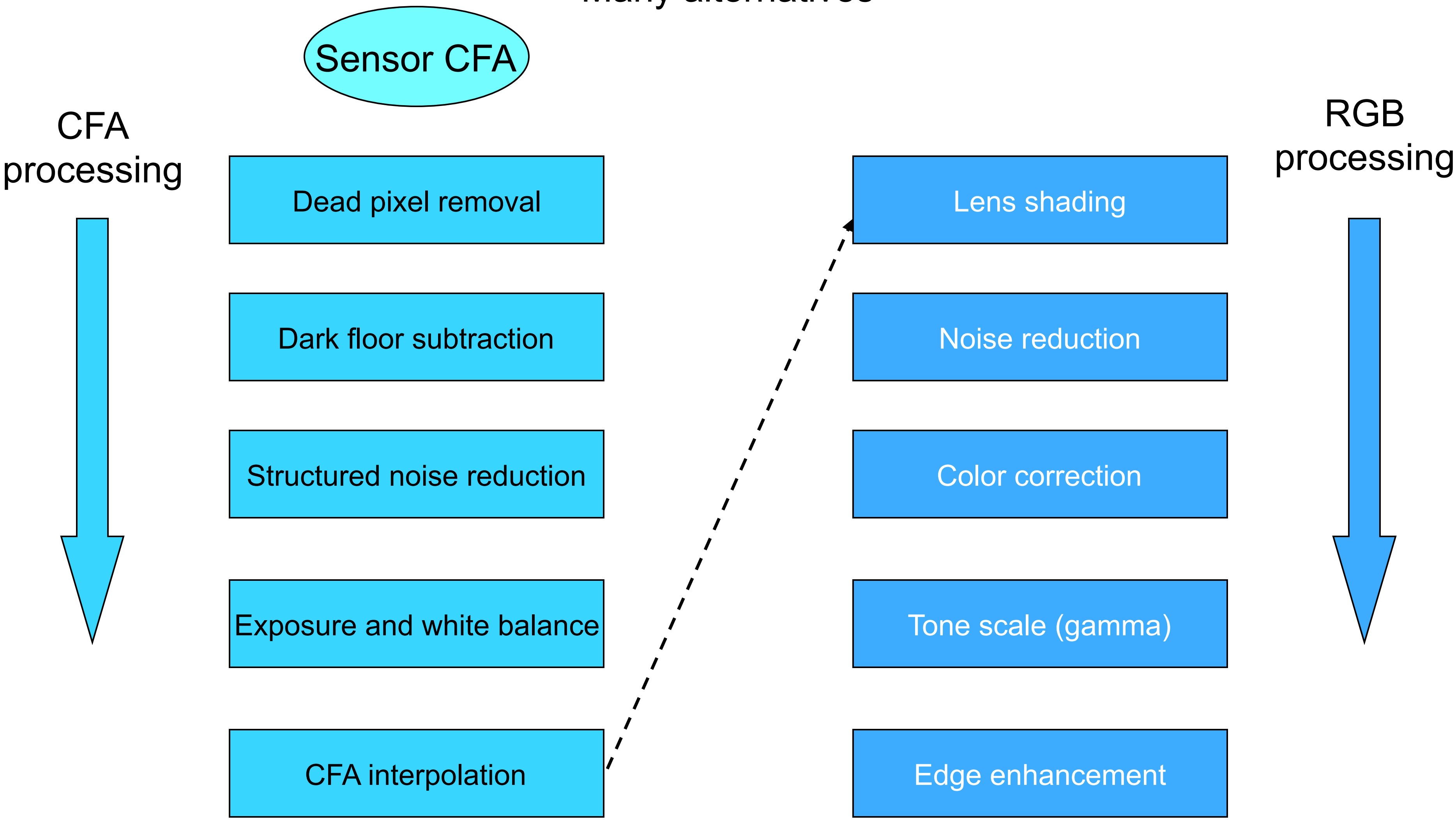
Image processing pipeline

- Correct hardware imperfections
- Address appearance (aesthetics) issues
    - Reproduce an image whose appearance matches the original up to display intensity and gamut limitations, or
    - Make a nice picture, guided by the original
    - Preserve certain image metrics (edges, geometry)

Reference pipeline

(after Adams and Hamilton, in Lukas, Fig. 3.1)

Many alternatives



Reference pipeline II

Design Considerations of Color Image Processing Pipeline for Digital Cameras

Wen-Chung Kao, *Member*, IEEE, Sheng-Hong Wang, Lien-Yang Chen, and Sheng-Yuan Lin

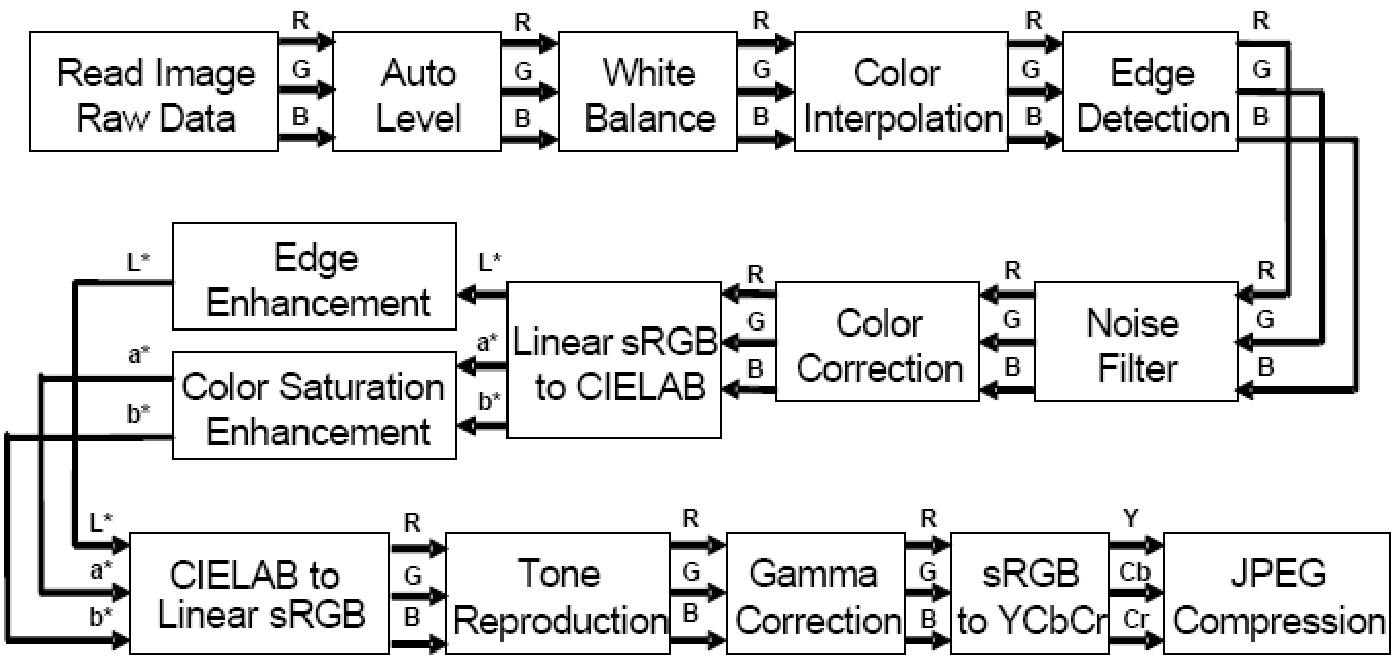


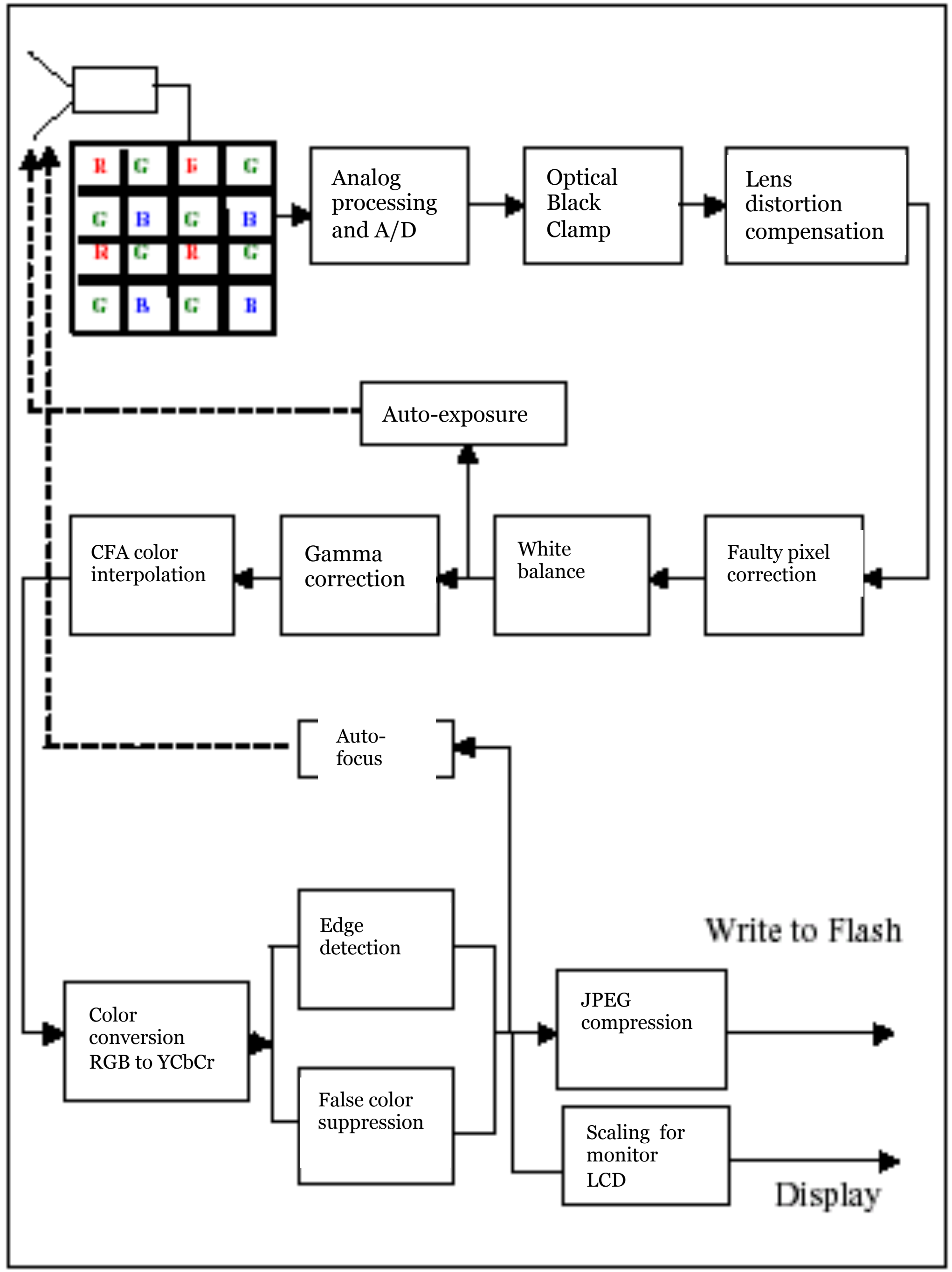
Fig. 1. The proposed image processing pipeline.

Color pipeline

Texas Instruments

TMS320DSC21

Chip controllers for the pipeline

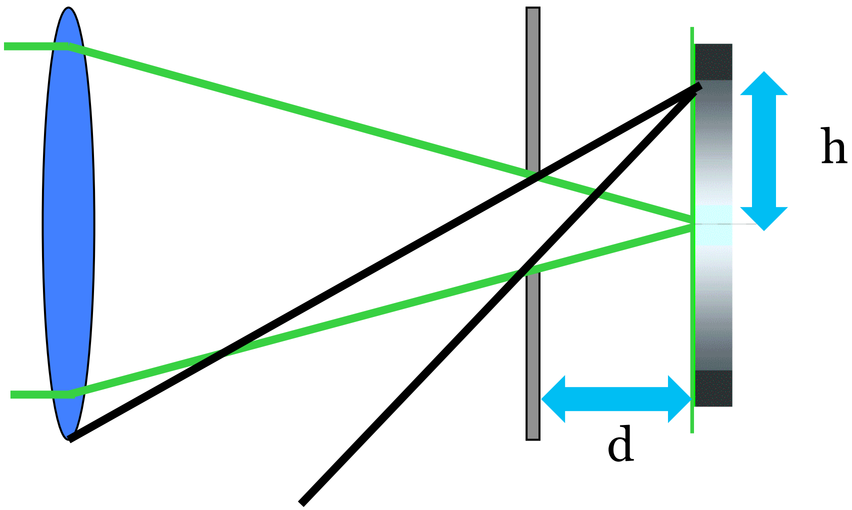


Relative illumination

- Vignetting - cos4th

Relative illumination

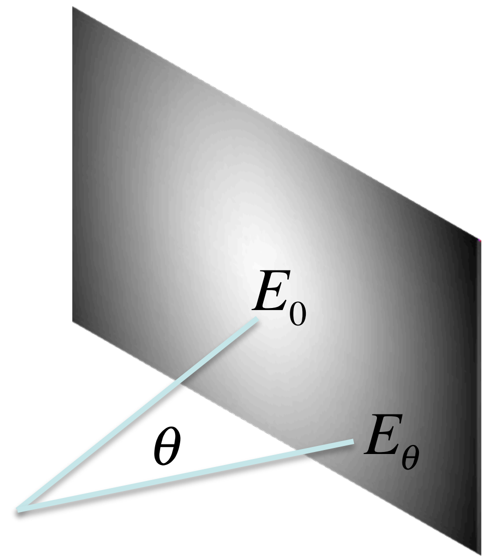
- Off-axis pixels receive a smaller bundle of incident rays compared to on-axis
- Relative illumination at the edge is lower



