Application Note

Akademie věd České republiky
Ústav teorie informace a automatizace AV ČR, v.v.i.

Stereo Demo

Zdenek Pohl, Lukas Kohout, Jiri Kadlec
[zdenek.pohl, kohoutl, kadlec]@utia.cas.cz

Revision history

Rev.	Date	Author	Description
0	11.10.2018	Z.P.	Description of demo

Contents

Acknowledgement

This work has been supported from project FitOptiVis, project number ECSEL 783162 and MSMT 8A18013.

1 Introduction

This document provides description of Stereo Demo. The demo implements “stereoPipeline” example from xfOpenCV library [1] which is applied on live video stream from stereo camera constructed using two Vita 2000 sensors connected to zc702 board from Xilinx. The zc702 provides live synchronous side by side stereo signal in resolution 3840x1080p30 (2x Full HD, half FPS). Demo implements:

1. Collecting of stereo pairs for camera calibration.
2. Detection of chessboard with preview of detected pattern.
3. Computation of camera intrinsic and extrinsic parameters.
4. Preview of undistorted and rectified images with canvas and epilines.
5. Conversion of parameters to proper format for hardware undistortion and rectification of images before stereo matching.
6. Live demonstration of hardware undistortion, rectification and stereo matching of images from video stream. Resulting estimated depth in image is shown in output HDMI. It presents side by side depth estimation and original input from one of camera sensors.

2 Used Tools and Resources

1. TEBF0808 Carrier board from Trenz.
2. TE0808 Module with Xilinx Zynq ULTRASCALE+.
3. Power source suitable for TEBF0808.
4. FMC Imageon extension card from Avnet.
5. Camera setup with Xilinx board ZC702:
    a. ZC702 development board.
    b. 2x FullHD camera with Vita 2000 image sensor
    c. 2x FMC Imageon populated to FMC connectors on zc702
    d. SD or SDHC card with FAT32 filesystem with stereo Vita 2000 camera design
    e. HDMI cable from zc702 to tebf0808
    f. Power source for zc702
    g. Optionally, mini USB cable and 3840x1080p30 capable LCD screen for adjustment of both cameras.
6. HDMI cables to connect video output from tebf0808 to LCD screen
7. 3840x1080p30 capable LCD screen with HDMI input.
8. SD or SDHC card with FAT32 filesystem.
9. Xilinx SDx 2017.1 design suite (for compilation from sources).
10. Petalinux 2017.1 for compilation of Linux image (optional, precompiled image from Trenz reference design can be used)
11. Serial terminal application on PC.
12. Mini USB cable, Ethernet cable (optional for debugging).

3 How to Run Demo

In the first step, stereo camera must be setup, cameras aligned and focused. To do so, please follow these steps:

1) Setup cameras and FMC Imageon boards as shown in the Figure 1.
2) Connect Mini USB cable to J17 (USB UART) on zc702.

3) Connect HDMI cable from HDMI out (both HDMI out connectors on FMC Imageon are providing the same signal) and LCD screen capable to display 3840x1080p30.
4) Insert SD card with stereo camera design to SD card slot on zc702.
5) Make sure that zc702 is set to boot from SD (SW10: 1 – OFF, 2 – ON, SW16: 1,2,5 – OFF; 3,4 – ON)
6) Power on board and connect PC terminal.
7) Use terminal menu to setup gains and manually set focus and iris. Both images left and right should look similar after correct setting.

