End to End: Part 3

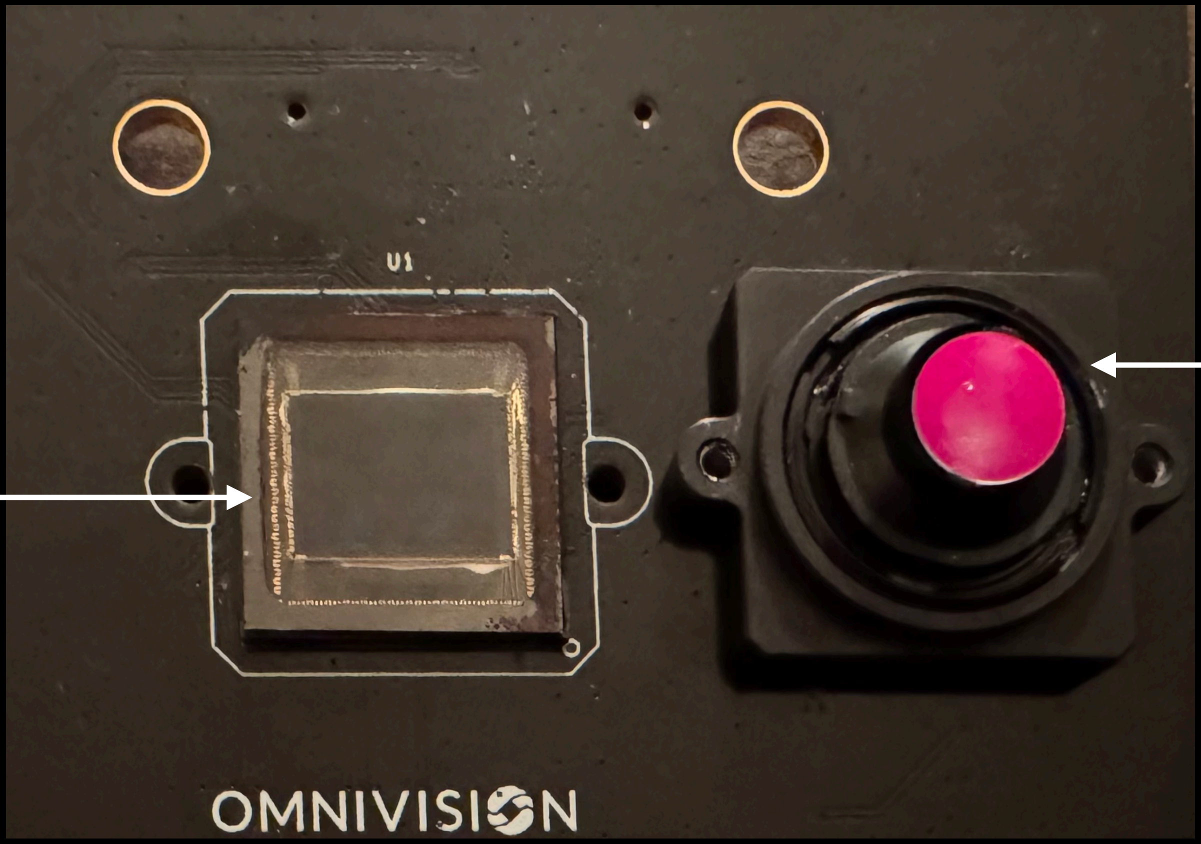
Earl Wong

Subjects

- Optics
- Sensor
- **ISP**
- GPU
- NPU

Pre-ISP / Creating ISP Input

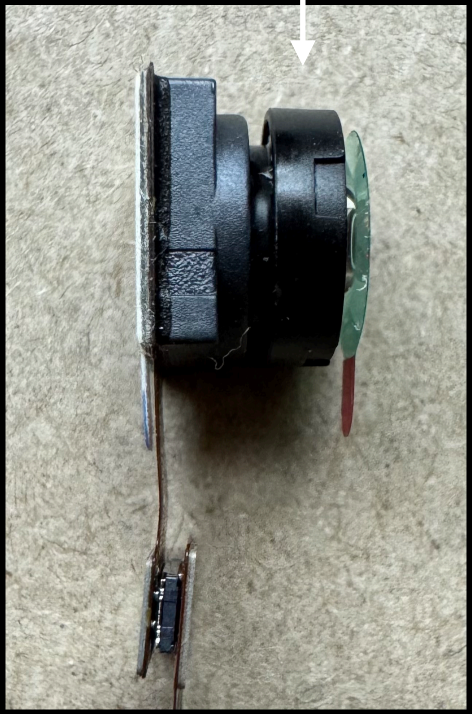
- The camera lens / optics gathers and focuses the incoming light rays.
- The sensor detects these light rays, converting the incoming photons into electrical signals.