Relative illumination

<u>General approximation</u>

$$\frac{E_\theta}{E_0} = \cos^4(\theta)$$



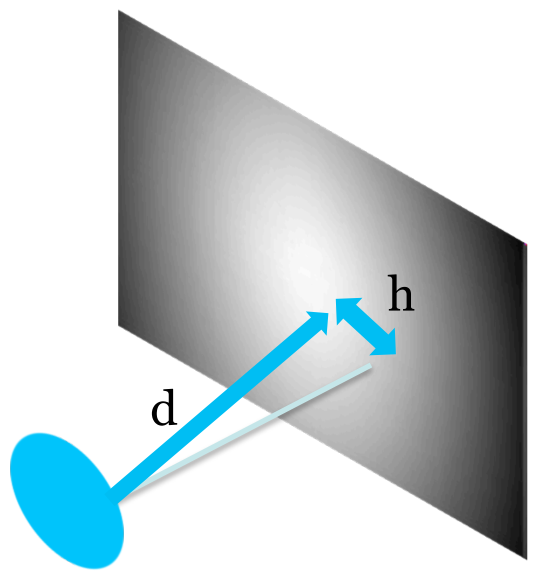
<u>Paraxial approximation (small h)</u>

$h =$ image field height

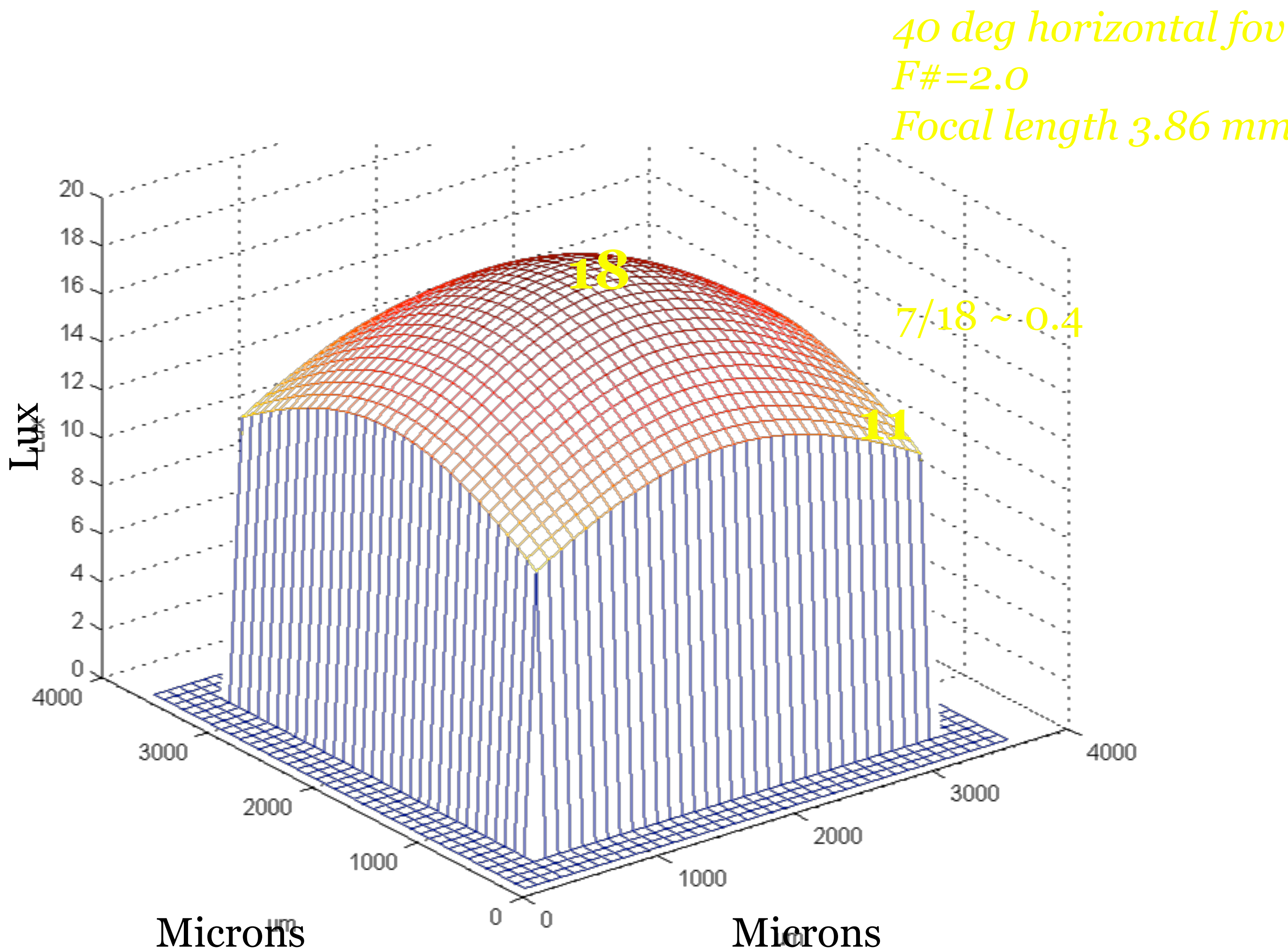
$d =$ distance from lens to image plane

$S = \sqrt{d^2 + h^2}$

Relative illumination $= (h / S)^4$

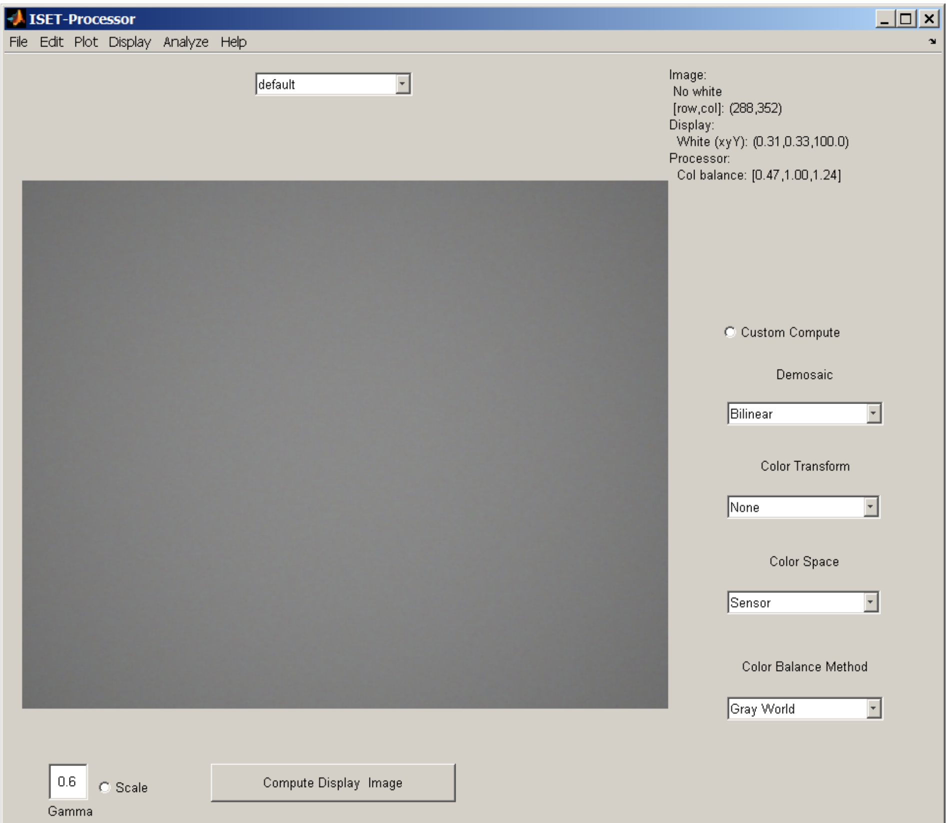


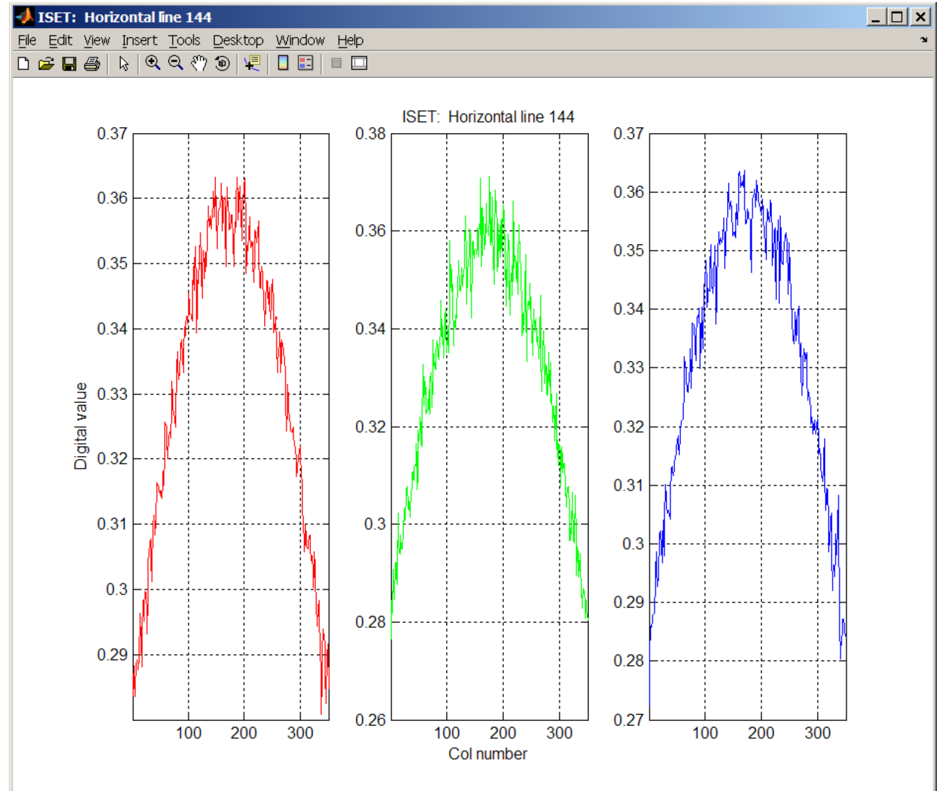
Relative illumination simulation (ISET)



Relative illumination is visible

Rendered image





Exposure algorithms

- Exposure value
- Impact of exposure on appearance

Exposure algorithms

Set the f-number (F) and exposure time (T) so that the image pixels accumulate a charge that reaches, but does not exceed, the capacity of the pixels (sensor range).

Exposure Value

$$EV = \log_2\left(\frac{F^2}{T}\right) = 2\log_2(F) - \log(T)$$

F : f-number

T : exposure time

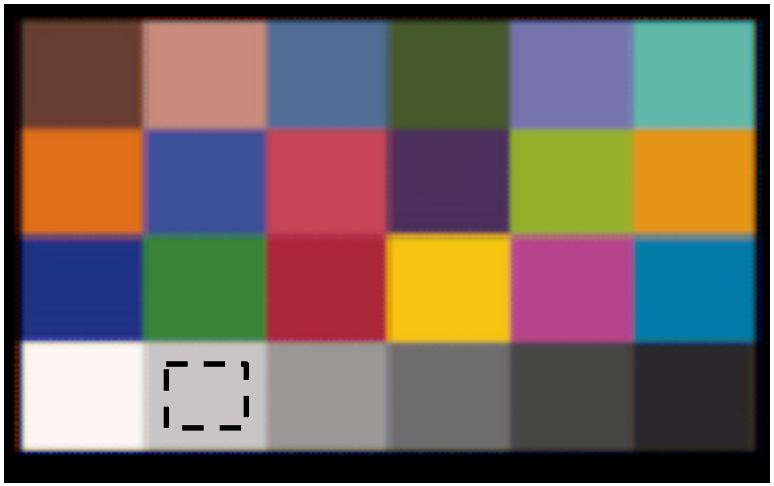
- EV accounts both for F-number and integration time (T)
- EV is zero for F-number 1 and integration time 1 sec
- EV becomes smaller as
  - Exposure duration increases
  - Aperture increases

Exposure algorithms use various image statistics

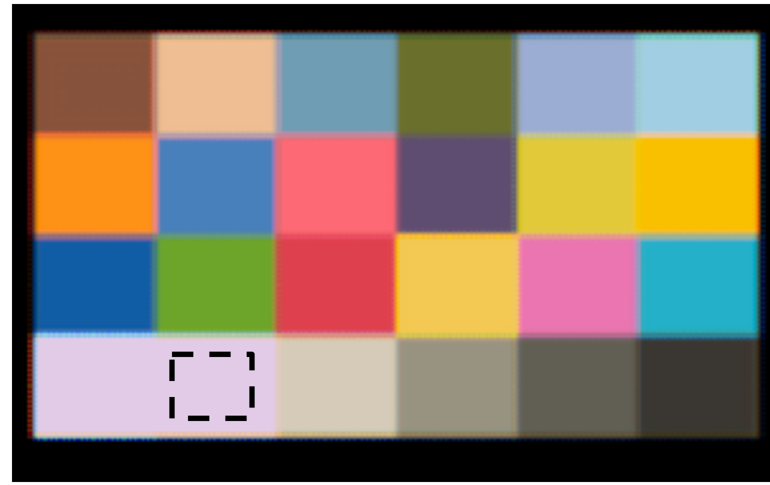
- *Mean:* B_{pre} is the mean image brightness
- ***Center-weighted Mean:*** B_{pre} is the weighted image brightness (more weight given to the image center)
- *Spot:* B_{pre} is the mean of the center (3%) area
- *Median:* B_{pre} is the median image brightness
- *Green:* B_{pre} is the mean of the green channel only

Poor exposure can produce incorrect color

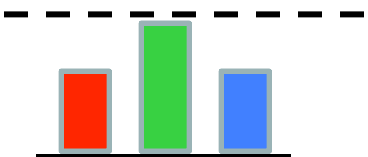
Correct exposure



Long exposure (too long)

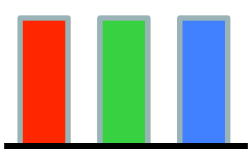


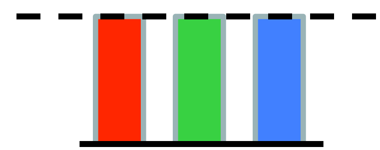
Raw data



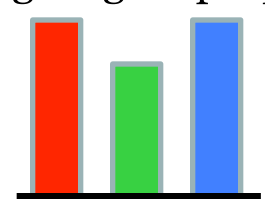
Scaled (1.2, 1.0, 1.2) to appear neutral

Processed data





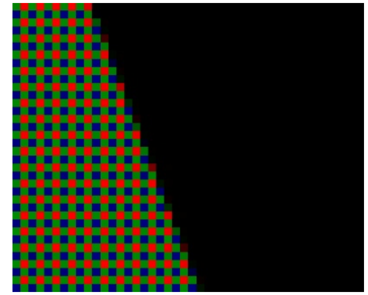
Scaling makes the saturated image region purple



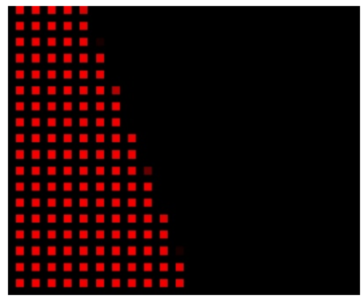
Demosaicking and CFAs

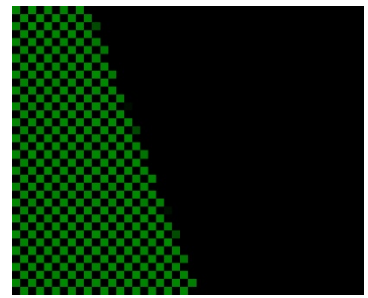
Demosaicking: The problem

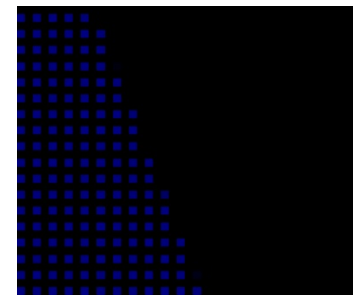
Sensor Array has Red, Green OR Blue
at every pixel



Leave measurements in place

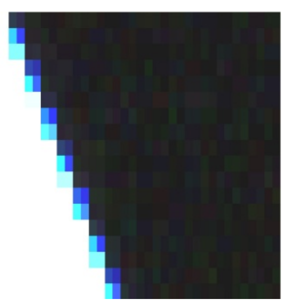




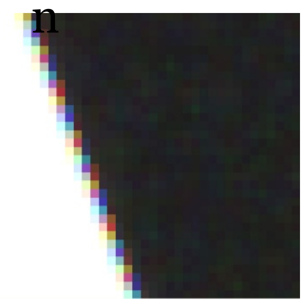


Interpolate the missing
Red, Green, and
Blue Pixel Values

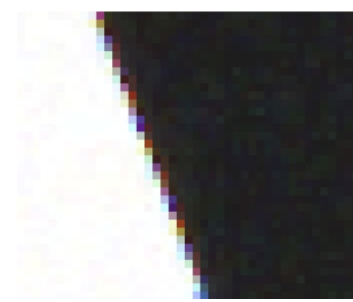
Nearest
Neighbor



Bilinear
Interpolatio
n



Adaptive
Laplacian



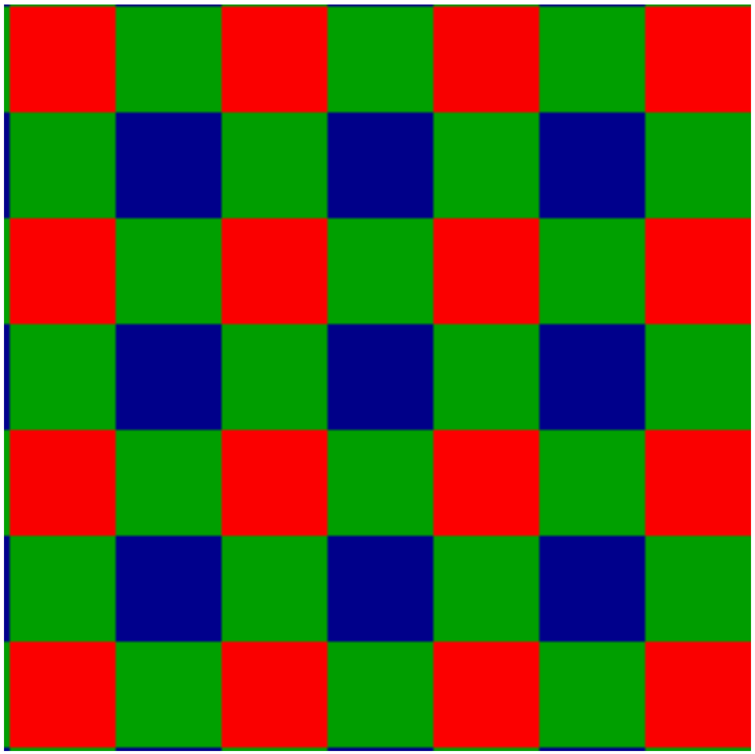
Create an image that has a
Red, Green AND Blue
at every pixel

Demosaicking depends on the mosaic

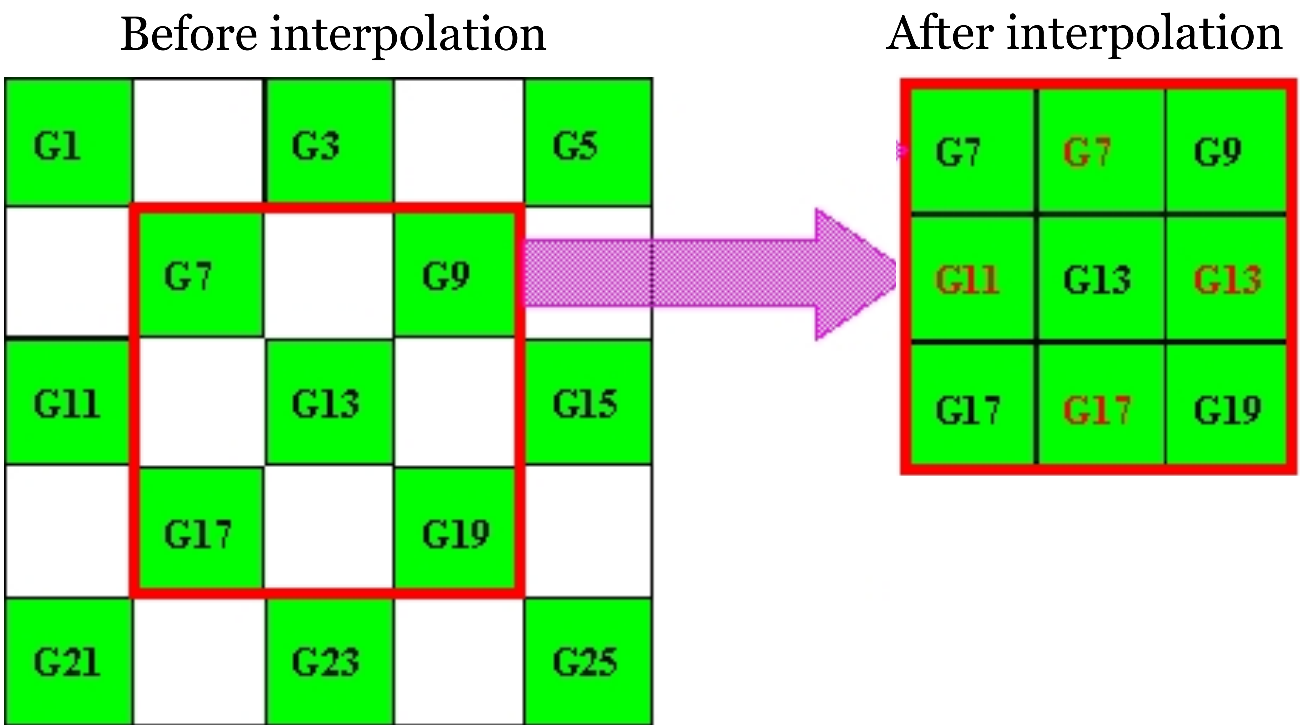
The Bayer mosaic is most common

•G is present at twice the spatial resolution of the R or B

•G is associated with luminance; G is always fully interpolated



Demosaicking: Nearest Neighbor

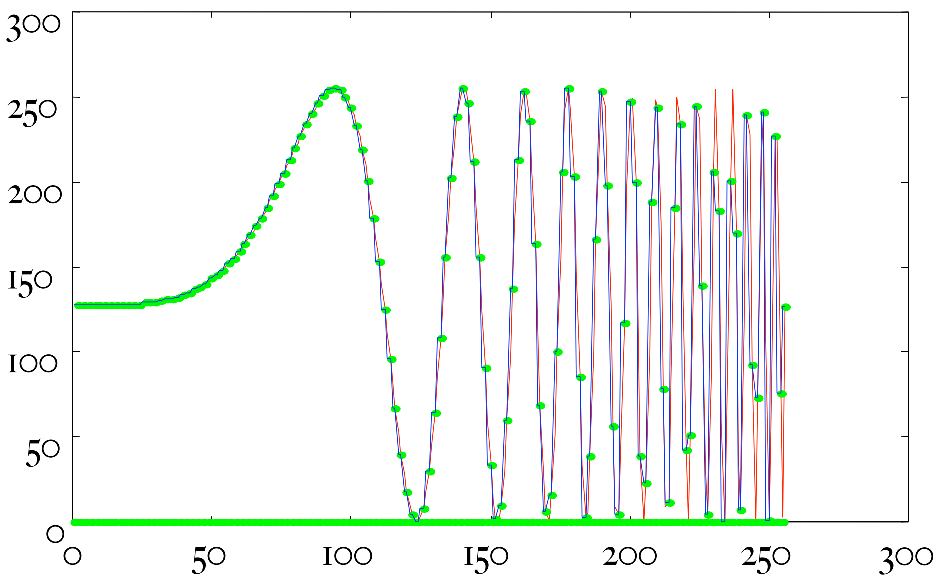


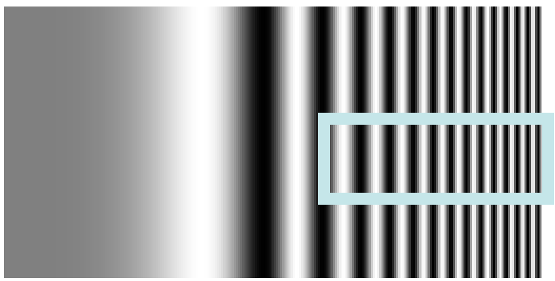
Similar – only more – for R and B channels

