Himax Technologies, Inc.

Software Package (SDK)

Himax Post-Processing

Linux Development Guide

Himax Technologies, Inc.

Revision History

Aug 21, 2019

Version	Date	Description of changes
1.0	2019/07/30	First draft.
1.1	2019/08/21	1. Add a new section of example code description. 2. Change default upscaled size from 864x496 to 640x400 in Figure 1. 3. Fix few typos.

List of Contents

List of Contents

List of Figures

1. System block diagram

Himax Post-Processing provide the quality improvement of depth image including Clipping, Outlier Removal, Hole Filling, Denoise, and Up-Scaling as the below diagram:

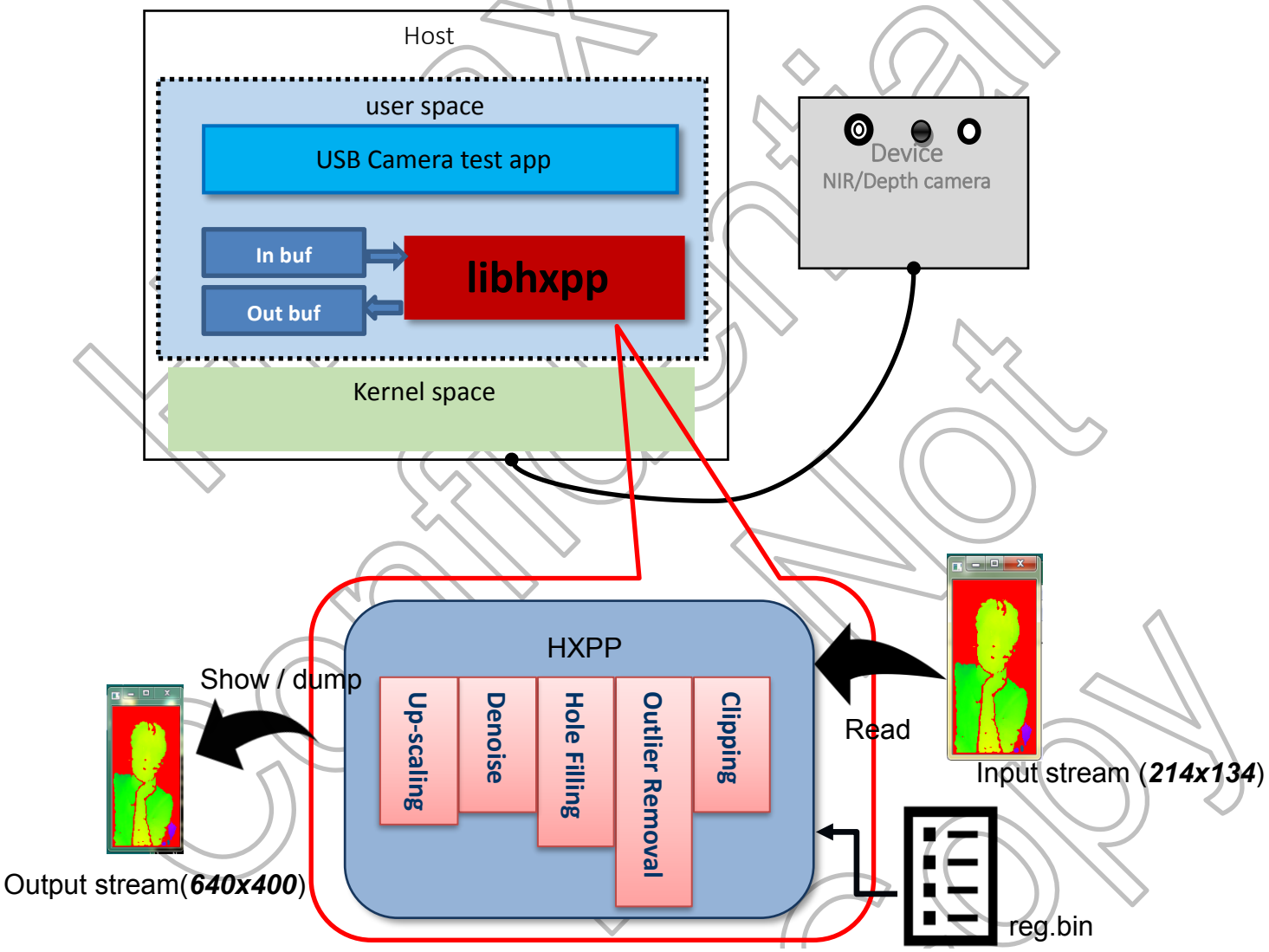

Figure 1 System block diagram

In this case, we will provide a prebuilt shared library “libhxpp.so” and binary configuration file “reg.bin”.

2. HxPP library API

In this section, we will describe the functions and arguments of each API provided by Himax Post-Processing library.

2.1 HxPP library API function and description

1. `int hx_init(const char *cfg_file);`
   Initialize and load the binary configuration file.
   ※ `cfg_file` should be assigned with the path of a binary configuration file “reg.bin”, it could be either relative path or absolute path, there is no restriction on the existence of `cfg_file`, i.e., it won’t be loaded if it isn’t existed.

2. `void hx_cfg_input(int input_width, int input_height);`
   Setup the dimension of input depth image.

3. `void hx_cfg_upscale(int upscale_width, int upscale_height);`
