a rectangular object.

Result

The result of Volume Estimation application is shown below.

© Copyright e-con Systems. 2017. All rights reserved.



```
Measurement in cm
X:9.8
Y:19.2
Z:4.2
```

Figure 9: Volume Estimation

TaraCAMViewer

The TaraCamViewer application is explained in detail as below.

Use Case

The OpenCV application is used to stream the Tara camera. This Stereo vision camera provides two synchronized sensor frame data interleaved side by side to the host machine over USB 3.0 interface. The TaraCamViewer application uses the Tara layer (libecon_tara.so) to grab the interleaved frame using OpenCV and splits the interleaved data into left and right camera output.

Environment

The application streams the scene whatever it is viewing under any environment conditions.

Running the Application

Please refer to Executing the Samples section and replace TaraCamViewer in BinaryName.

Result

The result of TaraCamViewer application is shown below.

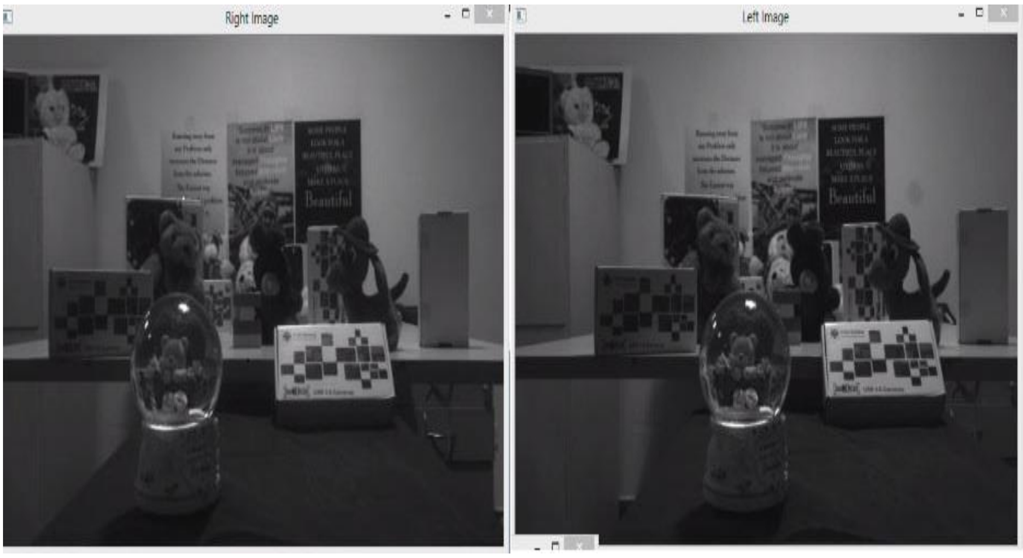


Figure 10: Right and Left Image of TaraCAMViewer

TaraDepthViewer

The TaraDepthViewer application is explained in detail as below.

Use Case

This application is used to measure the depth of the person selected point. The disparity map refers to the apparent pixel difference or motion between the left and the right image. The application displays the disparity map, and the left and right frame. When a point is selected in the disparity map, the depth of the point from the camera will be displayed.

Environment

To have an accurate depth measurement, the point selected must be textured. The depth deviation directly depends on the texture of the point selected.

Note: The operating depth range of the camera is from 50 to 300 cm.

Running the Application

Please refer to Executing the Samples section and replace TaraDepthViewer in BinaryName.

Result

The result of TaraDepthViewer application is shown below.

