Lecture 12: Digital Camera Image Signal Processing: The Basics

Yuhao Zhu

http://yuhaozhu.com
yzhu@rochester.edu

CSC 292/572, Fall 2022
Mobile Visual Computing

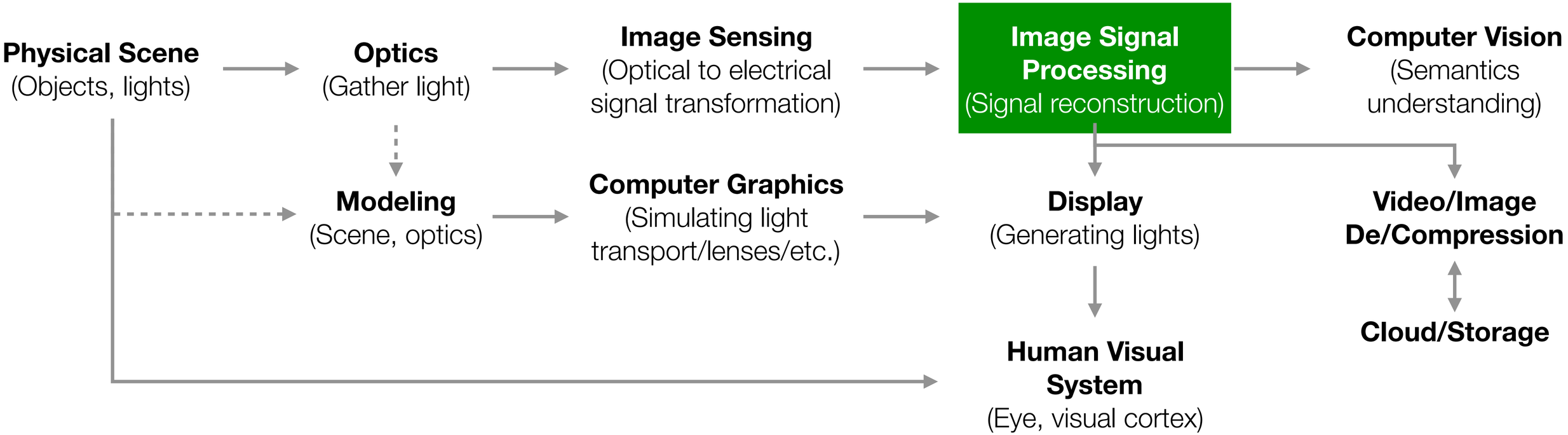
Logistics

WA 2 grades are posted.

Project idea document is posted. Feel free to work on your own idea too.

- The link is on the assignment page
- Can work in groups of 2
- Submit a one-page proposal describing what you want to work on by Oct. 26, 11:30 AM.

Where Are We


Physical Scene
(Objects, lights)
Optics
(Gather light)
Image Sensing
(Optical to electrical signal transformation)
Image Signal Processing
(Signal reconstruction)
Computer Vision
(Semantics understanding)
Modeling
(Scene, optics)
Computer Graphics
(Simulating light transport/lenses/etc.)
Display
(Generating lights)
Video/Image De/Compression
Cloud/Storage
Human Visual System
(Eye, visual cortex)

The Roadmap

Theoretical Preliminaries

Human Visual Systems

Color in Nature, Arts, Tech
(a.k.a., the birth, life, and death of light)

Digital Camera Imaging

Modeling and Rendering

Applications

Optics in Camera

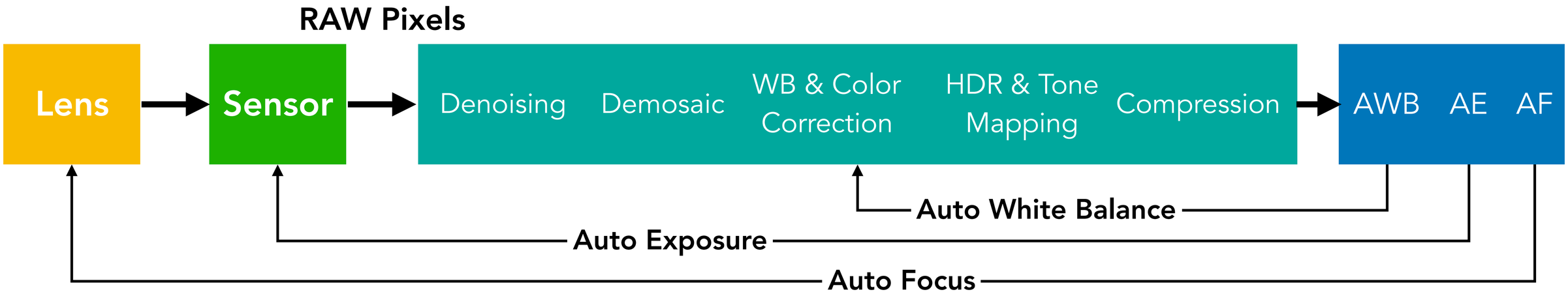
Image Sensor

Image Signal Processing

Image/Video Compression

Immersive Content

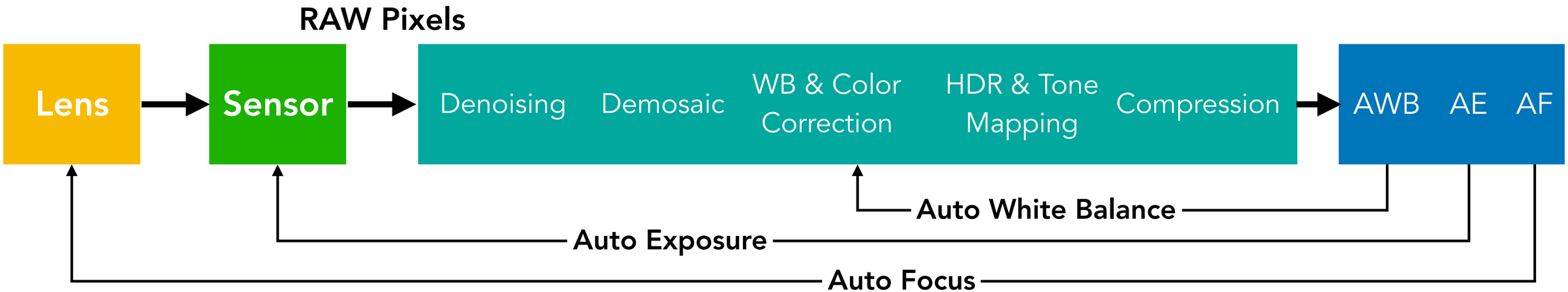
Image Signal Processing in Digital Cameras



While general stages are the same, the order may vary. The actual implementation in commercial cameras are mostly proprietary.

Forward path gathers various statistics, which enable the **"3A" algorithms**, which control the lens, sensor, and the forward path in a **feedback** fashion.

Image Signal Processing in Digital Cameras

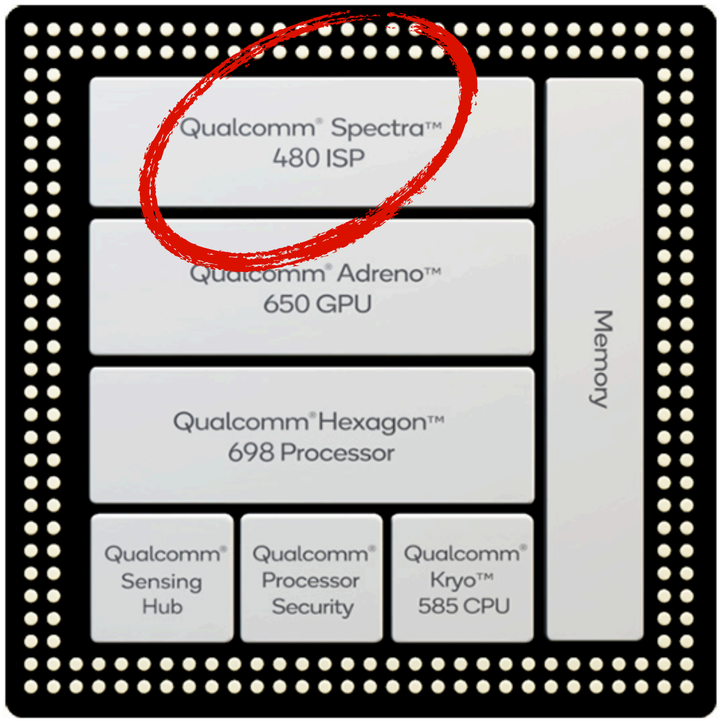


Speed and low energy consumption are critical, so the entire pipeline usually is executed in a special processor, Image Signal Processor (ISP).

ISP used to be fixed-function pipeline, but is becoming more programmable to flexibly support new computational photography algorithms.

Image Signal Processing Hardware

Qualcomm Snapdragon 865 Systems-on-a-chip (SoC)