   Setup the dimension of output depth image, this will automatically enable HxPP internal up-scaling filter.

4. `int hx_pp(uint16_t input_frame, uint16_t output_frame);`
   Executing the image post-processing on the input frame buffer and output the processed image on the output frame buffer.

5. `void hx_deinit(void);`
   De-initialize and release configuration of Himax Post-Processing library.

2.2 Default configuration vs. “reg.bin”

If the binary configuration file “reg.bin” is existed and loaded after your program running and initialized, then the built-in default configuration in the prebuilt library “libhxpp.so” will be overridden by the new configuration in the binary configuration file “reg.bin”.

Currently the default configuration only enables Up-scaling filter, and target resolution is “640x400”.

If you want to enable other filters (Outlier Removal, Hole Filling, Denoise), please contact us to obtain a new binary configuration file “reg.bin” with correct configuration settings you wanted.

3. Example code of HxPP APIs call flow

The example code of HxPP API usage and call flow.
Please refer to the external package “hxpp-test” tarball.

Please replace the prebuilt library “`libhxpp.so.0.0.6`” and the header file “`hx_reg.h`” in it before build the hxpp-test example code.

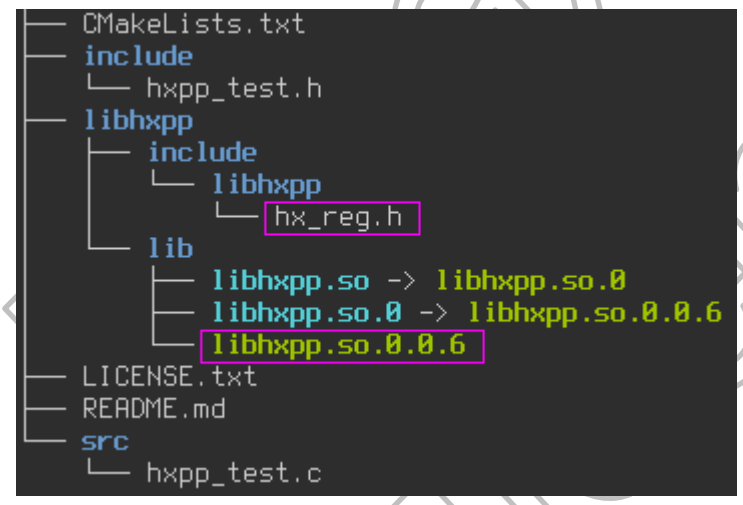

Figure 2 hxpp-test source tree

The CMake build commands are listed in “`README.md`” text file. We had tested that it could be built on Ubuntu 14.04 x86_64 and NXP i.MX 7ULP Yocto build environment.