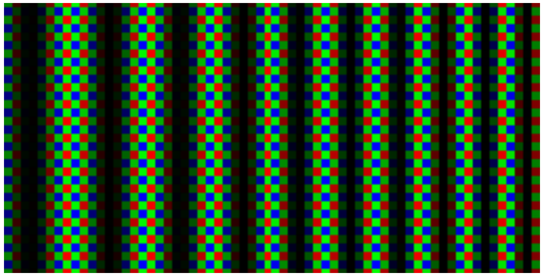
Nearest Neighbor - artifacts

Chromatic fringing on monochrome test patterns is a good visual test for demosaicking

Very significant fringing for nearest neighbor

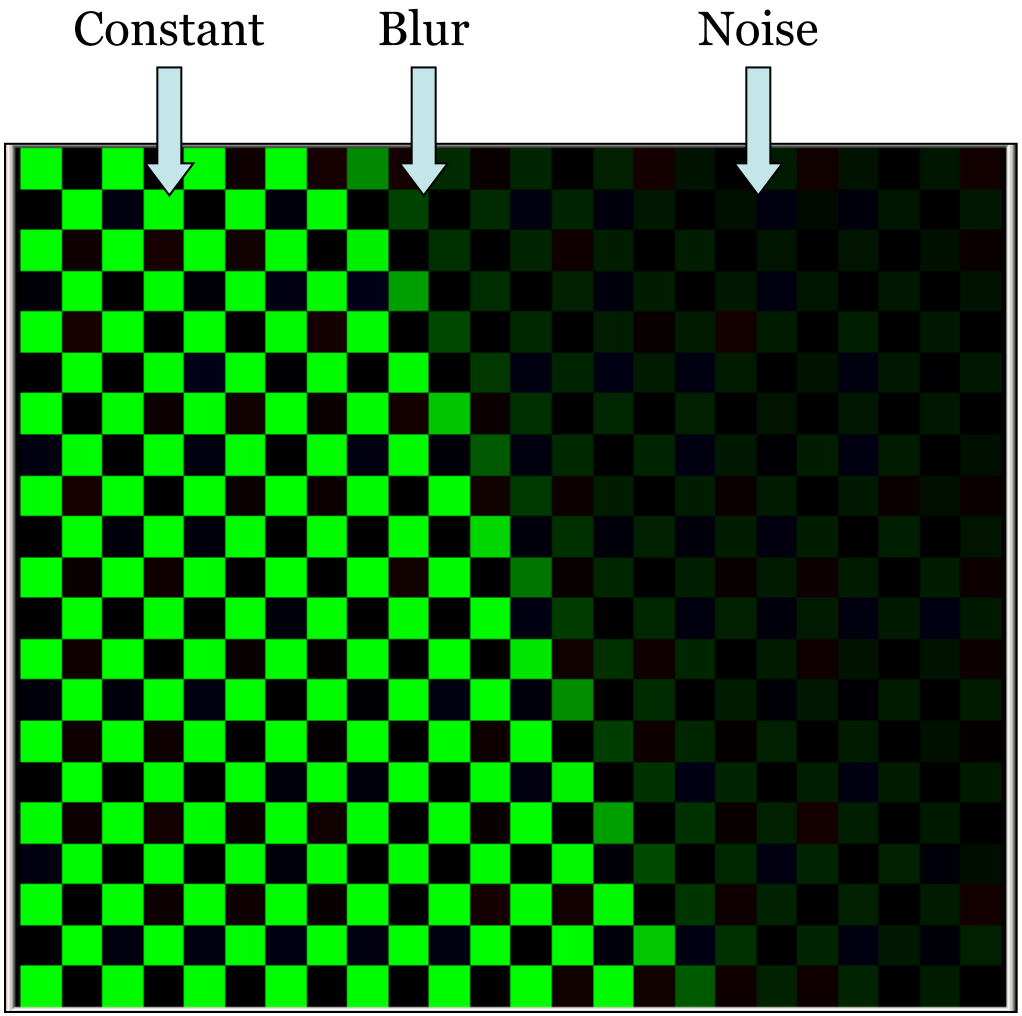






Bilinear interpolation
(within channel)

We use nearby measured G to interpolate the missing G; similarly for R and B

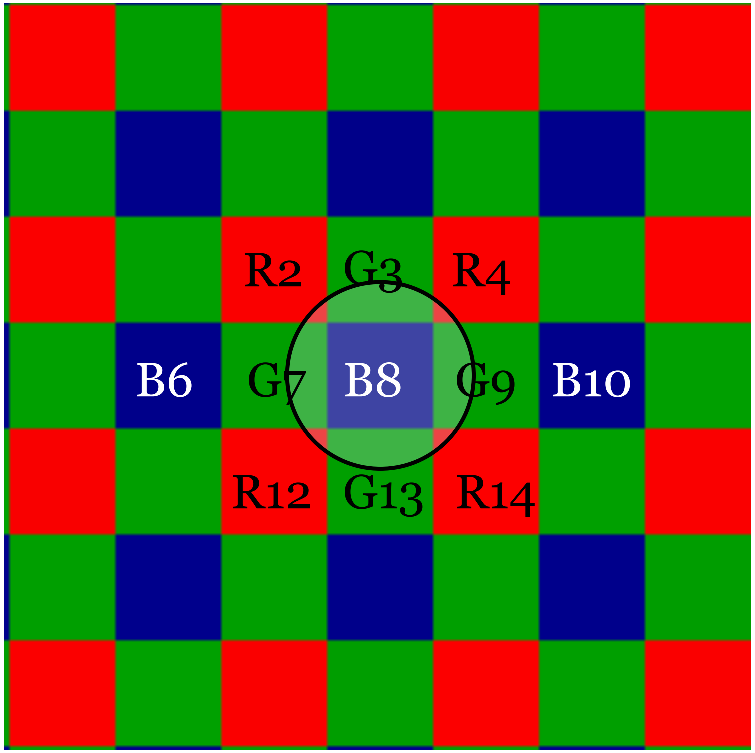


Bilinear interpolation (within channel)

Existing values are left untouched. The average of adjacent green pixel.

For example:

$$G8 = (G3+G7+G9+G13) / 4$$

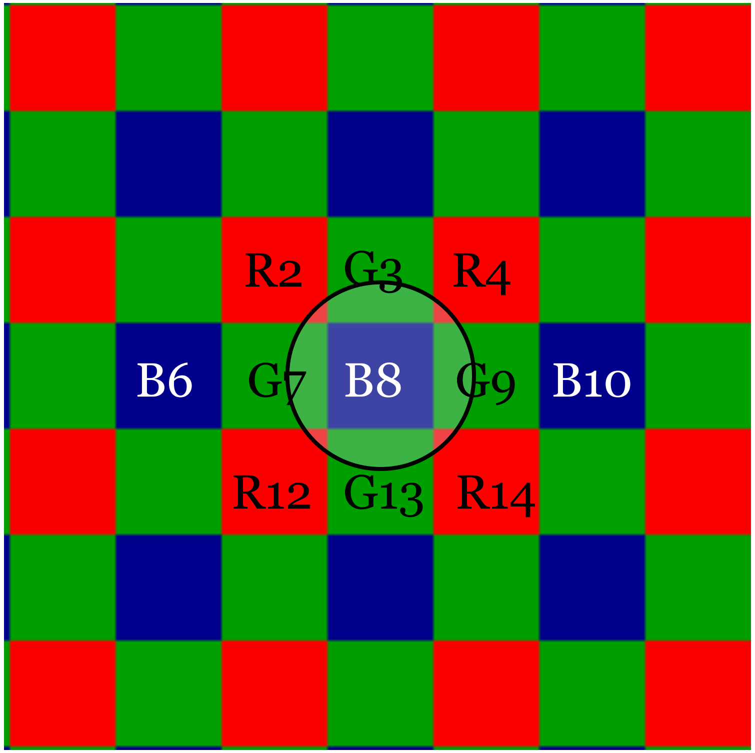


Bilinear interpolation (across channel)

Across channel uses weighted sum from other colors, in particular those at the same location – these are usually very correlated.

For example:

$$G8 = \alpha B8 + \beta \frac{G3 + G7 + G9 + G13}{4} + (1 - \alpha - \beta)\frac{R2 + R4 + R12 + R14}{4}$$



Chromatic demosaicing

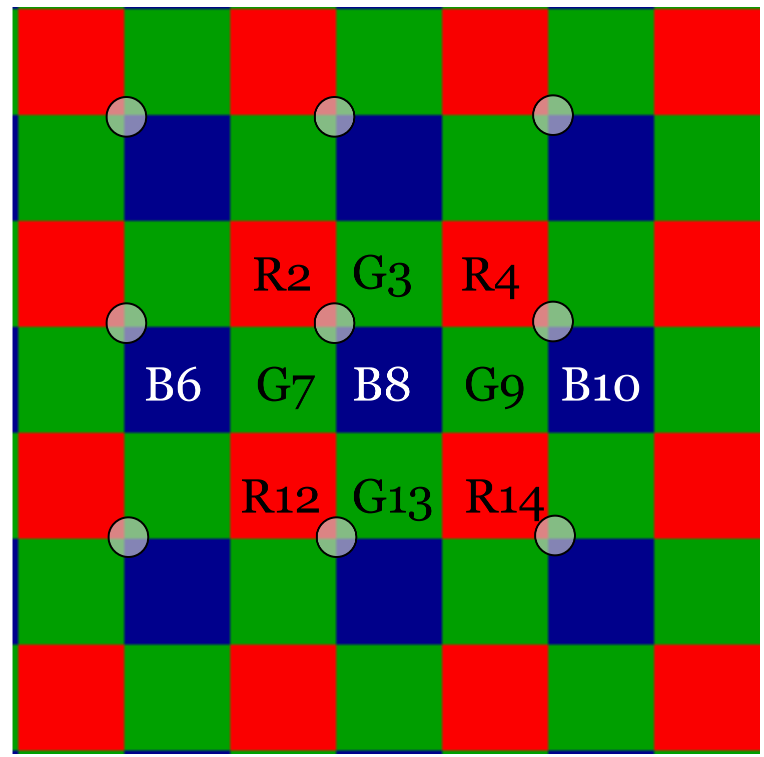
The RGB values are converted to two opponent-colors chromatic values (chrominance plane).

The chrominance plane has 1/4th the luminance resolution. This is consistent with common JPEG practice – it is OK for the reasons we discussed about human vision.

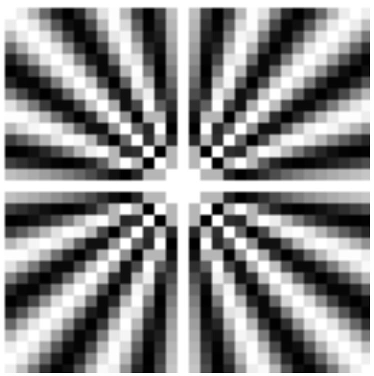
$$C_R = R - G$$
$$C_B = R - B$$

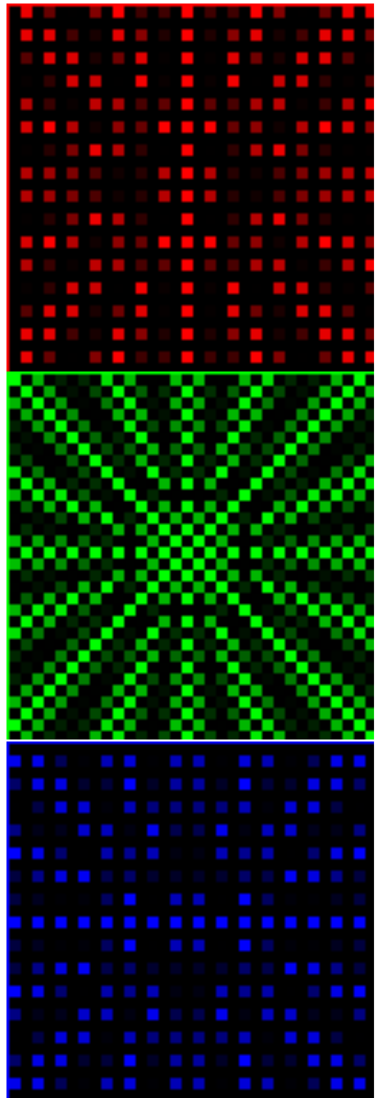
YCrCb

RGB Mosaic



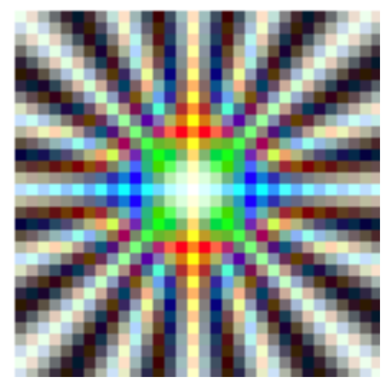
Demosaicking comparison
Nearest neighbor vs. bilinear





Nearest neighbor

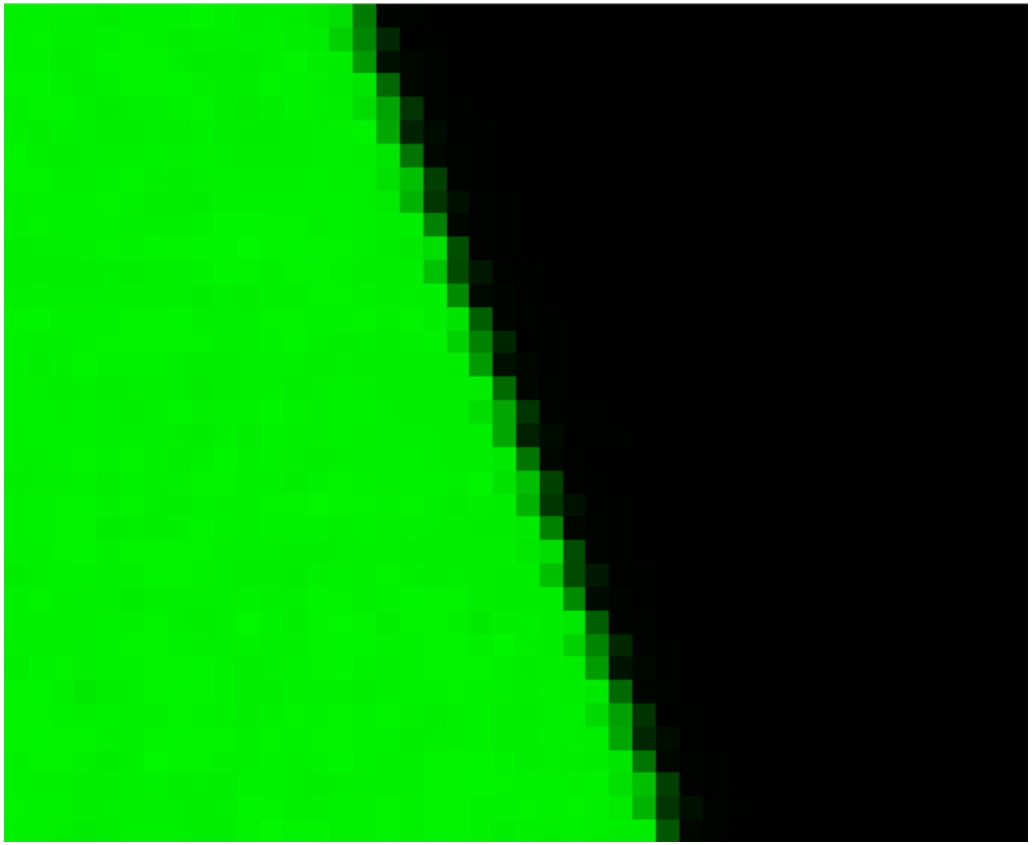
Bilinear



Analysis: bilinear interpolation

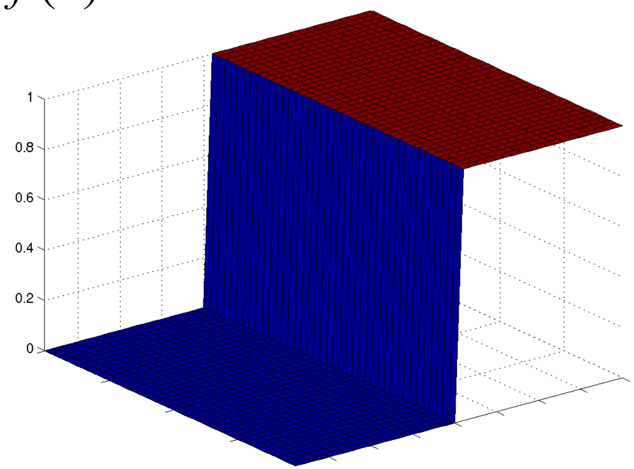
Most modern algorithms

- Combine information from color channels to make a decision
- Go beyond linear interpolation and try to preserve edges