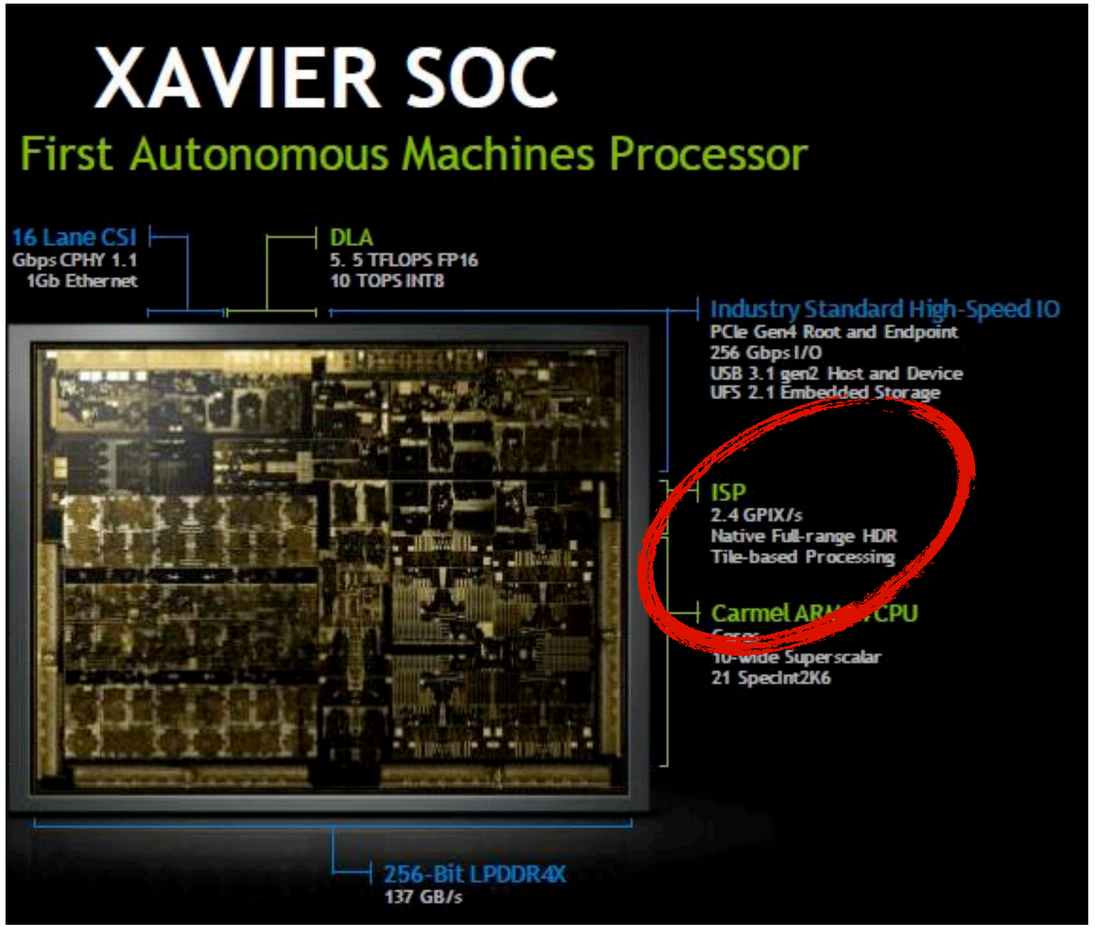


Samsung Galaxy S20

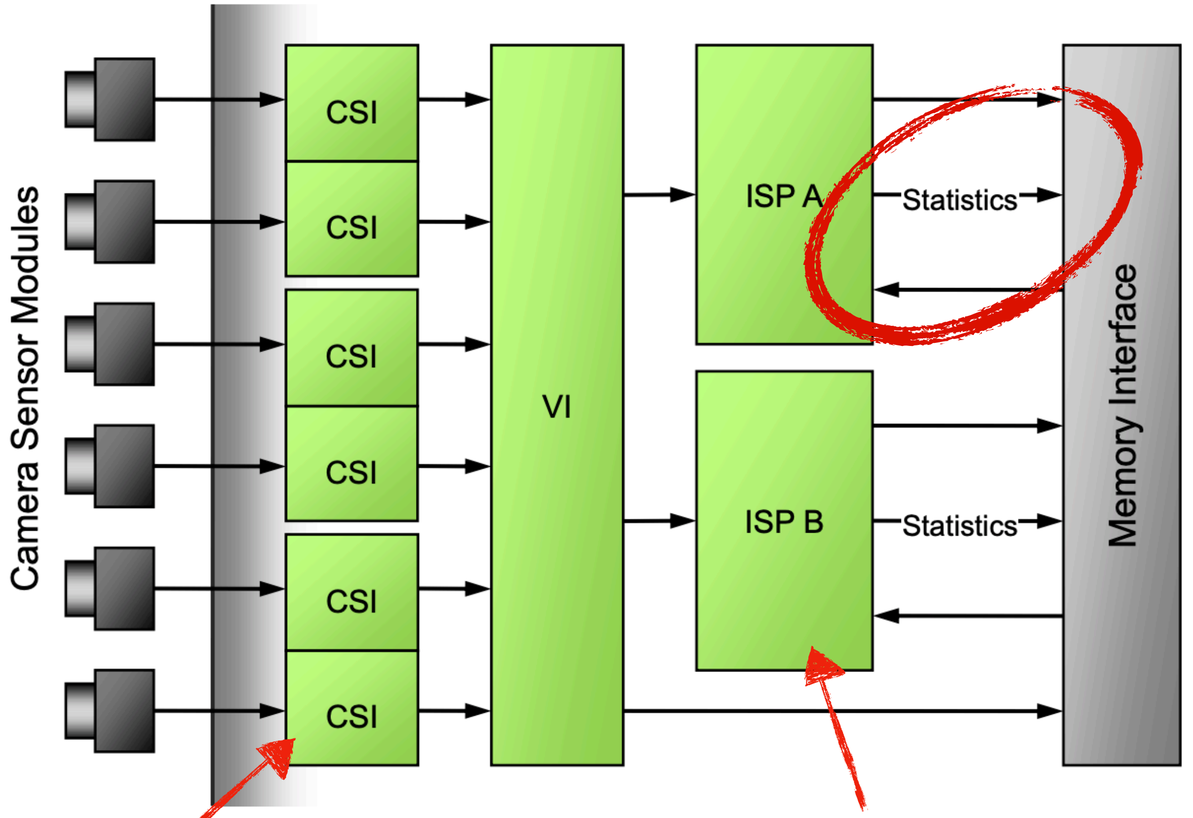
https://arstechnica.com/gadgets/2019/12/qualcomms-new-snapdragon-865-is-a-step-backwards-for-smartphone-design/

https://www.androidauthority.com/qualcomm-snapdragon-865-specs-1058483/ https://www.bhphotovideo.com/c/product/1549750-REG/samsung_sa_g980fdpnk_galaxy_s20_g980f_dual_sim.html

Image Signal Processing Hardware



Camera subsystem in Nvidia TX1 SoC



MIPI Camera Serial Interface, transferring data from the sensor to the SoC

2 ISPs processing 6 cameras simultaneously

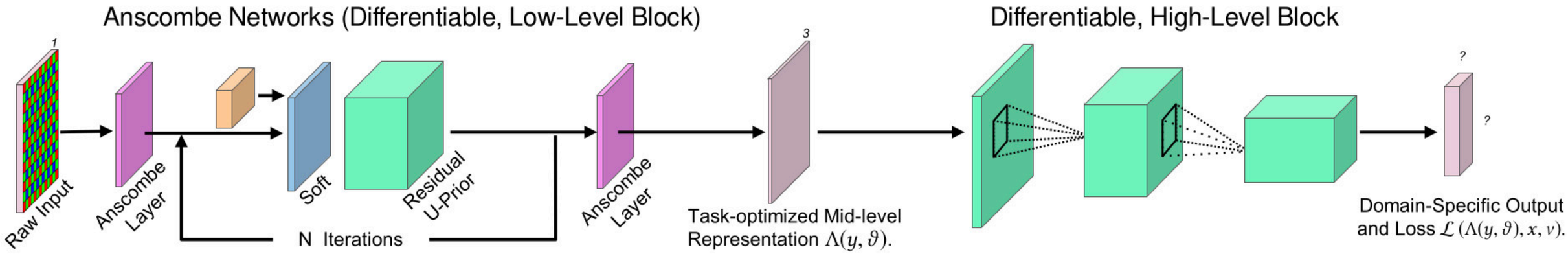
https://on-demand.gputechconf.com/gtc/2016/webinar/getting-started-jetpack-camera-api.pdf
https://www.servethehome.com/nvidia-jetson-agx-xavier-module-for-robotics-development/nvidia-jetson-agx-xavier-soc/

ISP for Photography vs. for Machine Vision

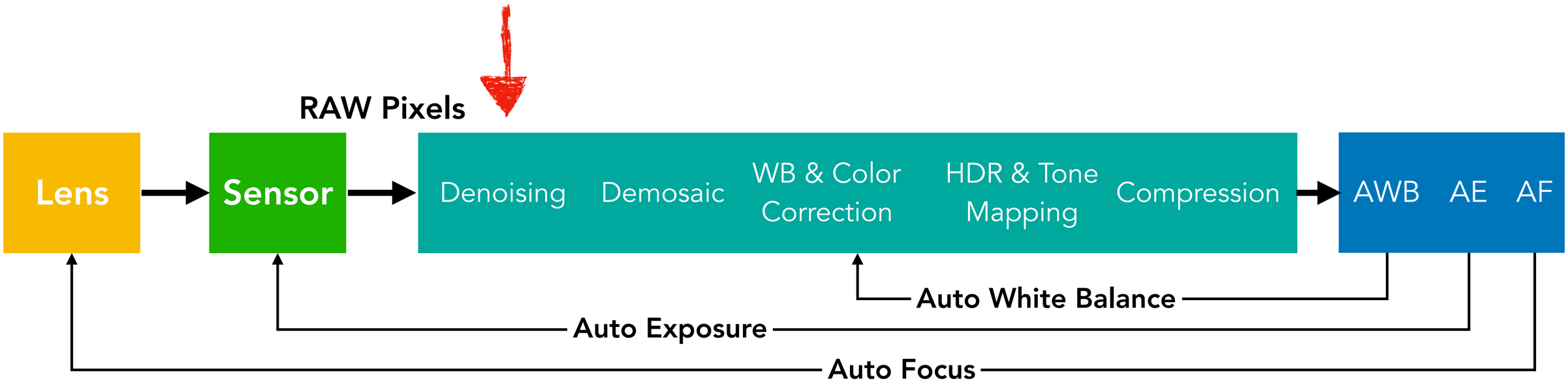
Image reconstruction algorithms designed to produce visually pleasing images for humans are not necessarily appropriate for computer vision tasks.

Co-design/co-train the ISP algorithms with down-stream vision tasks.

Rethink optics and sensor hardware design too.



Denoising


RAW Pixels
Lens
Sensor
Denoising
Demosaic
WB & Color Correction
HDR & Tone Mapping
Compression
AWB
AE
AF
Auto White Balance
Auto Exposure
Auto Focus

Denoising

Sensor/sensing introduces various sources of noise.

Post-processing stages impact (amplify) noise levels.

- Since they manipulate noisy signals.

