libcamera software ISP status update

Hans de Goede

Senior Staff Engineer, Qualcomm Technologies NL BV

Bryan O’Donoghue

Kernel Engineer, Linaro

Introductions

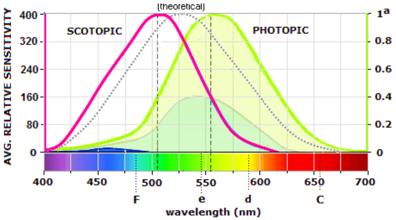
- Hans
  - Senior staff engineer @ Qualcomm Technologies NL BV
  - Kernel subsystem co-maintainer for UVC, drivers/platform/x86
  - https://git.kernel.org/pub/scm/linux/kernel/git/hansg/linux.git/
  - hansg @ #libcamera, “Fedora ARM”
- Bryan
  - Kernel engineer @ Linaro
  - Kernel media-committers maintainer - Qualcomm CAMSS, Iris, Venus
    - https://git.codelinaro.org/bryan.odonoghue/kernel
    - https://gitlab.freedesktop.org/linux-media/users/bodonoghue.git
    - https://github.com/0xB0D
  - bryanodonoghue @ #linux-media #linux #aarch64-laptops #libcamera #linux-msm

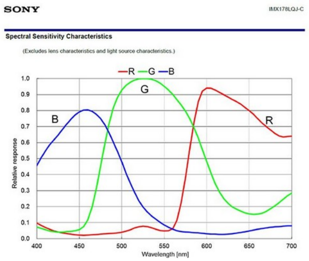
Topics

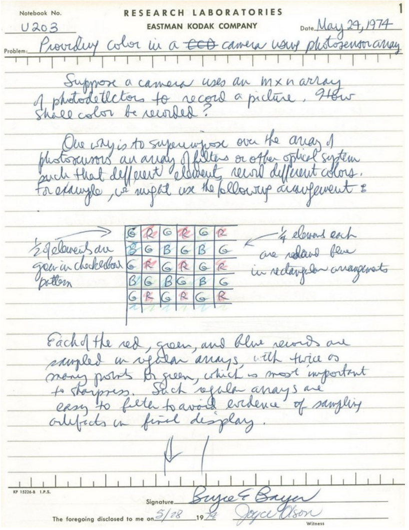
- Bayer Encoding
- CSI2 MIPI Cameras
- What is a Hard ISP - what are the 3As ?
- CPUISP
- GPUISP
- Color Correction Matrix support
- Linux distributions shipping softISP to support MIPI cams OOTB
- Demo

Bayer Encoding

- Human eye is composed of “rods” and “cones”
  - Rods Scotopic
  - Cones Photopic - sensitive to light around 550 nm
  - Most sensitive to “greenish yellow” colours
- Most RGB sensors don’t capture RGB
  - Contain monochrome sensors
- Sensors overlaid with a “mosaic” pattern of Red, Green, Blue
  - Bruce Bayer Eastman Kodak 1974
  - Bayer encoding
- Problem - each pixel contains only one colour
  - Approximate based on proximate pixels
  - Called “interpolation”
- Methods
  - Label
  - Nearest
  - Bilinear
  - Malar-He-Cutler







Notebook No. U203 RESEARCH LABORATORIES 1

EASTMAN KODAK COMPANY Date May 24, 1974

Problem: Providing color in a ~~CCD~~ camera using photosensor array

Suppose a camera uses an m×n array of photodetectors to record a picture. How shall color be recorded?

One way is to superimpose over the array of photosensors an array of filters or other optical system such that different elements record different colors. For example, we might use the following arrangement:

½ of elements are green in checkerboard pattern

¼ element each are red and blue in rectangular arrangements

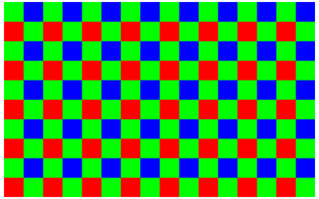
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R

Each of the red, green, and blue records are sampled in regular arrays, with twice as many points for green, which is most important to sharpness. Such regular arrays are easy to filter to avoid evidence of sampling artifacts in final display.