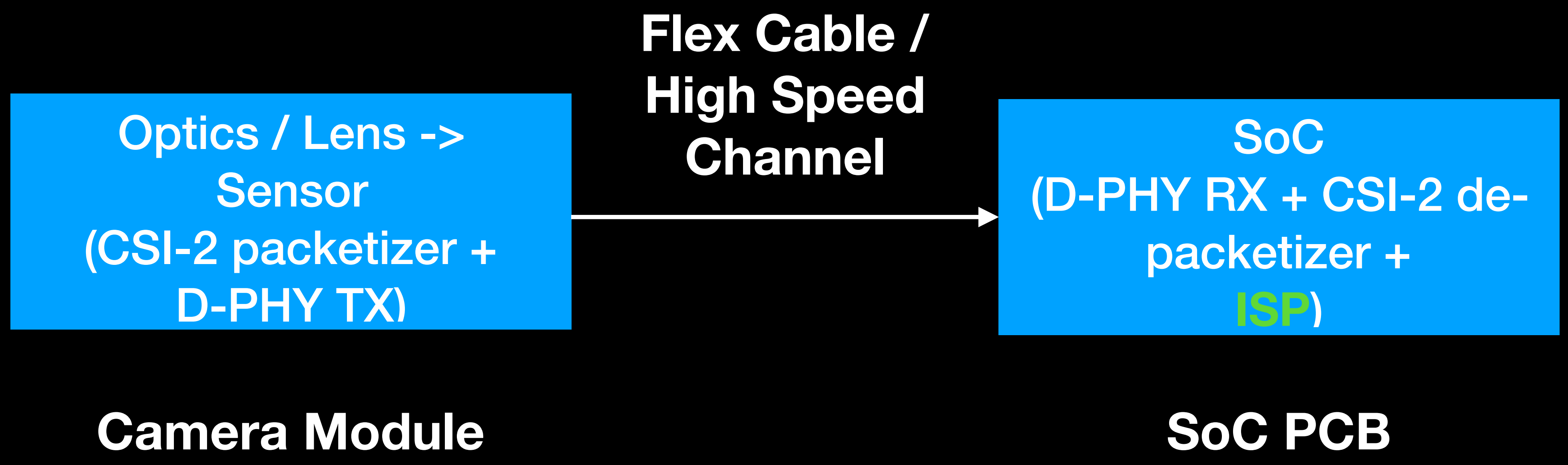
Pre-ISP / Creating ISP Input

- In many implementations, the lens / optics and sensor form a camera module.

- This camera module then interfaces with a printed circuit board via flex cable, transmitting pixel information to a SoC.



Overview



Optics / Lens -> Sensor (CSI-2 packetizer + D-PHY TX)

Flex Cable / High Speed Channel

SoC (D-PHY RX + CSI-2 de-packetizer + ISP)

Camera Module

SoC PCB

“Under the Hood”

MIPI (Mobile Industry Processor Interface)
Physical Layer: D-PHY or C-PHY
Protocol: CSI-2 (camera serial interface)

Physical Layer

- The physical layer refers to the hardware / transport mechanism used to move the data = high speed channel. (Analogy: the road / highway)

- There are two standards: D-PHY and C-PHY

- C-PHY resulted primarily from the demands of the smart phone market.

- Specifically, smart / mobile phones required better electromagnetic interference (EMI) performance and connectors with fewer wires.

D-PHY

D-PHY consists of lanes and clock signals.

- Each lane contains 1 differential pair = 2 wires.
- Each lane carries a serial bitstream.
- The clock signal consists of 2 wires - one for the + differential pair and one for the - differential pair.

Example

- D-PHY with 4 lanes
- 4*2 + 2 = 10 total wires

C-PHY

C-PHY consists of lanes.

- Each lane has three wires, instead of two.
- Each lane transmits symbols, instead of bitstreams.
- Every symbol / state change acts as an “effective” clock signal.
- Hence, clock signals are no longer needed.
- Each lane has higher bandwidth than their differential pair counterpart in D-PHY.

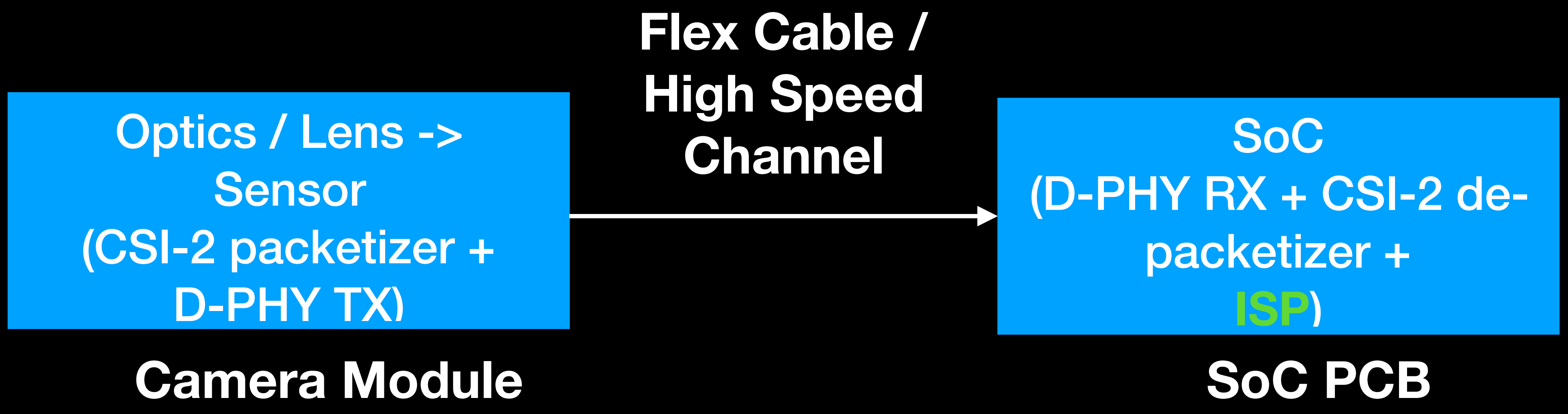
Why not always use C-PHY?