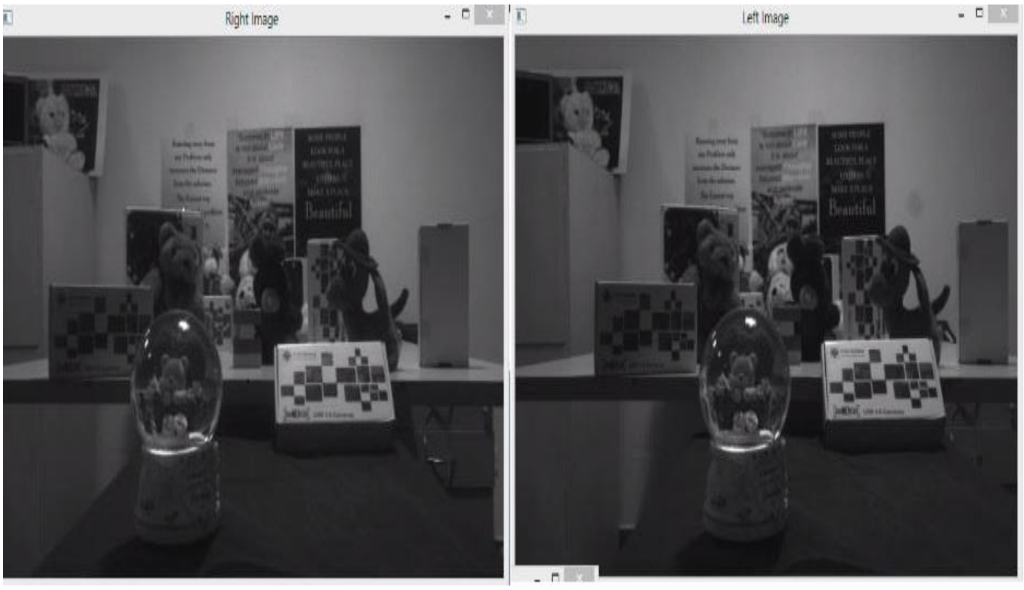


Figure 11: Right and Left Image of TaraDepthViewer

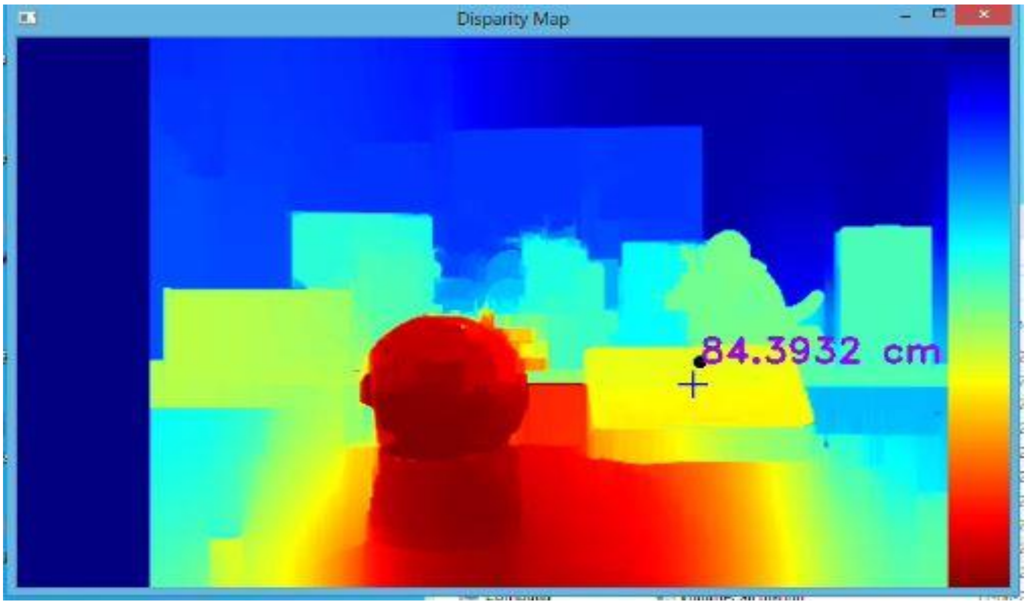


Figure 12: Disparity Map of TaraDepthViewer

TaraDisparityViewer

The TaraDisparityViewer application is explained in detail as below.

Use Case

This application is used to display the disparity map measured from the pair of stereo images. The disparity map refers to the apparent pixel difference or motion between the left and right images. This application displays the left, right and disparity map without any filtering being applied.

Environment

To have a better disparity map, the scene must have textured objects.

Note: The operating depth range of the camera is from 50 to 300 cm.

Running the Application

Please refer to Executing the Samples section and replace TaraDisparityViewer in BinaryName.