KP 15226-B I.P.S.

Signature Bryce E Bayer

The foregoing disclosed to me on 5/28 19 74 Joyce Olson Witness

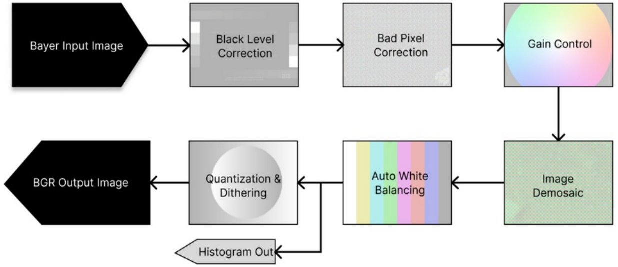


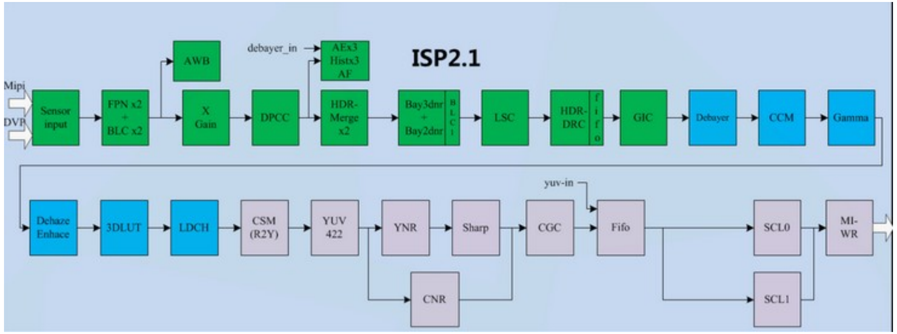
CSI2 MIPI Cameras

- USB - UVC Cameras
  - What you might consider a “standard” camera
  - Enormously expensive and power consuming
  - USB is an entire data communications protocol
- CSI2 MIPI Sensors
  - Proliferation of these sensors due to mobile phone economies of scale
- CSI2 standard
  - A PHY level analogue signal
- Unprocessed output images from most MIPI sensors
  - Various type of Bayer encoded data
  - Libcamera
    - Integration of MIPI Cameras into standard Linux user-space
- Raspberry Pi
- Intel IPU3 / IPU6 / IPU7
- Qualcomm Linux on upstream stacks

What is a HardISP?

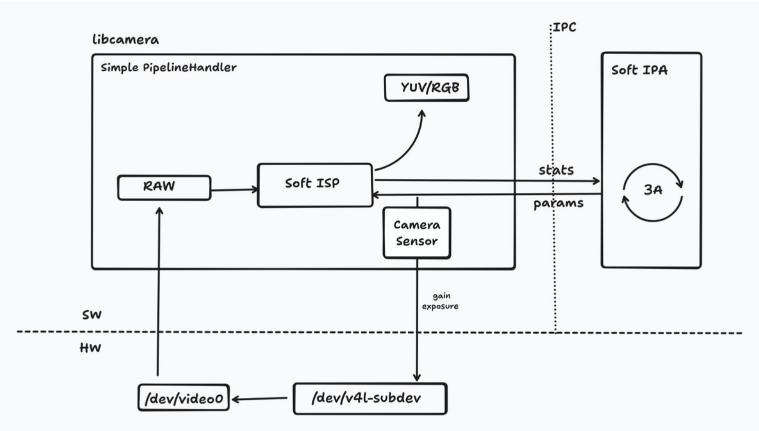
- HardISP
  - Hardware Image Signal Processor
  - A specialised silicon or firmware block - usually both
  - Principle of data locality - process close to sensor input
- Debayers
- Implements 3As
  - Autofocus (AF)
  - Auto White Balance (AWB)
  - Auto Exposure / Gain (AG)
- More advanced “secret sauce” algorithms
  - Sensor tunings (joking but not joking)
  - Skin tones
  - Low light noise reductions
  - Contrast
  - Lens flare reductions