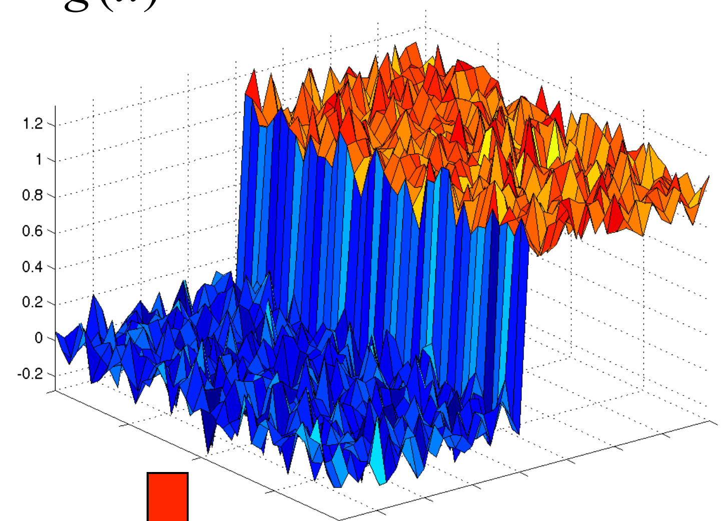


Linear filtering – 2D Gaussian kernel

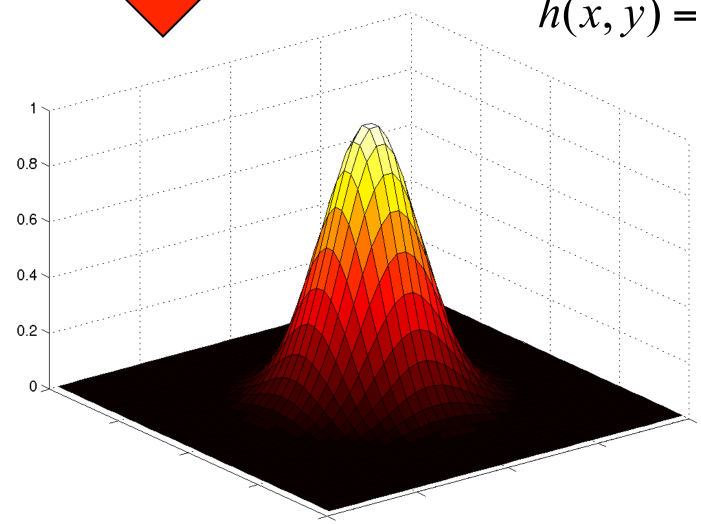
$f(x)$



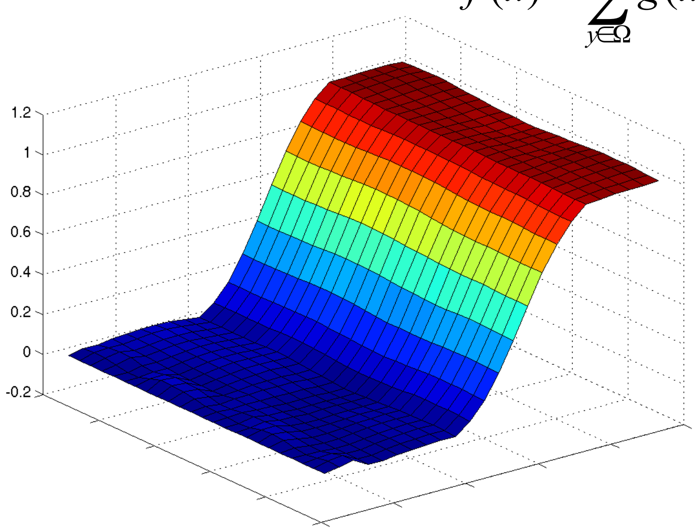
$g(x)$



$$h(x,y) = e^{\frac{(x-y)^2}{2\sigma^2}}$$

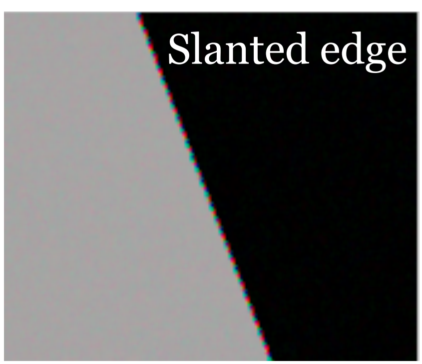


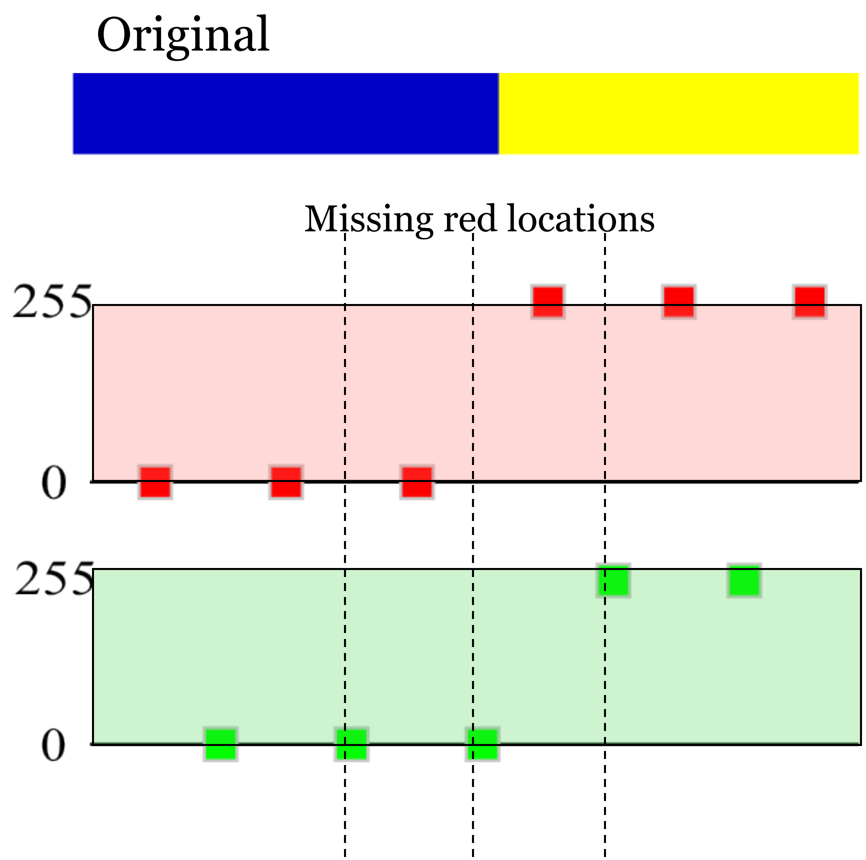
$$\hat{f}(x) = \sum_{y \in \Omega} g(x)h(x,y)$$



Bilinear color interpolation artifacts
(within-channel interpolation example)

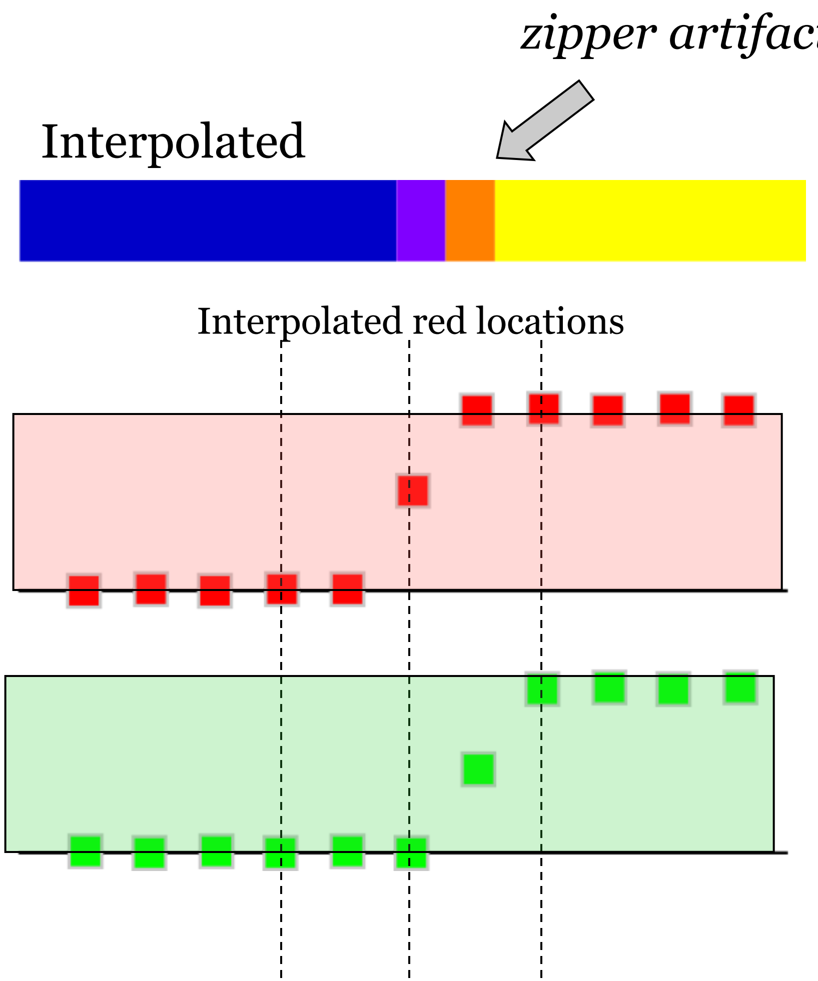
Mis-registration of the color channels, coupled with within channel interpolation, introduces unwanted color artifacts in the rendered display

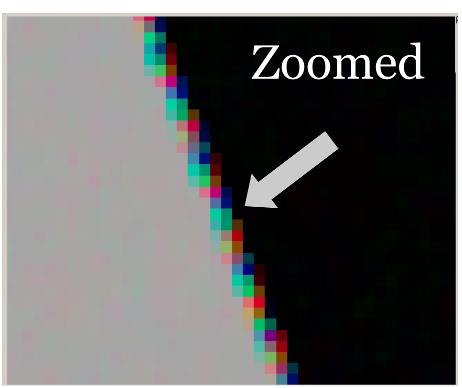




Color interpolation artifacts
(within-channel interpolation example)

Mis-registration of the color channels, coupled with within channel interpolation, introduces unwanted color artifacts in the rendered display





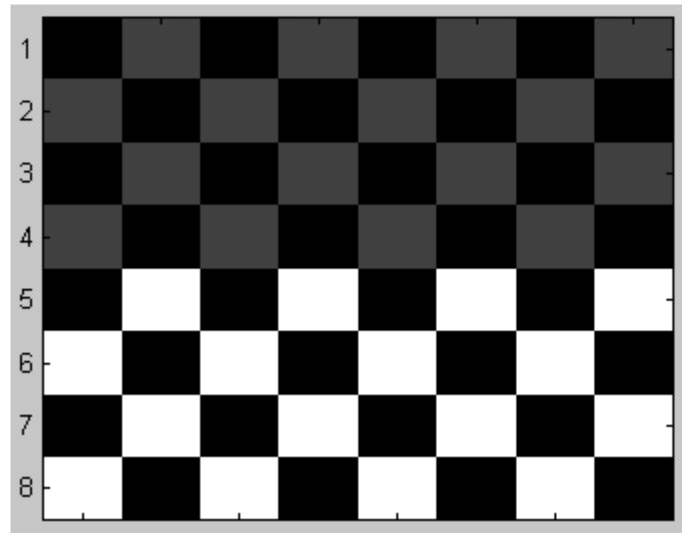
Across channel adaptive demosaicking

R G R

- - · - - · - - · - - · - - · - - · - - · -

G B G

R G R



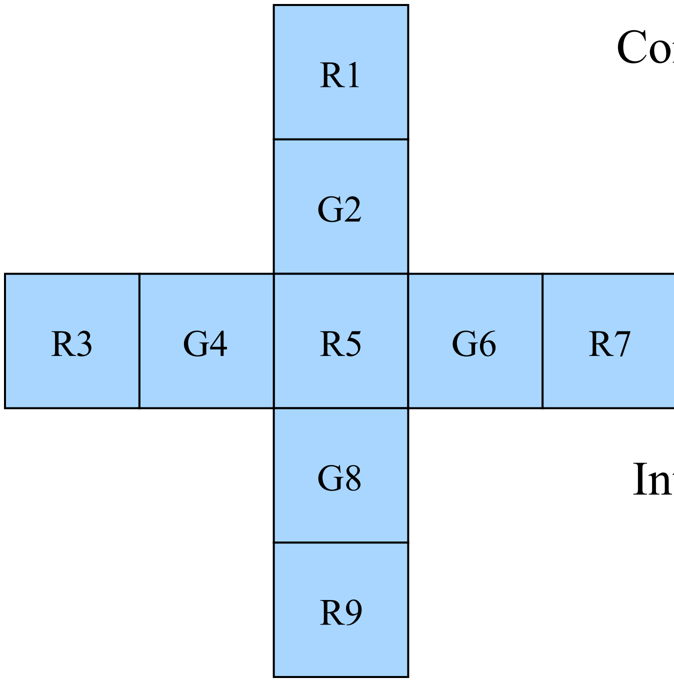
Goal: Don’t average across edges.

Method: Before averaging, check for outliers.

Average a G value if it is similar to others

Otherwise, exclude it from weighted sum

Adaptive Laplacian algorithm
(Hamilton and Adams, Kodak)



Control parameters $\alpha = |R3 - 2R5 + R7| + |G6 - G4|$

$$\beta = |R1 - 2R5 + R9| + |G8 - G2|$$

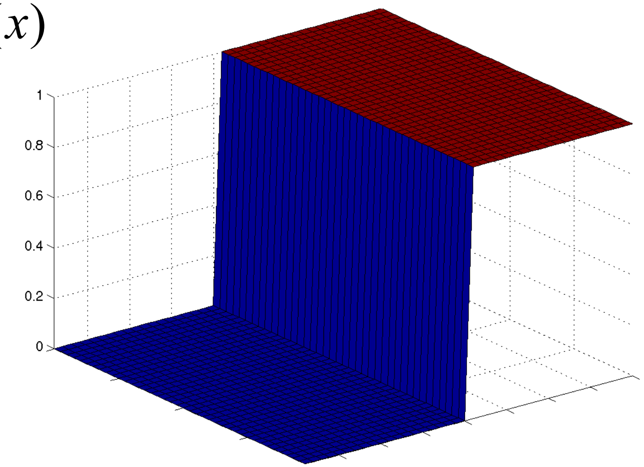
1st and 2nd derivatives

Interpolation rule

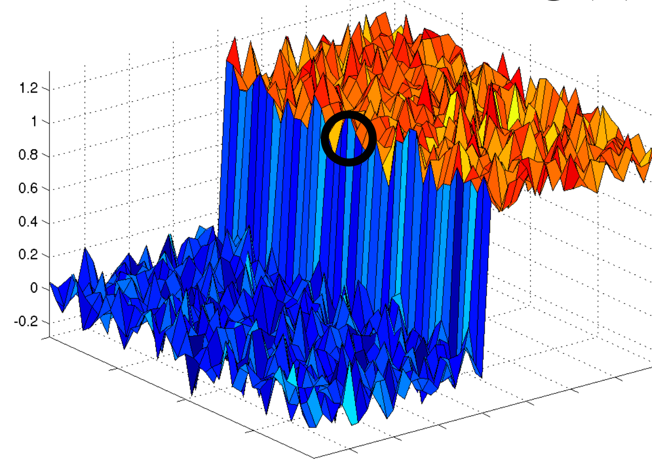
$$\mathrm{G5} = \begin{cases} \dfrac{G4 + G6}{2} - \dfrac{R3 - 2R5 + R7}{2}, & \mathit{if}\ \alpha < \beta \\ \dfrac{G2 + G8}{2} - \dfrac{R1 - 2R5 + R9}{2}, & \mathit{if}\ \alpha > \beta \\ \dfrac{G2 + G4 + G6 + G8}{4} - \dfrac{R1 + R3 - 4R5 + R7 + R9}{4}, & \mathit{if}\ \alpha = \beta \end{cases}$$

Adaptive filtering: Bilateral (Tomasi)