Test Cases

The objects closer to the camera appear Red in color whereas the farther objects appear in Blue color. The scale of color map in the right side of the disparity map is relative to the distance of the object from the camera (varies from Red to Blue color).

Result

The result of TaraDisparityViewer application is shown below.

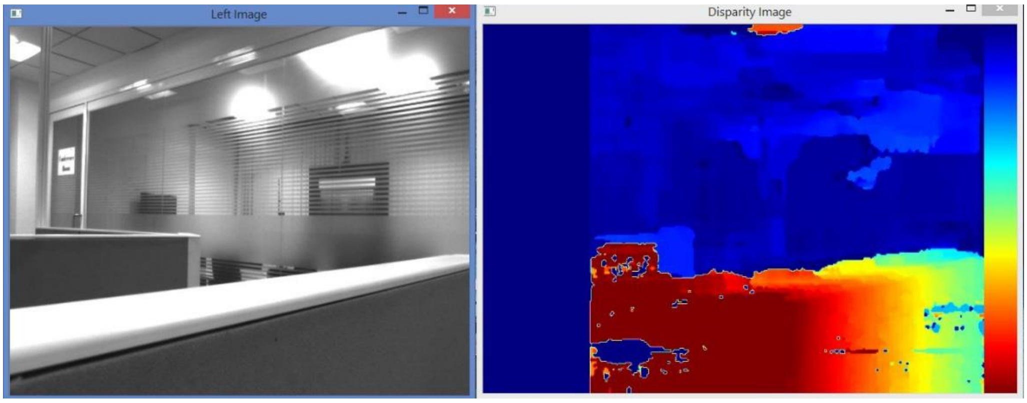


Figure 13: Left Image and Disparity Image of TaraDisparityViewer

Point Cloud

The Point Cloud application is explained in detail as below.

Use Case

This application is used to display the Point Cloud sample of the scene. Using the disparity map computed from the stereo pair images, a 3D point cloud is rendered in the window using the PCL libraries. The Point Cloud data in the window can be rotated.

Running the Application

Please refer to Executing the Samples section and replace PointCloud in BinaryName. In Ubuntu 16.04 version, launching the point cloud application, initially

© Copyright e-con Systems. 2017. All rights reserved.

displays zoomed in axis image. To get to the original data, either scroll the mouse key or press “ALT + r” in the **3d viewer window.**

Result

The result of Point Cloud application is shown below.

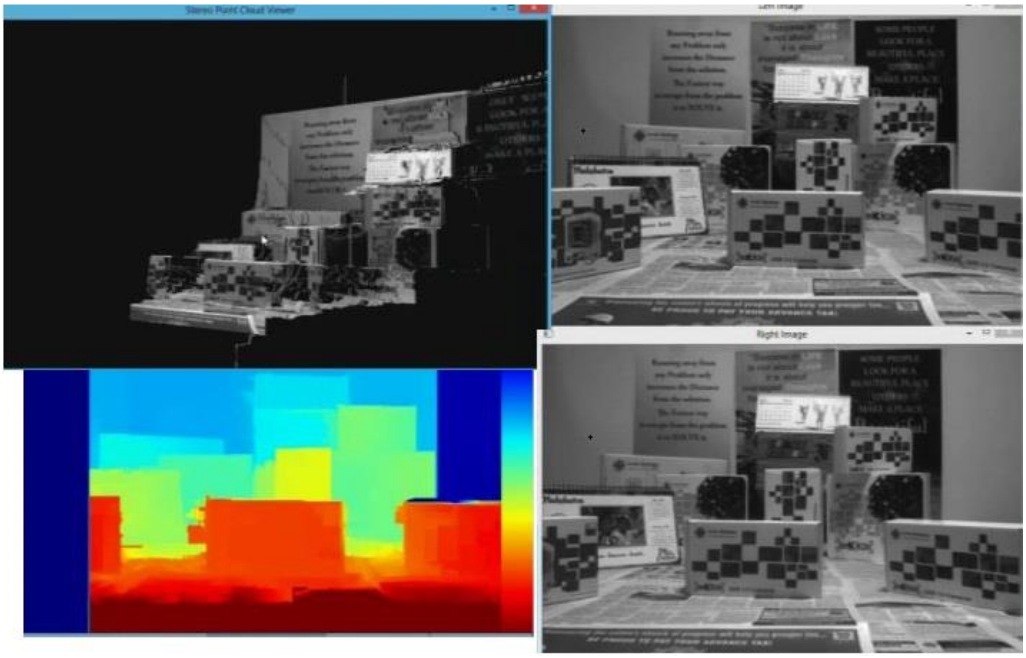


Figure 14: Point Cloud

IMU

The IMU application is explained in detail as below.

