Vision SDK

Surround Vision Demos

User Guide

Information in this document is subject to change without notice. Texas Instruments may have pending patent applications, trademarks, copyrights, or other intellectual property rights covering matter in this document. The furnishing of this document is given for usage with Texas Instruments products only and does not give you any license to the intellectual property that might be contained within this document. Texas Instruments makes no implied or expressed warranties in this document and is not responsible for the products based from this document

TABLE OF CONTENTS

1 Introduction

This document is a user guide for TI's Surround Vision Demos that are based on the same 4-camera surround configuration:

- Various Versions of **Surround View** (SRV):
  - 2D Surround View on TDA2xx and TDA3xx
  - 3D Surround View with SGX on TDA2xx
  - 3D Surround View with LDC on TDA3xx and TDA2Px
- **3D Perception** Demo using Structure from Motion (SfM).

IMPORTANT NOTE: In this document, only the 3D Surround View and 3D Perception demos are covered. For 2D Surround View demos, please refer to user guide in subfolder "2d_calibration_tool".

This document assumes that the reader is familiar with VisionSDK build procedure, board configurations, required hardware (such as EVMs, cameras, power supply etc.). This document provides details of calibration procedure, once the images have been collected from the Surround View setup.

1.1 Outline

Chapter 2 provides an overview of the various types of demos supported by this document. Chapter 3 covers the hardware & software setup, Chapter 4 describes how to get the demos up and running.

1.2 Abbreviations

DSP	Digital Signal Processor
EVM	Evaluation Module
EVE	Embedded Vision Engine (A vector processing core)
LDC	Lens Distortion Correction HWA
SDK	Software Development Kit
SfM	Structure from Motion
SGX	Graphics Processing Unit (GPU)
SoC	System of Chip
SRV	Surround View
TDA	TI Driver Assistance line of processors

2 Overview

TI's Surround Vision Demos utilize a set of 4 fisheye (wide field of view) cameras mounted around a car to monitor the 360 degree surroundings. The video streams are processed by TI's TDA (TI Driver Assistance) processors to provide advanced driver assistance (ADAS). Currently, these demos are Surround View (SRV) demos and the 3D Perception demo using Structure from Motion (SfM).

Surround View takes the 4 camera streams and stitches them together to provide a realistic image of the surroundings to the driver.

3D Perception uses the 4 camera streams for – what is commonly called – Structure from Motion (SfM) to reconstruct a 3-dimensional representation of the environment providing a map, e.g. of occupied versus free space. This information enables semi-autonomous driving, e.g. in a parking lot/deck.

All these demos share the same camera setup and calibration procedures, for which reason they are combined here in one user guide. However, their objectives, processing and algorithm requirements are quite different. In the following subsections, we provide brief introduction to each demo.

2.1 Surround View Demos

The surround view demos on the different TI platforms enable the capability of de-warping and stitching multiple live video feeds from fish-eye cameras to synthesize a virtual view of the surroundings of the vehicle. One key step in enabling such stitched videos is to ensure correct camera calibration. Once the cameras calibration parameters are estimated, we assume that they stay constant. This is true for all camera setups until the relative physical locations cameras change.

For a given set of cameras and fixed locations on the vehicle, we need to understand the intrinsic and extrinsic camera parameters. The intrinsic parameters such as focal length and fish-eye lens model are fixed for a given type of camera module and are typically provided by the lens vendor. The extrinsic camera parameters on the other hand are specific to camera locations on each vehicle that they are mounted on. The tool accompanying this document can be used to estimate the extrinsic parameter.

Once the camera calibration parameters are known, depending on the version of Surround view (2D/3D and GPU/DSP/LDC Hardware based), a Look-up table encoding the image pixel location mapping for given set of calibration matrices is generated and stored in DDR (memory). This LUT is read on-the-fly at run time along with the input video feeds from multiple cameras. The LUT and video feeds are then passed to a synthesizer module, whose exact implementation depends on the respective surround view incarnation, to create the stitched surround view output on the screen.

On TDA2xx platform, GPU based 3D surround view demo can be run using both 1 megapixel and 2 megapixel cameras. The calibration needs to be done separately for both demos.