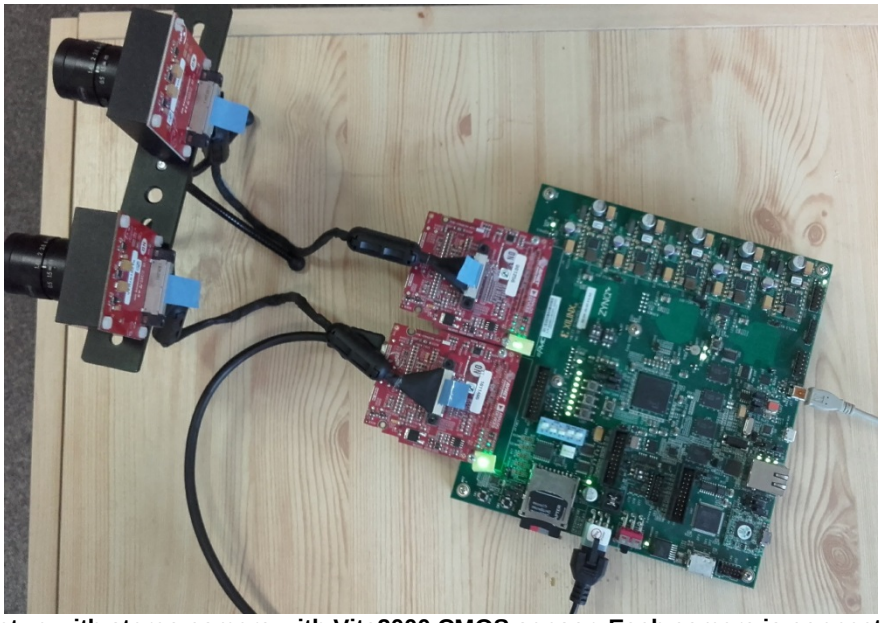

Figure 1: ZC702 setup with stereo camera with Vita2000 CMOS sensor. Each camera is connected to one FMC Imageon extension board on one of ZC702 FMC connectors.

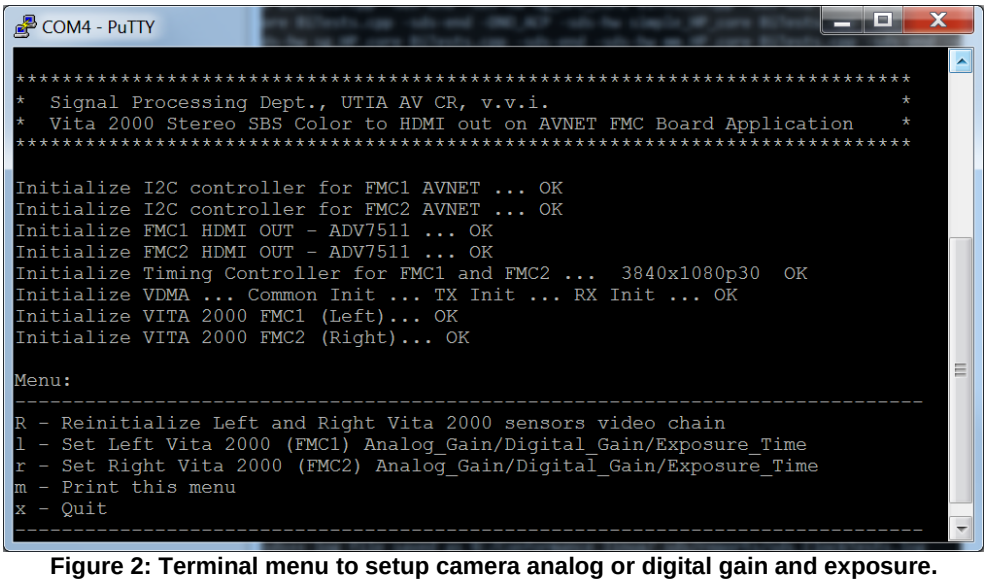

Figure 2: Terminal menu to setup camera analog or digital gain and exposure.

Figure 3: TEBF0808 + TE0808 ES1 module

When ZC702 produces correct side-by-side stereo video signal it is possible to proceed to Stereo Demo:

1. Connect HDMI input of TEBF board (Figure 3) to HDMI output from FMC IMAGEON on ZC702.
2. Copy Stereo Demo design to SD card and place it to SD slot on TEBF. Directory structure of Stereo Demo on SD card is following:
   - **a.** Root directory must contain boot.bin and xxxx.elf file for running the demo.
   - **b.** Folder “calib” stores calibration image pairs, i.e. pairs of images used to compute calibration parameters. It also contains file “info.txt” which stores the number of image pairs to be used for calibration. Example stereo pair for calibration is shown in Figure 4.