- C-PHY is more complex, and more difficult to debug / root cause.
- => D-PHY is simpler and lower risk.
- If D-PHY is adequate for the task at hand (bandwidth, EMI, mechanical / flex cable), D-PHY should be used.
- Every sensor support D-PHY.
- Some sensors do NOT support C-PHY.

CSI (Camera Serial Interface)

- CSI is a protocol layer.
- The protocol layer describes what the data looks like - data format, data frames, packets, etc.
- The same data in the protocol layer can be transmitted using D-PHY or C-PHY.
- Recall: D-PHY and C-PHY are physical layers.
- Recall: D-PHY and C_PHY describes how the data moves electrically.

Overview



An image sensor digitizes pixels, packs them into CSI-2 packets, and transmits the information across Mobile Industry Processor Interface (MIPI) lanes using D-PHY or C-PHY,

i.e. The sensor outputs CSI-2 over D-PHY TX or C-PHY TX.

The reverse occurs in the SoC PCB, resulting in an array of pixel data, ready to be consumed by the camera ISP.

ISP

Low Level Corrections

- Black Level
- Dead Pixel
- Lens Shading

ISP

Black Level

- What is the measured output for every pixel, when the sensor is covered = no light?
- Measure the output at every pixel location / average the output values for each pixel location and store the result.
- Subtract the stored value at every pixel location from the measured intensity.

Why black level correction?

- Without a proper “zero illumination” reference, every subsequent calculation in the pipeline will have an unwanted brightness error or bias.

ISP

Dead pixel

- Dead pixels are pixels that always read too low, too high or flicker.
- To find dead pixels, ask: “How much does the pixel deviate from its neighbors?”

Why dead pixel correction?

- Consider what would happen, if you performed de-mosaicing using information from an unknown dead pixel.

ISP

Lens shading

- In a perfect lens, there is no falloff in light intensity at the periphery of the sensor.
- In practice, for a constant light source / constant intensity surface (or a lens covered with a diffuser), the brightness measured at the center of the sensor is larger than at the periphery.
- To correct: “Measure the difference in brightness, determine the gain needed for correction and store / apply the gain.”

Why lens shading correction?

- Without it, all subsequent downstream calculations will be “under gained” at the periphery.
- Without it, images will be darker at the periphery.

ISP

High Level Algorithms

- Auto Exposure (AE)
- White Balancing (AWB)
- Color correction
- Auto Focus (AF)
- De-mosaicing
- Noise reduction (NR)
- Sharpening
- HDR & Tone Mapping

ISP

3A is the term used describe the following 3 camera functions:

- Auto Exposure (AE)
- Auto White Balance (AWB)
- Auto Focus (AF)

AE and AWB involve adjusting camera gains.

AF involves adjusting the lens focus position.

Auto Exposure (AE)

Goal of Auto Exposure

- Create a sharp, well exposed image with low noise and no clipping.

Auto Exposure (AE)

How?

- Methods include scene averaging, center weighted, spot and matrix metering.
- The NxN image is divided into mxm patches, where $\frac{N}{m} \in Int$
- The average intensity in each patch is computed.
- Next, the average intensity from the different patches are assigned different weights and combined, yielding AE_Measured.

Auto Exposure (AE)

- Scene averaging - each patch is assigned the same weight.
- Center weighted - the weight assigned to each patch follows an overlaid gaussian distribution.
- Spot - the weight for a small number of neighboring patches is assigned a value of 1. All other patches are assigned a weight of 0.
- Matrix - Patches which are clipped (0 or 255) are assigned weights of 0, regardless of patch location. Weights for non-clipped patches are determined using a heuristic.

Auto Exposure (AE)

How?

- An AE_Target value is set.
- An AE_Measured value is computed, based on the selected metering mode.
- For example, for pixel intensities ranging from [0, 255], AE_Target might be 128 and AE_Measured might be 50.
- The sensor gain and exposure are adjusted, to drive the AE_Measured value to the AE_Target.
- Short exposures increase noise, but produce sharper images.
- Longer exposures decrease noise, but may result in blurry images.

Auto Exposure (AE)

- Implicit in the described process, is a luma histogram for the image.

- The histogram contains the “big picture” overview of the image.

- Sans patch weighting, the goal is to “fit” the histogram within the [0, 255] output intensity range.

- If the dynamic range of the sensor exceeds that of the scene, many AE_Target values will work, with different tradeoffs.

Auto Exposure (AE)

Worse case scenario:

- The scene exceeds the dynamic range of the sensor.
- As a result, some portion of the scene will always be clipped.
- Exposure is then set for “what is important”.
- i.e. Expose for the shadows or the highlights?

White Balancing (AWB)

- For a deeper dive into white balancing, please see the presentation: VISION_ColorConstancy_WhiteBalance (49 slides)

Goal of AWB

- Replicate the WB capabilities of the human visual system.
- i.e. Regardless of scene illumination (Tungsten, Fluorescent, Daylight, etc.) neutral (gray) objects appear neural.
- => White objects are white.

White Balancing (AWB)

How?

- Apply the correct gains to the R, G and B channels, under different illumination conditions.

