Application Note

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

Live Canny Edge Detection Demo for TE0808+TEBF0808 Trenz Board

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

Revision history

Rev.	Date	Author	Description
0	10.10.2018	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 Live Canny Edge detection demo for Trenz Zynq Ultrascale+ TE0808 and TEBF0808 Carrier assembly. The design is based on Trenz SKHio_zusys_SDSoC package downloadable from Trenz web pages. It provides basic HDMI in/out chain for Avnet FMC Imageon extension card. Sources for Canny edge detection filter are available free in xfOpenCV library [1]. The xfOpencv is Xilinx library of OpenCV cores rewritten to be synthesizable by Vivado HLS tool. The library provides one example for each supported function including Canny edge detector. Each example reads input from file(s) and writes output file with result. The result is also compared with original OpenCV function as golden model.

2 Used Tools and Resources

1. TEBF0808 Carrier board from Trenz.
2. TE0808 Module with Xilinx Zynq ULTRASCALE+.
3. FMC Imageon extension card from Avnet.
4. 2 HDMI cables to connect video input to PC and output to LCD screen
5. FHD capable LCD screen with HDMI input.
6. SD or SDHC card with FAT32 filesystem.
7. Xilinx SDx 2017.1 design suite.
8. Petalinux 2017.1 for compilation of linux image (optional, precompiled image from Trenz reference design can be used)
9. Serial terminal application on PC.
10. Mini USB cable, Ethernet cable (optional for debugging), power source

3 How to Test xfOpenCV Examples

Follow these steps to test xfOpenCV examples:

1. Clone xfOpenCV library from [1].

   Checkout branch “2017.1_lkdensepyrof”
2. Download SDSoC reference design package with HDMI in/out support from Trenz download page [2]. Download version for SDSoC version 2017.1 https://shop.trenz-electronic.de/en/Download/?path=Trenz_Electronic/TE0808/Reference_Design/2017.1/SKHio0808_SDSoC
3. Configure the package from step to your particular Zynq module and generate SDSoC platform. Alternatively prebuilt SDSoC platform may be used.

   IMPORTANT: New Petalinux build must be performed to support OpenCV library in both cases (regardless building from sources or using prebuilt). Use Petalinux BSP provided in the package and run: petalinx-config –c rootfs. Check OpenCV library support. Alternatively you can download, cross compile and install OpenCV 3.0.1 library separately from Petalinux build.
4. Open xfOpenCV and visit examples directory. Choose one or more provided examples and edit their “description.json” file. Add your SDSoC platform name to the list of supported platform and remove libraries not needed for HDMI in/out platform.

For example:

Original “description.json” fragment	After modification
{ "example" : "canny - File I/O", "overview" : "canny", "board" : ["zcu102_es2_reVISION"], "os" : ["Linux"],	{ "example" : "canny - File I/O", "overview" : "canny", "board" : ["SKHio0808_zusys_SDSoC"], "os" : ["Linux"],
"libraries" : ["video", "glib-2.0", "drm", "v4l2subdev", "mediactl", "opencv_imgcodecs", "opencv_core", "lzma", "tiff", "png16", "z", "jpeg", "opencv_imgproc", "dl", "rt", "webp", "opencv_features2d", "opencv_flann", "opencv_calib3d"],	"libraries" : ["opencv_imgcodecs", "opencv_core", "lzma", "tiff", "png16", "z", "jpeg", "opencv_imgproc", "dl", "rt", "webp", "opencv_features2d", "opencv_flann", "opencv_calib3d"],

5. Go to directory where your SDSoC platform is located and remove all contents from samples folder.
6. Copy all xfOpenCV with updated “description.json” file to your SDSoC platform samples folder.
7. Create environment variable “SYSROOT”. Its value must be the path to petalinux filesystem copy.
8. Start Xilinx SDx 2017.1 tool and create new design from added samples.
9. Before building please check and update clock source for hardware accelerator if necessary.
10. Build example and copy the result to SD card. Test the example running on hardware.
11. Once File I/O based examples are tested, live video demo can be implemented starting from some of original samples which demonstrate video processing (one of motion_det_xxx, optical_flow_xxx or sobel_xxx)

4 How to Run Canny Demo

Using approach outlined in previous section we have implemented live video demo of Canny edge detector for FHD (1920x1080p60) input video signal. Output is the signal of identical size and rate which shows white edge pixels and black pixels otherwise.

1. Copy content of sd_card folder to SD card.
2. Place SD card to J27 socket on carrier.

3. Connect HDMI input from PC to “HDMI IN” connector on Avnet FMC IMAGEON extension board.
4. Connect HDMI out to LCD screen capable to display 1920x1080p60
5. Connect mini USB cable to XMOD1 module on carrier board. It will allow you to login and run canny application. Alternatively, the graphical terminal using USB keyboard and Display Port monitor can be used.

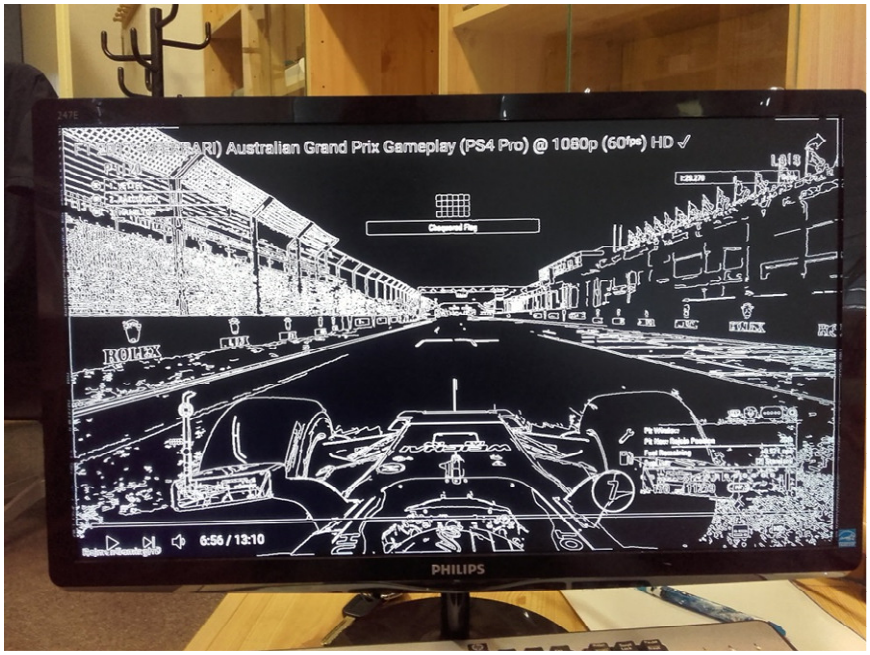

Figure 1: Example of Canny Edge Detector output, 1920x1080, 51 FPS

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