Computer Vision
Light and Color

Ali Hosseininaveh

Tools and references for this course

- Image Processing in Matlab:
  - Peter Corke's Book – Robotics, Vision and control
  - Peter Corke's Toolbox

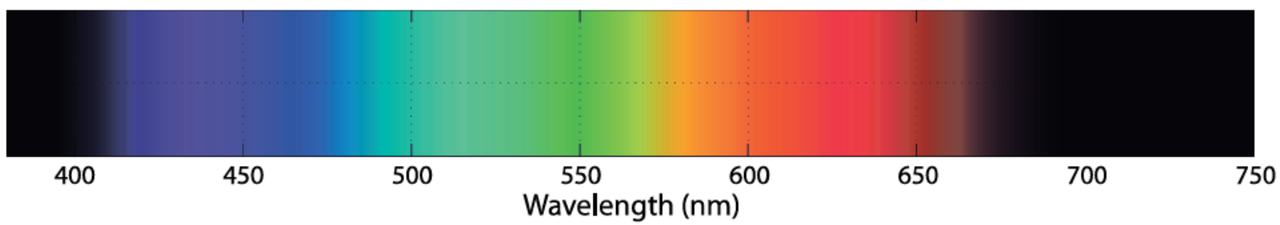
www.petercorke.com/Toolbox_software.html

Outline

- Spectral representation of Light
  - Absorption
  - Reflection
- Color
  - Reproducing colors
  - Chromaticity space
  - Color names
  - Other color spaces
  - Transforming between different primaries
  - What is white
- Advanced topics
  - Color consistency
  - White balancing
  - Color change due to absorption
  - Gamma
  - Applications: color image

Spectral representation of Light

- Isaac Newton (1670):



- Planck's Radiation Formula:

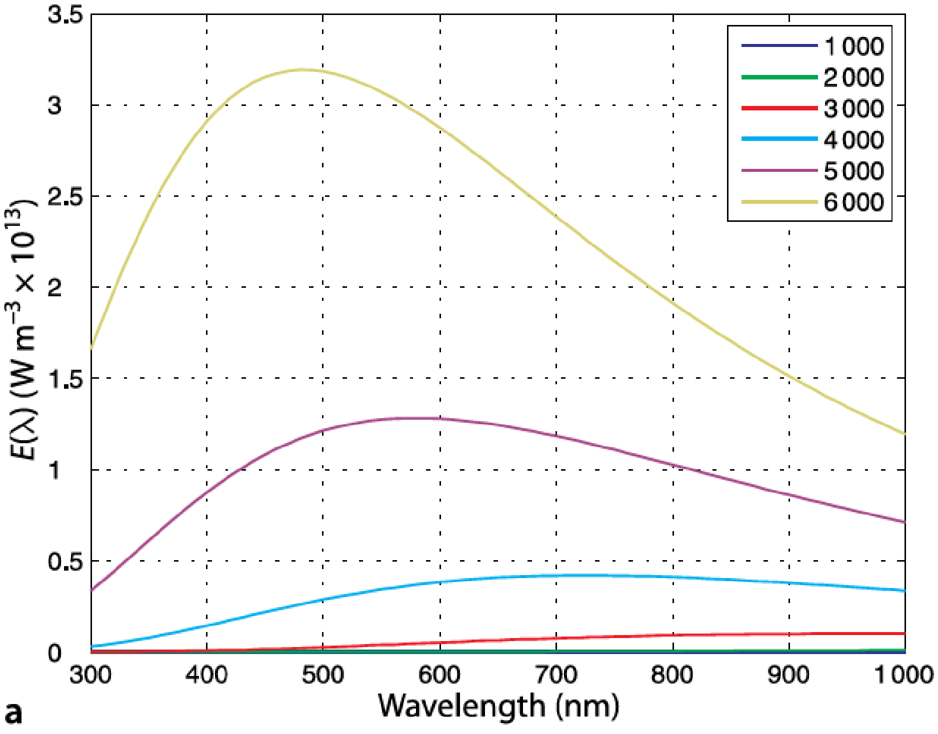
$$E(\lambda) = \frac{2\pi hc^2}{\lambda^5(e^{hc/k\lambda T} - 1)}\ \mathrm{W\,m^{-2}\,m^{-1}}$$

$c = 2.998 \times 10^8\ \mathrm{m\,s^{-1}}$,
$h = 6.626 \times 10^{-34}\ \mathrm{Js}$,
$k = 1.381 \times 10^{-23}\ \mathrm{J\,K^{-1}}$.