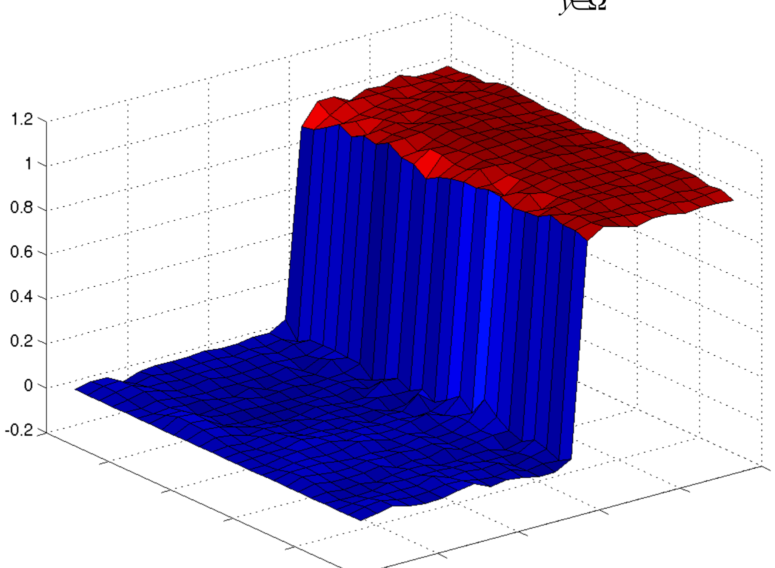
$f(x)$



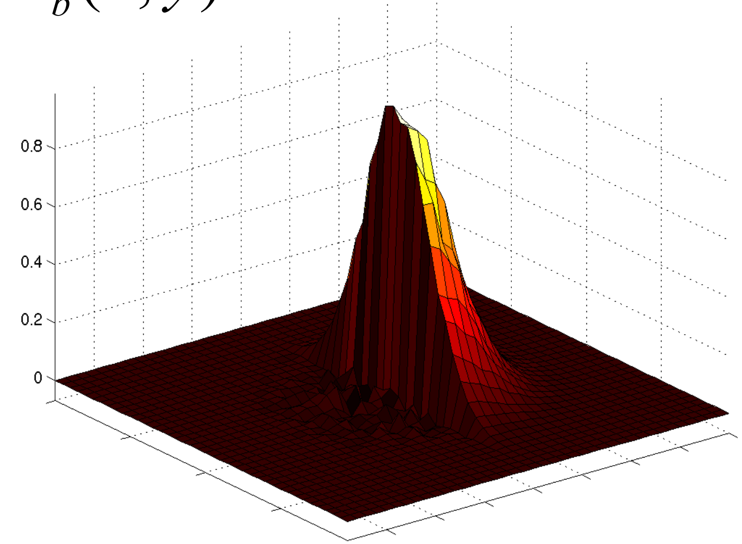
$g(x)$



$$\hat{f}(x) = \sum_{y \in \Omega} g(x) h_b(x, y)$$



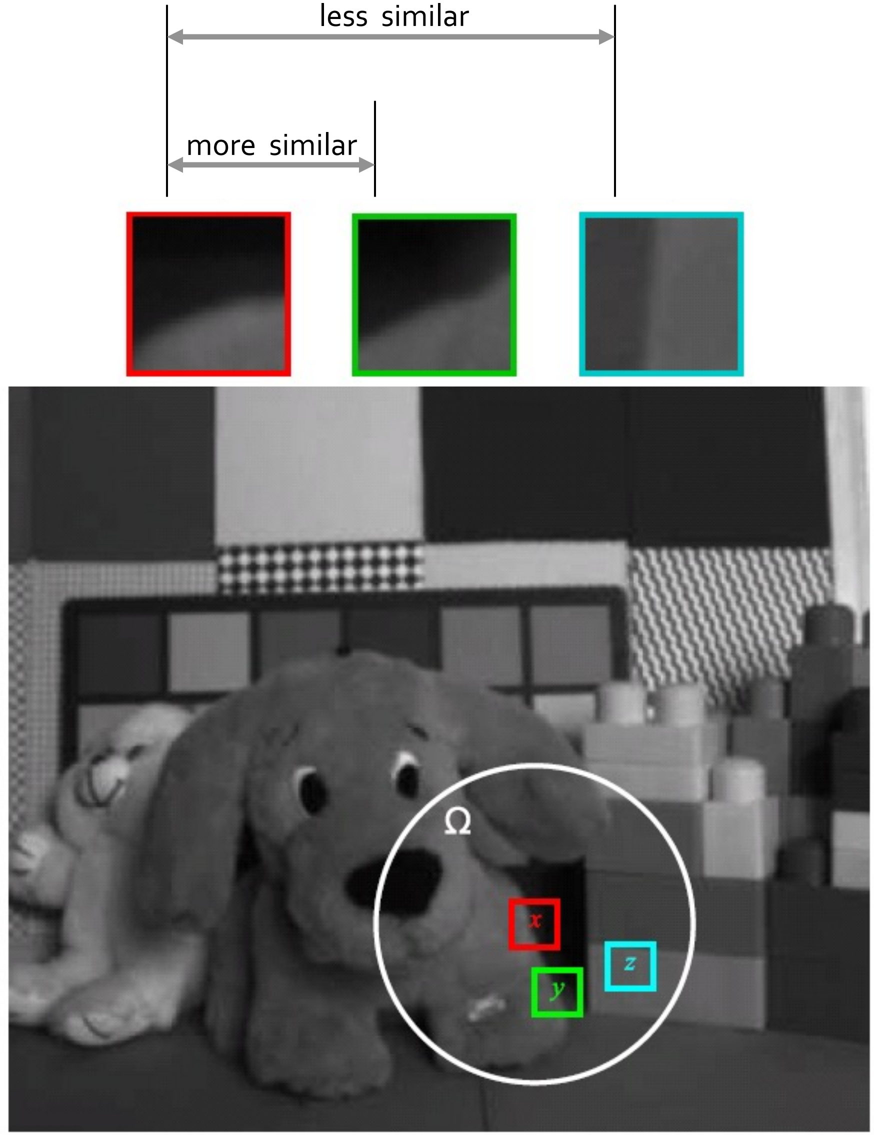
$$h_b(x, y) = e^{-\frac{(x-y)^2}{2\sigma_d^{\,2}}} e^{-\frac{(g(x)-g(y))^2}{2\sigma_r^{\,2}}}$$



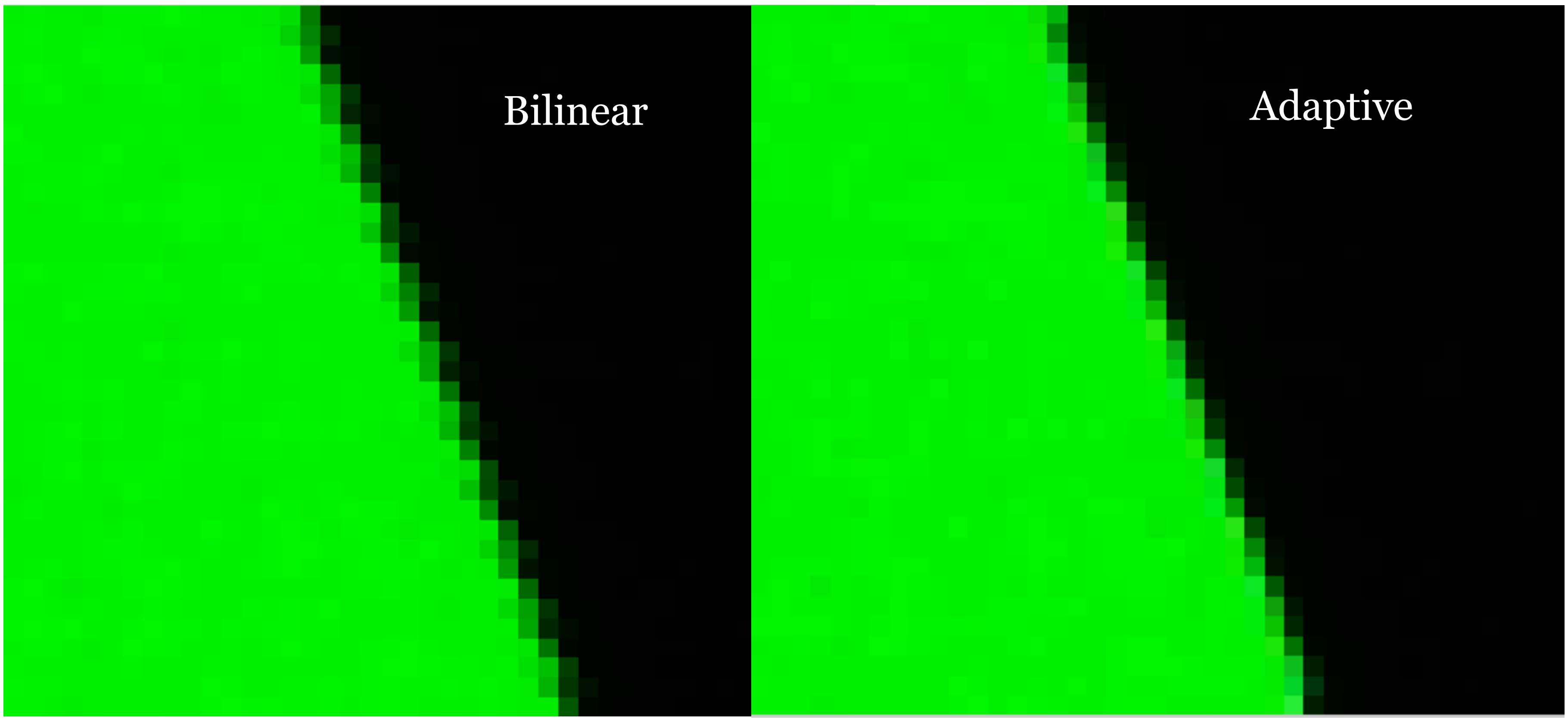
Pixel-similarity (nonlocal means)

(Buades et al., 2005)

- Each pixel is associated with a small surrounding patch
- Pixel-similarity is determined by similarity of associated patches
- The pixel-similarity between x and y is higher than the pixel-similarity between x and z
- Hence, x and y, but not x and z, are averaged



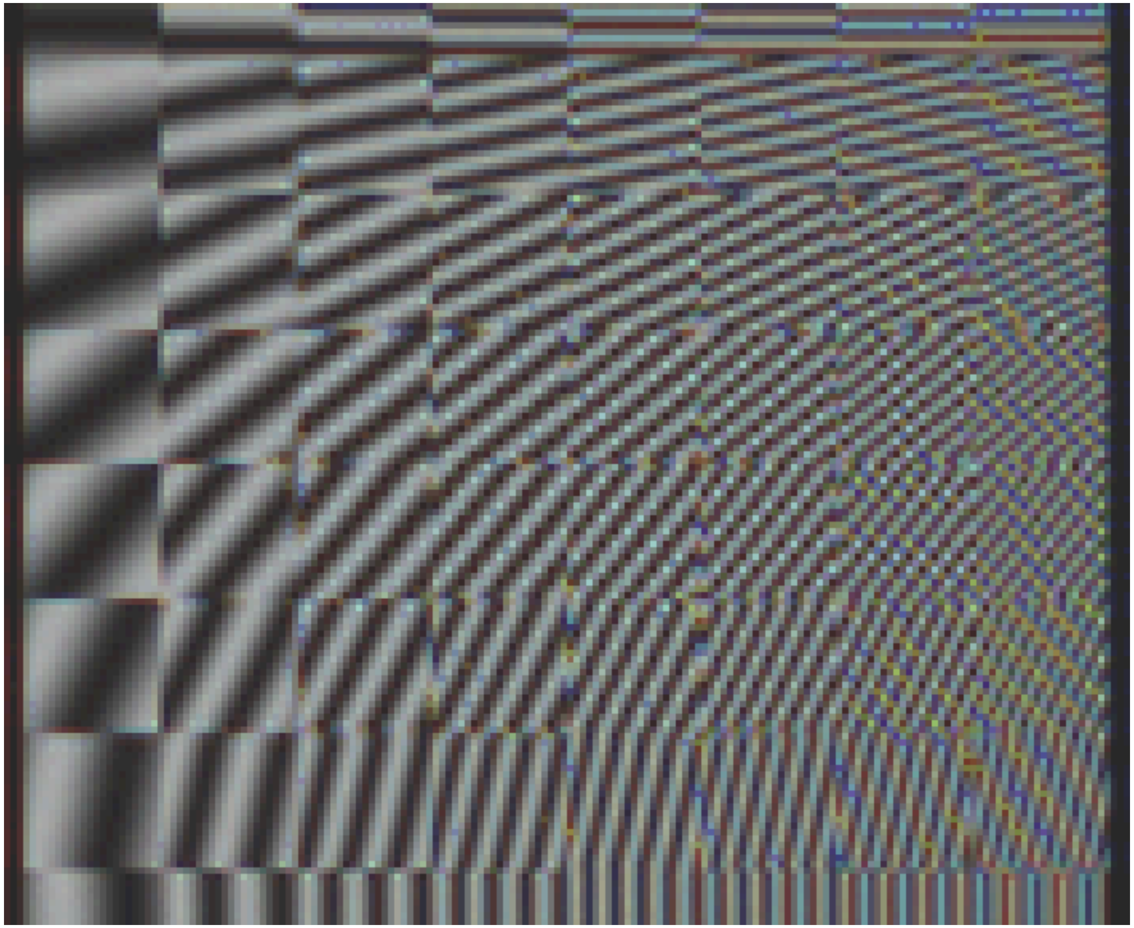
Demosaic comparisons
Bilinear and adaptive



Demosaic comparisons

Bilinear and adaptive

Adaptive Laplacian



Chromatic
artifacts

Sharper

Chromatic
artifacts

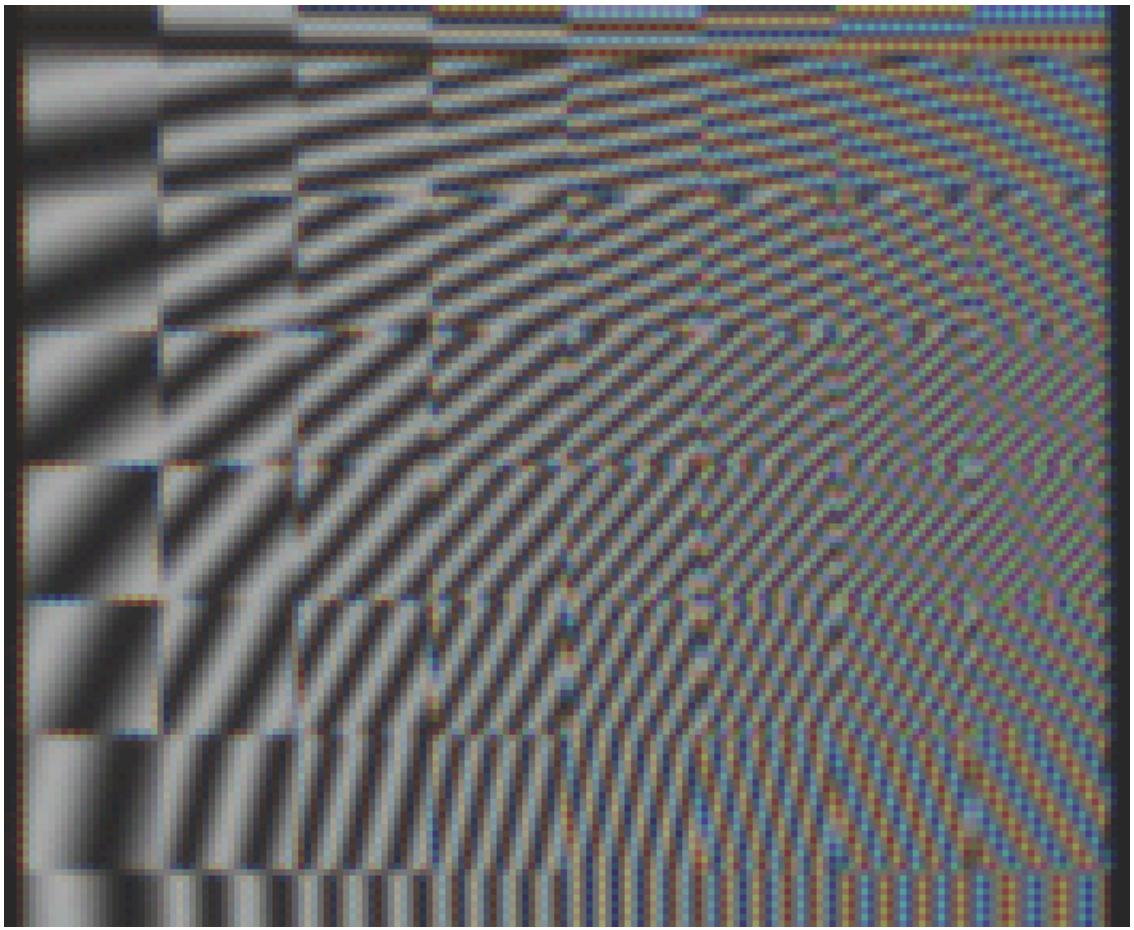
Demosaic comparisons

Bilinear and adaptive

Bilinear

Chromatic artifacts

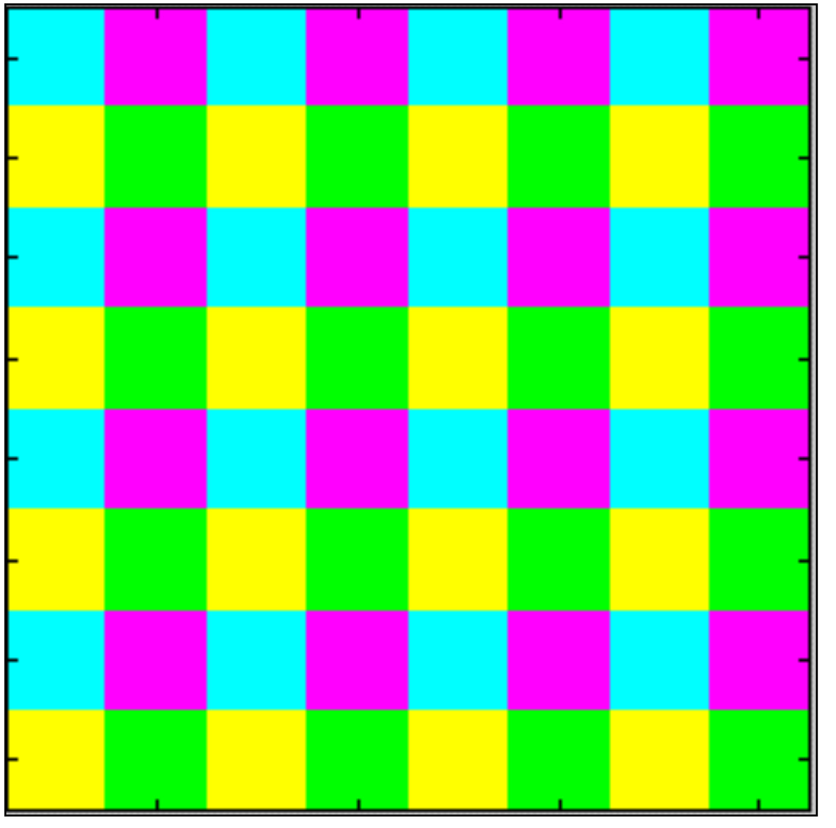
Blurred



Chromatic artifacts

Camera CFA CMYG example

- CMYG – video cameras
- Fast demosaicking
- Broad filters, more photons
- Poor noise properties

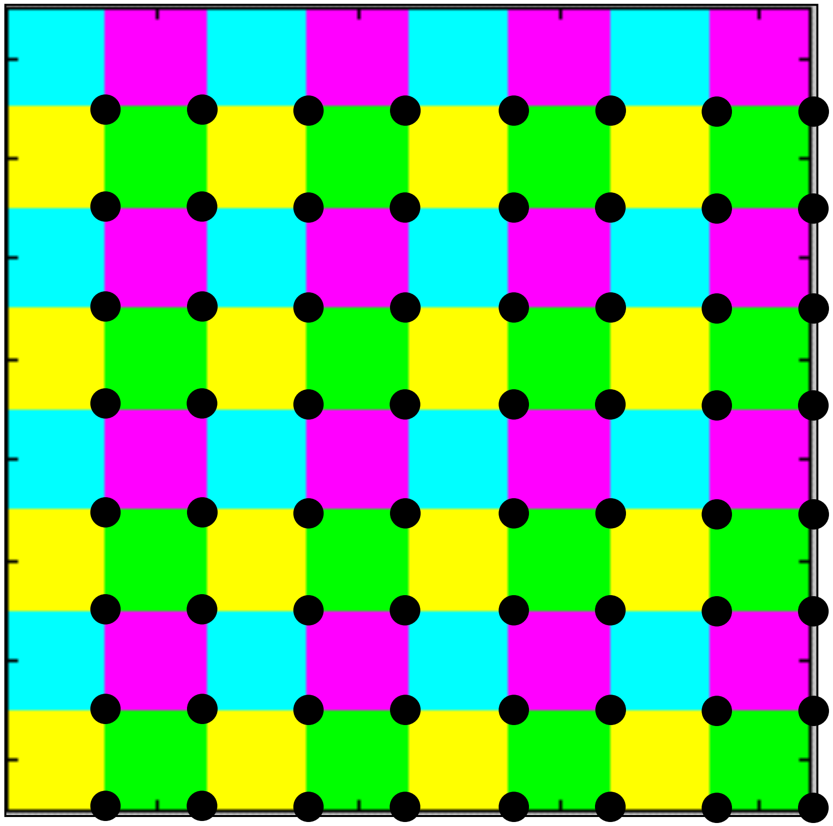


Nice features of the CMYG mosaic

• The sum of the neighbors at each corner estimates luminance (black dots)