CPUISP

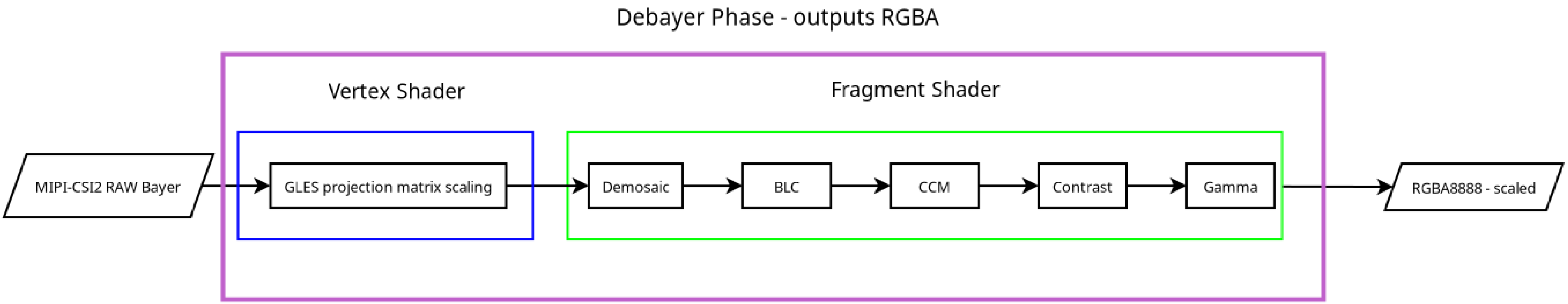
- Libcamera an open source library to handle v4l2 complex cameras
- Different “pipelines” handle different hardware implementations
  - RPI has a full hardware ISP pipeline
  - A generic simple pipeline handler for RAW bayer cameras exists
- A software ISP implemented in the CPU has been upstreamed
  - Debayer histogram and parameters generated and applied in CPU
  - Auto White Balance
  - Auto Gain
  - Gamma correction
- Performance and features
  - Works on all tested Qualcomm platforms
    - sdm845, sm8250, sc8280xp, x1e80100
  - Works on x86 IPU6
  - Too much for lower-end ARM systems
    - TI Sitara AM64x
    - i.MX8M