2.2 3D Perception Demo (Structure from Motion)

3D Perception uses the motion of a camera to reconstruct points in 3D space. This concept is commonly called Structure from Motion (SfM). As the car drives along the road, the cameras on the car are moving with respect to the static scenery. SfM can then be used to reconstruct the scenery and create a map of drivable (free) versus non-drivable (occupied) space.

The basic concept of SfM is a "temporal stereo view". Stereo reconstruction uses the fact that an object seen by two different cameras will create larger disparity as the object gets closer. In SfM, this same concept is employed except that the object is not viewed by two cameras at the same time but by one camera that takes two images in time while moving, thus creating the same stereo view images.

Given these temporal images (video), SfM consists of three steps:

1. Feature point tracking (Sparse Optical Flow) on the video input providing image point correspondences
2. Camera motion estimation
3. 3D point reconstruction (Triangulation)

Sparse Optical Flow is provided in VisionSDK as a library on the EVE (Embedded Vision Engine) core of TDA SoC. In 3D Perception, one camera stream is associated in software with one EVE, so 4 EVEs are required for the 4-camera setup.

The tasks of camera motion estimation and triangulation, as well as mapping are performed on TDA's DSPs.

3 Hardware and Software Setup

TDA2xx Demos:

A detailed description of the demo setup and hardware requirements is given in the user guide in subfolder "2d_calibration_tool" under Section 2 "Demo Setup" for the TDA2xx demos.

IMPORTANT NOTES:

1. 3D Surround View and 3D Perception demos on TDA2xx require EVM Revision E or better. Older revisions do not support these demos.
2. TDA2xx Demos do not require the analytics front cam.

TDA2xx demos require Linux version of VisionSDK 2.9 or higher.

TDA3xx and TDA2Px Demos:

A detailed description of the TDA3x and TDA2Px Surround View demo setup and hardware requirements are given in the user guide in subfolder "./TDA3x_3D_SurroundView/".

TDA3xx and TDA2Px demos require VisionSDK 2.9 or higher.

All demos described in this document are tested on a setup with dimensions listed as per the aforementioned user guide.

4 Running a Demo Usecase

This chapter describes how to get the demo of your choice up and running. We will go through 3 steps:

1. *Run uncalibrated demo usecase*
2. *Calibrate the camera system*
3. *Run calibrated demo usecase*

4.1 Run Uncalibrated Demo Usecase

This sequence of steps ensures that the hardware and software setup is correct and the demo usecase can be executed as expected.

1. Install VisionSDK. Follow instructions in VisionSDK documentation.
2. Compile VisionSDK. Follow instructions in VisionSDK documentation.
3. Make bootable SD card from compiled binaries. Follow instructions in VisionSDK documentation.
4. Connect HDMI display device to EVM.
5. Boot binaries through SD card
6. Choose appropriate usecase through UART terminal prompt
7. You should see live camera streams in some form on the HDMI display. (They do not need to be calibrated yet, but you should see, for example, your hand moving when you waive it in front of one of the cameras).

4.2 Calibrate the Camera System

Camera calibration is necessary for two reasons.

First, the camera model (intrinsic camera parameters) changes with the type of camera and lens being used. **Intrinsic camera calibration** ensures that the correct camera model parameters are being used. In most practical cases, intrinsic calibration needs to be done only once for a setup, as long as the camera modules do not change.

Second, the positions and angles of the cameras relative to the car and to each other vary from setup to setup and change slightly over time or when the cameras are touched, e.g., during transport. The positions and angles of the cameras are called extrinsic camera parameters. The objective of **extrinsic camera calibration** is to estimate the position and angles of the cameras at the current time. Extrinsic calibration for a setup needs to be performed once and then whenever camera positions and angles have deviated from the state when extrinsic calibration was performed last. Typically, "bad" extrinsic calibration is noticed by geometric misalignment in the Surround View output. If this misalignment cannot be fixed by manually moving camera positions and angles back to their calibrated state, extrinsic calibration procedure needs to be performed as described below.

4.2.1 Intrinsic Camera Calibration

Please follow instructions in document

"manual_TI_3D_SurroundVision_CalibTool.docx"

in subfolder "3d_calibration_tool" section 7.1 ("Create File ldc_lut.c").