C+M+Y+G

• High spatial sampling, but spatially correlated



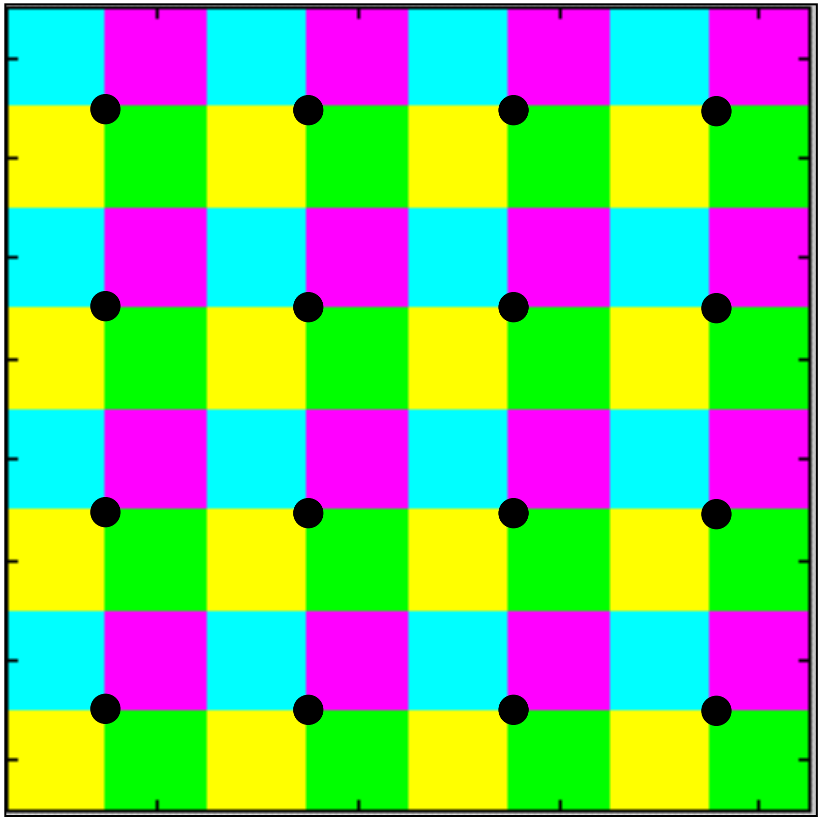
Chromatic Demosaicing CMYG

- The chrominance plane can be calculated at half the spatial resolution using these formula:

$$C_R = (M + Y) - (G+C)$$
$$C_B = (M + C) - (G+Y)$$

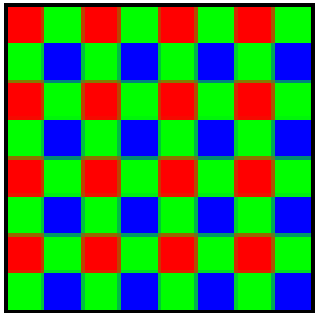
Note:

$$R = 1 - Cyan$$
$$G = 1 - Magenta$$
$$B = 1 - Yellow$$

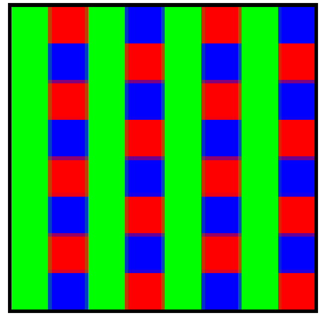


New algorithms for new CFAs

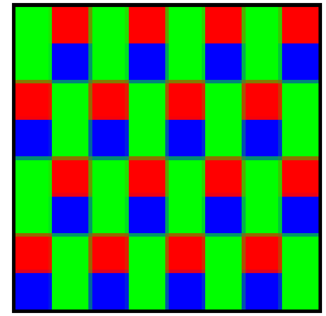
Most widely used



Bayer

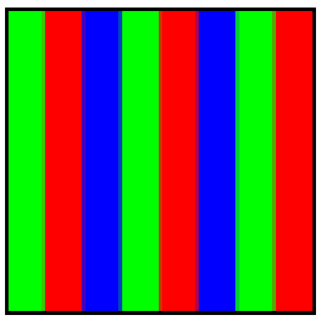


Yamanaka

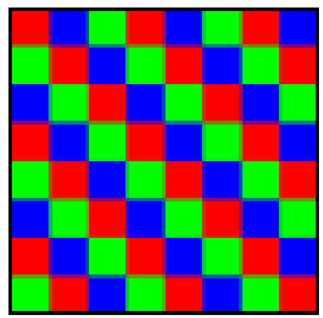


Lukac

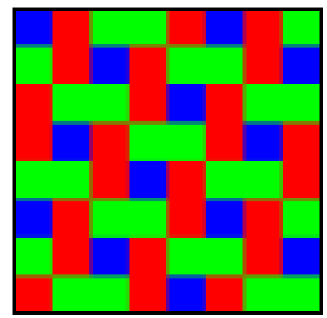
L3



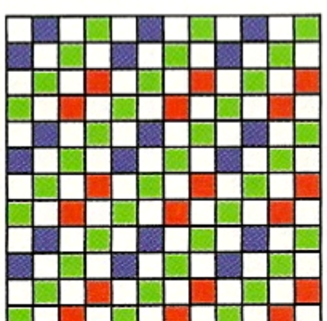
Striped

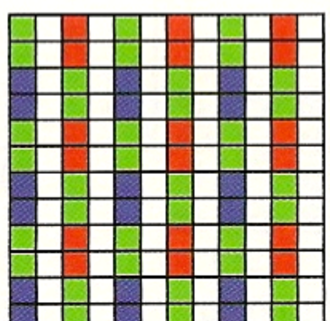


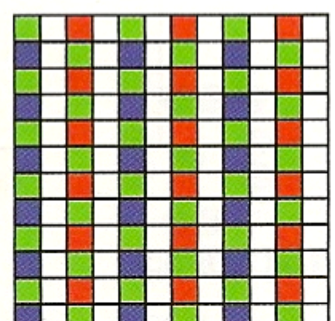
Diagonal striped



Holladay







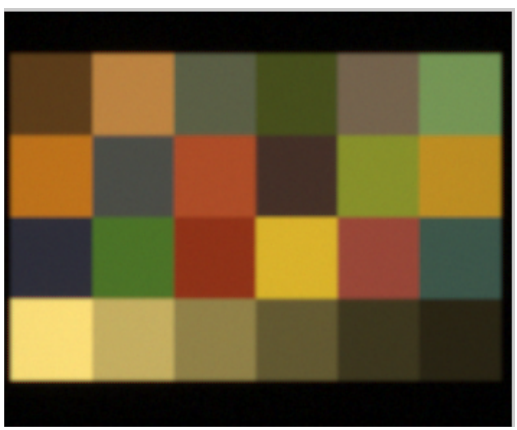
Color management

Color management: The problem

Copying sensor values directly to display graphics card or printer driver is a bad idea

- Devices (cameras and displays) differ; need calibration
- Illuminant change influences camera more than it influences the human

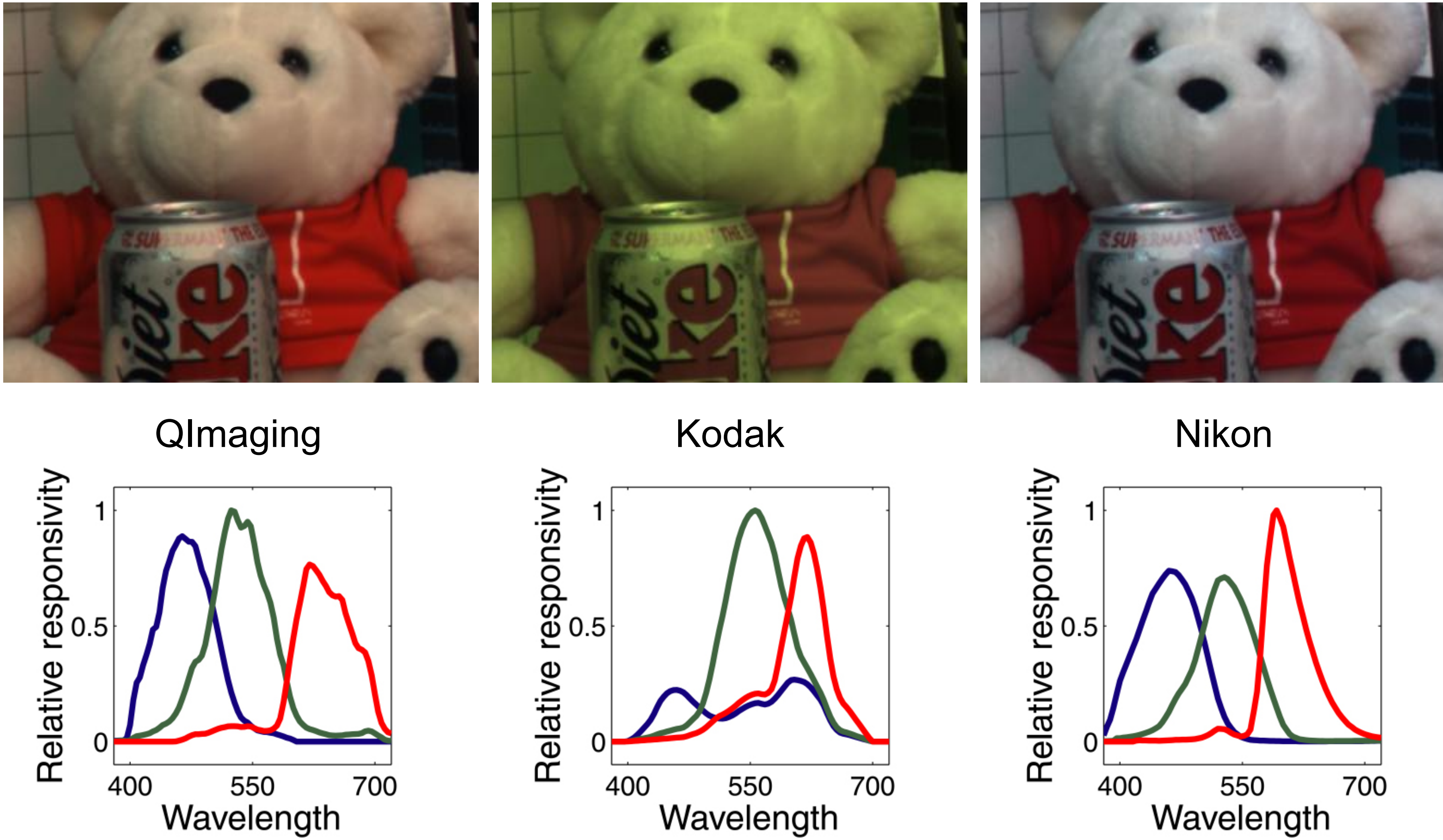
Fluorescent



Tungsten

D65

Camera sensors differ between one another



Color management

- Informal terms describe color processing
  - Terms are used in different ways
  - I won't try to explain them all

- My preference is to use
  - Sensor correction
  - Illuminant correction

White balance

Color conversion

Color balance

Color correction

Color rendering

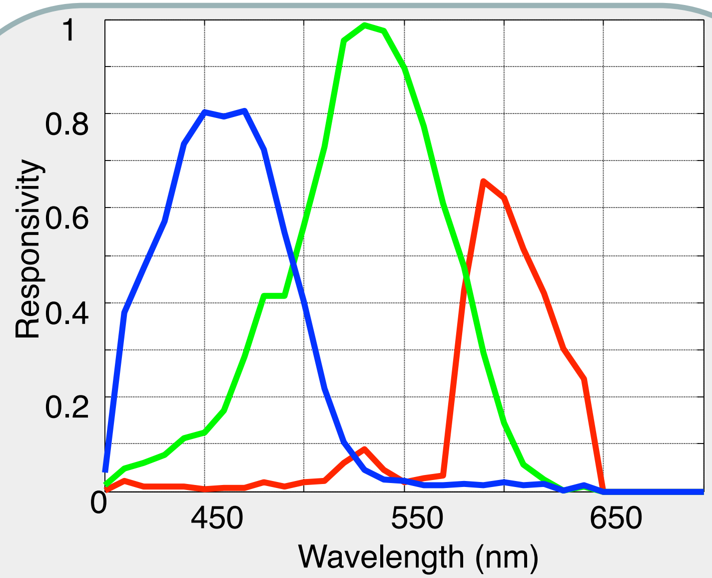
Illuminant transformation

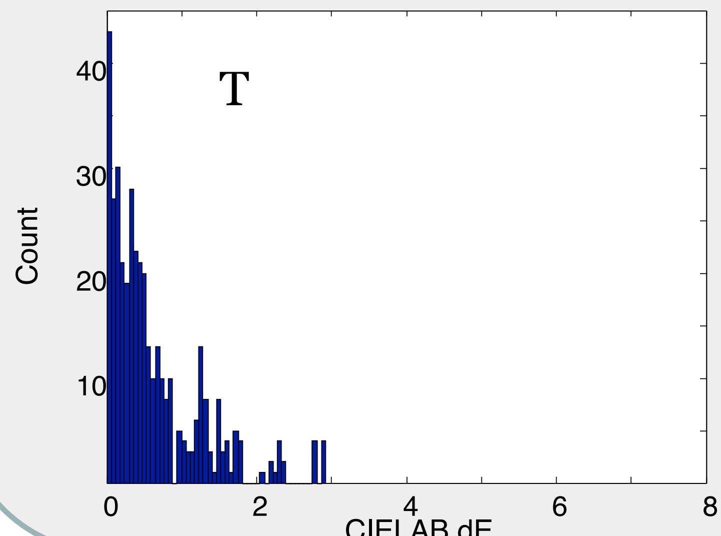
Color constancy

Sensor correction
(to CIE XYZ)

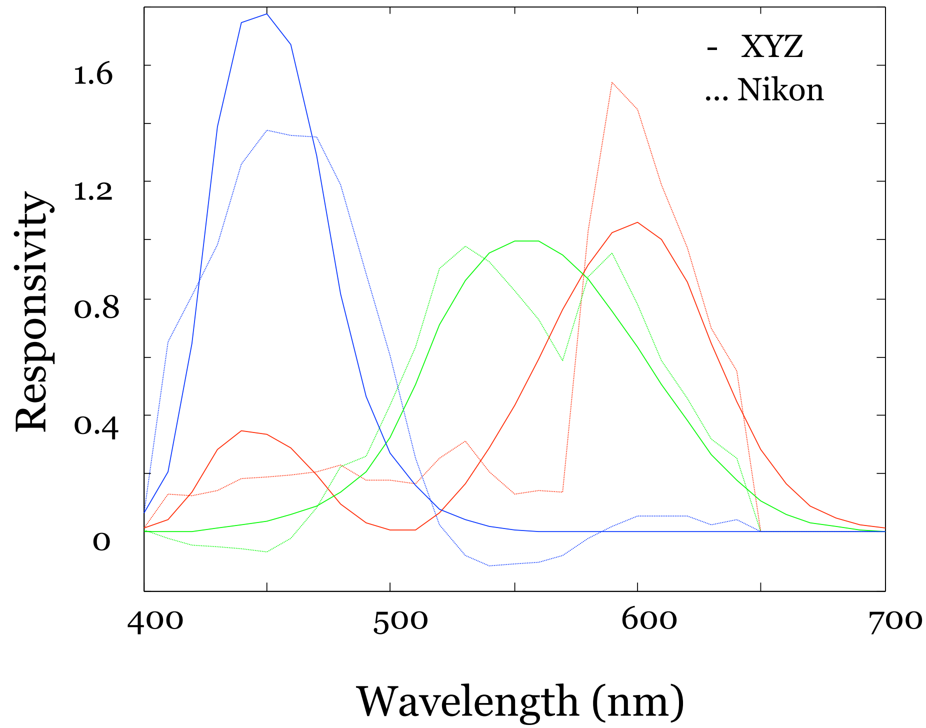
ISET: s_sensorCorrection

$$xyz = T * sensor$$





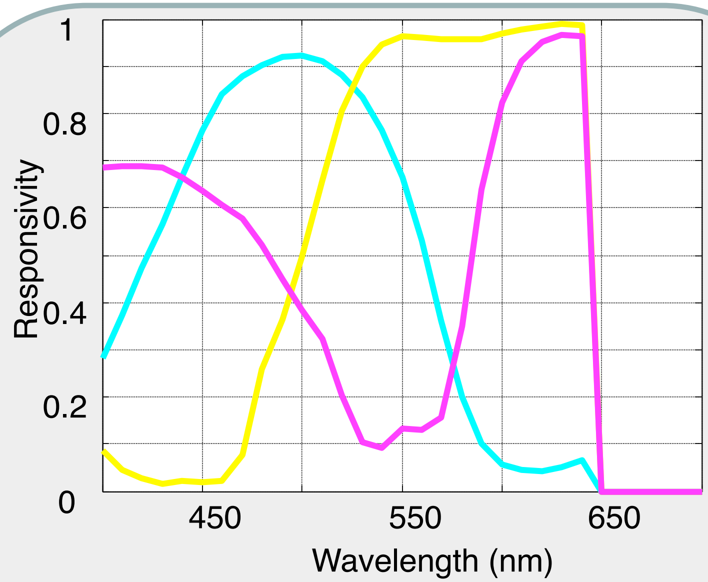
$$T = \begin{bmatrix} 2.2591 & 0.1328 & 0.1867 \\ 1.0159 & 0.9371 & -0.2491 \\ 0.0156 & -0.1201 & 1.7118 \end{bmatrix}$$