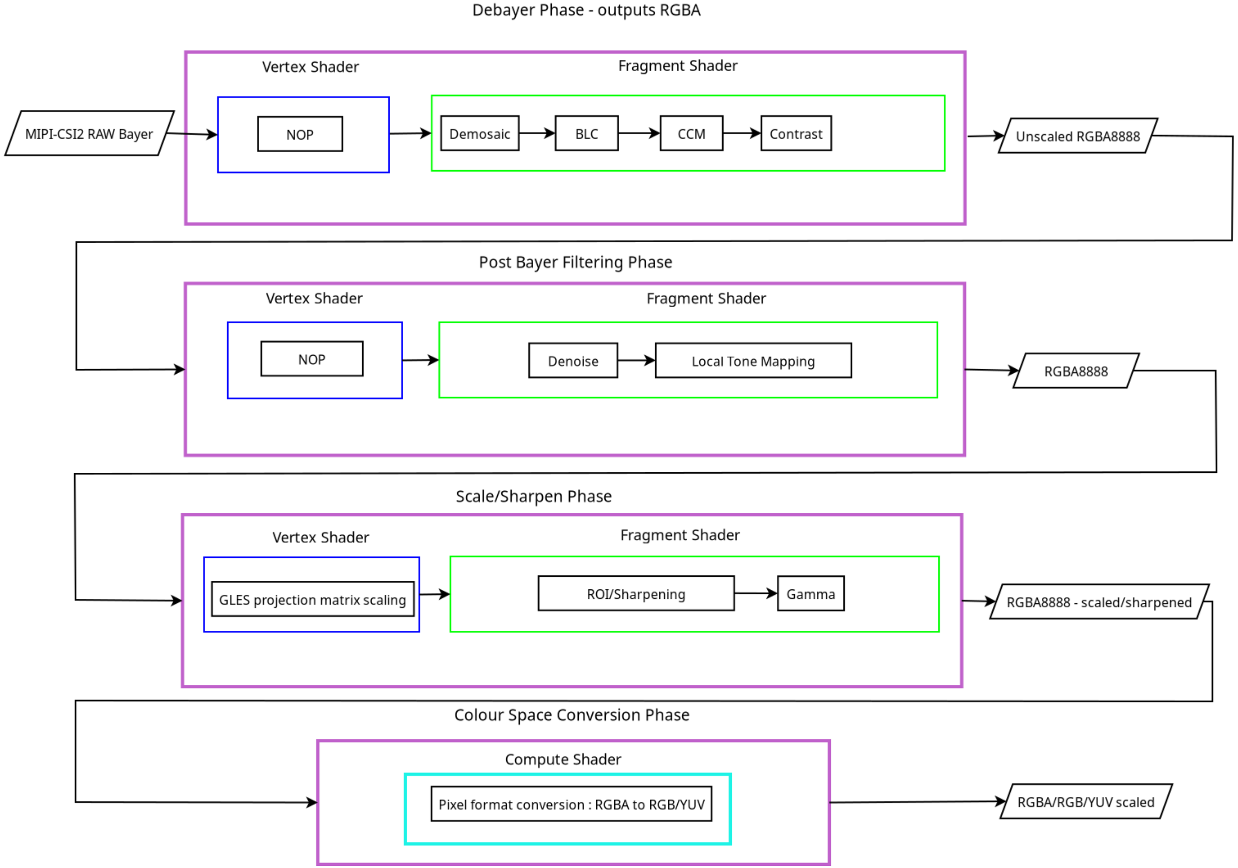
GPUISP

- GPUs are massively parallel pixel pixel processing engines
- Libcamera already contains simple debayer shaders in a QT application
- Shaders debayer 8 bit RAW as well as 10/12 bit MIPI
  - Vertex/Fragment shaders already upstream and somewhat validated
  - Objective to reuse these shaders unmodified
- GLES 2.0 not OpenGL
  - To support the greatest range of hardware possible
  - Places constraints on some of the techniques we can use to attack the problem
  - eGL 1.0 was our first target but for a number of reasons it is undesirable
  - GLES 2.0 released March 2007 => nearly 20 years of supported hardware
- Allows implementing features CPU can’t handle
  - Denoise
  - Dead pixel correction
  - Lens Shading