-

Blackbody spectra

```
lambda = [300:10:1000]*1e-9;
for T=1000:1000:6000
    plot( lambda*1e9, blackbody(lambda, T)); hold all
end
```



Spectrum laws

- The total amount of radiated is the area under the blackbody curve (Stefan Boltzman law):

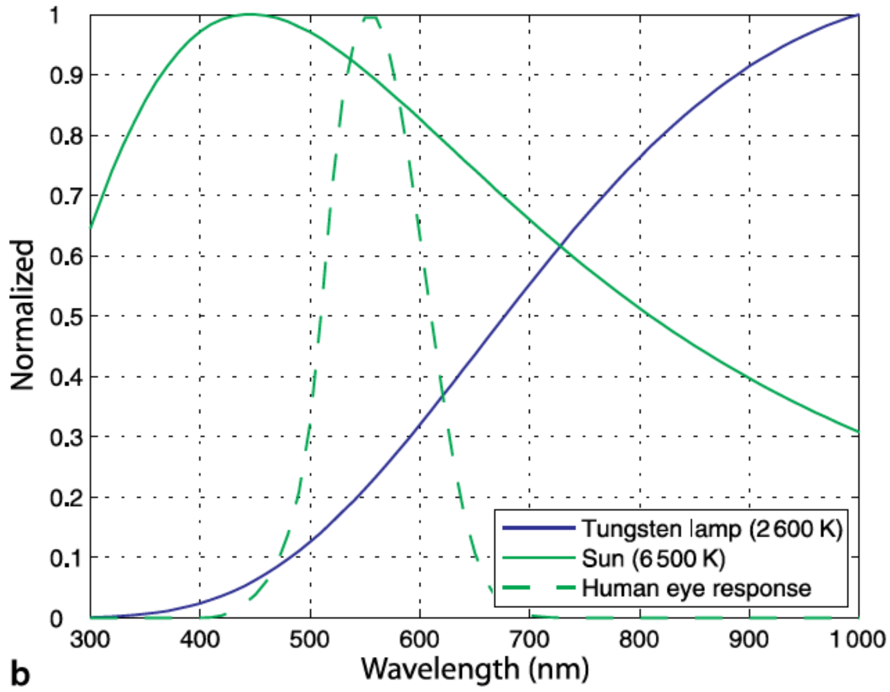
$$\frac{2\pi^5 k^4}{15c^2h^3}T^4 \ \mathrm{W\,m^{-2}}$$

- The peak of blackbody curve (Wien's Displacement law):

$$\lambda_{\mathrm{max}} = \frac{2.8978 \times 10^{-3}}{T}\ \mathrm{m}$$

Blackbody emission for Sun and a lamp

```
lamp = blackbody(lambda, 2600);
sun = blackbody(lambda, 6500);
plot(lambda*1e9, [lamp/max(lamp) sun/max(sun)])
```

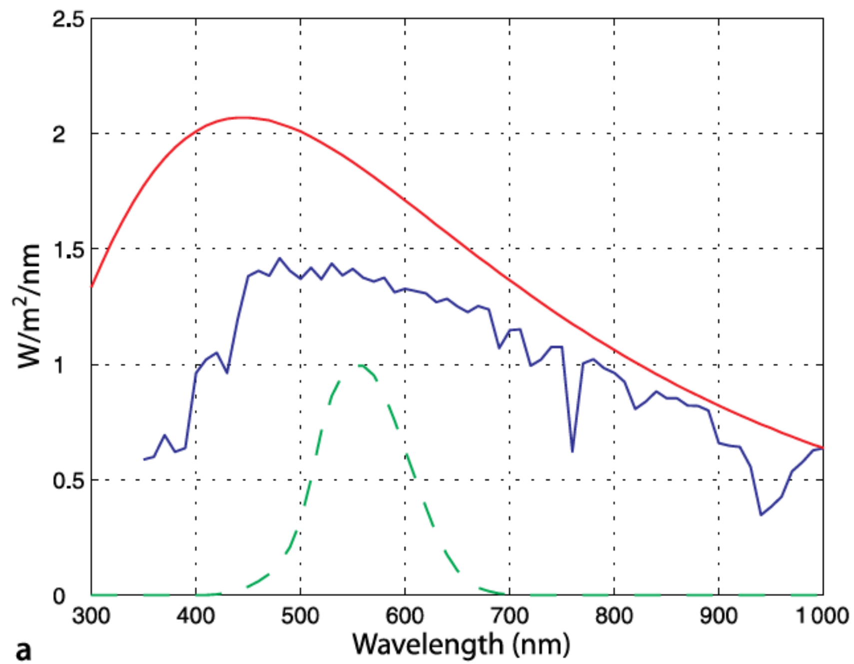


Absorption

- The Sun's spectrum at ground level on the earth:

```
sun_ground = loadspectrum(lambda, 'solar.dat');
plot(lambda*1e9, sun_ground)
```

- CO_2 absorbs radiation in the infra-red region,
- Ozone(O_3) absorbs strongly in the ultra-violet
- Region.