- i.e. if R, G and B represent the captured values of a gray card under lighting “xyz”, apply $g_R, g_G, g_B\ so\ that\ R^{'} = G^{'} = B^{'}$

- $$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} g_R & 0 & 0 \\ 0 & g_G & 0 \\ 0 & 0 & g_B \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

White Balancing (AWB)

Worse case scenario:

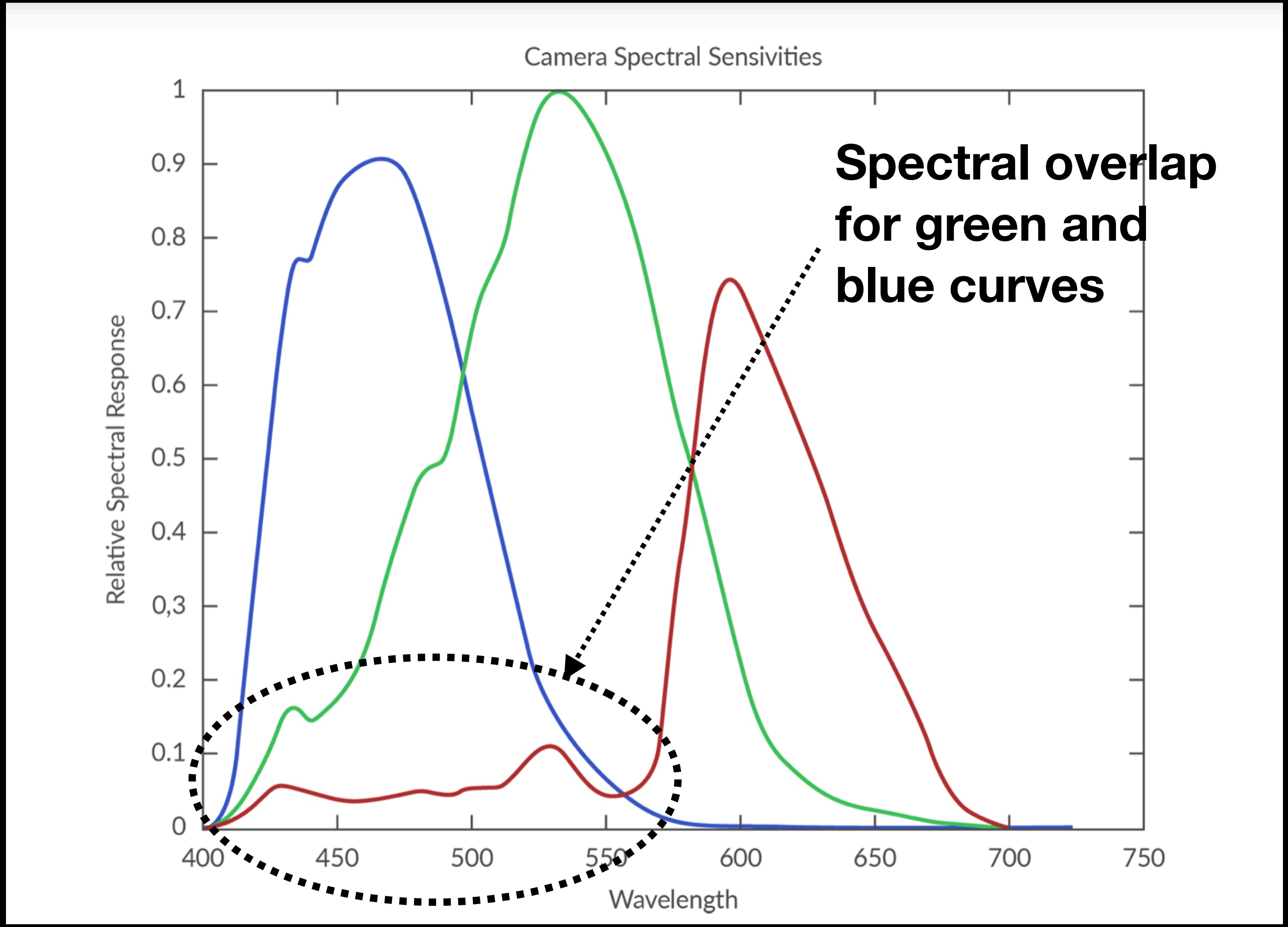
- The scene consists of a single color - a green wall.
- Without additional information, it is impossible to determine (from the captured pixel values) whether 1) the wall is white and contains a green cast or 2) the wall is actually green.