Going through these steps will provide a file called "ldc_lut.c". In VisionSDK source code, replace file

\vision_sdk\examples\tda2xx\src\alg_plugins\commonutils\ldc_lut.c

with "ldc_lut.c" produced by calibration tool.

Recompile VisionSDK and create bootable SD card.

4.2.2 Extrinsic Camera Calibration

Calibration Setup

The document will explain the calibration procedure for demos that utilize cameras in the scale of the setup suggested in Section 3. This calibration procedure is common for all 3D Surround View and 3D Perception demos. For 2D Surround View demos, please refer to user guide in subfolder "2d_calibration_tool".

Create a 36 inch x 48 inch poster print-out of the chart placed in "./docs/poster_calib_chart.pdf". Place the demo setup with installed cameras at the center of the chart as shown below in Figure 1. The goal for calibration is to ensure that two complete square are visible in each camera image.

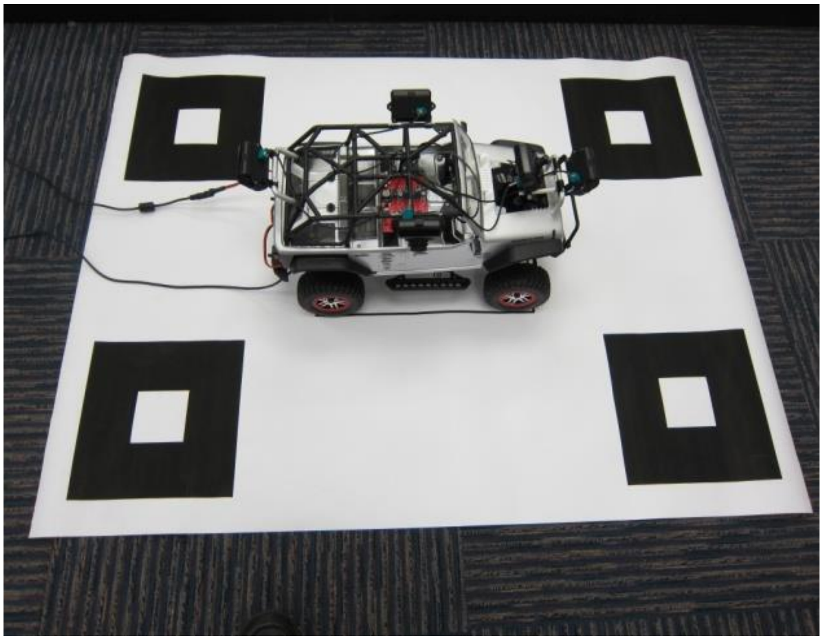


Figure 1: Surround view and 3D Perception Demo Calibration setup

The cameras should be placed such that the patterns lie in the box highlighted in red for each camera image to ensure good performance. An illustration is shown below:

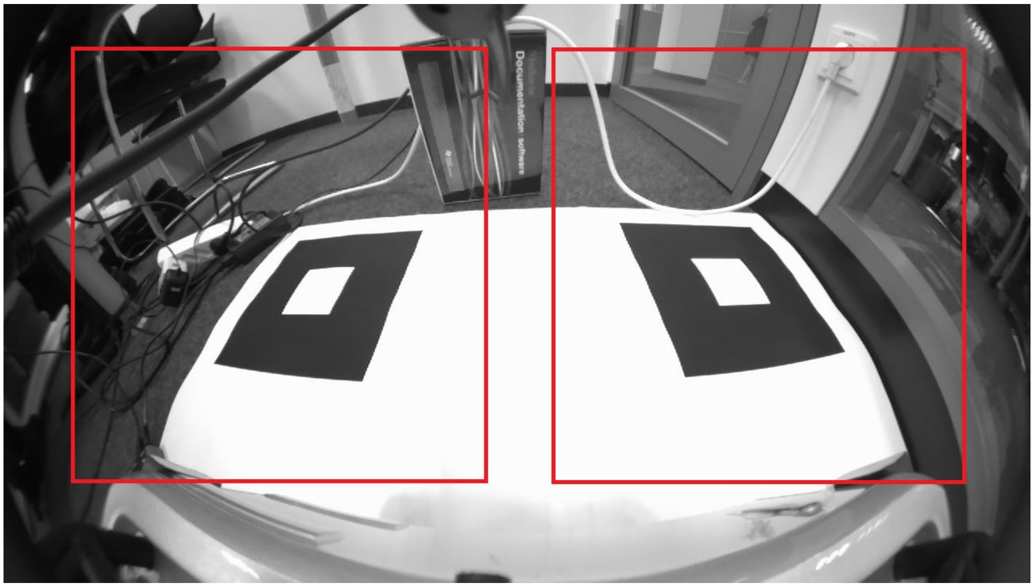


Figure 2: Sample input image with calibration chart and red guidelines

The next steps in the calibration procedure will vary based on the demo. Please follow instructions specific to your demo usecase.