Figure 4: Stereo calibration image pair example

   - **c.** Files “intrinsics.yml” and “extrinsics.yml” are holding pre-computed intrinsic and extrinsic parameters computed form present calibration image set.
   - **d.** Folder “snapshots” is used for storing stereo snapshots.
3. Connect HDMI output from TEBF to LCD screen capable to view 3840x1080p30 video.
4. Connect mini USB from XMOD1 module to PC USB port.
5. Power on TEBF board and connect UART terminal to TEBF board.
6. Wait until Linux boots, login as “root”, password “root”, change directory to “/media” and run:

   > stereoPipeline.elf

7. The application will initialize video chain and in case of success following menu on text console is shown, see Figure 5.

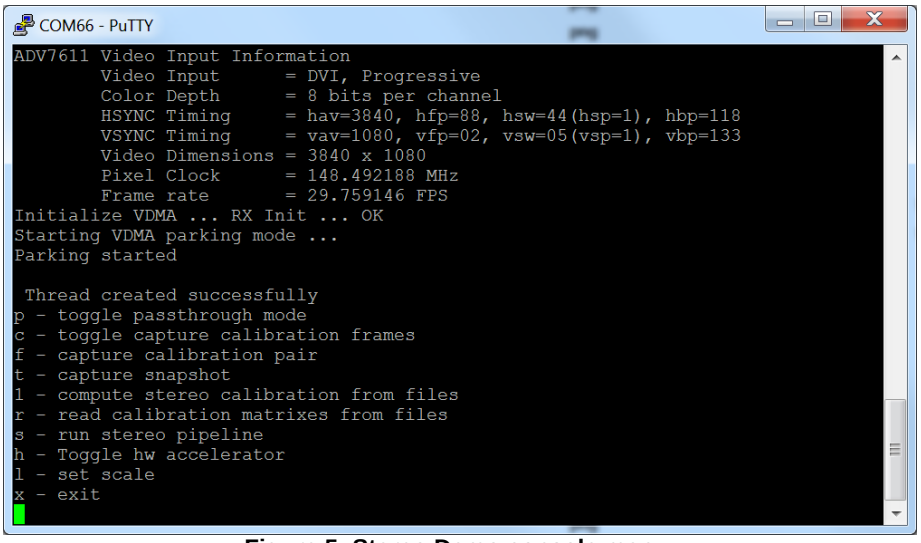

Figure 5: Stereo Demo console menu.

8. Now there are several options how to proceed:

   a. Check input video signal

   Use “p” to turn on passthrought mode. In this mode SBS input from stereo camera is shown directly to output. It is good to check your video signal before trying other steps.

   b. Capture stereo snapshot to files

   You can use option “t” to capture stereo snapshot to “snapshots” folder.

   c. Quick Start of Stereo Demo

   Choose ‘r’ to read pre-computed calibration from “yml” files. Then enter ‘s’ to run Stereo Demo. By default, the demo starts by software stereo matching. Use ‘h’ to toggle to hardware processing.

   NOTE: Software matching produces black/white tones to display estimated depth in image. Hardware accelerator uses color palate to do the same. Frame rate is approx. 23.5 FPS for hardware version.
   IMPORTANT: In this case, the camera extrinsic parameters are reused from file. Thus your camera alignment must be set identically to the configuration when calibration was computed. Check correct alignment by moving camera(s) until good results are shown in output screen – violet color for close objects going faint to distance and turning to green darker for the most distant objects. See example output in Figure 6.

Figure 6: Sample Stereo Demo output, hardware accelerated video processing 23.5 FPS.

d. Recompute calibration from stored stereo pairs

Option ‘1’ starts calibration process using images stored in folder ‘calib’. At the end of the process new ‘intrinsics.yml’ and ‘extrinsics.yml’ file is written to SD card. Process starts by detection of chessboard pattern. Each detected chessboard is shown with preview of corner points to be used for calibration, see Figure 7. When all points are collected, the calibration is computed. In the end of the process, the undistorted and rectified stereo pairs with canvas and epilines are shown, example can be found in Figure 8.