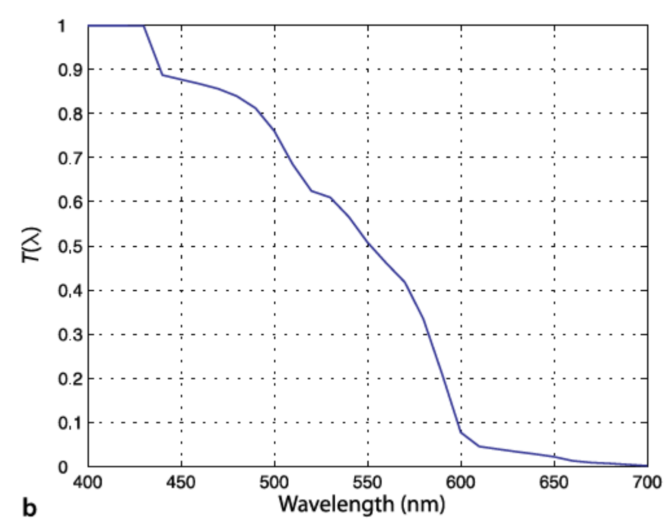
Absorption

- Transmission is the inverse of Absorption (Beer's Law): $T = 10^{-Ad}$

A is the absorption coefficient in units of m^{-1} and d is the path length

- Example: Transmission through 5 meter water

```
[A, lambda] = loadspectrum([400:10:700]*1e-9, 'water.dat');
d = 5;
T = 10.^(-A*d);
plot(lambda*1e9, T);
```

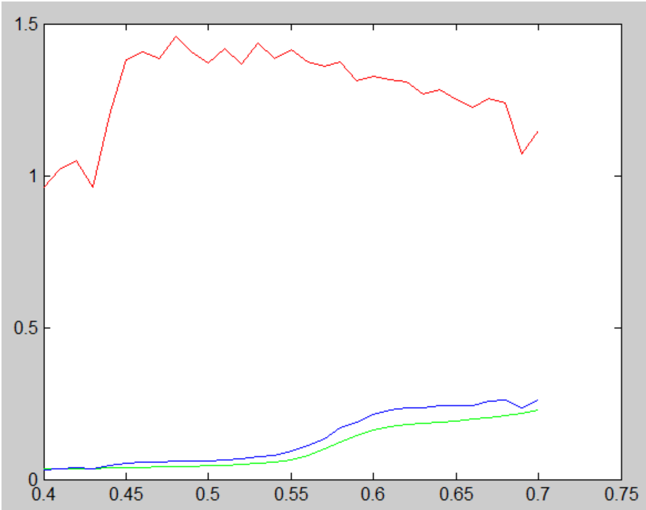


Reflection

- The spectrum of light reflected from a surface:

$$L(\lambda) = E(\lambda)R(\lambda) \text{ W m}^{-2}$$

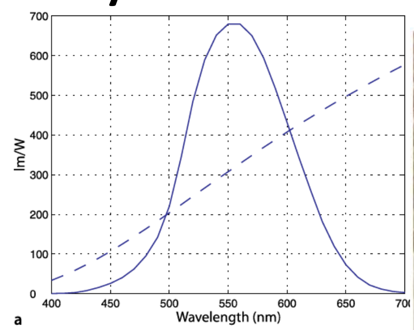
```
lambda = [400:10:700]*1e-9;
E = loadspectrum(lambda, 'solar.dat');
R = loadspectrum(lambda, 'redbrick.dat');
L = E .* R; plot(lambda*1e9, L,’b’);
```

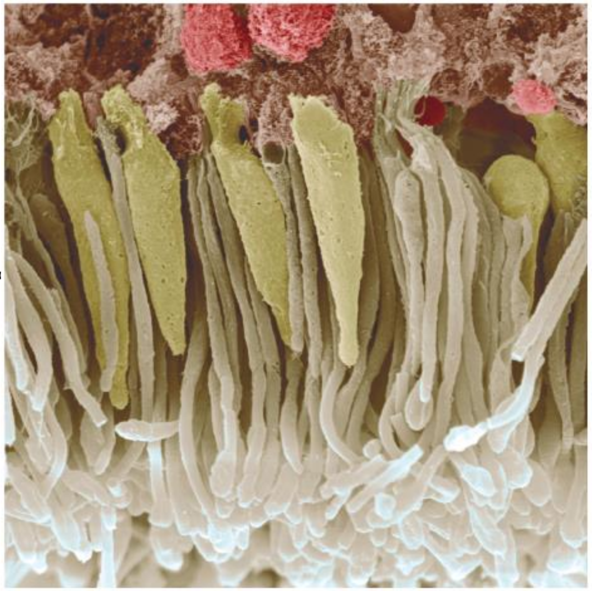


Color

- Our eyes contain two types of light sensitive cells
- Cone cells respond to particular colors and provide us with our normal daytime vision
- Rod cells are much more sensitive than cone cells but respond to intensity only and are used at night.

```
human = luminos(lambda);
plot(lambda*1e9, human)
camera = ccdresponse(lambda);
hold on
plot(lambda*1e9, camera*max(human), '--')
```