GPUISP current – single pass vertex/fragment shader



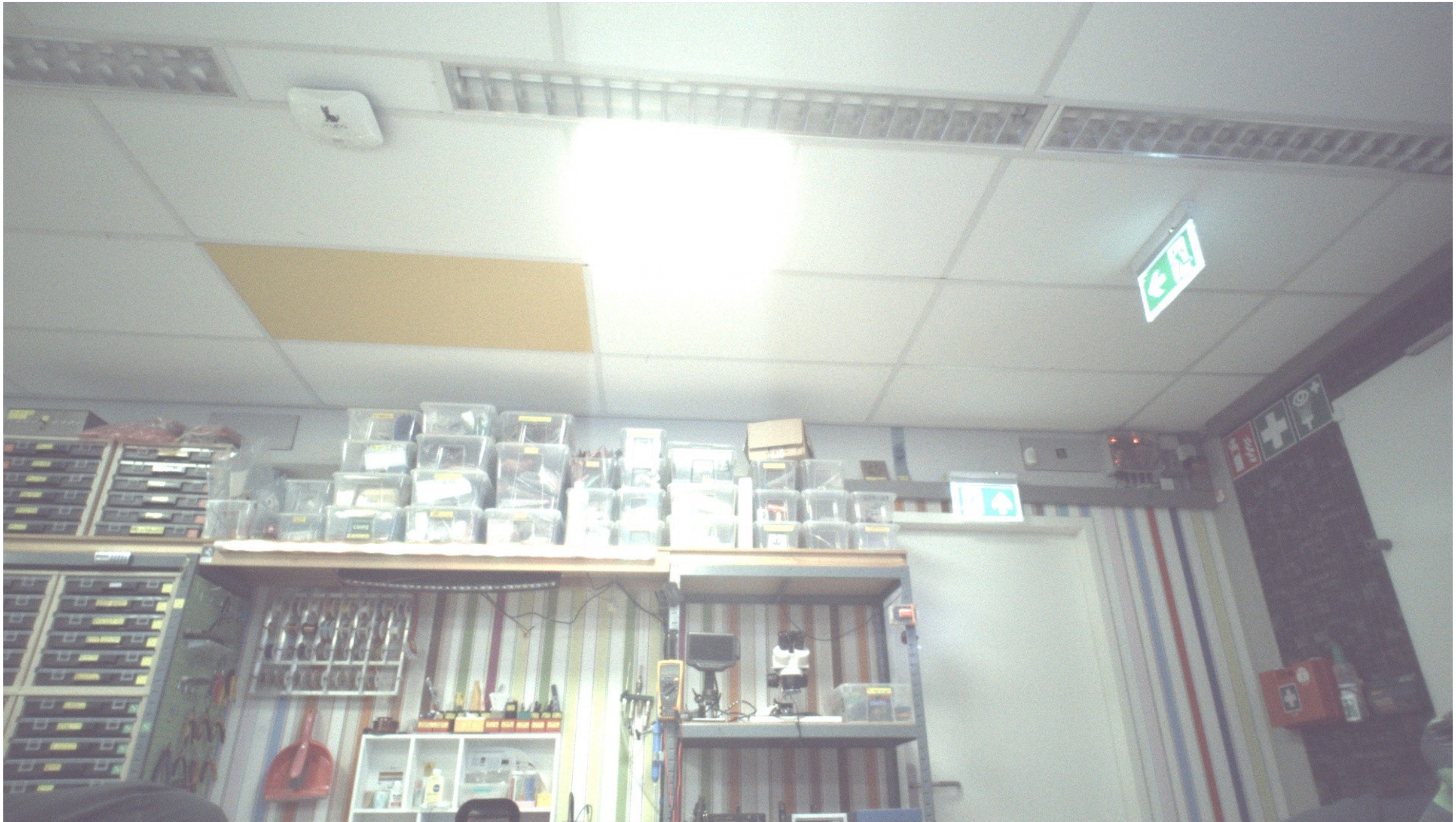
GPUISP next – multi pass



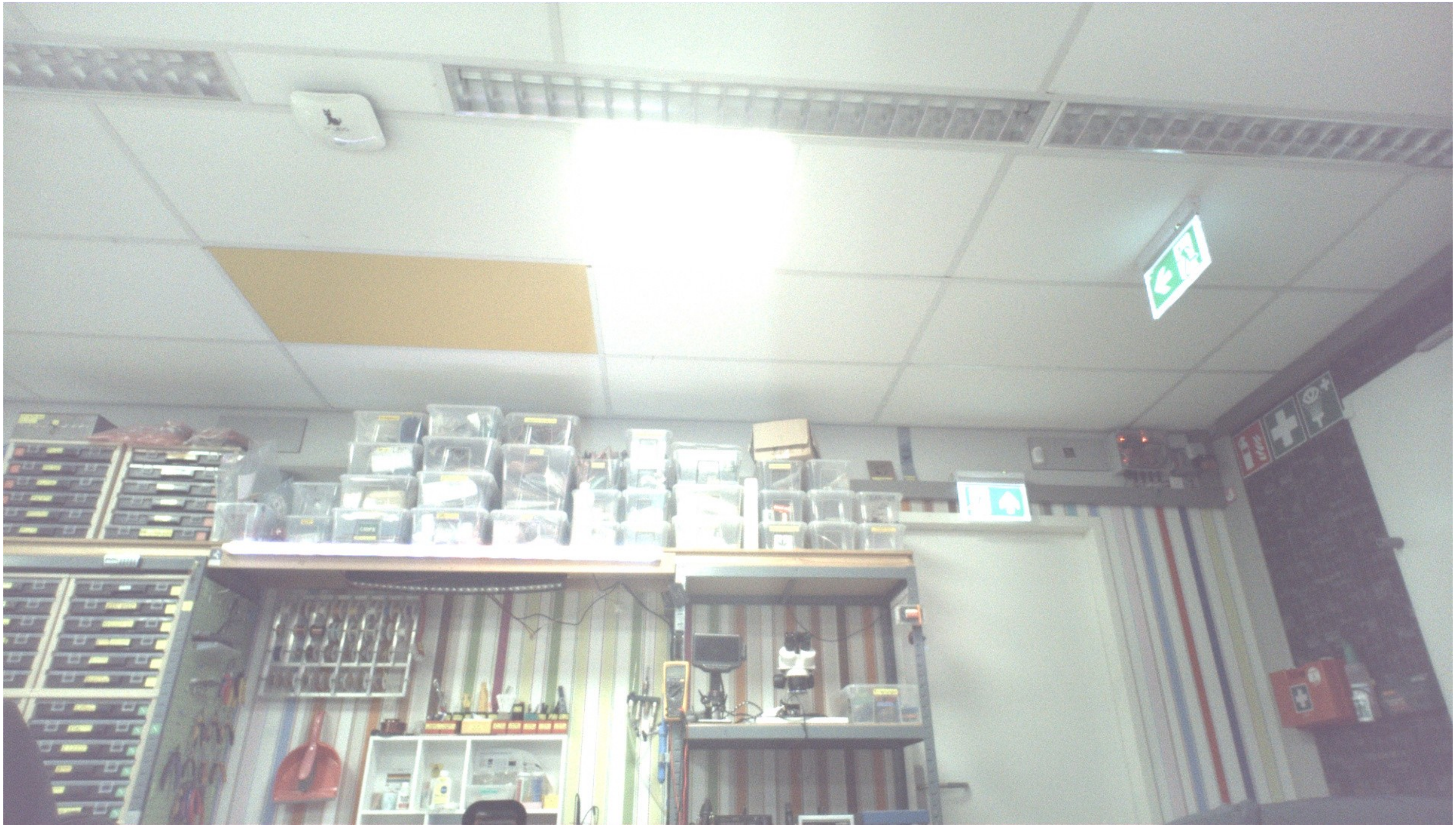
Next steps

- GPUISP features being worked on:
  - Lens shade correction (HHS Delft student group)
  - Multi pass GPUISP (Bryan)
  - Denoise (Milan Zamazal, Red Hat)
- GPUISP possible future features:
  - Multi-pass support
  - Lens distortion correction
  - Dead Pixel correction
  - Sobel Filter/Digital autofocus
  - Lens flare correction
- Calibrate camera modules for better colors (per module CCMs)

Example image without LSC



Example image with LSC



Color Calibration Matrix support

- GPUISP uses an identity CCM by default
- Camera module (sensor + lens) calibration can give us per color-temperature matrices for more accurate controls
- Camera module calibration also give us Lens Shading Correction (LSC) tables
- Using a CCM also allows implementation a saturation control to make colors “pop” more
- Saturation control goes from 0.0 to 2.0 with 1.0 being neutral colors
- May change the softISP default to 1.2 for more vibrant colors to get libcamera closer to proprietary stacks which seem to also do this

Linux distros shipping softISP for OOTB MIPI cam support

- Fedora has been shipping the softISP since Fedora 42 for OOTB support of Intel IPU6 and IPU7 MIPI cameras
- Postmarket OS and other mobile distros are using the softISP for camera support
  - Models on which PMOS + GPUSIP has successfully been tested:
    - Pixel 3a
    - One Plus 6
    - Fairphone 5
    - Librem5
    - PinePhone

Demo

Thank you !

Bedankt !

Go raibh maith agaibh!

Thank you !

Bedankt !

Go raibh maith agaibh!

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

References in this presentation to “Qualcomm” may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes our licensing business, QTL, and the vast majority of our patent portfolio. Qualcomm Technologies, Inc., a subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of our engineering, research and development functions, and substantially all of our products and services businesses, including our QCT semiconductor business.

Snapdragon and Qualcomm branded products are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patenteds are licensed by Qualcomm Incorporated.