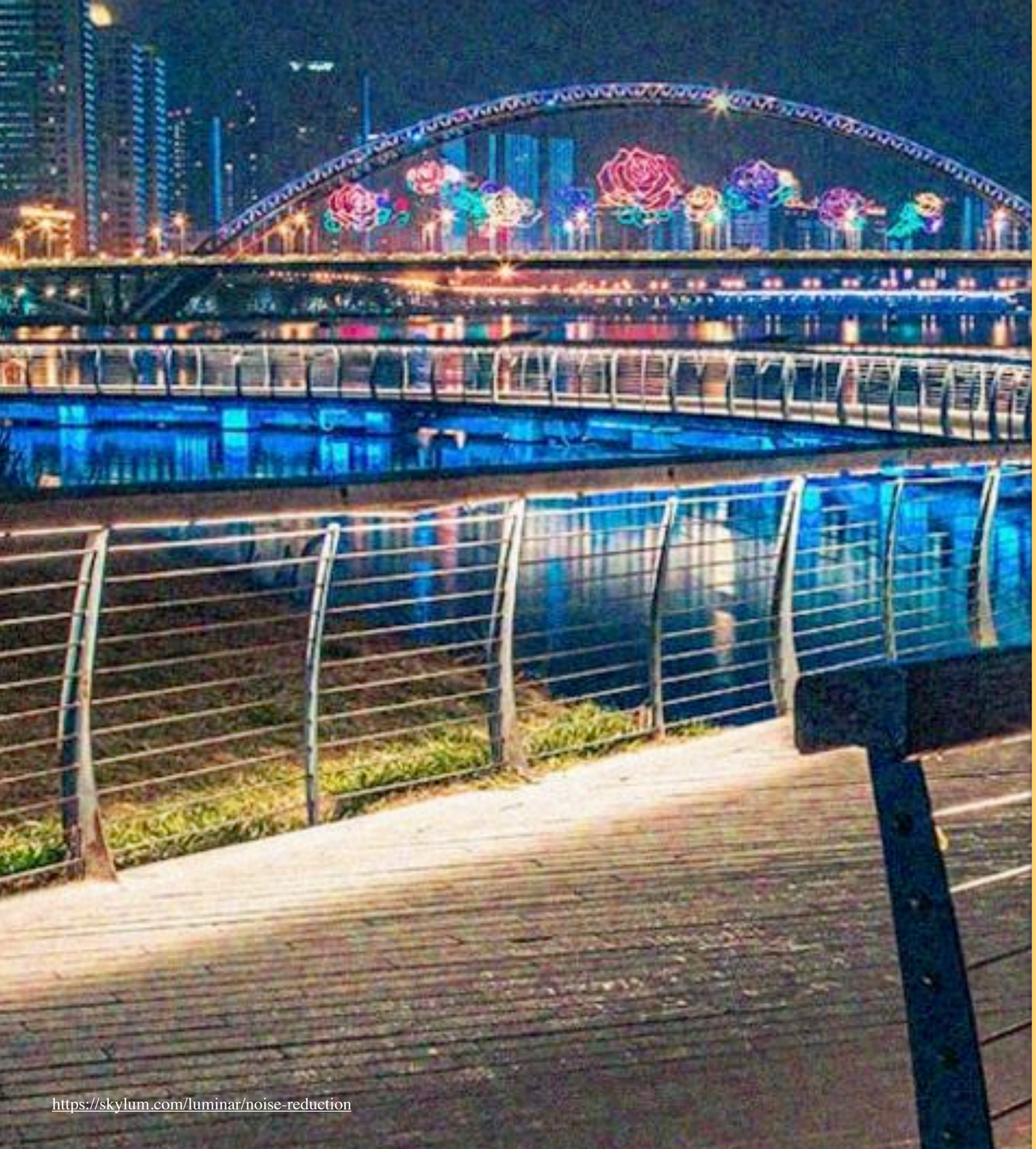
Denoise as early as possible, and can denoise multiples times.

The easiest way to denoise an image is to **blur** the image.

- Always a trade-off between image details retained vs. noise removed.

Spatial denoising vs. temporal denoising

- Single-image denoising vs. video denoising

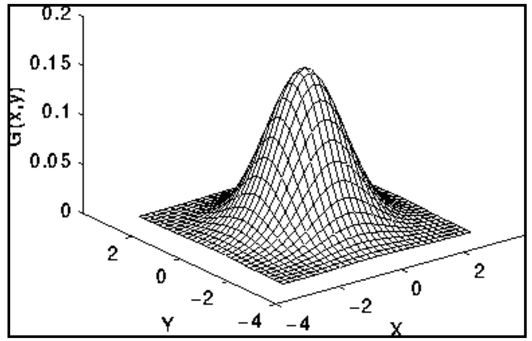


https://skylum.com/luminar/noise-reduction


Looks
152%
Library
Edit
Before
After
Essentials
nl_slider_1_b.jpeg
Light
AI Enhance
AI Structure
Color
B&W Conversion
Details Enhancer
Denoise
Luminosity Denoise
83
Color Denoise
82
Advanced Settings
Boost
33
Edit Mask
https://skylum.com/luminar/noise-reduction

Gaussian Filter

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



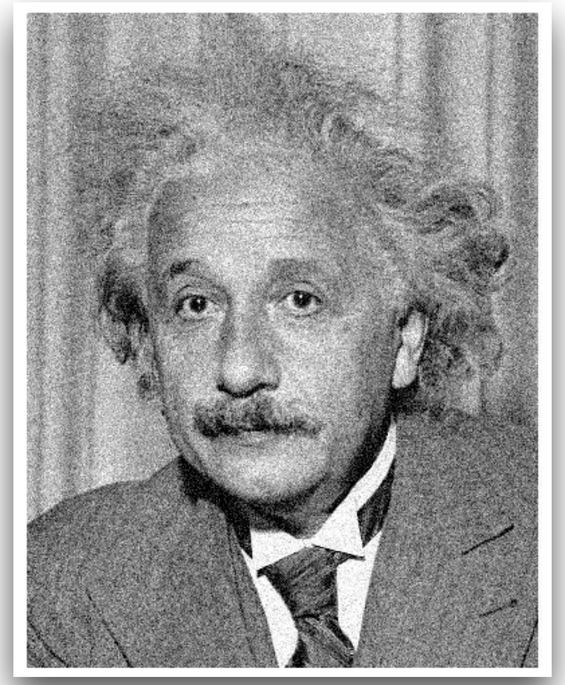
2D Gaussian distribution

$$\frac{1}{273}$$

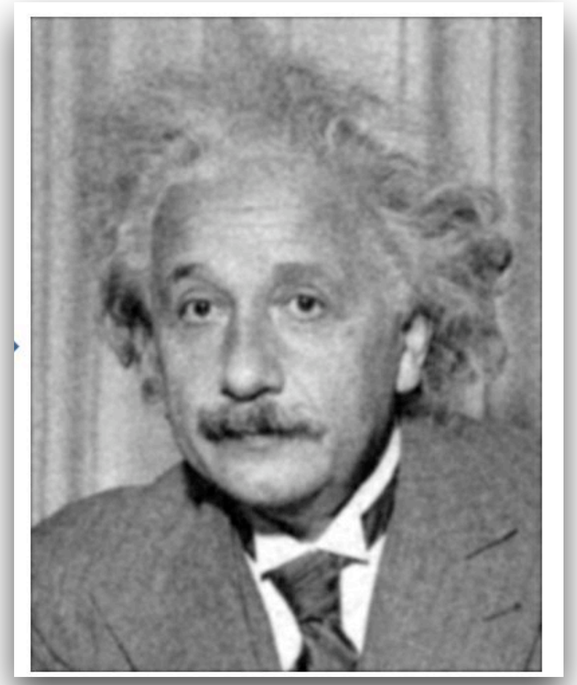
1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

A sample 2D Gaussian kernel with mean [0, 0] and σ=1

Original Image



Gaussian-filtered Image



A Gaussian filter also averages neighboring pixels, but gives more weight to closer neighbors. It's still a low-pass filter.

https://slideplayer.com/slide/2489074/ https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

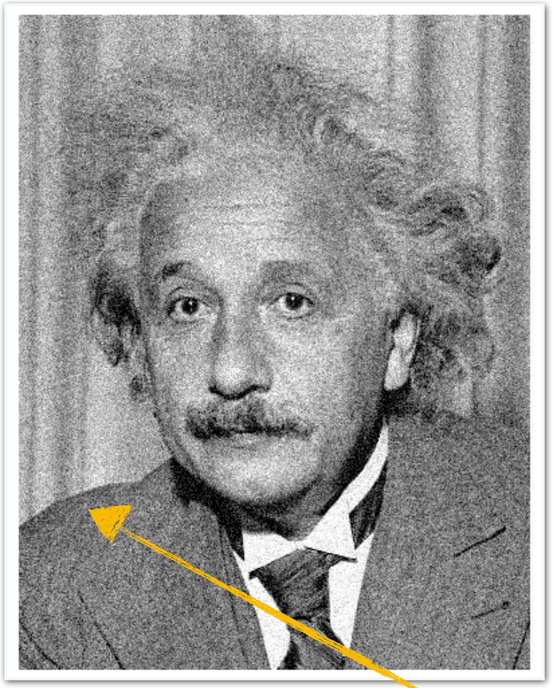
Issue with Simple Filtering

Issue: denoising using blurring works only when the scene is smooth (low spatial frequency). When scenes are not smooth (high frequency), i.e., with edges, blurring will destroy edges and the image looks less sharp.

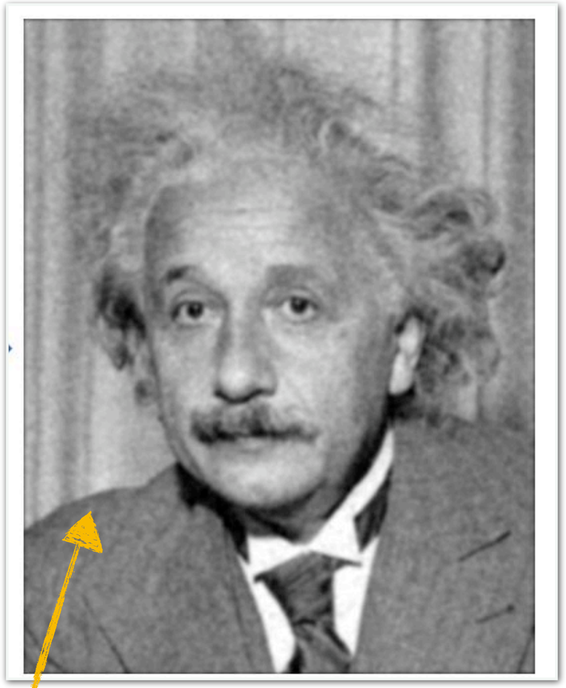
- Very similar to naive demosaic artifacts.

Solution: don't blur across the edge!

Original Image



Gaussian-filtered Image



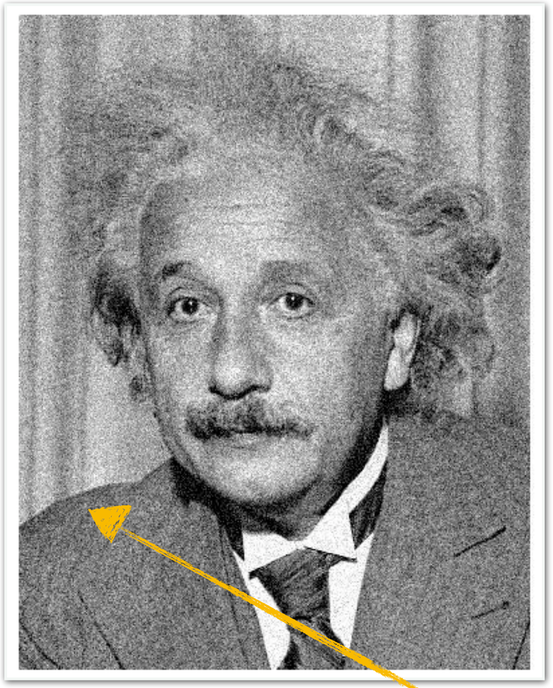
Edges become less sharp after Gaussian filtering

Bilateral Filter: Edge-Preserving Denoising

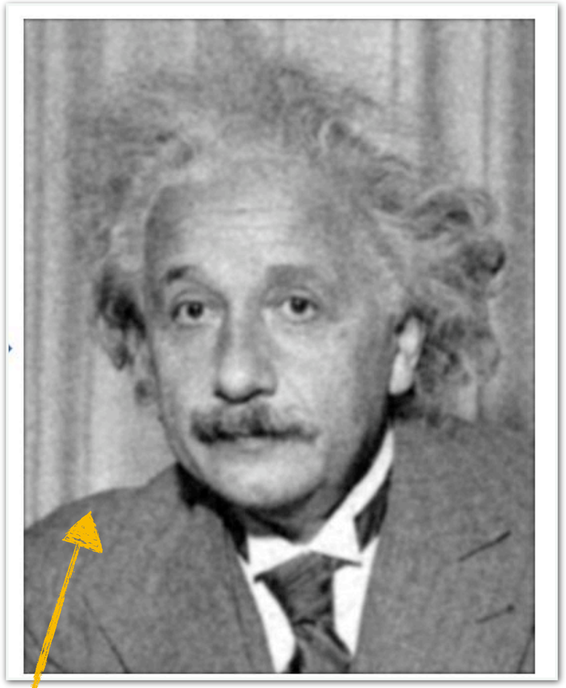
Solution: don't blur across the edge!