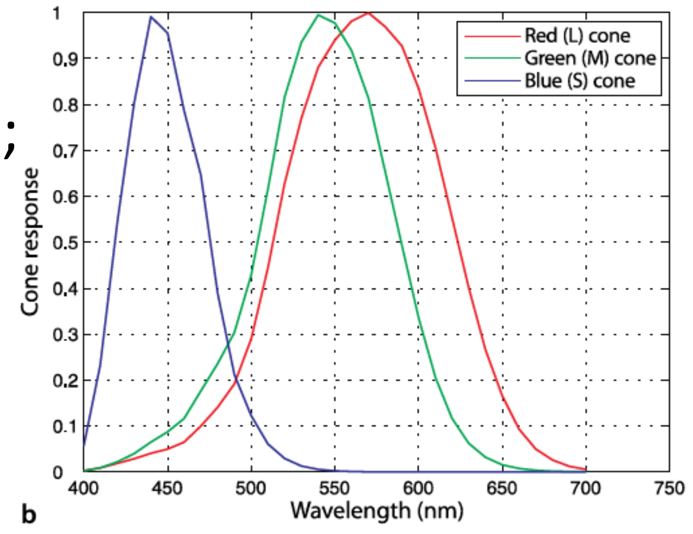




Color

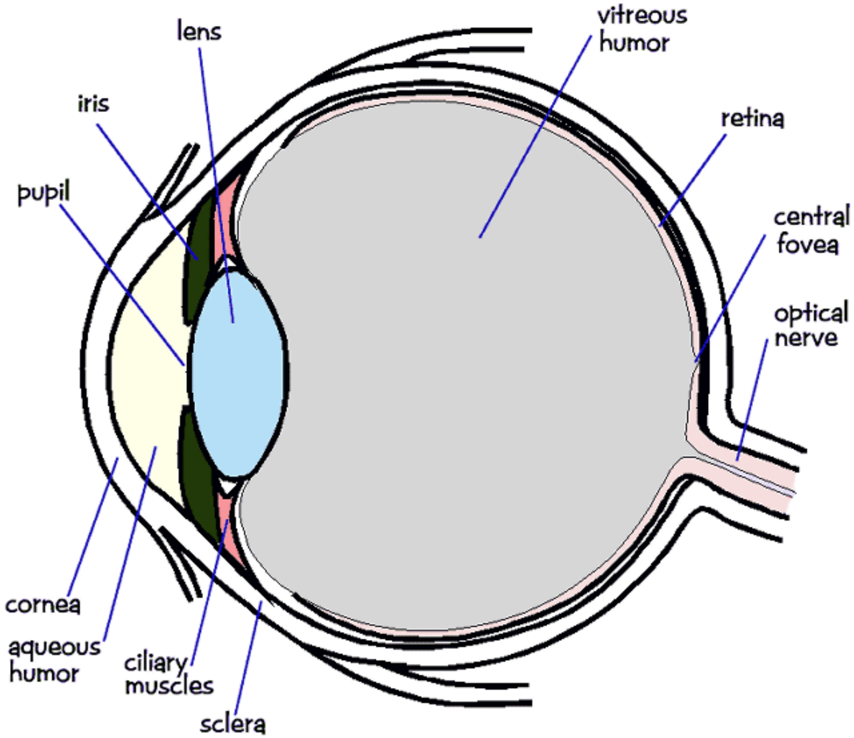
- The brightness we associate with a particular wavelengths is known as luminosity and is measured in units of lumens per watt.
- Humans are trichromats and have three types of cones (long (L), medium (M) and short (S)) or (RGB)

```
cones = loadspectrum(lambda, 'cones.dat');
plot(lambda*1e9, cones)
```



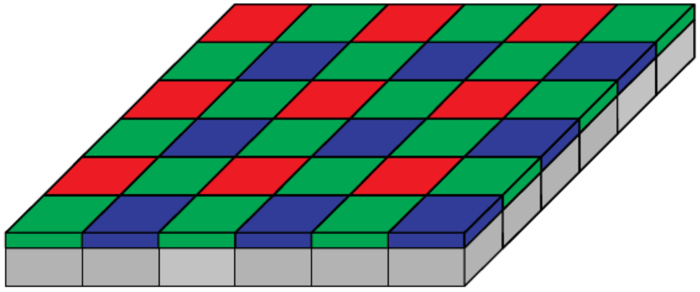
Human eye

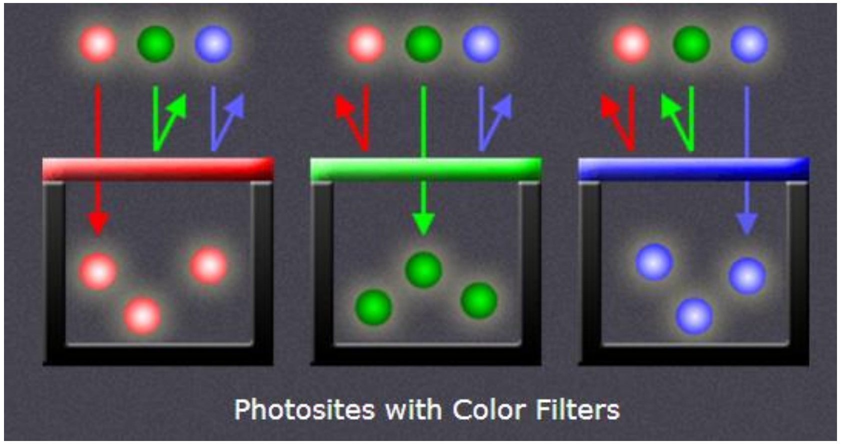
- The retina of the human eye has a central or foveal region which is only 0.6 mm in diameter and contains most of the 6 million cone cells:
- 65% sense red,
- 33% sense green and
- only 2% sense blue.
- 120 millions rod cells