Color Correction

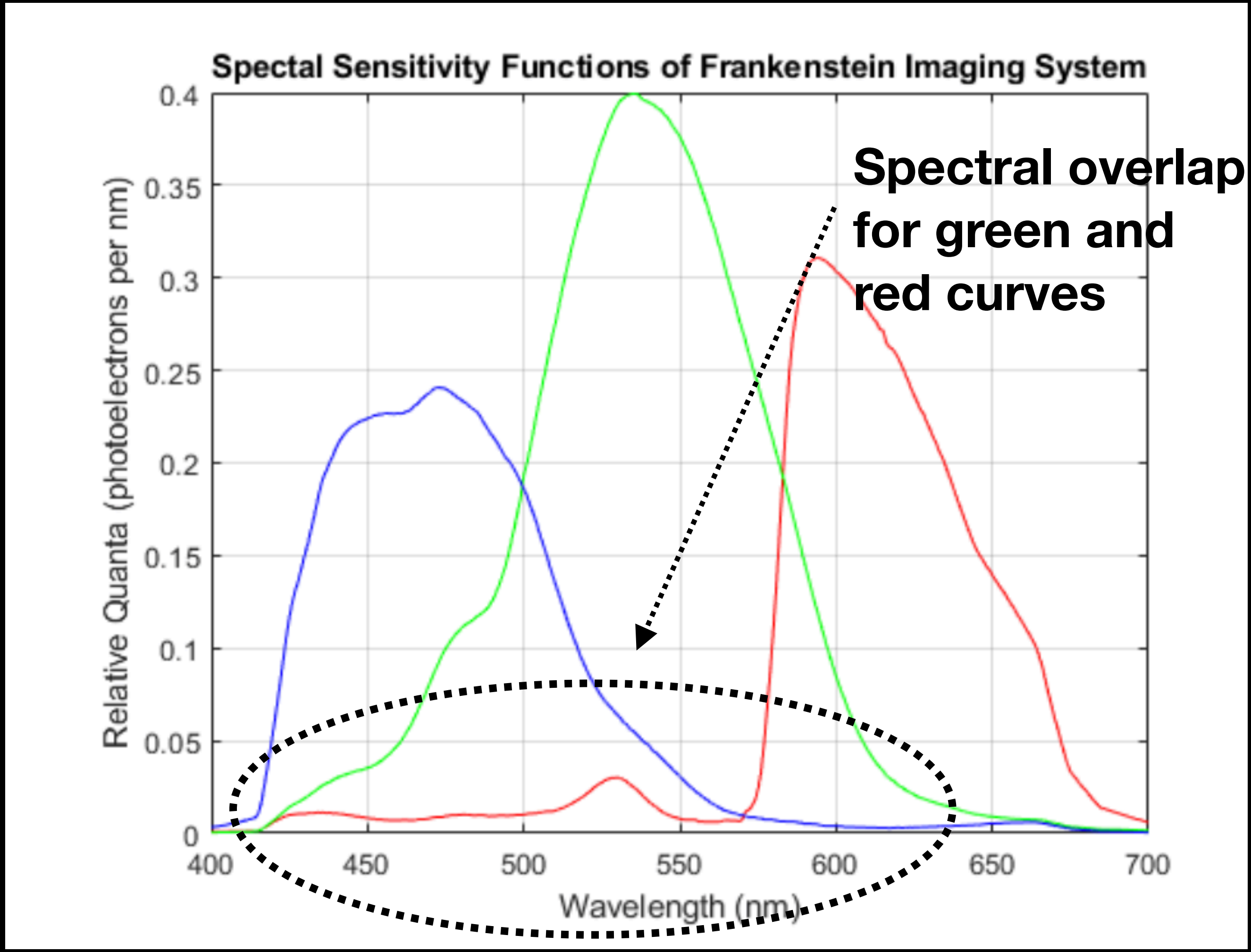
Goal of Color Correction

- White balancing makes neutral grays neutral, under a host of different illuminants / color temperatures.
- However, white balancing cannot / does not fix cross channel contamination.
- Cross channel contamination results from spectral overlap in the spectral sensitivity curves. (See next slide.)
- The degree of overlap, the physical shape of curves and their locations are sensor dependent.
- Because R, G and B AWB gains (g_r, g_g, g_b) cannot correct spectral overlap, different colors can still be incorrect.

Color Correction



Spectral overlap for green and blue curves



Spectral overlap for green and red curves

Spectral sensitivity curves for two different camera sensors.

Color correction matrices are both illumination and sensor dependent.

Color Correction

How do we fix cross channel contamination / spectral overlap?

- We apply an illuminant dependent 3x3 linear transform to the previously white balanced pixels.

$$\begin{bmatrix} R_c \\ G_c \\ B_c \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

- In practice, a 24 patch Macbeth color checker is captured for two reference illuminants - A (tungsten) and D65.
- The needed 3x3 linear transforms are then determined using a least squares fit.
- The required matrix (for every other illuminant / color temperature) is interpolated from these two 3x3 linear transformation matrices by estimating α : $\mathbf{M}(\alpha) = \alpha\,\mathbf{M}_{D65} + (1-\alpha)\,\mathbf{M}_A$

Auto Focus (AF)

- For a deeper dive into auto focus, please see the presentation: VISION_AutoFocus (52 slides)

Goal of Auto Focus

- Rapidly move / drive the lens to focus at certain distance in the scene.

Auto Focus (AF)

How?

- In contrast based AF, hill climbing is employed.
- i.e. The lens position is swept through its entire focus range.
- The position yielding the greatest sharpness for a given scene location, is deemed the “in focus” position.
- In more advanced AF, depth information and scene content is used, to drive the lens to the desired focus position.

Auto Focus (AF)

Worse case scenarios:

- Fast lens (small F number lens) has a shallow depth of field, making accurate focusing more challenging.
- Since depth of field decreases for close objects, macro photography is a challenging use case.

Best case scenarios:

- Pinhole camera like apertures have wide depth of field, thereby decreasing the need for a precise focus position.
- Since depth of field increases for distant objects, landscape photography becomes an easy use case.