What's an edge? A simple but effective heuristic: **pixel intensity** or **color difference**!

Original Image



Gaussian-filtered Image



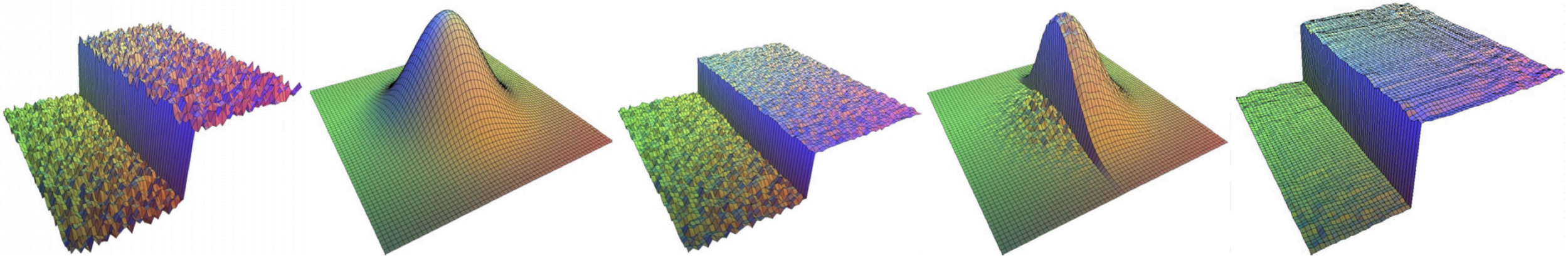
Edges become less sharp after Gaussian filtering

Bilateral Filter: Edge-Preserving Denoising

Idea: Use the product of two kernels as the filtering kernel

- Weights in first kernel are dictated by pixel **distance**. Closer pixels have higher weights.
- Weights in second kernel are dictated pixel **color**. Similar pixels have higher weights.

Color difference usually calculated as pixel value difference. The original bilateral filtering paper [ICCV 98] uses perceptual difference.

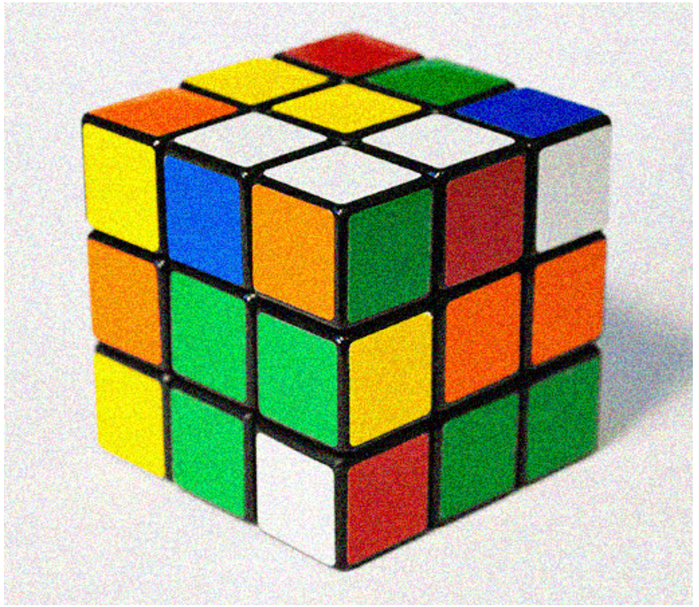


input | spatial kernel f | influence g in the intensity domain for the central pixel | weight $f \times g$ for the central pixel | output

Fast Bilateral Filtering for the Display of High-Dynamic-Range Images, SIGGRAPH 2002 Bilateral Filtering for Gray and Color Images (ICCV 1998)

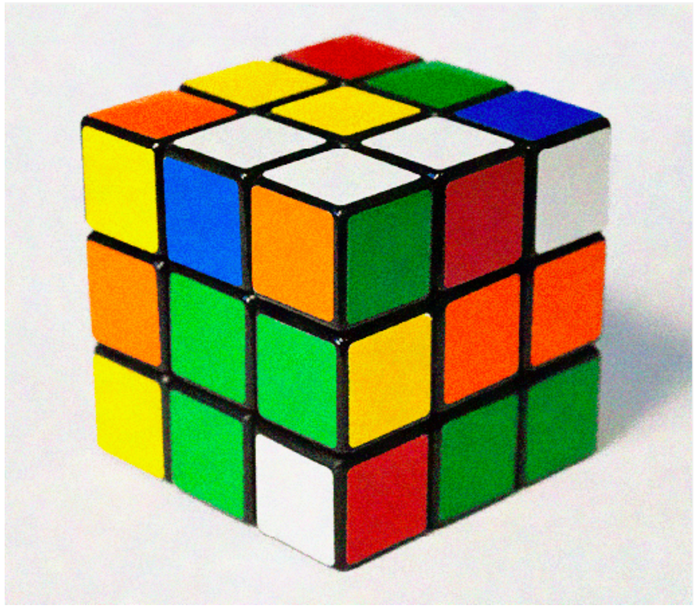
Bilateral Filter: Edge-Preserving Denoising

Noisy image



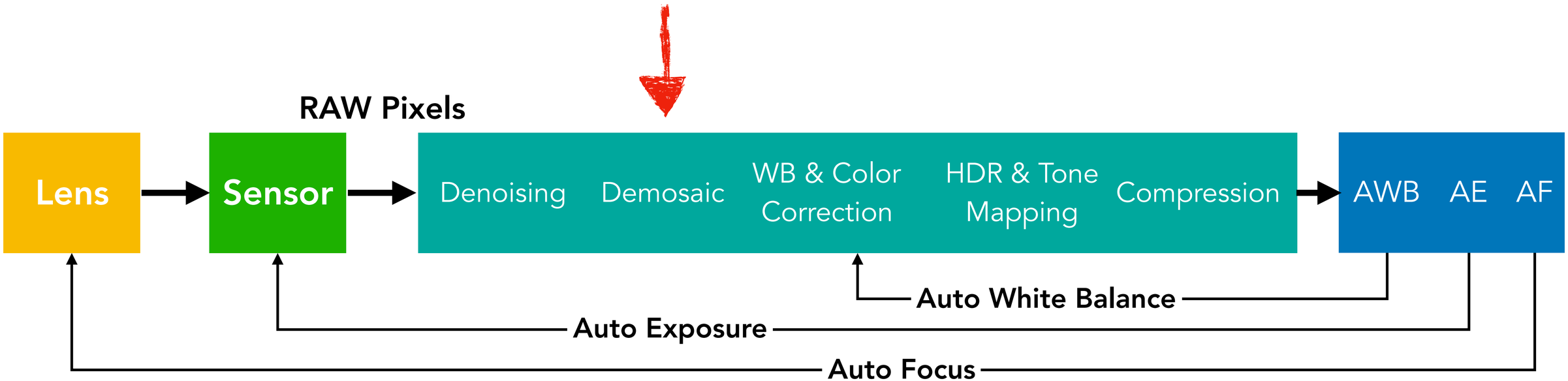
Gaussian filtered image

Bilateral filtered image



https://cs.brown.edu/courses/cs129/labs/lab_bilateral/index.html

Demosaicing



Need For Demosaicing (Spatial Color Reconstruction)

24	?	24	?	24	?
?	36	?	36	?	36
12	?	19	?	28	?
?	13	?	27	?	30
11	?	12	?	31	?
?	14	?	28	?	32

?	24	?	25	?	24
?	?	?	?	?	?
?	17	?	23	?	32
?	?	?	?	?	?
?	11	?	12	?	32
?	?	?	?	?	?

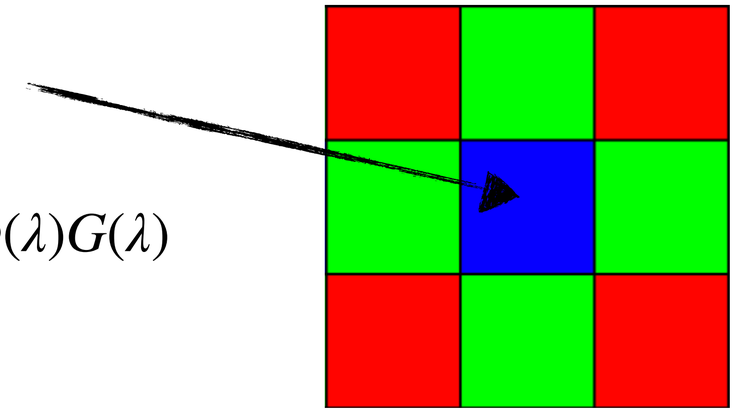
?	?	?	?	?	?
33	?	24	?	30	?
?	?	?	?	?	?
12	?	20		32	?
?	?	?	?	?	?
12	?	12	?	32	?

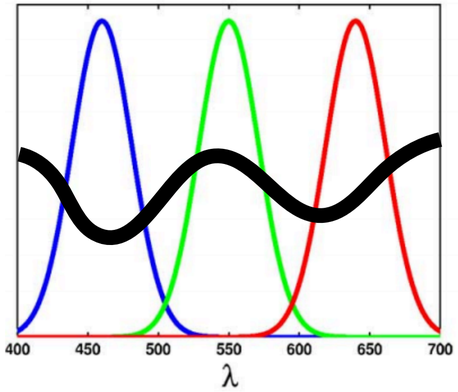
Figure 7-2 The pattern of green, red and blue pixels in a raw image. The question marks represent unknown data needed for full color channel resolution.