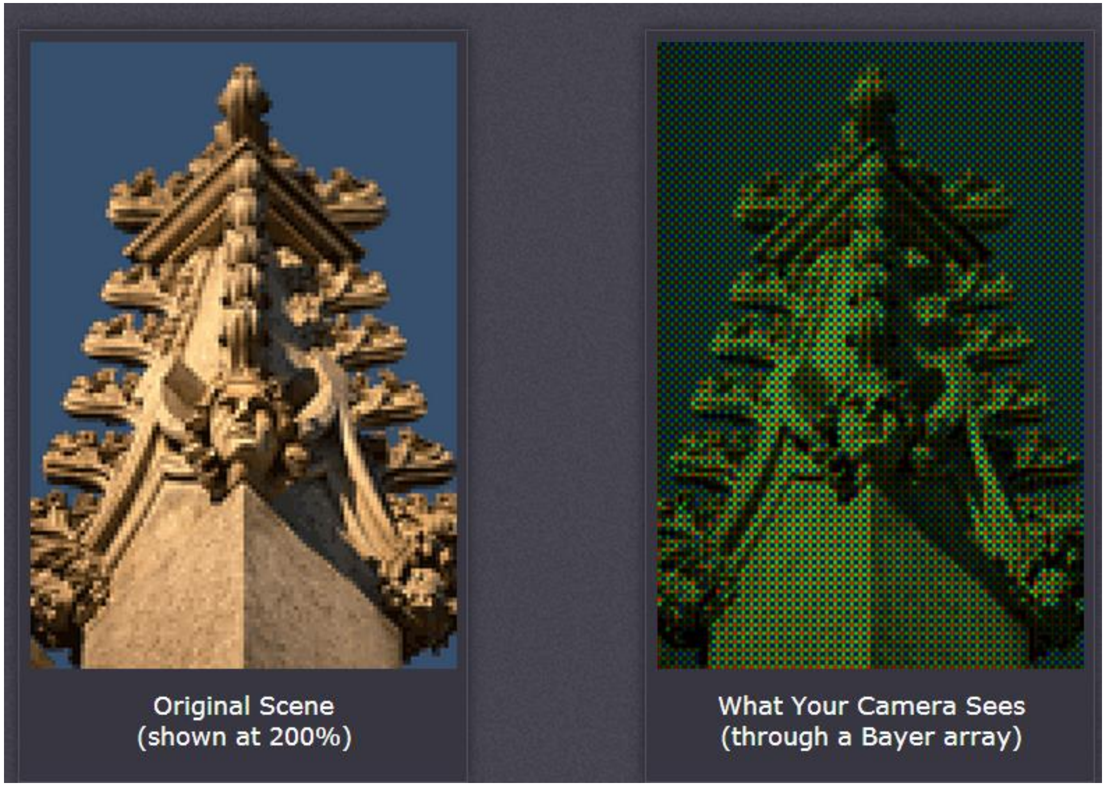
Digital Camera Sensor

- The sensor in a digital camera is analagous to the retina, but instead of rod and cone cells there is a regular array of light sensitive photosites on a silicon chip.
- A very common arrangement of color filters is the Bayer pattern which uses a regular 2× 2 photosite pattern comprising two green filters, one red and one blue.