Use Case

This application is used to illustrate the IMU unit (LSM6DS0 (or) LSM6DS33) that is integrated with the stereo camera-Tara. The IMU sample application is a basic example demonstrating the rotations of camera around x-axis and y-axis. The output rotation angles calculated from the IMU values are limited to the range from -90 to +90 degrees for illustration. It displays a window where the rotation of the device is plotted for all the three axes respectively.

Environment

This IMU application does not depend on the scene, and it is based on the camera rotation around x-axis and y-axis, so there is no need for any specific environment.

Running the Application

Please refer to Executing the Samples section and replace ImuApplication in BinaryName.

Result

The result of IMU application is shown below.

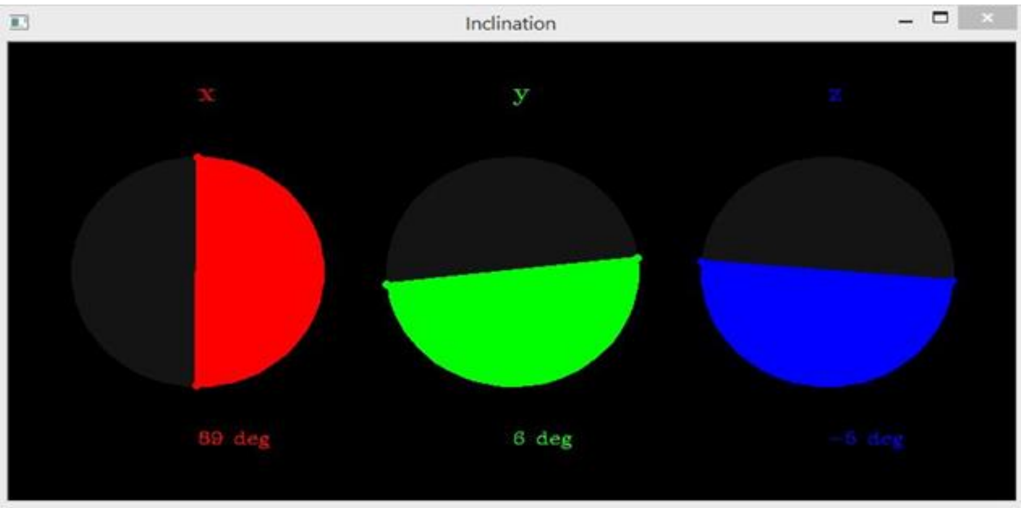


Figure 15: Inclination Screen of IMU

Known Issues

Tara camera also has an option to output the 10-bit monochrome format. The format is given as RGB-24 from the camera.

Pixel arrangement: Byte 1 - X X X X M1 M0 S1 S0 Byte 2 - M9 M8 M7 M6 M5 M4 M3 M2 Byte 3 - S9 S8 S7 S6 S5 S4 S3 S2.

Where, Key - X - Don't care, M - Master, S - Slave.

PointCloud application will initially display the zoomed in image of the XYZ-axis point.

The UVC supports BGR3 format from the kernel version 4.0. To get the 10-bit data, the kernel must be updated to the higher versions above 4.0.

Commands to Update the Kernel

The commands to update the Kernel are as follows:

1. Update the kernel version to 4.2 to support the RGB format in uvc layer using the following command.

```
$ sudo apt-get update
$ sudo apt-get install linux-image-generic-lts-wily
```

2. Reboot the system to reflect the changes using the following command.

```
$ sudo reboot
```

© Copyright e-con Systems. 2017. All rights reserved.