Need For Demosaicing (Spatial Color Reconstruction)

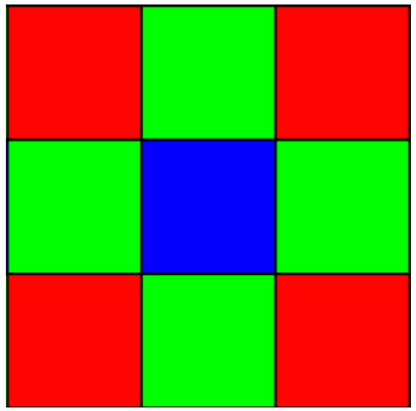
What we have $\int \Phi(\lambda)B(\lambda)$

What we need to reconstruct $\int \Phi(\lambda)R(\lambda)$ $\int \Phi(\lambda)G(\lambda)$

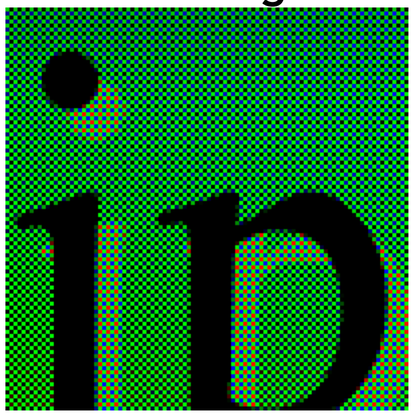




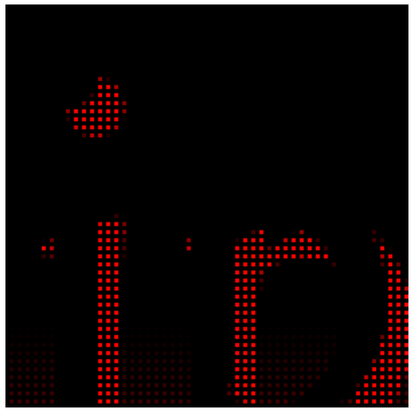
Bayer filter



A bayer-filtered raw image



Red channel



Green channel

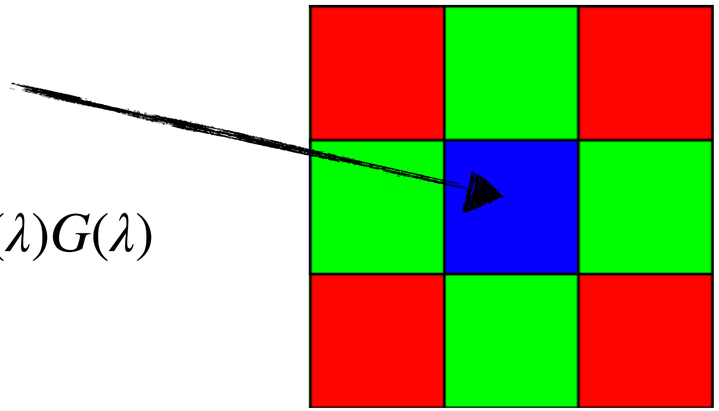
Blue channel

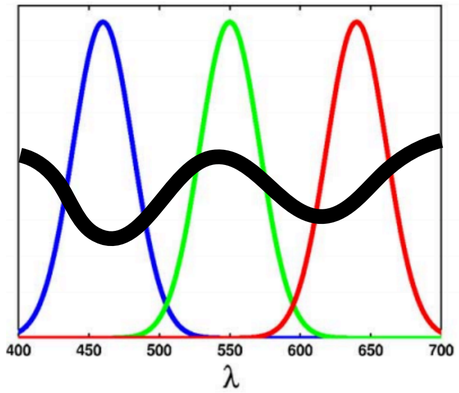
Selection of Optimal Spectral Sensitivity Functions for Color Filter Arrays, IEEE Transactions on Image Processing, 2010

Need For Demosaicing (Spatial Color Reconstruction)

What we have $\int \Phi(\lambda)B(\lambda)$

What we need to reconstruct $\int \Phi(\lambda)R(\lambda)$ $\int \Phi(\lambda)G(\lambda)$





Goal of demosacing: reconstruct the missing color values of each pixel as if the corresponding color filters were there.

An artifact of using CFA for color sensing (spatial color sampling). Not needed in monochromatic sensors or in sensors that use three chips.

Demosaicing Idea

Ideally: reconstruct the continuous signal first and then resample.

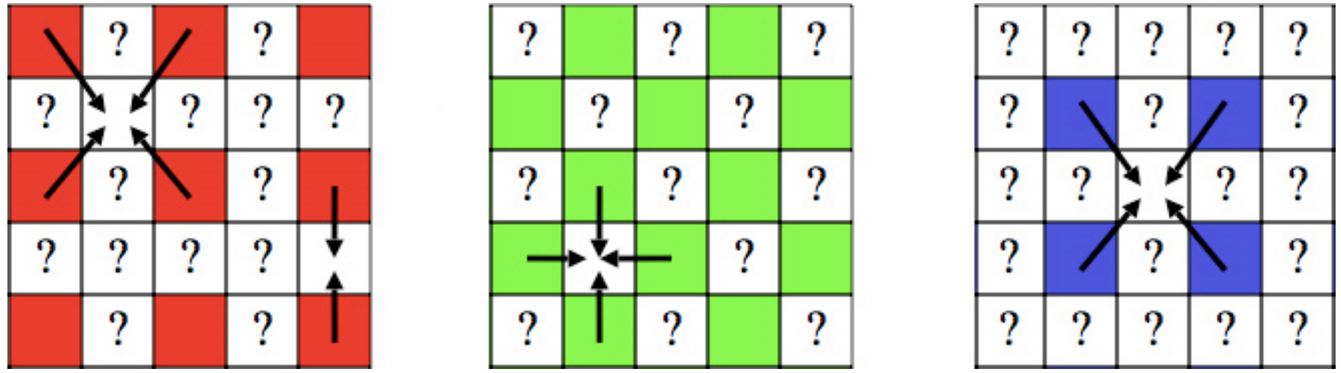
- If scene is smooth (band-limited), reconstruction and sampling becomes interpolation.

Intuitively: nearby pixels likely receive similar incident lights; missing pixel color can be interpolated from the same channel in nearby pixels.

Reality: scene has high-frequency components so reconstruction will mostly be aliased. Anti-aliased demosaicing.

Demosaicing Using Spatial Correlation

Assumption: scene is smooth,
i.e., low spatial frequency.

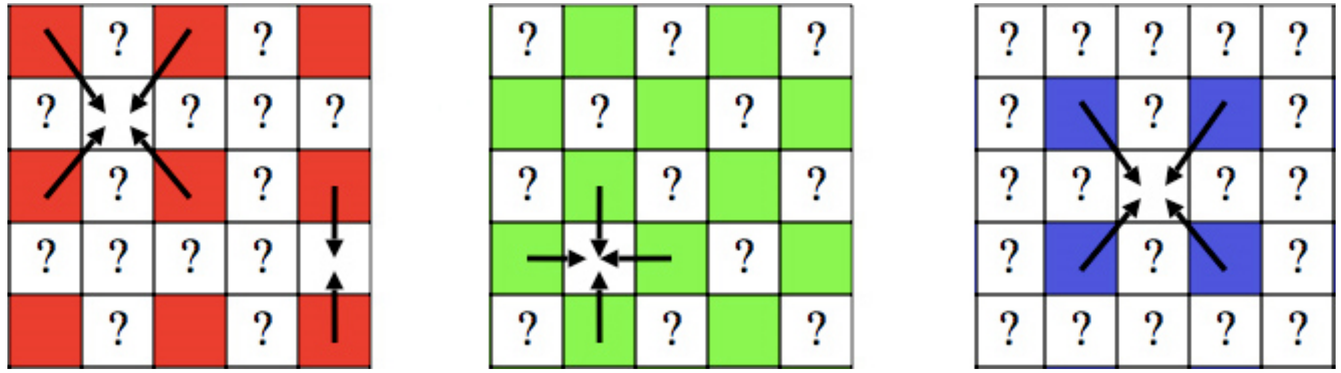


http://dmmd.net/main_wp/research/demosaicing/

Demosaicing Using Spatial Correlation

Assumption: scene is smooth, i.e., low spatial frequency.

Fails when the spatial frequency in the scene is high, i.e., local details are not smooth.

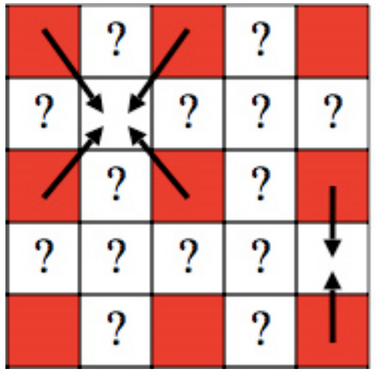


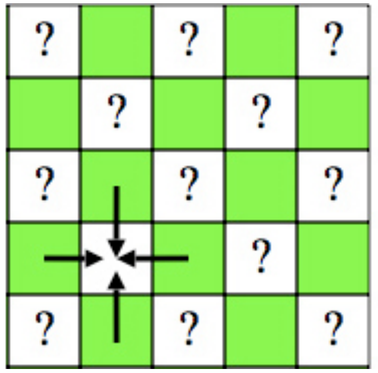
http://dmmd.net/main_wp/research/demosaicing/

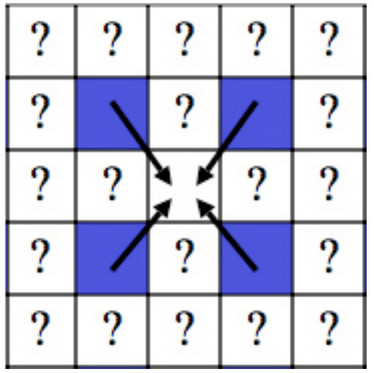
Demosaicing Using Spatial Correlation

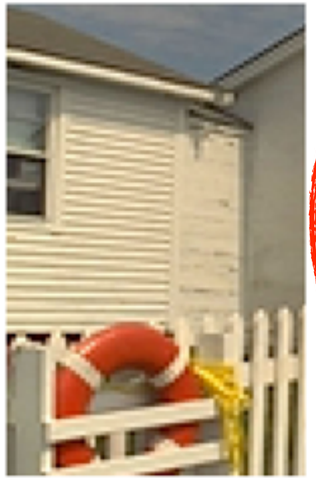
Assumption: scene is smooth, i.e., low spatial frequency.

Fails when the spatial frequency in the scene is high, i.e., local details are not smooth.