IMPORTANT: Correct camera parameters must be found before using them for stereo matching. Following properties of undistorted and rectified images must be carefully inspected:
- pincushion distortion of images (result of barrel distortion compensation) must be near symmetric, similar for both cameras (as they are using identical lenses), the maximal vertical displacement of pixels must be less than 256 pixels. For orientation epilines are 50 pixels apart. See Figure 9.

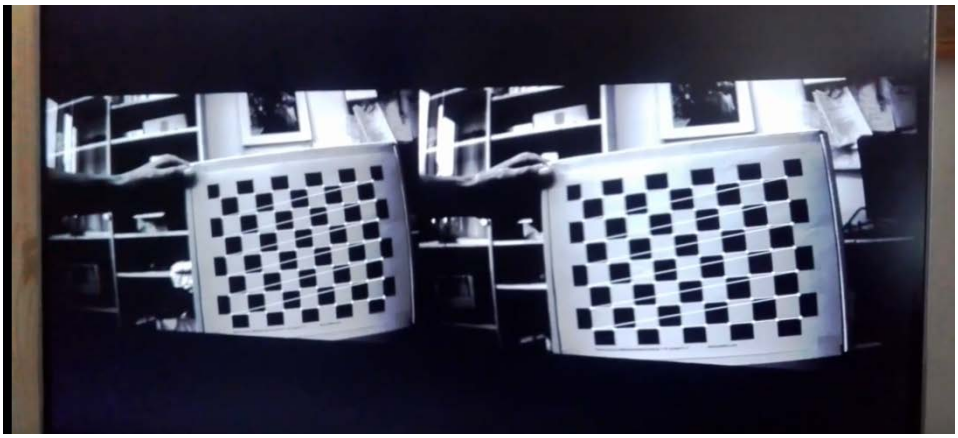

Figure 7: Example chessboard detection.

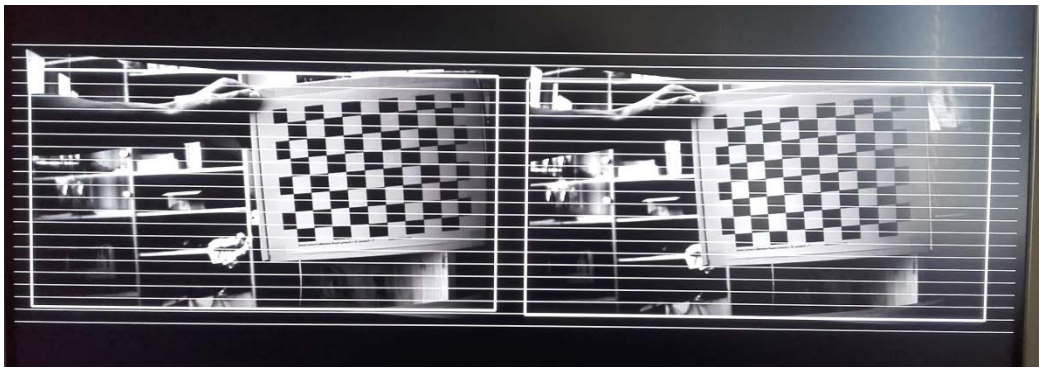

Figure 8: Undistorted rectified stereo pair with canvas and epilines.