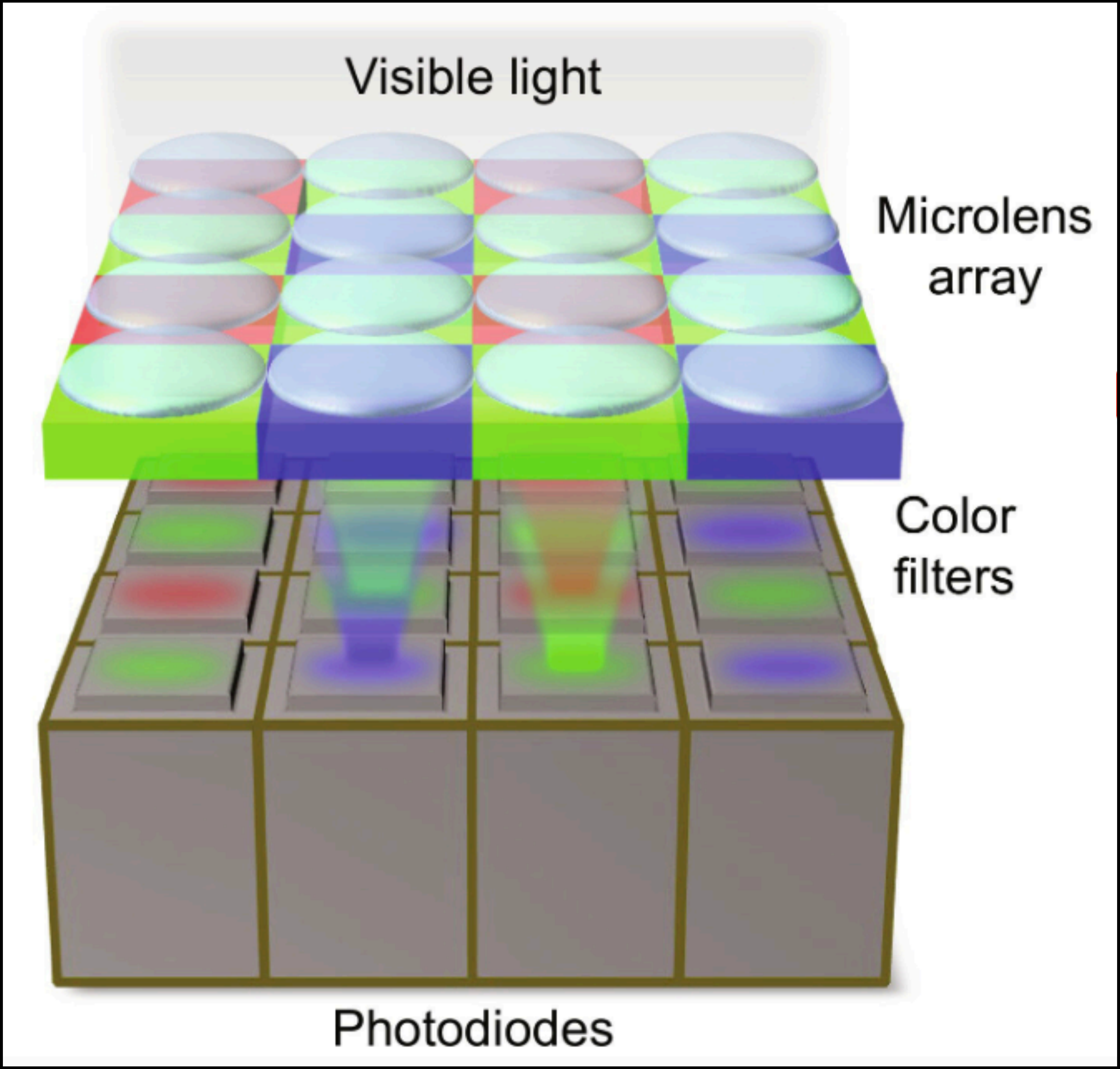
De-mosaicing

- By construction, monochrome sensors and sensor technology introduced by Foveon, do not need de-mosaicing.

- However, the majority of sensors image either an R, G, or B channel at every photo diode site / pixel site using a Bayer pattern, or something similar.

Goal of De-mosaicing

- Create R, G and B output planes that closely mimic the equivalent capture three separate sensors with R, G and B color filters.


Visible light
Microlens array
Color filters
Photodiodes

De-mosaicing

How?

- Bilinear interpolation can be used to fill in the “missing” R, G and B values, respectively.

- Bilinear interpolation is simple and fast.

- However, it does not respect image structure.

De-mosaicing

How?

- Edge aware / gradient based de-mosaicing can be applied, thereby respecting image edges.

- The green channel benefits the most, since its pixel density and intensity are noticeably higher than the red and blue channels.

- Other: Joint de-mosaicing and denoising

Noise Reduction

- Noise reduction algorithms fall into two categories: spatial and temporal
- Spatial noise reduction can be applied to both the luma and chroma channels (YUV, YCbCr, etc.)

Goal of Noise Reduction

- Remove sensor noise, while simultaneously preserving image structure.

Noise Reduction

How?

- For an overview of classical noise reduction algorithms, please see the presentation: VISION_NoiseReduction (38 Slides)

- Many temporal noise reduction algorithms are extensions of their 2D spatial counterparts.

- Although temporal noise reduction algorithms are more powerful than their spatial counterparts, care must be taken to address potential artifacts resulting from scene motion between neighboring temporal frames.

Noise Reduction

Worse case scenarios:

- Spatial and temporal denoising: images with low contrast texture and / or fine texture, like human hair

- Temporal denoising: scenes rich in local motion

Sharpening

Goal of Sharpening

- Perceptually enhance the edge information / the high frequency structure in an image.

Sharpening

How?

- A tried and true approach that works remarkably well, is unsharp masking.

- Unsharp masking takes a given image, and blurs the image with a low pass filter.

- The un-blurred and blurred images are then subtracted, yielding an “unsharp mask”.

- The unsharp mask is then added back to the original image, producing a new image with increased perceptual sharpness.

Sharpening

Worse case scenarios:

1) The image undergoing sharpening still has a noticeable amount of noise.

- After sharpening, the residual noise is amplified by the unsharp mask.

2) The original image lacks details = low resolution.

- Excessive unsharp masking is applied, producing noticeable ringing and / or halos around the edges in the image.
- Example: Movies captured on VHS / analog medium (low resolution), that are subsequently upscaled and digitized.