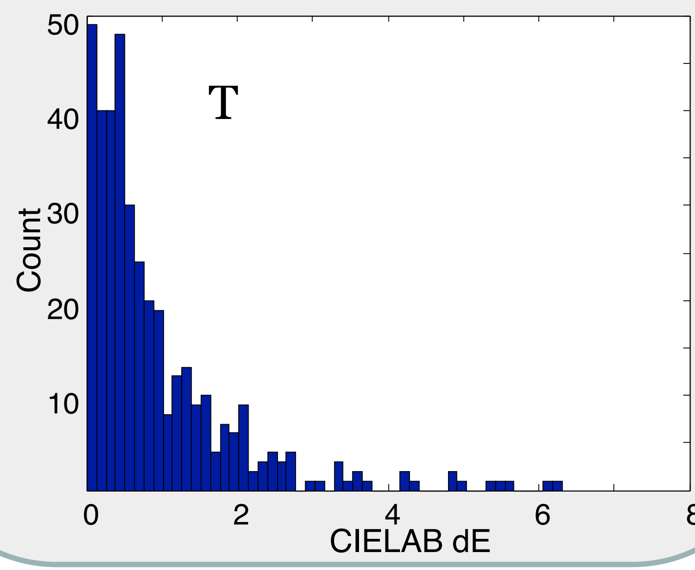


Sensor correction
(to CIE XYZ)

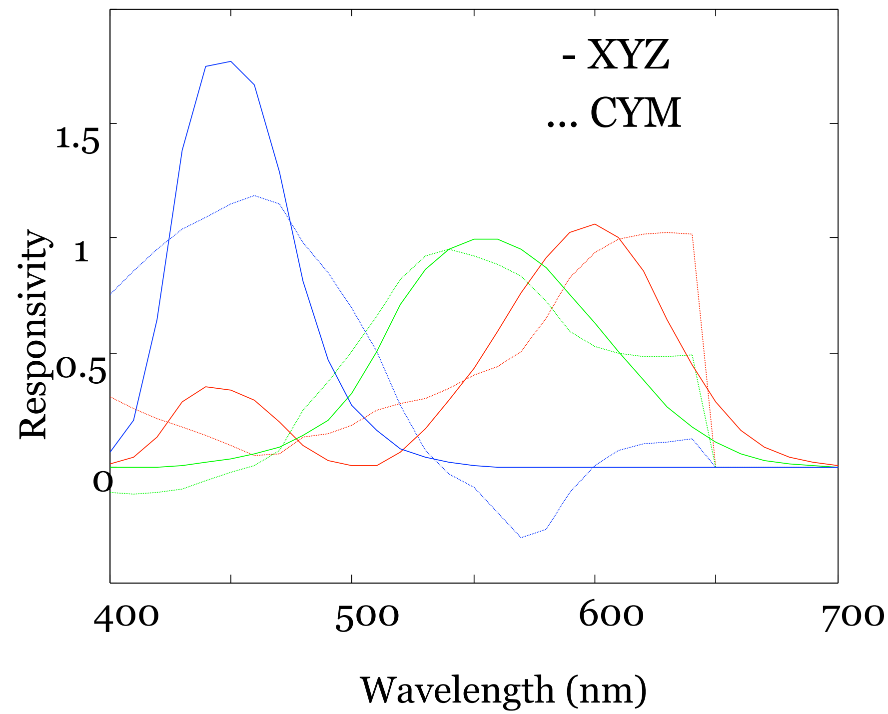
ISET: s_sensorCorrection

$$xyz = T * sensor$$



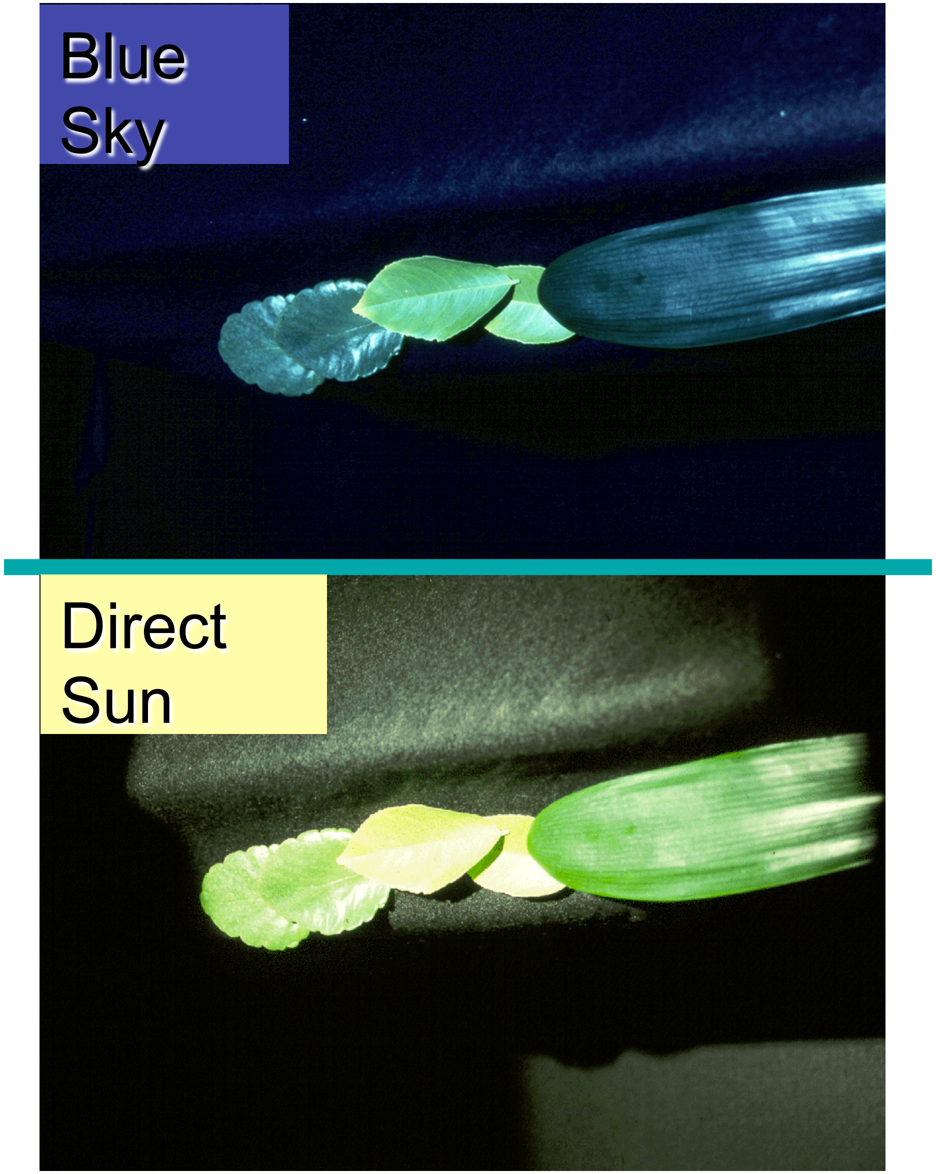


$$\begin{matrix} -0.2970 & 0.5812 & 0.4605 \\ 0.2581 & 0.8520 & -0.3963 \\ 0.8259 & -0.7436 & 0.8038 \end{matrix}$$



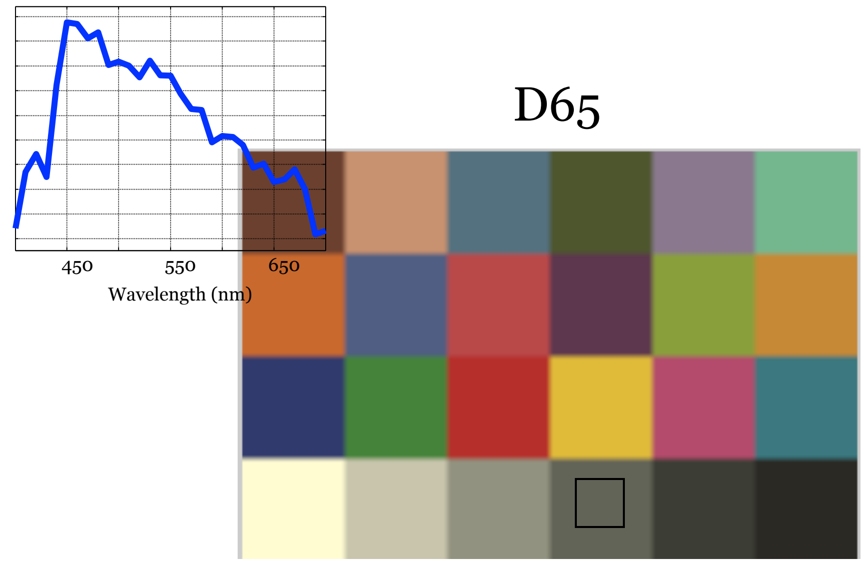
Illuminant correction

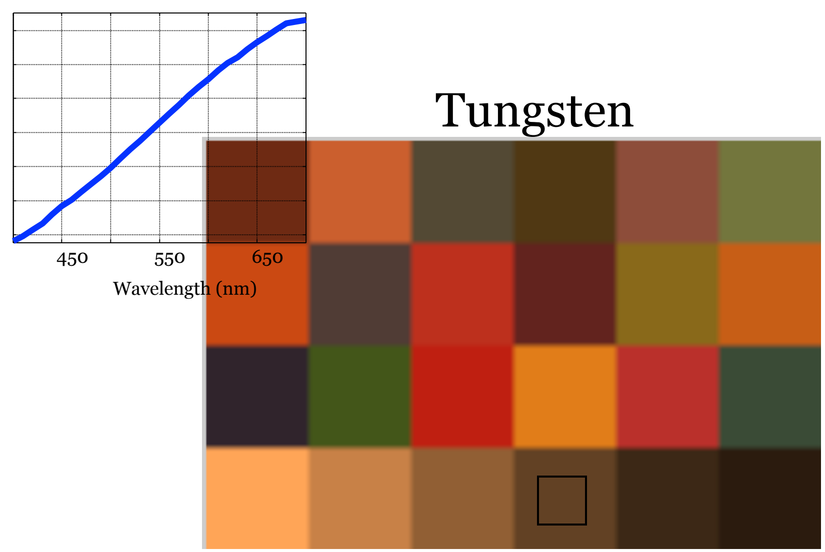
- Illuminant changes cause the scattered light to change
- The visual system adjusts to the illuminant (color constancy)
- If the pipeline doesn't adjust, the rendering has the wrong color balance
- Illuminant correction is a very important aspect of color balancing



Illuminant correction

- Illuminant changes cause the scattered light to change
- The visual system adjusts to the illuminant (color constancy)
- If the pipeline doesn't adjust, the color appears wrong
- Illuminant correction is important



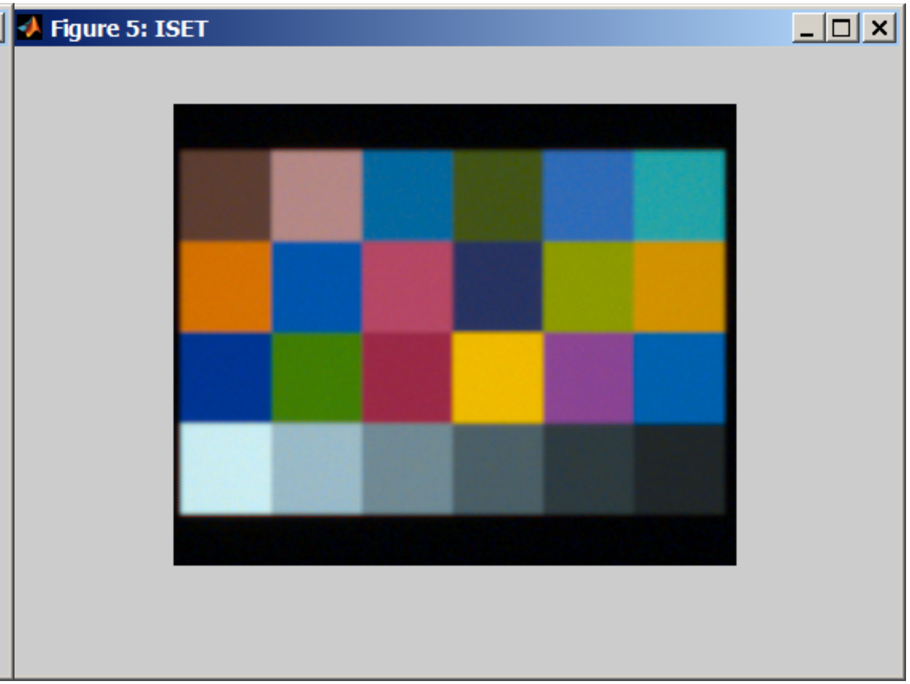


Color management: Illuminant

Color management: Example

Copied fluorescent data

Corrected fluorescent data



Color management computations

- Linear transformations (global)
- Look-up tables (locally linear)
- Statistical characterizations of the image
- Knowledge of the output device

Matrix based color adjustment methods

Device-dependent diagonal

$$\text{Display} \quad \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} s_r & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s_b \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad \text{Camera}$$

Choose s_r and s_b, so that a neutral (gray) surface in the image is rendered as a neutral display output

You must make an educated guess about the camera RGB to a neutral surface. Example ideas:

- The average of the image is neutral
- The brightest elements of the image average to neutral
- Your idea goes here ....

Matrix based color adjustment methods

Display

Device-independent diagonal

Camera

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} & & \\ & M & \\ & & \end{pmatrix} \begin{pmatrix} s_r & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s_b \end{pmatrix} \begin{pmatrix} & & \\ & L & \\ & & \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Transform the device data into a calibrated space (cones, XYZ), (**L**).

Perform the diagonal transformation. again choosing s_r and s_b, so that a neutral (gray) surface in the image is rendered as a neutral signal in that space

Convert to the display representation from the calibrated space to display space (**M**)

Matrix based color adjustment methods

Device-independent linear

$$\underset{\text{Display}}{\begin{pmatrix} R_1^{'} & & R_N^{'} \\ G_1^{'} & \mathrm{L} & G_N^{'} \\ B_1^{'} & & B_N^{'} \end{pmatrix}} = \begin{pmatrix} & & \\ & C_E & \\ & & \end{pmatrix} \underset{\text{Camera}}{\begin{pmatrix} R_1 & & R_N \\ G_1 & \mathrm{L} & G_N \\ B_1 & & B_N \end{pmatrix}}$$

For each light condition, E, choose a matrix, C_E, that transforms multiple measurements to desirable display values.

Find these matrices for 30-50 likely lights.

The matrix C_E may be chosen by MSE or perceptual error minimization.

Matrix based color adjustment methods

Device-independent locally linear

Display

Camera

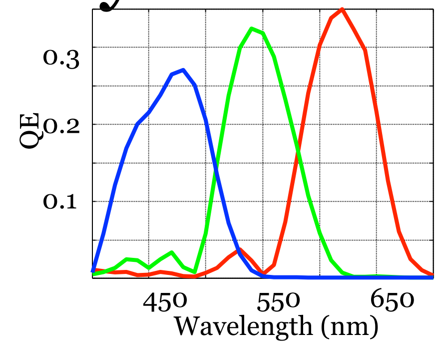
$$\begin{pmatrix} R_1' & \mathrm{L} & R_N' \\ G_1' & \mathrm{L} & G_N' \\ B_1' & & B_N' \end{pmatrix} = \boxed{\begin{matrix} & \mathbf{E} \\ C_1 & \\ C_2 & \\ C_3 & \\ C_4 & \\ \ldots & \\ C_N & \end{matrix}} \begin{pmatrix} R_1 & \mathrm{L} & R_N \\ G_1 & \mathrm{L} & G_N \\ B_1 & & B_N \end{pmatrix}$$

Create lookup tables for each light condition, E, that map surfaces into a desired display for that light.

Such tables, which are essentially a collection of local linear transformations; they may be chosen by MSE or perceptual error minimization.

Illuminant correction summary

Sensor



Transform sensor data to calibrated color (XYZ)