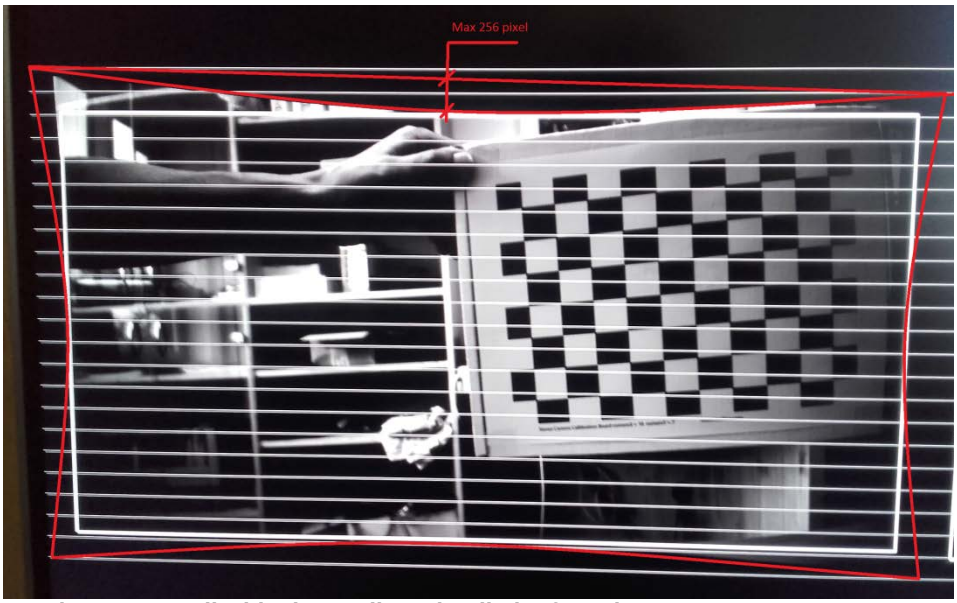

Figure 9: Detail with shown distortion limit of used HW core.

After computation of correct calibration parameters we can continue by step c) **Quick Start of Stereo Demo**.

e. Capture new set of calibration pairs

For this step it is needed to prepare chessboard pattern as shown in calibration images. Pattern must have exactly 10 corners along horizontal axis and 7 along vertical. Pattern must be planar. Then select ‘c’ to start capture calibration mode. In this mode small preview shows image of both cameras with pre-detected pattern to see if pattern can be seen and properly detected from both cameras at the same time. After pattern is properly detected in both images you can enter ‘f’ to capture and store pair to SD card. Only valid pairs with detectable pattern on both sides are stored. When enough calibration pairs were captured we can continue by step d) **Recompute calibration from stored stereo pairs** (previous item of this list).

f. Other possibilities

Option ‘m’ prints menu again to terminal.
Option ‘l’ sets scaling factor for computation of color tone from estimated disparity (depth).

4 Known Issues and Solution

Problem	Solution
Demo freeze after some time using different options in menu	Can be resetted-restarted. GUI is debugged only for setting up demo in the flow : (optional) capture calibration frames -> (optional) compute calibration -> restore parameters -> run stereo pipeline (and toggle hw/sw solution)

5 Package contents

```
Release     – doc           this document
            - sd_card       SDSoC generated SD card content with demo
```

6 Licensing

Evaluation License

The evaluation version of the package can be downloaded from UTIA www pages free of charge for evaluation.

The evaluation package includes only precompiled bitstreams which has been compiled with valid full IP licenses of used IP components.

Full License

To obtain license to full package with sources please contact Jiri Kadlec, UTIA (kadlec@utia.cas.cz).

7 Disclaimer

This disclaimer is not a license and does not grant any rights to the materials distributed herewith. Except as otherwise provided in a valid license issued to you by UTIA AV CR v.v.i., and to the maximum extent permitted by applicable law:

(1) THIS APPLICATION NOTE AND RELATED MATERIALS LISTED IN THIS PACKAGE CONTENT ARE MADE AVAILABLE "AS IS" AND WITH ALL FAULTS, AND UTIA AV CR V.V.I. HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and

(2) UTIA AV CR v.v.i. shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under or in connection with these materials, including for any direct, or any indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or UTIA AV CR v.v.i. had been advised of the possibility of the same.
Critical Applications:

UTIA AV CR v.v.i. products are not designed or intended to be fail-safe, or for use in any application requiring fail-safe performance, such as life-support or safety devices or systems, Class III medical devices, nuclear facilities, applications related to the deployment of airbags, or any other applications that could lead to death, personal injury, or severe property or environmental damage (individually and collectively, "Critical Applications"). Customer assumes the sole risk and liability of any use of UTIA AV CR v.v.i. products in Critical Applications, subject only to applicable laws and regulations governing limitations on product liability.

8 References

[1] xfOpenCV git repository: https://github.com/Xilinx/xfopencv.git
[2] Trenz Reference Design for SDSoC tool: https://shop.trenz-electronic.de/en/Download/?path=Trenz_Electronic/TE0808/Reference_Design/2017.1/SKHio0808_SDSoC