3D SRV (SGX), 3D Perception

Ensure Section 4.2.1 has been completed.

Load binaries with calibration mat present.
Choose usecase (c: 4CH VIP LVDS capture + Auto Calibration + 3D SRV (SGX/A15) + SGX/DRM DISPLAY(A15) - Only HDMI 1080p display supported) for automatic calibration.

Automatic calibration is being executed and should take less than 2 minutes to complete. You will see a gray screen while calibration is performed. Autocalibration usecase detects the chart corners, estimates the extrinsic camera parameters (camera positions and angles) and stores them on the SD card.

If Autocalibration detects the chart pattern, you will see the 3D SRV (SGX) usecase after ca. 2 minutes. This can be used to visually inspect the calibration quality. The calibration patterns should be well aligned at the stitching lines. If result is satisfying, press "0", "Enter", "3", "Enter", or simply wait for 1 minute. Now calibration result is stored on the SD card and will be picked up automatically by 3D SRV (SGX) or 3D PERCEPTION usecases upon reboot. You are done here, continue with Section 4.3.

If Autocalibration fails (abort or unsatisfying quality), ensure calibration chart setup instructions above are followed closely and reiterate on this subsection.

If Autocalibration keeps failing, offline calibration on PC is available as a backup:

Please follow instructions in document "manual_TI_3D_SurroundVision_CalibTool.docx" in subfolder "3d_calibration_tool" section 7.2 ("Create Extrinsic Calibration Results"). This will provide a file called "calmat.c". Please note that this tool requires the user to dump a calibration frame from each camera to PC. Please refer to user guide in "./2d_calibration_tool", Sec. 2.5.1 for instructions on how to dump frames to PC.

In VisionSDK source code, replace file
\vision_sdk\examples\tda2xx\src\alg_plugins\commonutils\calmat.c
with "calmat.c" produced by calibration tool.
Recompile VisionSDK and make SD card.

On SD card's rootfs partition, delete file "rootfs/home/root/.calibtable"

Run binaries from SD card and choose the demo usecase. The extrinsic camera parameters will now be picked up from the calmat.c file and then be stored again on the SD card. Be sure to <u>NOT</u> execute the Autocalibration usecase if Autocalibration is known to fail, since this might overwrite the calibration tables on the SD card with a bad calibration table.

3D SRV (LDC)

Ensure Section 4.2.1 has been completed. The subsequent steps for TDA3x 3D SRV differ significantly from TDA2x demos. Please refer to the document user guide in "./TDA3x_3D_SurroundView/" for detailed instructions.

NOTE: User guide in "./TDA3x_3D_SurroundView/" contains all necessary steps for TDA3x based 3D Surround View using LDC. One can ignore the remainder of this document for TDA3x 3D Surround View.

4.3 Run Calibrated Usecase

Now that the demo is calibrated, your demo usecase should be in full working condition. You can now run the demo usecase as before by booting from SD card and choosing appropriate usecase option.

IMPORTANT NOTE:
Always ensure that the cameras have not moved after calibration for all demos described herein. If the cameras have moved, you will need to re-calibrate the cameras (extrinsic calibration only).

In this section, some additional demo-usecase-specific instructions are provided, for example:

- Expected output and explanation of the visualization
- Additional features & requirements
- How to use the demo
- Limitations
- Common issues and solutions

Please refer to the section covering your specific demo usecase.

4.3.1 Surround View Demos

3D SRV on TDA2X (SGX)

The 3D SRV on TDA2x additionally has the option of connecting a "Logitech Extreme™ 3D Pro Joystick" to control the virtual viewpoints on the output screen. Connect the Joystick to the USB port on the TDA2x EVM as shown in Figure 3 , below. Figure 4 shows the configuration of the various buttons on the Joystick. Once can transition between certain preset views or switch manually by enabling the on/off switch indicated in the picture below.