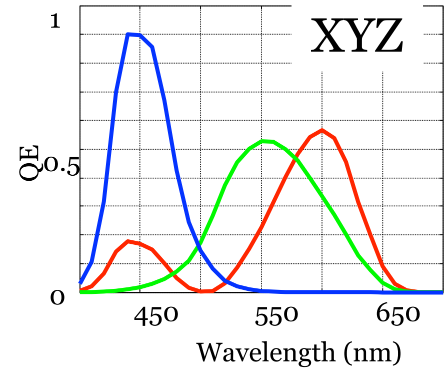
L

Estimate illuminant

E

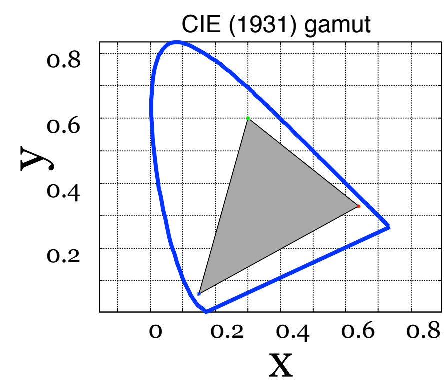
Apply transformation for illuminant E

C_E



Convert to display space, adjusting for gamma and gamut

M

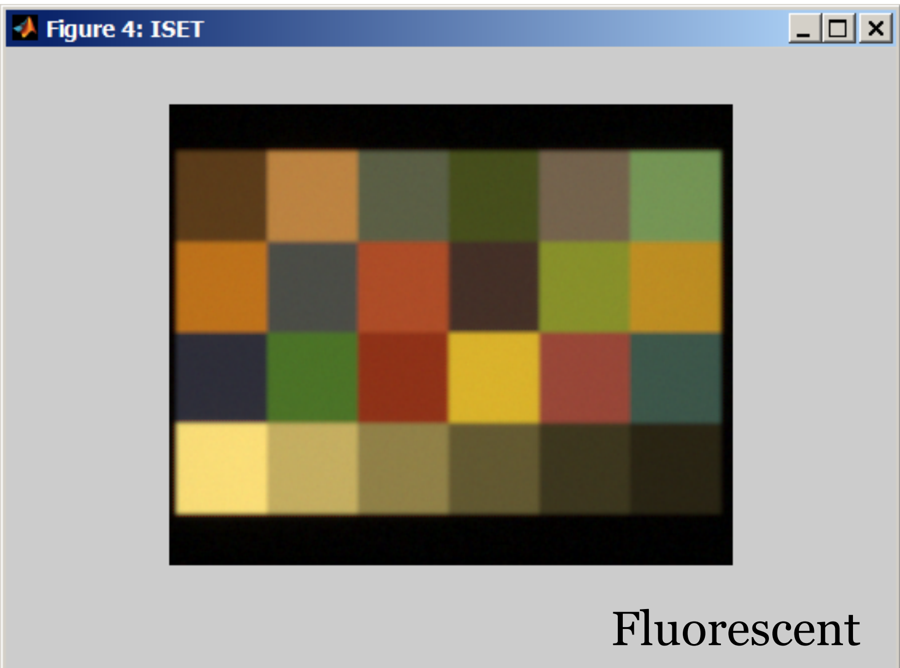


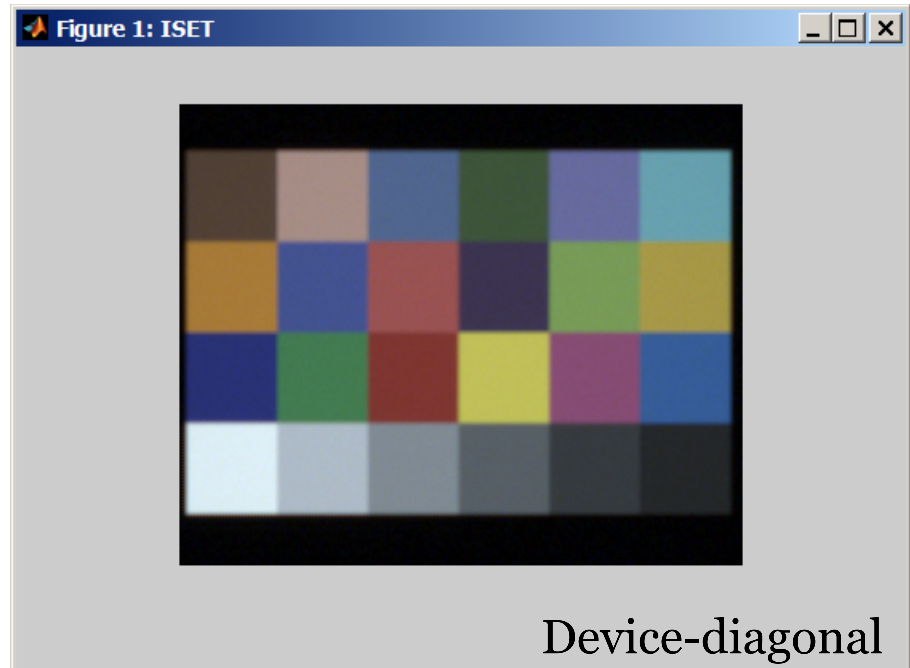
Display

Device-dependent diagonal

<u>Conversion matrix</u>

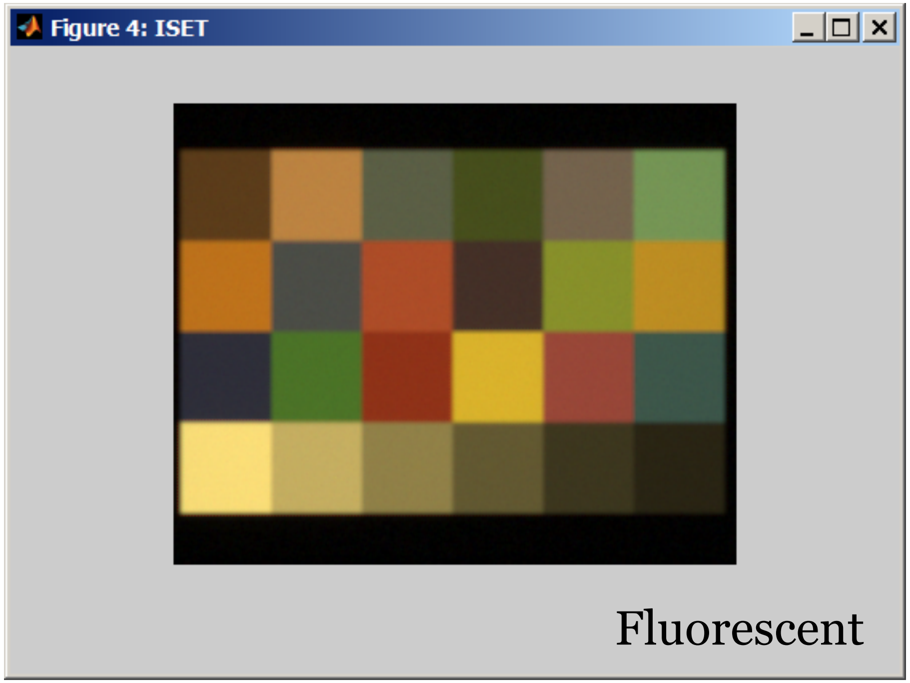
0.1123	-0.0179	0.0207
0.1041	0.2094	-0.1924
-0.3055	-0.0102	1.0000

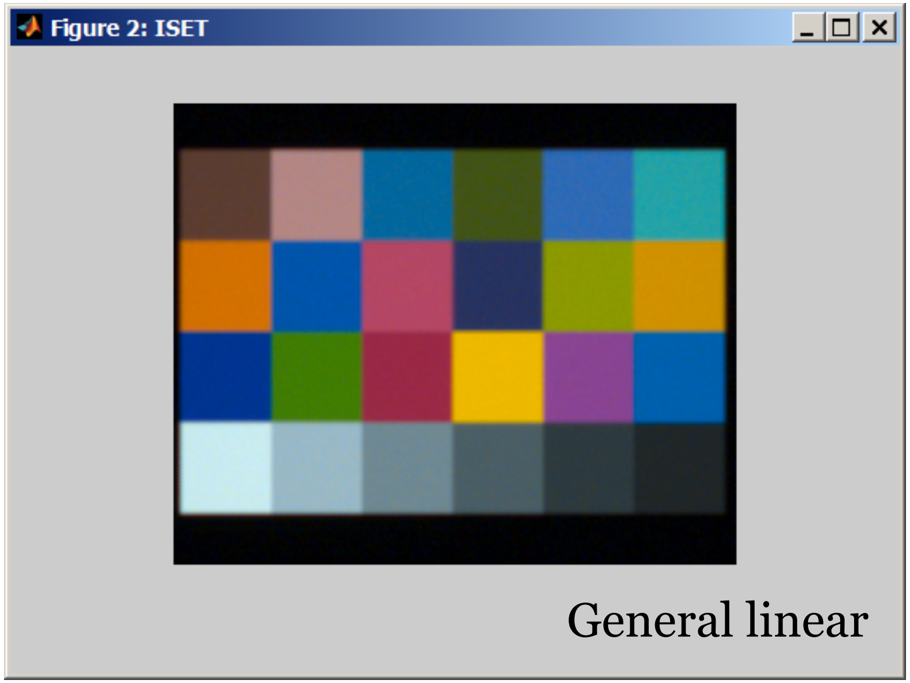




Illuminant estimation algorithms

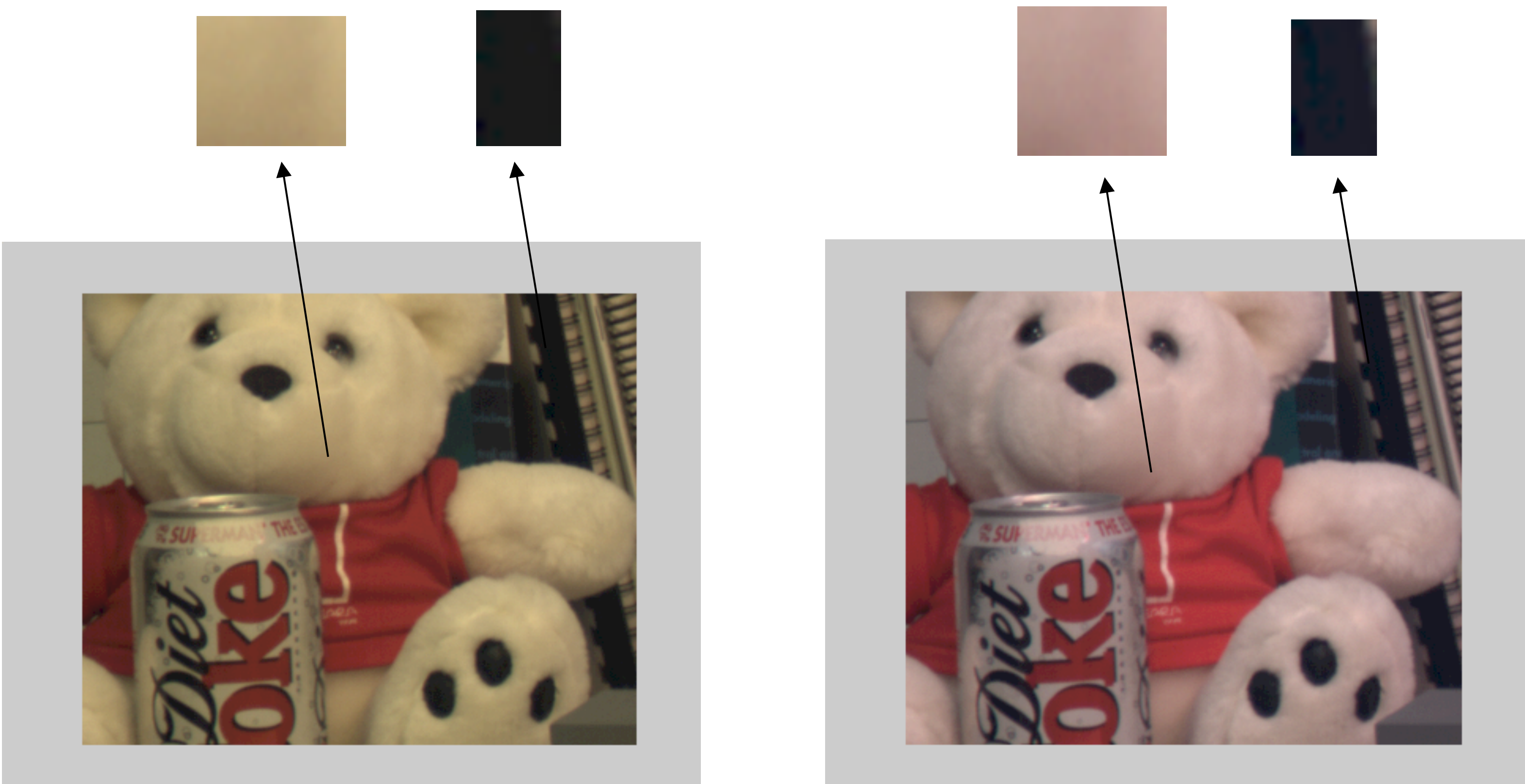
Device-dependent diagonal





Illuminant correction advice

Best Information In high intensity – black is black



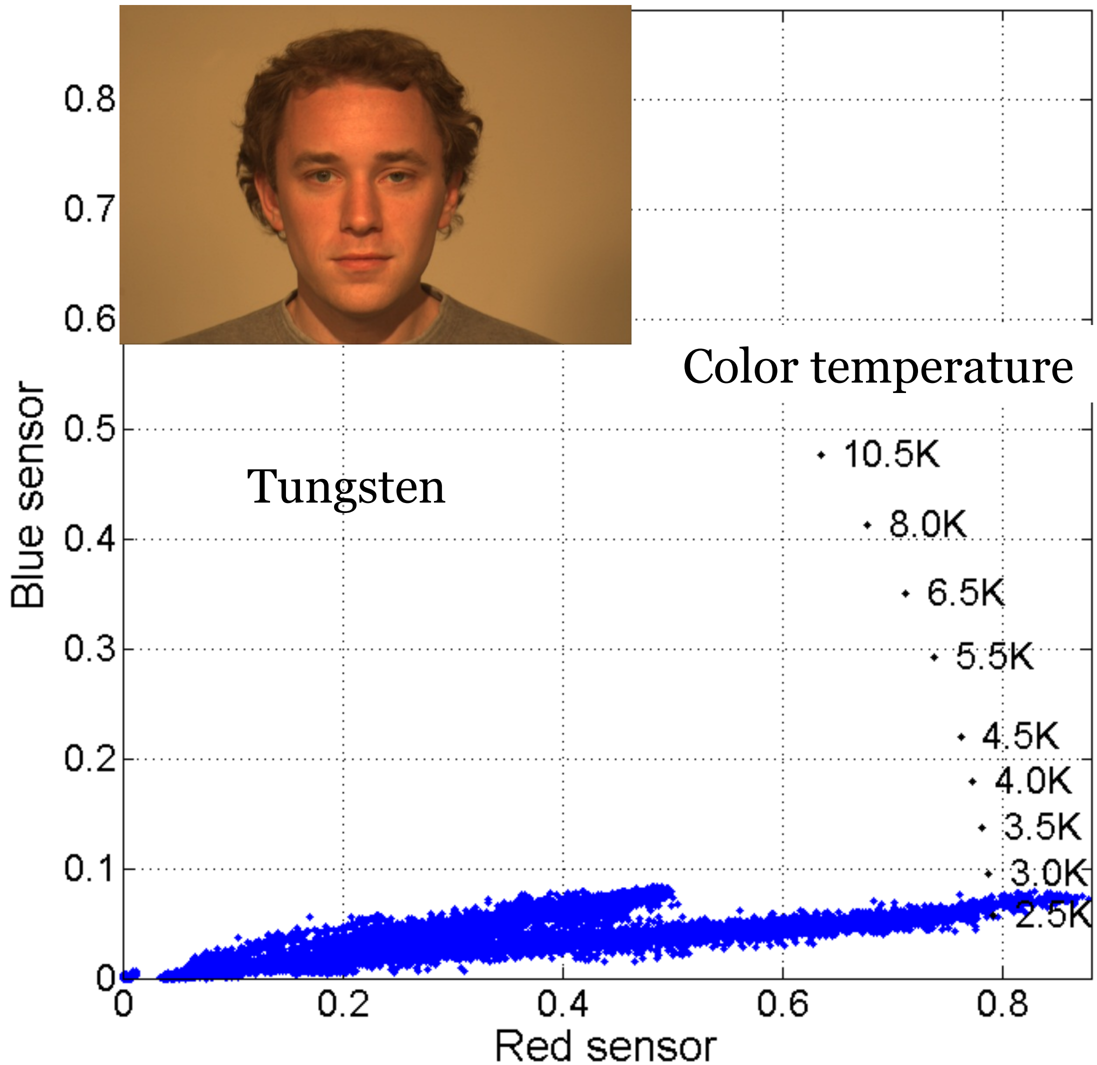
Examples of image statistics used in illuminant estimation

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 23, NO. 11, NOVEMBER 2001 1209

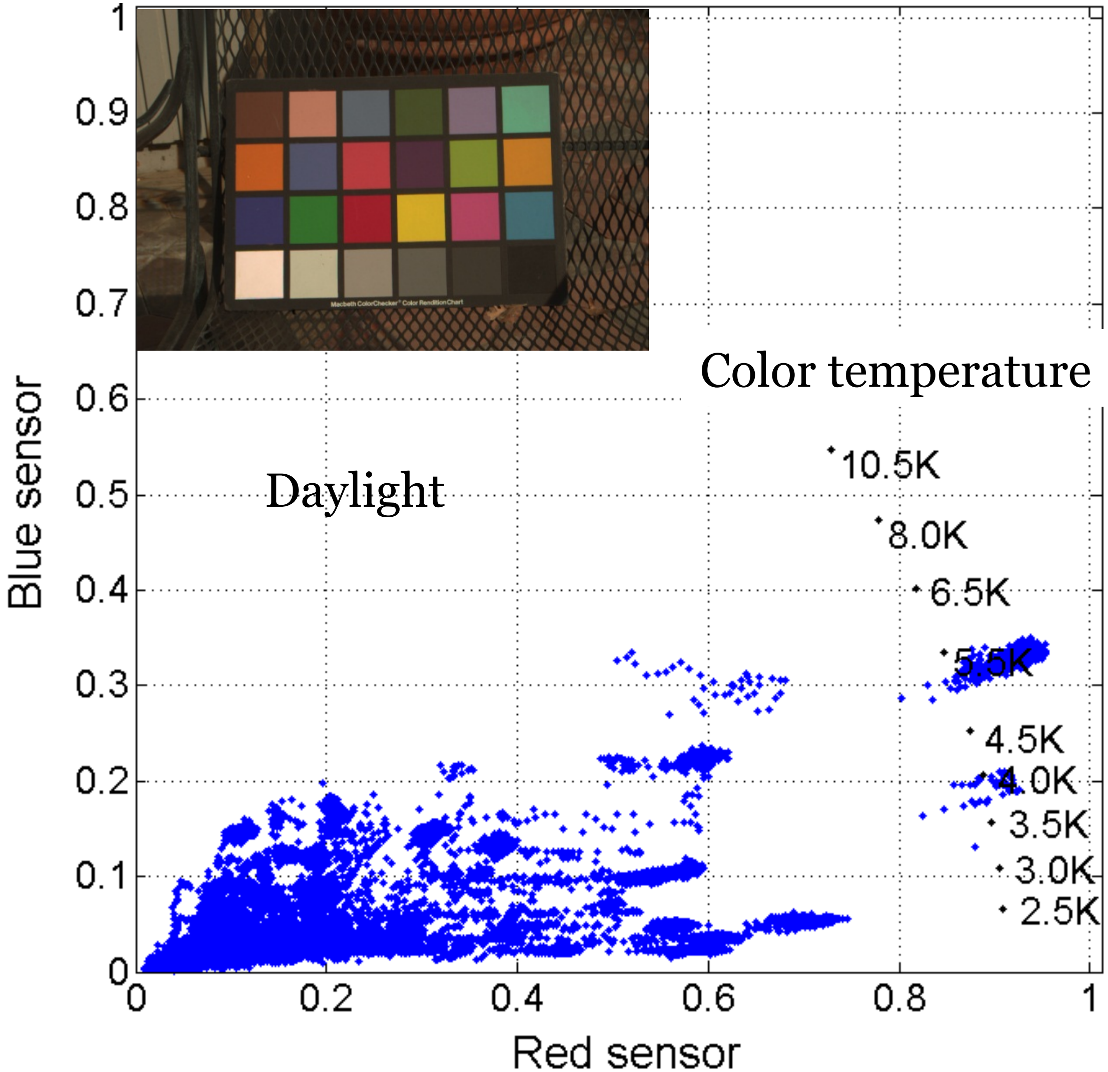
Color by Correlation: A Simple, Unifying Framework for Color Constancy

Graham D. Finlayson, Steven D. Hordley, and Paul M. Hubel

Color by correlation image statistics



Color by correlation image statistics



Flash/No Flash color balancing
(DiCarlo et al., CIC, 2001)

Multiple captures

Illuminant bracketing

Single image

Flash is known illuminant

Flash/No Flash color balancing
(DiCarlo et al., CIC, 2001)

Use the known illuminant case to estimate the surface reflectance

Single image

Flash is known illuminant

Flash/No Flash color balancing
(DiCarlo et al., CIC, 2001)

Use the estimated surfaces to estimate the surface illuminant SPD.

Single ambient illuminant

Surfaces are estimated

Space-varying illuminant estimation