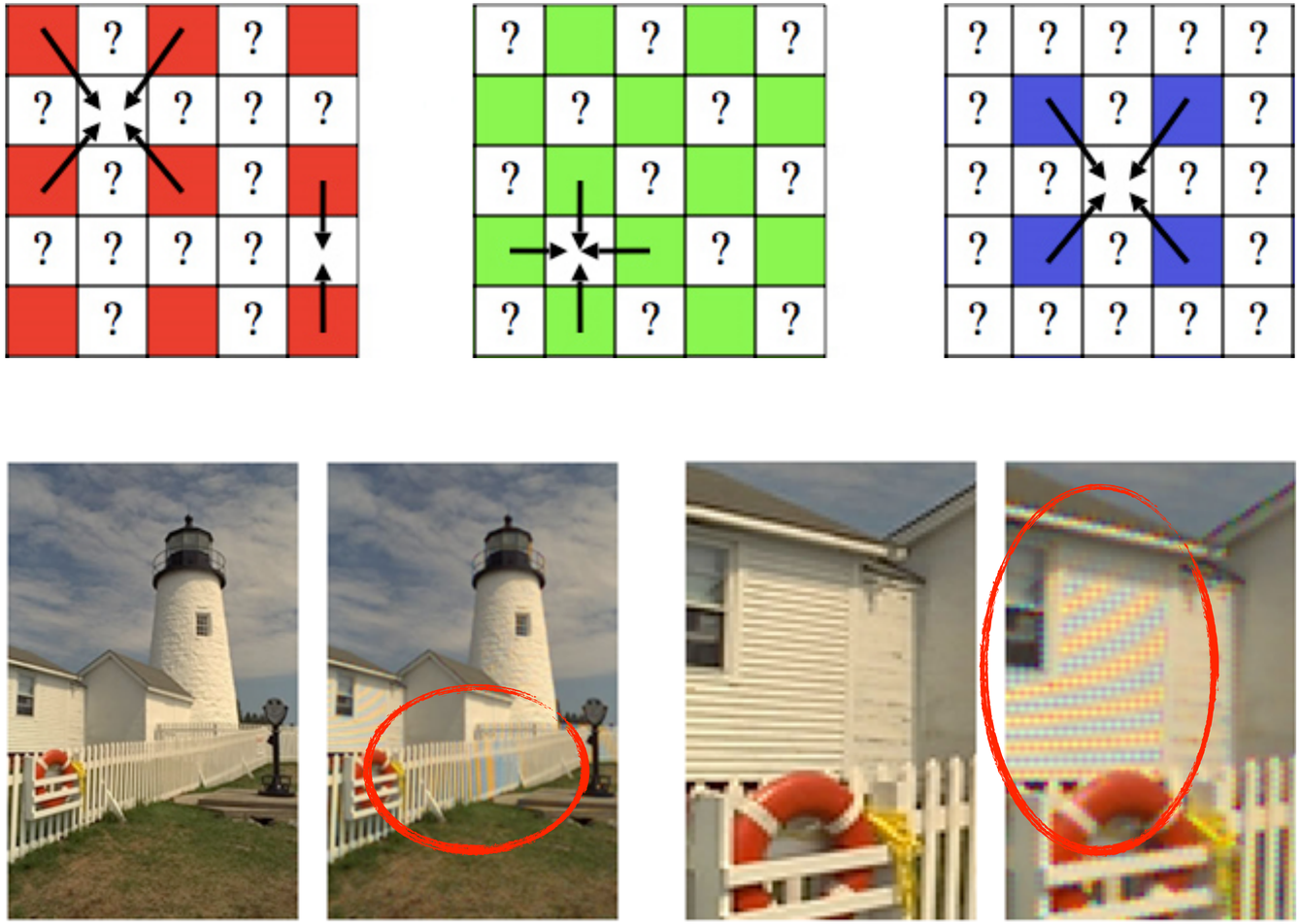
http://dmmd.net/main_wp/research/demosaicing/

Demosaicing Using Spatial Correlation

Assumption: scene is smooth, i.e., low spatial frequency.

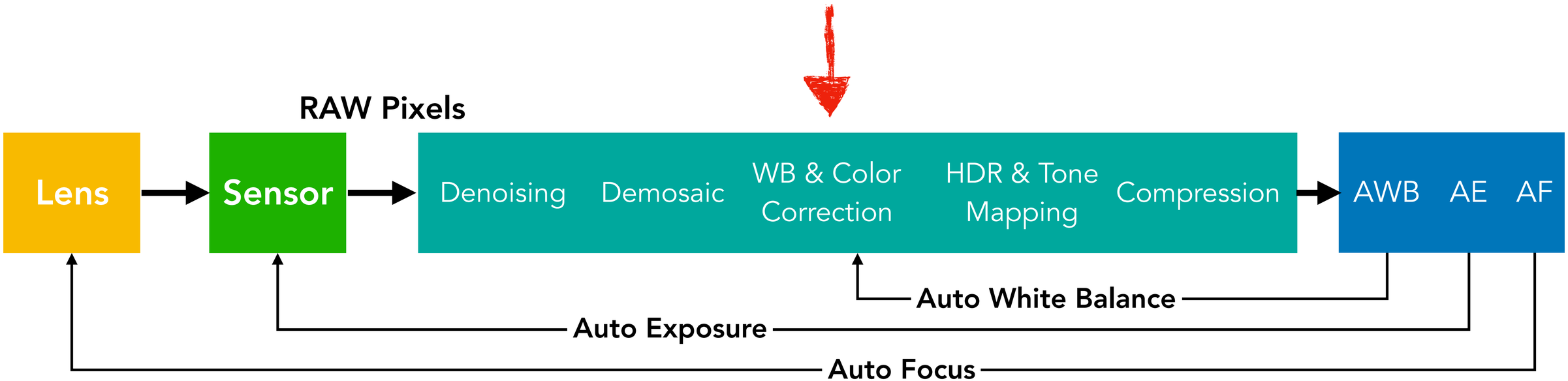
Fails when the spatial frequency in the scene is high, i.e., local details are not smooth.

Mitigation: detect edges, and then interpolate *along the edges* but *not across* the edges.



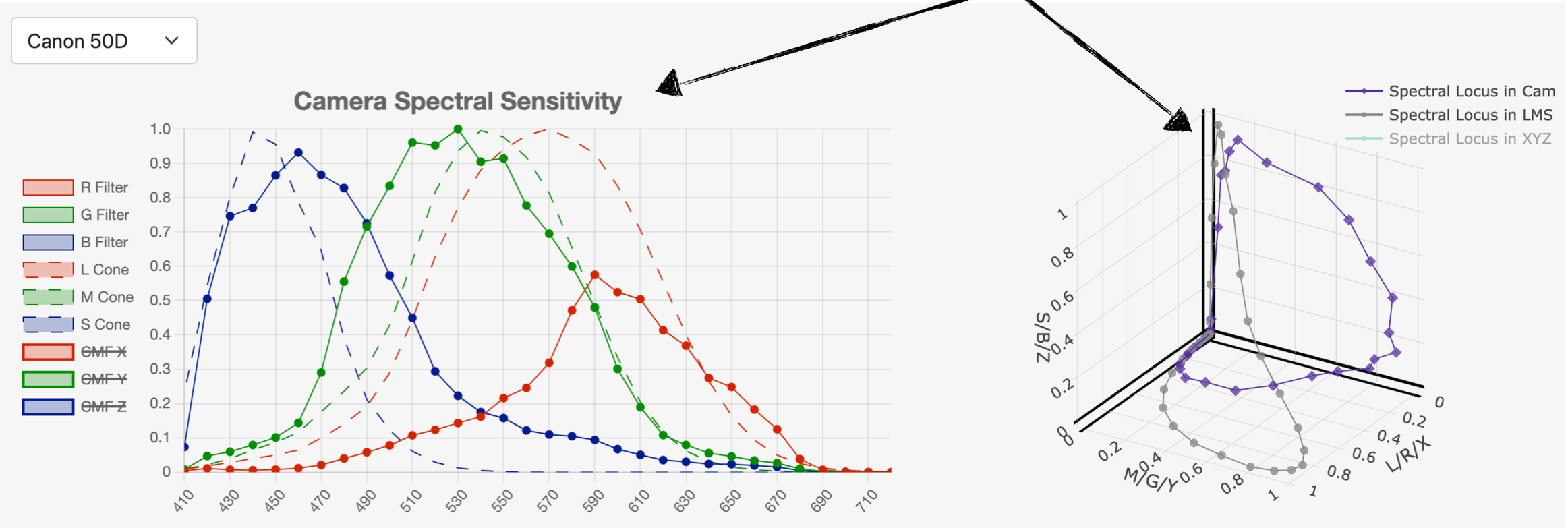
http://dmmd.net/main_wp/research/demosaicing/

Color Correction



The Need for Color Correction

Camera SSFs do not overlap with LMS sensitivities (or any other CMFs). Need to find a transformation, ideally linearly, to other color spaces.

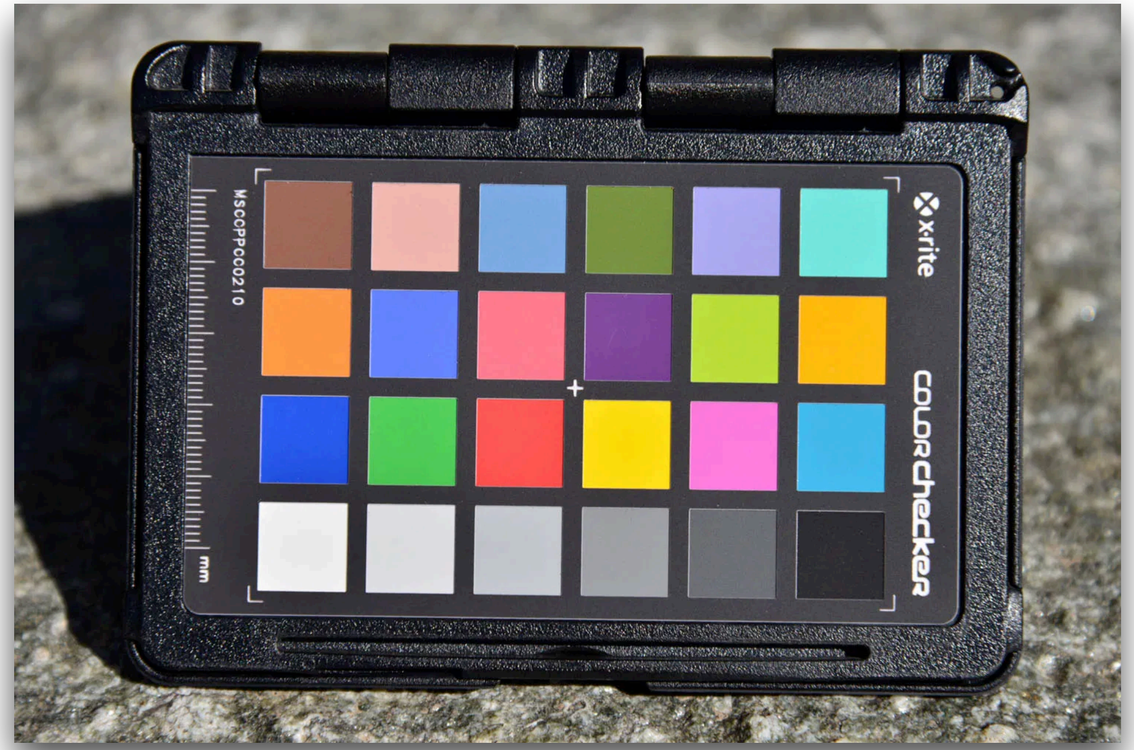


https://www.cs.rochester.edu/courses/572/colorvis/camcolor.html

The Goal

Calculate the transformation matrix that converts the raw RGB values of a color in the camera-internal space to the true tristimulus values of a known color space (e.g., XYZ). Minimize the conversion error.