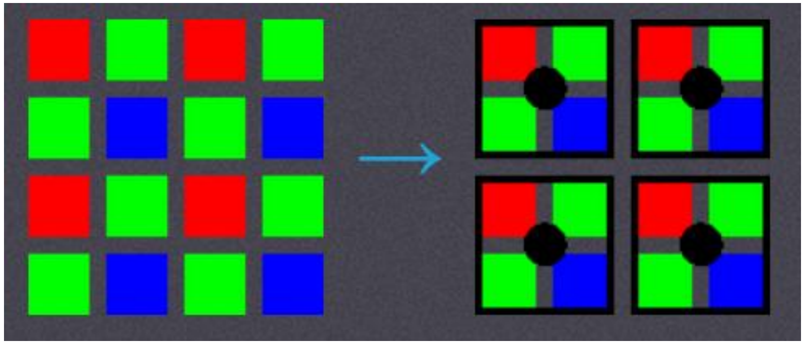




Digital Camera Sensor



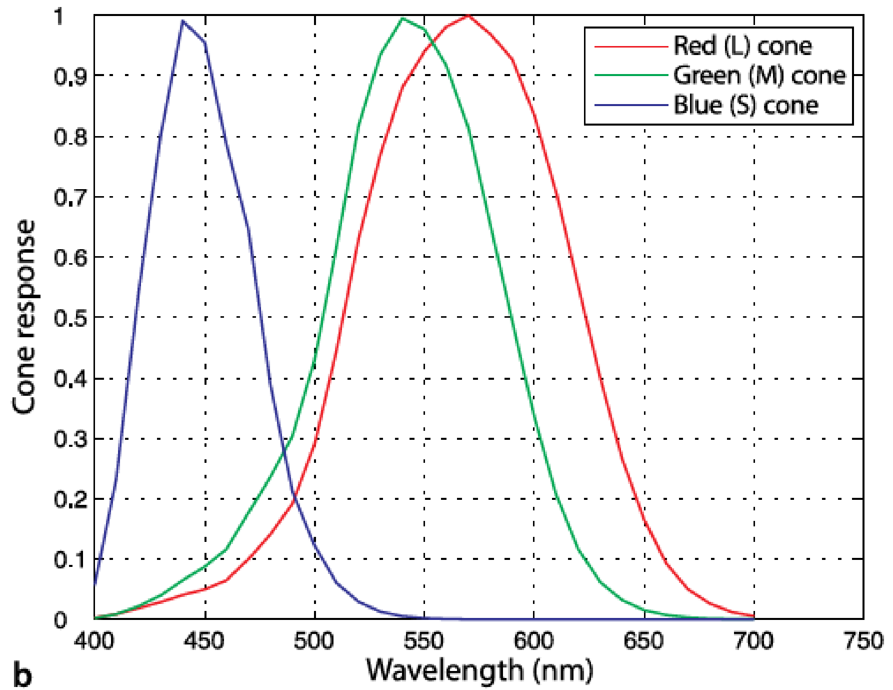
BAYER DEMOSAICING



Color

- Cones' respond to the luminance of an object:

$$\rho = \int_\lambda L(\lambda) M_r(\lambda) d\lambda$$
$$\gamma = \int_\lambda L(\lambda) M_g(\lambda) d\lambda$$
$$\beta = \int_\lambda L(\lambda) M_b(\lambda) d\lambda$$



b

sum((L*ones(1,3)) .* cones)

16.3578 10.0702 2.8219

where $M_r(\lambda)$, $M_g(\lambda)$ and $M_b(\lambda)$ are the spectral response of the red, green and blue cones respectively

The response is a 3-vector (ρ, γ, β) which is known as a tristimulus.

Color (last sentences)

Primary colors are not a fundamental property of light – they are a fundamental property of the observer. There are three primary colors only because we, as trichromats, have three types of cones. Birds would have four primary colors and dogs would have two.

Reproducing color

- How much of each primary is required to match a given tristimulus?

$$\rho = L_\lambda M_r(\lambda_S) \qquad \rho = RM_r(\lambda_r) + GM_r(\lambda_g) + BM_r(\lambda_b)$$

$$\gamma = L_\lambda M_g(\lambda_S) \qquad \gamma = RM_g(\lambda_r) + GM_g(\lambda_g) + BM_g(\lambda_b)$$

$$\beta = L_\lambda M_b(\lambda_S) \qquad \beta = RM_b(\lambda_r) + GM_b(\lambda_g) + BM_b(\lambda_b)$$

$$L_\lambda \begin{pmatrix} M_r(\lambda_S) \\ M_g(\lambda_S) \\ M_b(\lambda_S) \end{pmatrix} = \begin{pmatrix} M_r(\lambda_r) & M_r(\lambda_g) & M_r(\lambda_b) \\ M_g(\lambda_r) & M_g(\lambda_g) & M_g(\lambda_b) \\ M_b(\lambda_r) & M_b(\lambda_g) & M_b(\lambda_b) \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

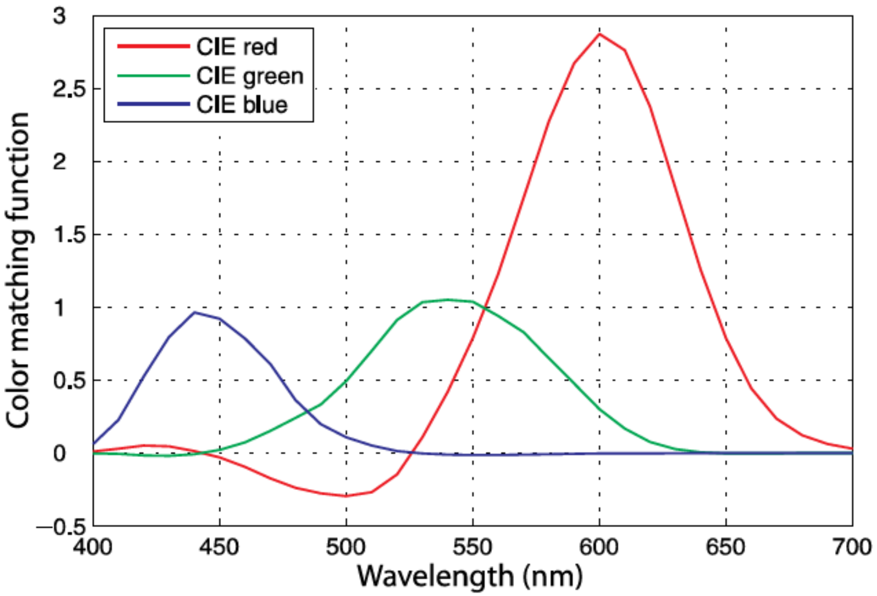
$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = L_\lambda \begin{pmatrix} M_r(\lambda_r) & M_r(\lambda_g) & M_r(\lambda_b) \\ M_g(\lambda_r) & M_g(\lambda_g) & M_g(\lambda_b) \\ M_b(\lambda_r) & M_b(\lambda_g) & M_b(\lambda_b) \end{pmatrix}^{-1} \begin{pmatrix} M_r(\lambda_S) \\ M_g(\lambda_S) \\ M_b(\lambda_S) \end{pmatrix}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} \bar{r}(\lambda_S) \\ \bar{g}(\lambda_S) \\ \bar{b}(\lambda_S) \end{pmatrix}$$