Troubleshooting

In this section, you can view the list of commonly occurring issues and their troubleshooting steps.

Error: haarcascade_frontalface_alt2.xml File Missing in the path /usr/local/tara-sdk/bin/Face! in Face Detection Application.

The steps to be followed are:

1. You need to create a **Face** folder in **/usr/local/tara-sdk/bin** binary location, if the folder is not present.

   ```
   $ mkdir /usr/local/tara-sdk/bin/Face
   ```

2. You must copy the **haarcascade_frontalface_alt2.xml** file from the package to **/usr/local/tara-sdk/bin/Face location**.

   ```
   $ cp <Extracted Package
   Location>/Tara_SDK_LINUX_REL_PACKAGE_xxxx/Prebuilts/Ub
   untu-
   16.04/binary/Face/haarcascade_frontalface_alt2.xml
   /usr/local/tara-sdk/bin/Face
   ```

   Then you must run the application.

Error: BaseHeight.txt file creation in the path /usr/local/tara-sdk/bin/Height failed!! in Height Calibration Application.

You need to create a **Height** folder in **/usr/local/tara-sdk/bin** binary location, if the folder is not present.

```
$ mkdir /usr/local/tara-sdk/bin/Height
```

Then you must run the application.

Error: BaseHeight.txt Reference File Missing in the path /usr/local/tara-sdk/bin/Height in Height Estimation Application.

The steps to be followed are:

1. You need to create a **Height** folder in **/usr/local/tara-sdk/bin** binary location, if the folder is not present.

   ```
   $ mkdir /usr/local/tara-sdk/bin/Height
   ```

2. You must run the Height Calibration Application to get the text file.
3. Then run the Height Estimation Application**.**

GrabFrame: No Frame Received! Camera is Unavailable! error.

You must check whether the Camera is not being used by other applications.

Launching point cloud application shows the zoomed in image of the XYZ-axis.

Yes, initially while launching the point cloud application, the 3d viewer will show the zoomed in image of the axis point. Press “**ALT + r**” key or Scroll the mouse key in the 3d viewer window to get the original data from the camera.

FAQ

1. **Is the Tara camera pre-calibrated?**

   Yes, the camera is factory calibrated. Do not disturb the casing or the lens, which would alter the calibration.

2. **Where is the calibrated data of the camera stored?**

   The calibrated data is stored in the flash memory of the camera. The SDK applications get access through the extension unit APIs.

3. **What is IMU? How do we configure it?**

   Inertial Measurement Units (IMUs) is a self-contained system that measures linear and angular motion with a triad of gyroscopes and triad of accelerometers and sometimes the magnetic field surrounding the body, also called as magnetometers.

   Tara supports a LSM6DS0 or LSM6DS33 chip which is a 6dof (degree of freedom) IMU featured with triaxial accelerometer and triaxial gyroscope and supports different modes of configuration. To configure the IMU please refer to the *Extension Unit API* document.

4. **What is the calibrated file's format?**

   The camera is calibrated using OpenCV. The calibrated files are in the yml format.

5. **What are the output formats supported in the Tara Camera?**

   The output formats supported in Tara are Y16 and RGB24.

   In Y16 (8bpp) format, the Pixel arrangement are:

   - Byte 1 - M9 M8 M7 M6 M5 M4 M3 M2.
   - Byte 2 - S9 S8 S7 S6 S5 S4 S3 S2.

   In RGB24(10bpp) format, the Pixel arrangement are:

   - Byte 1 - X X X X M1 M0 S1 S0.
   - Byte 2 - M9 M8 M7 M6 M5 M4 M3 M2.
   - Byte 3 - S9 S8 S7 S6 S5 S4 S3 S2.

   Note: Key - X - Don't care, M - Master, S - Slave.

6. **What is Tara SDK?**

   Tara SDK package, built on OpenCV (opencv-3.4.1) Image Processing Library is bundled with Tara - Stereo Vision USB 3.0 Camera. SDK currently uses C++ APIs of OpenCV, hence the applications are written in C++. Tara SDK includes the commonly used stereo camera functions such as disparity, depth measurement, and few real-time applications implemented in OpenCV.