Correct colors for common materials seen in nature. MacBeth ColorChecker board is a common calibration target.



https://www.youtube.com/watch?v=Q7Fh54VZcgE https://en.wikipedia.org/wiki/ColorChecker

Color Correction Procedure

1. Calculate the XYZ tristimulus values of all the patches under a known illuminant (e.g., D65). These will be our “ground truth” or “actual color.” The spectral reflectance of each patch in the checker (24 patches in this check board) is known.

2. Take a picture of a Macbeth “ColorChecker” under the same illuminant.

Raw RGB

True RGB

Color Correction Procedure

1. Calculate the XYZ tristimulus values of all the patches under a known illuminant (e.g., D65). These will be our “ground truth” or “actual color.” The spectral reflectance of each patch in the checker (24 patches in this check board) is known.

2. Take a picture of a Macbeth “ColorChecker” under the same illuminant.

Raw RGB

True RGB

Color Correction Procedure

3. Read the raw RGB values of each color patch generated by the camera. Those are the tristimulus values in the raw camera color space, and most likely don’t match the tristimulus values in the CIE XYZ space.

Raw RGB

True RGB

Color Correction Procedure

4. Now calculate the transformation matrix. This is an overdetermined system (3x3 variables but 24x3 equations). Formulate it as an optimization problem
   - usually in the form of linear least squares (LLS).
   - can use Euclidean distance as loss but CIELAB ΔE* is more common (and better)

$$\begin{bmatrix} T_{00}, T_{01}, T_{02} \\ T_{10}, T_{11}, T_{12} \\ T_{20}, T_{21}, T_{22} \end{bmatrix} \times \begin{bmatrix} R_1 & \dots & R_{24} \\ G_1 & \dots & G_{24} \\ B_1 & \dots & B_{24} \end{bmatrix} = \begin{bmatrix} X_1 & \dots & X_{24} \\ Y_1 & \dots & Y_{24} \\ Z_1 & \dots & Z_{24} \end{bmatrix}$$

Raw RGB

True RGB

Color Correction Procedure

5. Optimizations are not a perfect. Usually a non-linear step in the end (through look-up tables) brings the transformed color further closer to the actual color.

6. The transformation matrix depends on the illuminant. Usually cameras supply a different matrix for a different illuminant and interpolate for new illuminants.

Raw RGB

True RGB

Exploring Camera Color Space

cs.rochester.edu/courses/572/colorvis/camcolor.html

Exploring Camera Color Space and Color Correction

Introduction

Do cameras see the same color as us? Can cameras always accurately reproduce colors that our eyes see? This interactive tutorial explores these questions and many more interesting aspects of camera raw color space. In particular, we will walk you through an important concept in both color science and camera signal processing: color correction, the process of correcting the color perception of a camera such that it is as close to ours as allowed. In the end, you will get to appreciate why you should never trust the color produced by your camera and how you might build your own camera that, in theory, out-performs existing cameras in color reproduction.

Caveats. 1) This tutorial demonstrates the principle of color correction with many important, but subtle, engineering details omitted; we will mention them when appropriate. 2) Color correction is one of the two components in camera color reproduction, the other being white balance (or rather, camera's emulation of chromatic adaption of human visual system). We have a post that discusses the principles of chromatic adaptation and its application in white balance. The relationship of color correction and white balance is quite tricky, but Andrew Rowlands has a fascinating article that demystefies it for you.

Step 1: Exploring Camera Color Space

In principle, cameras work just like our eyes. Our retina has three types of cone cells (L, M, and S), each with a unique spectral sensitivity, translating light into three numbers (the L, M, S cone responses) that give us color perception. Similar to the three cone types, (most) cameras use three color filters (commonly referred to as the R, G, and B filters), each with a unique spectral sensitivity and, thus, also translate light into three numbers. In this sense, you can really think of a camera as a "weird" kind of human being with unconventional LMS cone fundamentals. Not all cameras use three filters though. Telescope imaging cameras use five filters, just like butterflies!

The left chart below shows the measured camera sensitivity functions of 48 cameras in two recent studies. The 48 cameras are classified into four categories: DSLR, point and shoot, industrial cameras, and smartphone cameras. The sensitivities are normalized such that the most sensitive filter (usually the green filter) peaks at 1. What should be noted is that the sensitivities functions measured here are not just the spectral transmittances of the color filters; rather, they are measured by treating the camera as a black box, and thus reflect the combined effects of everything in the camera that has a spectral response to light, such as the anti-aliasing filter, IR filter, micro-lenses, the photosites, etc.

The default view shows the average sensitivities across the 48 cameras, but you can also select a particular camera from the drop-down list. As a comparison, we also plot the LMS cone fundamentals in the same chart as dashed lines. As is customary, the LMS cone fundamentals are, each, normalized to peak at 1. As is usually the case in color science, these normalizations merely introduce some scaling factors that will be canceled out later if we care about just the chromaticity of a color (i.e., the relative ratio of the primaries).

Color Reproduction vs. Noise

Color reproduction is concerned with:

- how easy is it for demosaicing algorithm to recover the missing colors in pixels?
- How easy is it for color correction to recover the tristimulus values of the light?

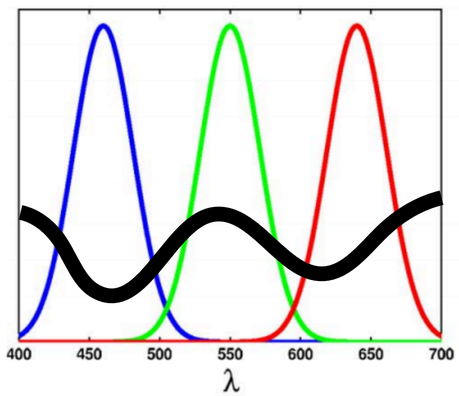
SNR consideration:

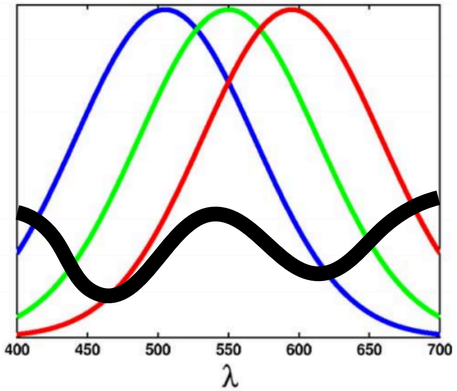
- how much noise is introduced in color reproduction?

SSFs of CFA affect both the accuracy of color reproduction and noise reduction of the sensor.

- Trade-offs need to be carefully evaluated.

For Demosaicing





When the SSFs overlap, the three channels are highly correlated, and one channel's information can be used to infer other channels in demosacing.

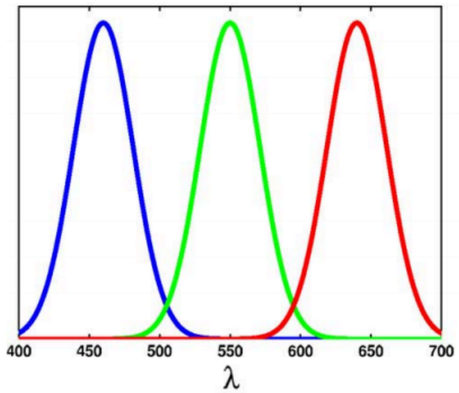
- Constant color difference heuristic: R/G (B/G) ratio is smooth across pixels. Usually G is first interpolated spatially, and then R (B) is interpolated from R/G (B/G).

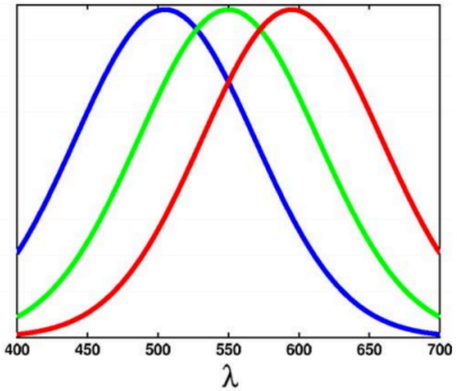
SSFs that are too wide are bad too. In the extreme case, the SSFs completely overlap: the three channel values are the same for any pixel.

- "Perfect" demosaicing! But…
- The camera is "color blind" — can detect only light intensity but not color.

Selection of Optimal Spectral Sensitivity Functions for Color Filter Arrays, IEEE Transactions on Image Processing, 2010

For Color Correction





First design: SSFs are narrower and have little overlap.

- Slight difference in light SPD might not be captured, making color correction harder.
- Extreme case: the three SSFs are delta functions, then two lights having the same power at the three detected wavelengths but differing widely elsewhere will have the same raw RGB values — color correction is impossible.

Second design: wider SSFs that overlap significantly.

- RGB values of the two lights are different in all three channels. Easier to correct color.
- Too wide is bad too, since raw RGB values will be too similar. The color correction matrix will have very large coefficients (amplify noise).