$\bar{r}(\lambda)$, $\bar{g}(\lambda)$, $\bar{b}(\lambda)$ are known as color matching functions.

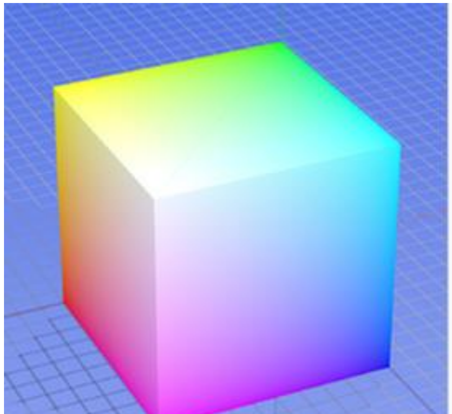
Color matching functions

```
lambda = [400:10:700]*1e-9;
cmf = cmfrgb(lambda);
plot(lambda*1e9, cmf);
```



- For example to create the sensation of light at 600 nm (orange) we would need

```
orange = cmfrgb(600e-9)
orange = [2.8717    0.3007    -0.0043]
```



Chromaticity space

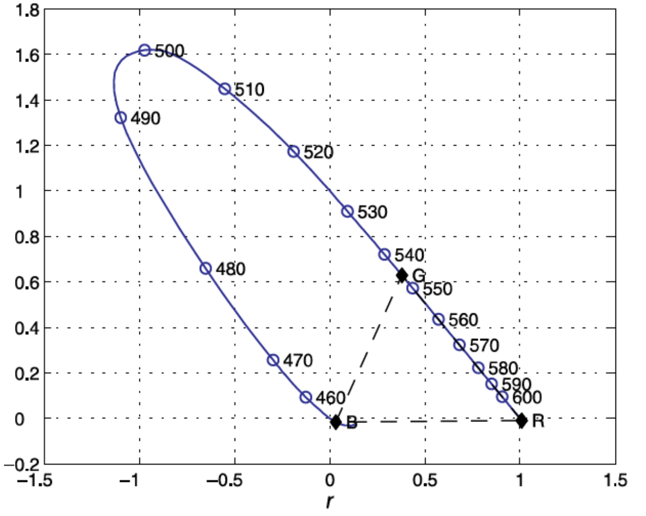
- normalizing the tristimulus values results in chromaticity coordinates:

$$r + g + b = 1$$

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}$$

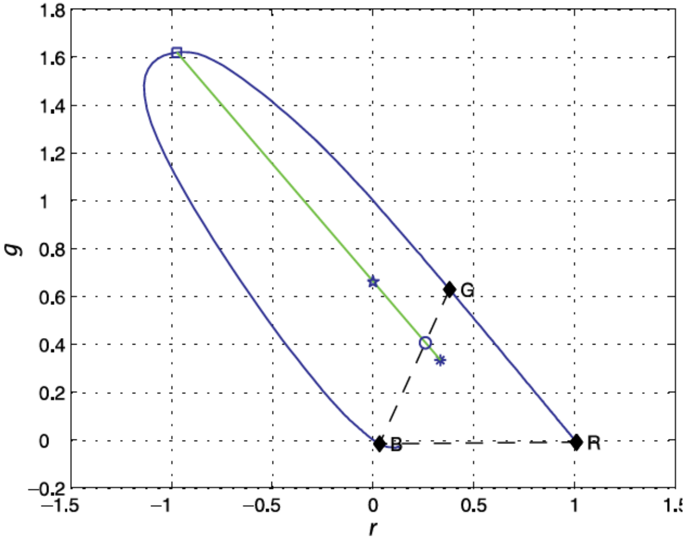
- Since the effect of intensity has been eliminated the 2-dimensional quantity (r, g) represents color.

```
[r,g] = lambda2rg( [400:700]*1e-9 );
plot(r, g)
rg_addticks
```