HDR

- In the presentation ARCHITECTURE_Sensor, we mentioned that dynamic range could be increased by using larger sensors, resulting in larger pixels and larger well capacity.

- With an increased dynamic range, exposure clipping from AE can be eliminated or minimized.

Goal of HDR

- Increase the dynamic range of the existing sensor, using computational photography.

HDR

How?

- Capture multiple images / a burst of several images at different exposures.
- Fuse the captured images to create a new image with a larger dynamic range.
- Tone map the image for a targeted display.

HDR

Worse case scenario:

- The captured images are not linear with respect to exposure.
- Ghosts are still visible after fusion due to inadequate de-ghosting.
- The tone curve is incorrectly designed for the fused image, producing a “plastic like” output image.

Tone Mapping

Goal of Tone Mapping

- Compress the image brightness in a pleasing fashion, without creating hue shifts and / or creating an un-natural looking image.

Tone Mapping

How?

- Transform the R, G and B image to brightness Y.
- Y = .2126R + .7152G + .0722B
- Apply a tone mapping curve T(Y).
- Compute a gain factor k = T(Y) / Y.
- Apply k to the color channels: $(R^{'}, G^{'}, B^{'}) = k * (R, G, B)$

Tone Mapping

Worse case scenario:

- The tone curve is “not correctly designed”, resulting in noise amplification, excessive and un-natural flatness on known structures (like human faces) etc., and lost / “compressed” detail.

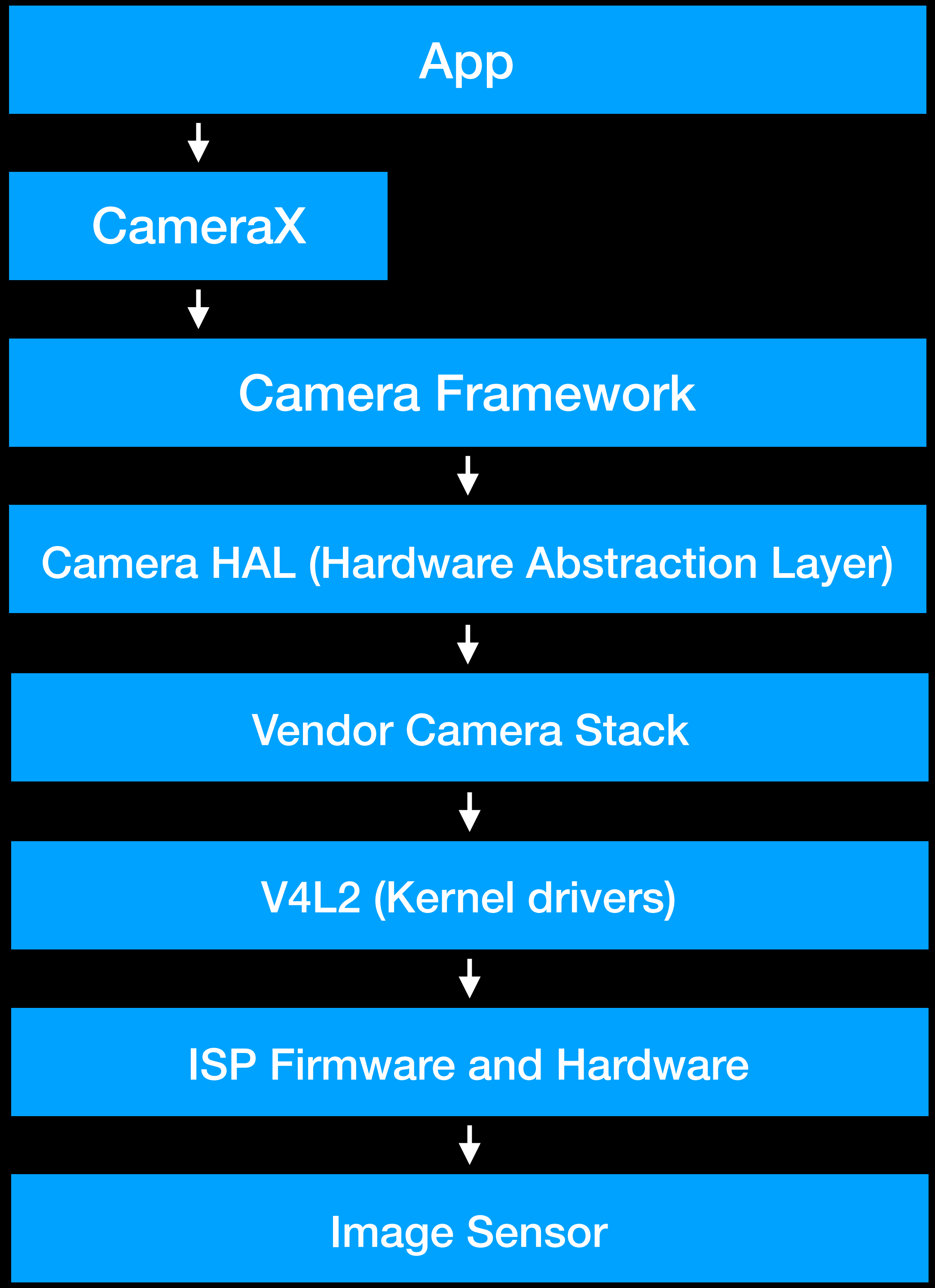
The Role of ML

- In 2025, scene detection, scene segmentation, object detection, etc. (using deep learning models based on CNN’s) produce state of the art results.