7. **Is the SDK provided with the source code?**

   Yes, the Tara SDK package is provided with the source code.

8. **What is the Disparity algorithm used in the SDK?**

   The two algorithms for computing the Disparity are as follows:

   - Best quality, Lower frame rate - Stereo_SGBM (Semi-global block matching) - 3 Way generic Left and Right.
   - Less quality, Higher frame rate Stereo_BM (Block matching) - generic Left and Right.

   There is a define (DISPARITY_OPTION) in the header file which can be altered to switch between the Disparity algorithms. After changing the disparity option, the source must be built again to reflect the changes.

   Note: The Default value is set to 1.

Disparity Option Value	Disparity Algorithm
1	Stereo_SGBM 3-way generic Left and Right
0	Stereo_BM-generic Left and Right

9. **What is the procedure to be followed while building OpenCV?**

   The procedure to build OpenCV is specified in *Building_SDK_Solutions.pdf* document which is given along with the Tara SDK package.

10. **Is it mandatory to have OpenCV, to test the SDK samples?**

    No, the library files provided in Tara SDK package is enough to run the applications. The executable and the library files must be placed in the same location.

11. **Is the SDK applications use rectified images?**

    Yes, the applications in the SDK use rectified images. Rectified such that there is only displacement in the x axis.

12. **In the SDK, can the Exposure and Brightness of the camera be altered in run time?**

    Yes, the Exposure and Brightness of the camera can be altered. On the image window of the application, press e/E to change the exposure and the Exposure range is from 10 to 1000000 micro seconds or press b/B to change the brightness and the brightness range is from 1 to 7.

13. How to modify the Tara SDK to auto select the Device ID and Resolution ID?

The source of Tara library must be modified to start streaming directly with Tara. The steps to build the library file are as follows:

1. Select the Device and Resolution from the API named CameraEnumeration::GetDeviceIDeCon in Tara.cpp.
2. The last two lines of the method is the resolution selection. Replace the Resolution ID with the value of the resolution required as listed as follows:

   - Resolution ID 0 for 752 x 480
   - Resolution ID 1 for 640 x 480
   - Resolution ID 2 for 320 x 240
3. To select the Stereo camera by default, you can compare the name of the device with the friendly name See3CAM_Stereo and assign that ID to the variable Device ID of that method. The Snippet to Compare the device name with the friendly name is also in the same method for your reference.

   Note: Setting the Stereo Camera to stream by default will have another drawback when multiple Stereo Cameras are connected that is the first enumerated Stereo Camera will be initiated by default.

What’s Next?

After understanding the usage of SDK, you can refer to the following documents to understand more about Tara.

- *Tara SDK Installation Manual Windows*
- *Tara Linux Extension Unit API Document*

Glossary

QVGA: Quarter Video Graphics Array (320 x 240 resolution).

USB: Universal Serial Bus.

USB 2.0: Universal Serial Bus High speed.

USB 3.0: Universal Serial Bus Super speed.

UVC Compliant: USB Video Class Compliant.

VGA: Video Graphics Array (Industry name for 640 x 480 resolution).

WVGA: Wide Video Graphics Array (752 x 480 resolution).

Support

Contact Us

If you need any support on Tara product, please contact us using the Live Chat option available on our website - https://www.e-consystems.com/

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - https://www.e-consystems.com/create-ticket.asp

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - https://www.e-consystems.com/RMA-Policy.asp

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - https://www.e-consystems.com/warranty.asp

© Copyright e-con Systems. 2017. All rights reserved.

Revision History

Rev	Date	Description	Author
1.1	12-Jul-2016	Initial Draft	Karthikeyan.K
1.2	09-Aug-2016	Added Build system	Karthikeyan.K
1.3	12-Sep-2016	Added Auto Exposure and Trigger Support	Karthikeyan.K
1.4	15-Dec-2016	Added Exposure Supports for all applications	Karthikeyan.K
1.5	08-06-2018	Added the support for Ubuntu 16.04, Opencv version 3.4.1 and IMU Revision B change	Chandra Sekar.V
1.6	04-07-2018	Added pointcloud behavioural key commands to get the original data	Chandra Sekar.V