Impact of Color Correction on Noise

Linear sRGB values

Raw RGB values

$$\begin{bmatrix} SR \\ SG \\ SB \end{bmatrix} = \begin{bmatrix} a_{00}, a_{01}, a_{02} \\ a_{10}, a_{11}, a_{12} \\ a_{20}, a_{21}, a_{22} \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$SNR_R = \frac{\mu_R}{\sigma_R}$$

$$SR = a_{00} \times R + a_{01} \times G + a_{02} \times B$$

$$\mu_{SR} = a_{00} \times \mu_R + a_{01} \times \mu_G + a_{02} \times \mu_B$$

$$\sigma_{SR}^2 = a_{00}^2 \times \sigma_R^2 + a_{01}^2 \times \sigma_G^2 + a_{02}^2 \times \sigma_B^2$$

$$SNR_{SR} = \frac{\mu_{SR}}{\sigma_{SR}} = \frac{a_{00} \times \mu_R + a_{01} \times \mu_G + a_{02} \times \mu_B}{\sqrt{a_{00}^2 \times \sigma_R^2 + a_{01}^2 \times \sigma_G^2 + a_{02}^2 \times \sigma_B^2}}$$

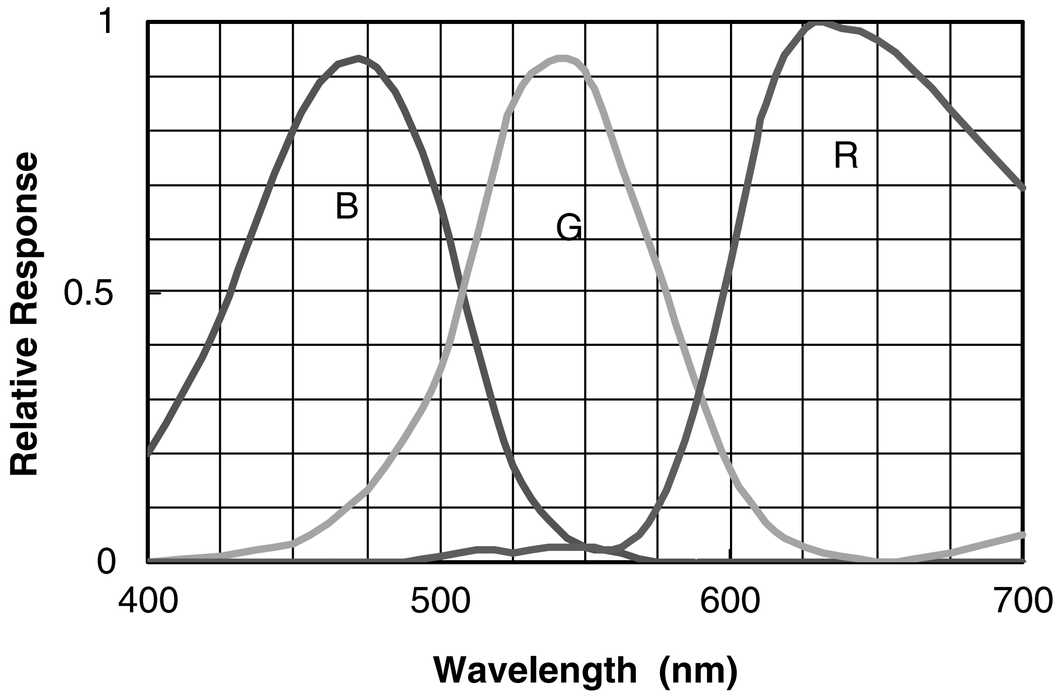
$$\approx \frac{(a_{00} + a_{01} + a_{02}) \times \mu_R}{\sqrt{a_{00}^2 + a_{01}^2 + a_{02}^2} \times \sigma_R}$$

Assuming same mean and noise characteristics across the three channels

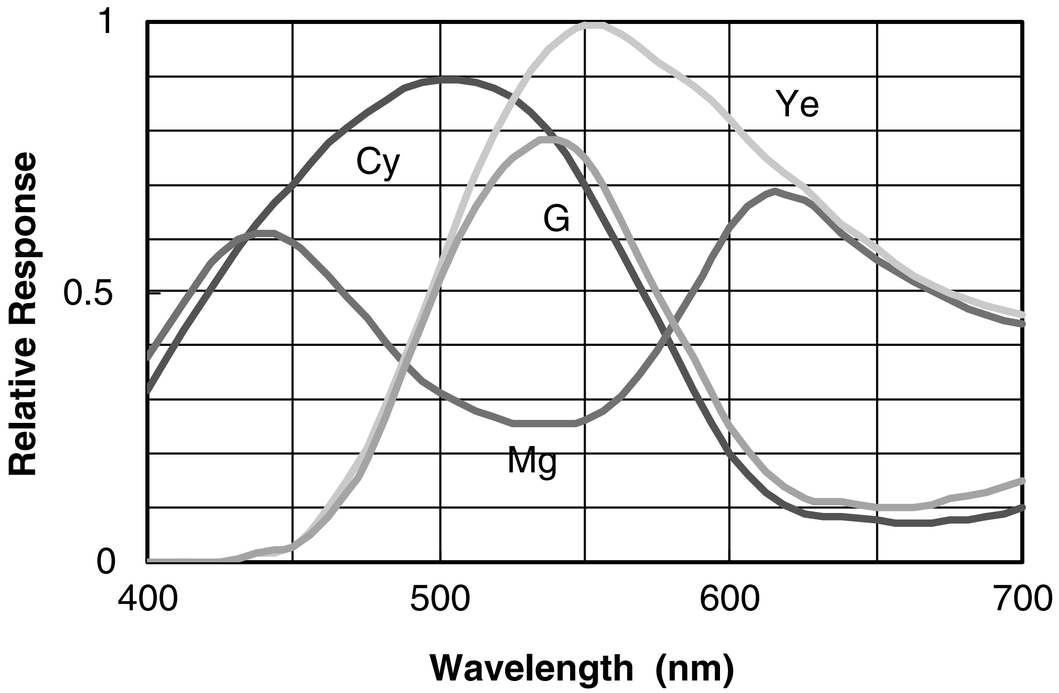
If < 1, SNR is worse

Two Color Filters

The nice thing about wider SSFs: they allow in more lights, which helps SNR



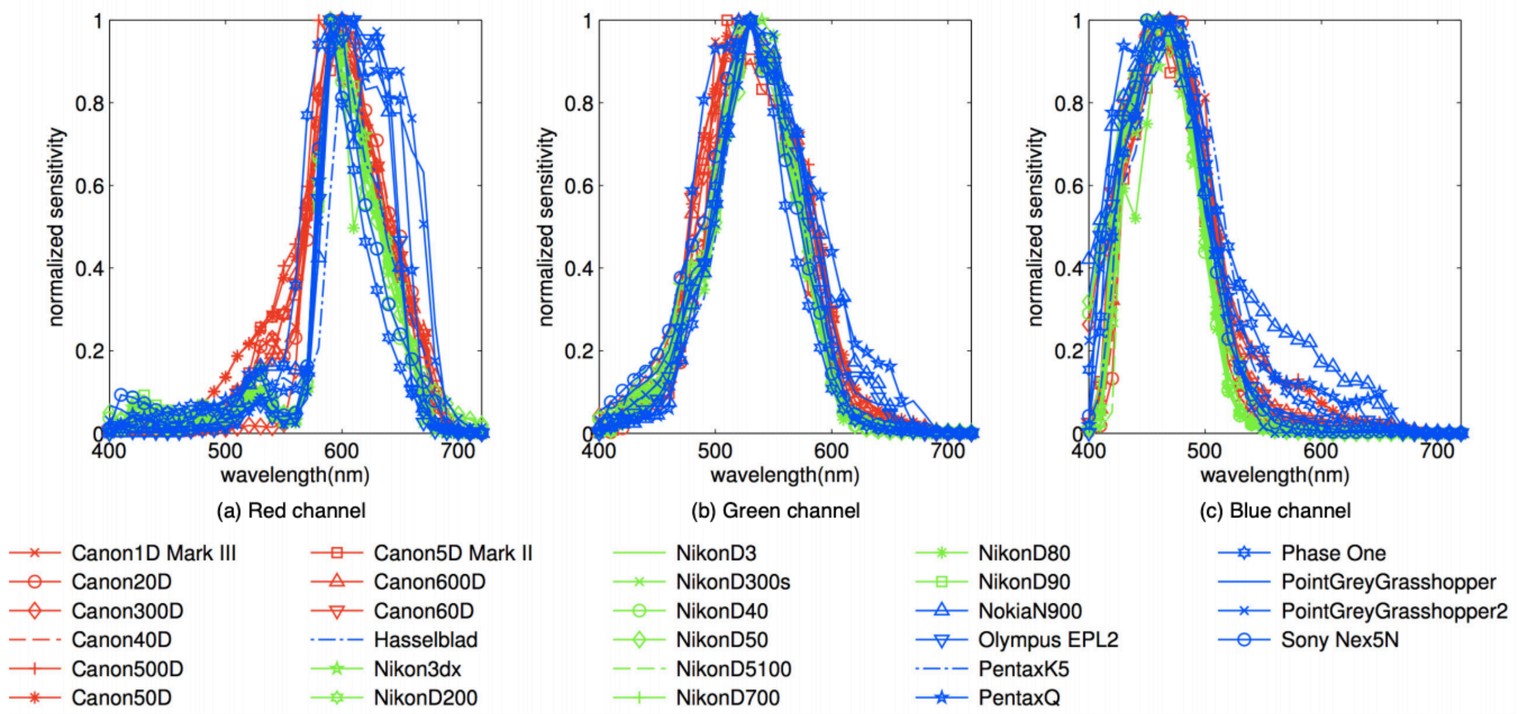
(a)



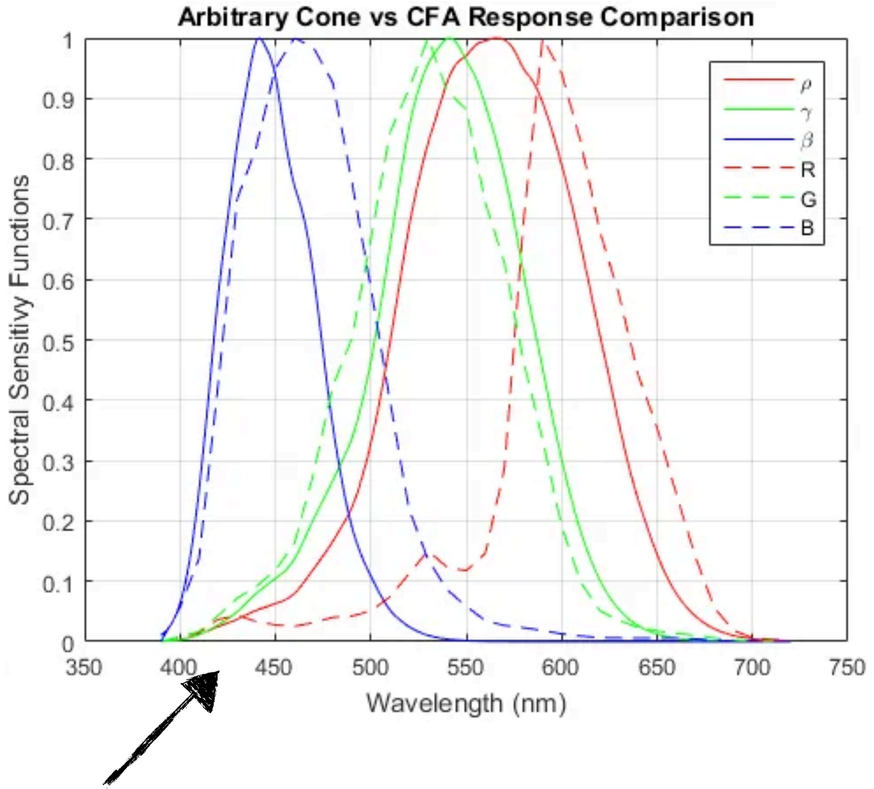
(b)

Real Spectral Sensitivity Functions

SSFs of 28 real cameras (DSLR, point-and-shoot, industrial and mobile cameras).



Average SSF vs. LMS cone response functions.



Actual camera SSFs resemble CMFs, but they are less overlapped, presumably to improve SNR in the color correction process.