- The additional information is (then) fed into / and consumed by different blocks in the camera ISP, to improve the efficacy of the traditional algorithms.

- Direct replacement of the traditional algorithms will eventually occur in the coming years.

Bonus Material
(Android Camera Stack)



App

CameraX

Camera Framework

Camera HAL (Hardware Abstraction Layer)

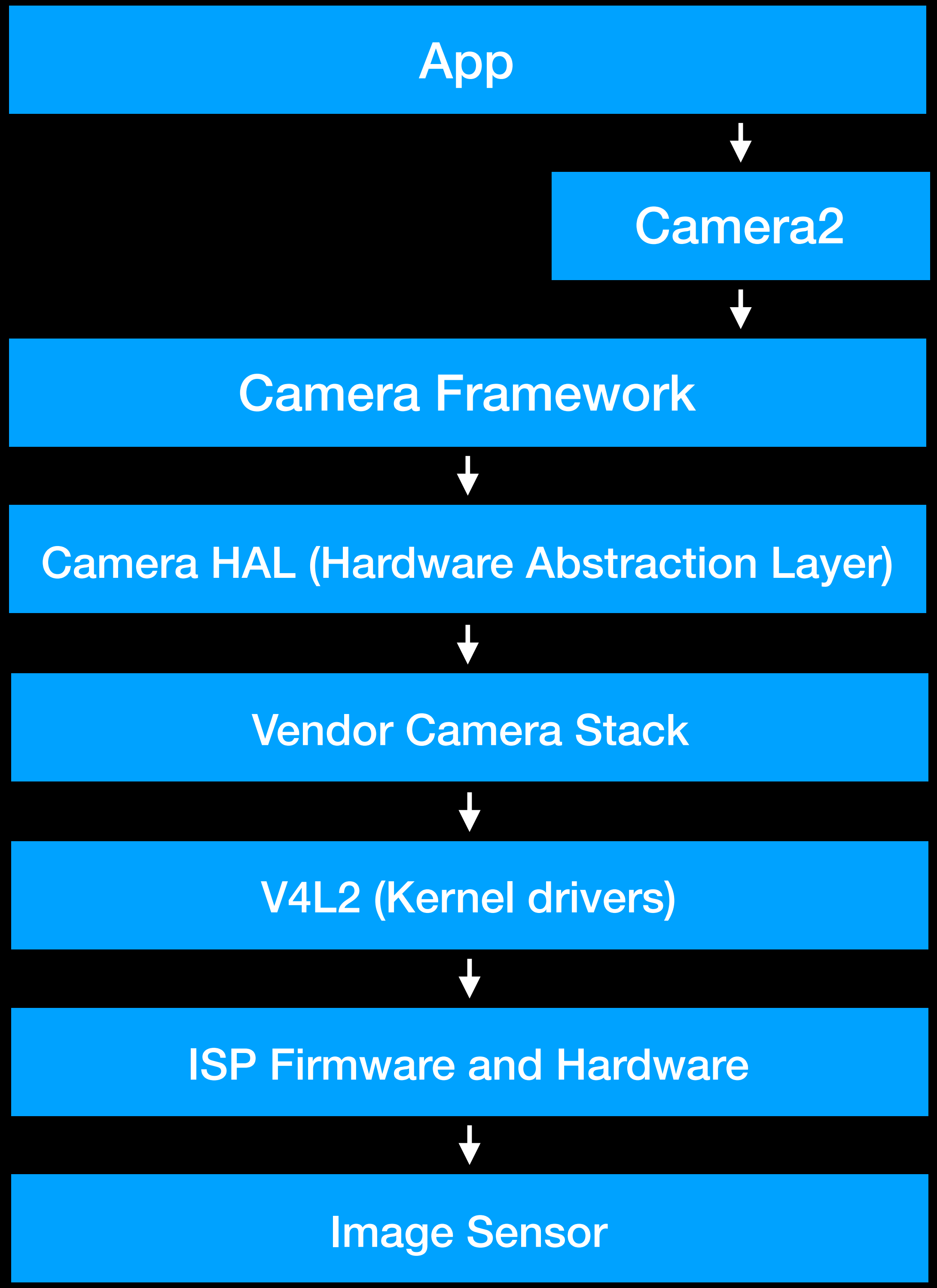
Vendor Camera Stack

V4L2 (Kernel drivers)

ISP Firmware and Hardware

Image Sensor

Bonus Material (Android Camera Stack)



App

Camera2

Camera Framework

Camera HAL (Hardware Abstraction Layer)

Vendor Camera Stack

V4L2 (Kernel drivers)

ISP Firmware and Hardware

Image Sensor

Bonus Material
(Android Camera Stack)

- CameraX and Camera2 are high level API's that interface with Camera Framework.

- CameraX supports all camera vendors, but only provides limited camera functionality. (i.e. Fully compatible, but only supports a small number of features.)

- Camera2 provides the developer with full control of any / all available camera functionality. (i.e. You need to "make it work" for your application.)

- Hardware Abstraction Layer (HAL) provides a generic interface into the lower levels of the hardware stack, used by all camera vendors.

- V4L2 contains the kernel drivers needed for different firmware functionality.

Bonus Material
(Android Camera Stack)

- Vendor camera stack provides the secret sauce for vendor specific algorithms.

- Example: HAL says “take picture” … vendor camera stack …. V4L2 then executes the low level functionality.

- Vendor camera stack contains proprietary algorithms for different functionality like Night Mode, 3A, etc.