Sample output is shown in Figure below, which includes a live stream of the four input fish-eye images and in the central region the rendered 3D Surround View output.

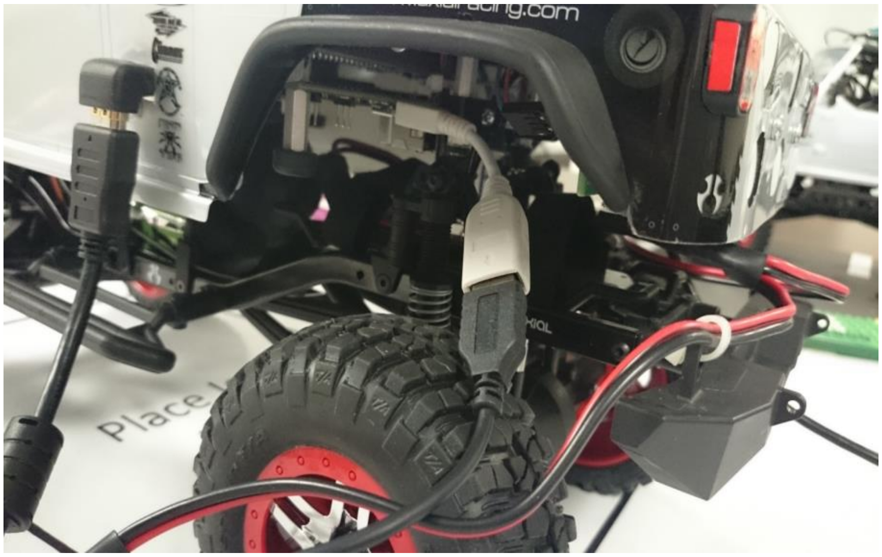


Figure 3: Connect Joystick to USB port on the EVM

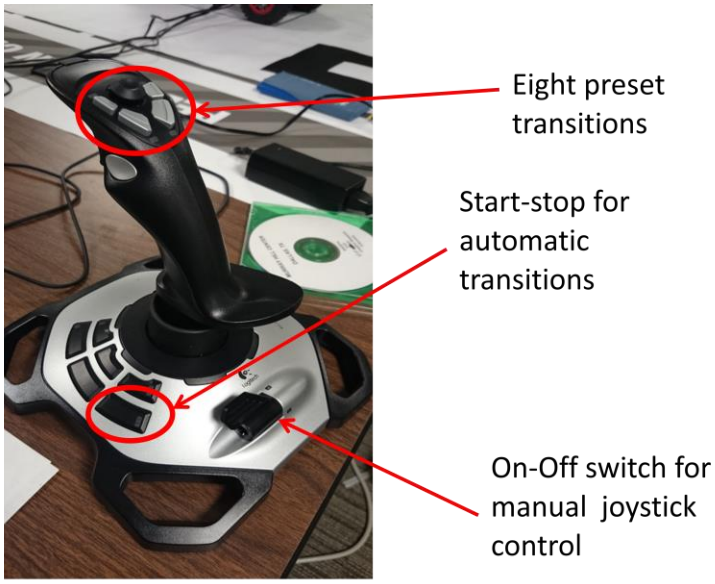

Eight preset transitions
Start-stop for automatic transitions
On-Off switch for manual joystick control

Figure 4: Preset settings on the joystick

Use below the Instructions to get SGX load bar:

1. Run 3D SRV demo
2. Open another terminal (telnet)
3. cd /opt/vision_sdk
4. run “./pvrscope -f 0”

4.3.2 3D Perception Demo

Setup/Environment

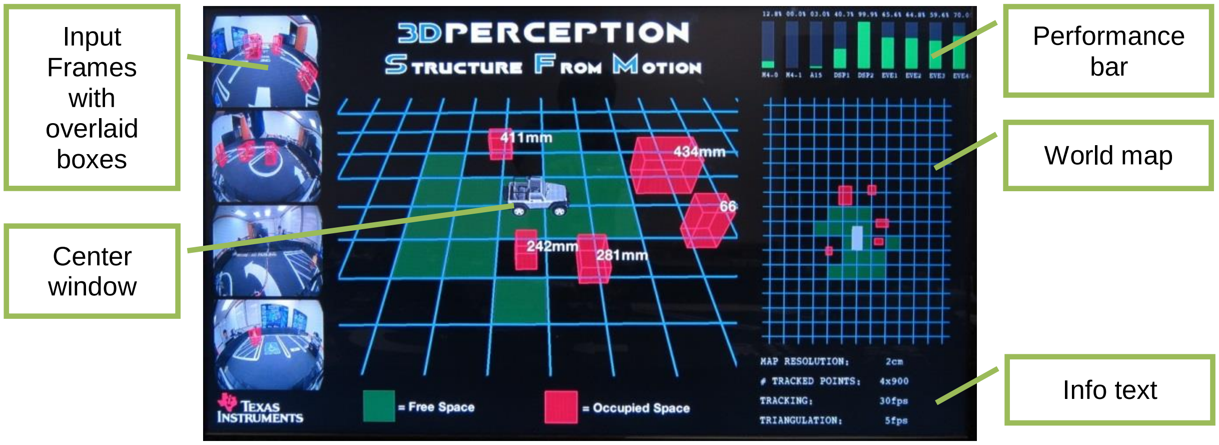
The demo needs a certain amount of visual features in the environment, especially on the ground. It is recommended to use a print of the parking lot design in "./docs/Parking_Poster.pdf", but other textured ground with strong visual features may be OK, too. Also, the environment should not be too bleak (e.g. all uniform walls with no texture, or black curtains are difficult and may result in failure).

Objects/Obstacles can be placed as desired on the ground. There is no "minimum distance" to the vehicle, but the obstacle should be at least 7cm high and have visual features (One-colored uniform texture, or stripes are difficult) to be detected as such.

Make sure that all cables connecting to the vehicle (e.g., power, HDMI, UART) extend from the rear of the vehicle and are not seen in the front, right and left camera views.

Demo Procedure & Output

In Figure 5, the demo output screen is shown. Please refer to this figure for nomenclature of the different windows.



```
MAP RESOLUTION:       2cm
# TRACKED POINTS:   4x900
TRACKING:           30fps
TRIANGULATION:       5fps
```

Figure 5: 3D Perception Demo Output (visual design may vary)

The 3D Perception demo relies on the motion of the vehicle. If the vehicle stands still, the demo is idling while maintaining its state. It is recommended to use a remote control to get the vehicle in motion. Pushing the lower frame of the vehicle may be acceptable, but may adversely affect results. Slow speeds are typically easier for the demo, but fast motion and turns are OK.

As the vehicle is moving, one should observe the vehicle on the center window and the world-map window (gray rectangle) to be moving accordingly. Simultaneously, the demo is collecting information about 3D points using Structure from Motion (SfM). After some time, typically within seconds, it has accumulated enough information to build a map, which is shown in the center and world-map windows as green tiles (free space) and red boxes (occupied space). As the vehicle keeps moving, more and more information is added to the map. If the map does not show up after one drive, try stopping the vehicle and start moving again. The demo was designed for a “stop-and-go” type of motion, which is typical for demo environments.

After 30 seconds of standstill, the map is reset (in software). So, let the vehicle rest for 30 seconds if you want the map to reset and start from scratch. Note that the algorithm is fairly sensitive to motion: Make sure the car is completely at rest for 30 seconds for the timer to work.

The demo screen also shows the input camera streams on the left and overlays detected occupied space (red boxes) on them.

The distance readings in the center window show the closest distance from each red box to the vehicle’s bounding box, it is a conservative estimate for the ”impact distance”.

In general, it is recommended to try a few different scenarios and drive the car for a while to get a feeling of how the demo works before demoing to other people.

Assumptions/Constraints

The demo assumes a static world, so moving world points/objects (through people interfering with the demo) are rejected from consideration. This static world assumption also implies that occupied space will remain classified as occupied space until a reset of the map occurs (after 30 seconds of standstill).

Revision History

Version #	Date	Revision History
00.10	11/March/2016	First draft
00.11	15/March/2016	Updated for release 2.xx
00.12	5/July/2017	Updated for release 3.xx
00.13	16/August/2017	Updated for release 3.01

««« § »»»