Chromaticity space

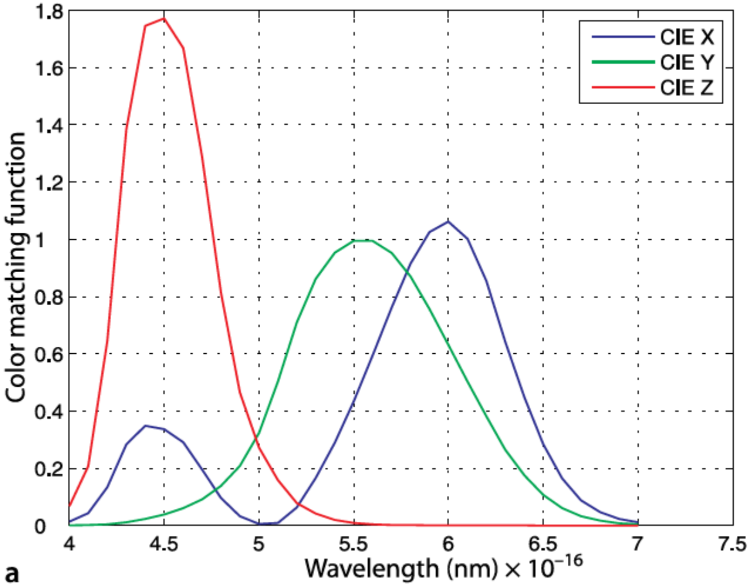
- a mixture of two colors lies along a line between those two colors on the chromaticity plane
- A mixture of N colors lies within a region bounded by those colors
- since all color stimuli are combinations of spectral stimuli all real color stimuli must lie on or inside the spectral locus.
- any colors we create from mixing the primaries can only lie within the triangle bounded by the primaries – the color gamut.
- many real colors cannot be created using these primaries
- >> green_cc = lambda2rg(500e-9);
- green_cc =
- -0.9733 1.6187
- >> plot2(green_cc, 's')
- >> white_cc = tristim2cc([1 1 1])
- white_cc =
- 0.3333 0.3333
- >> plot2(white_cc, '*')



Imaginary non-physical primaries (XYZ)

- The XYZ color matching functions defined by the CIE:

```
cmf = cmfxyz(lambda);
plot(lambda*1e-9, cmf);
```



a

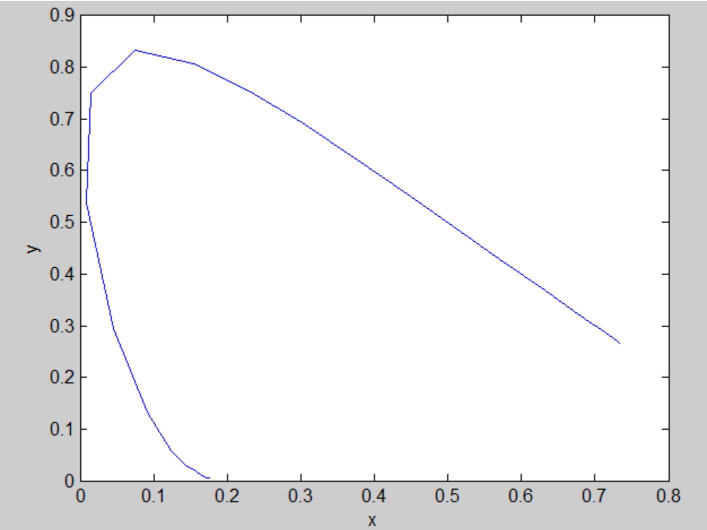
- The corresponding chromaticity coordinates:

$$x = \frac{X}{X+Y+Z}, y = \frac{Y}{X+Y+Z}, z = \frac{Z}{X+Y+Z}$$

$$x + y + z = 1$$

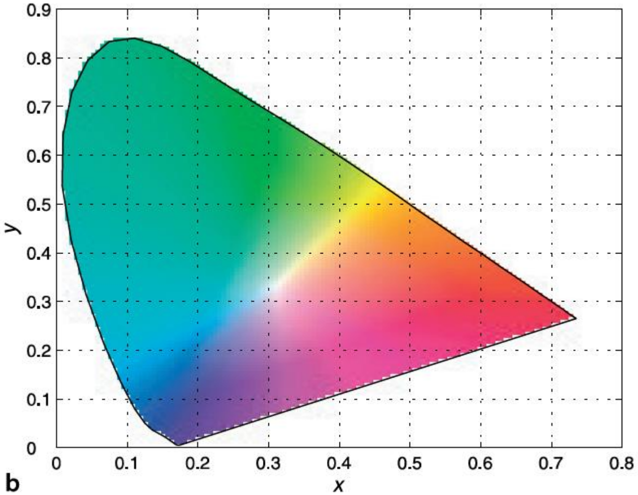
Spectral locus

```
[x,y] = lambda2xy(lambda);
plot(x, y);

xycolorspace

lambda2xy(550e-9)
ans = 0.3016 0.6924
```



- The chromaticity coordinates of a standard tungsten illuminanted at 2600 K is

```
lamp = blackbody(lambda, 2600);
lambda2xy(lambda, lamp)
ans =
0.4679 0.4126
```



Color names

colorname('pink')

ans=   1.0000   0.6824   0.7255

colorname('red','xy')

ans =   0.6400   0.3300

colorname([1,0,0])

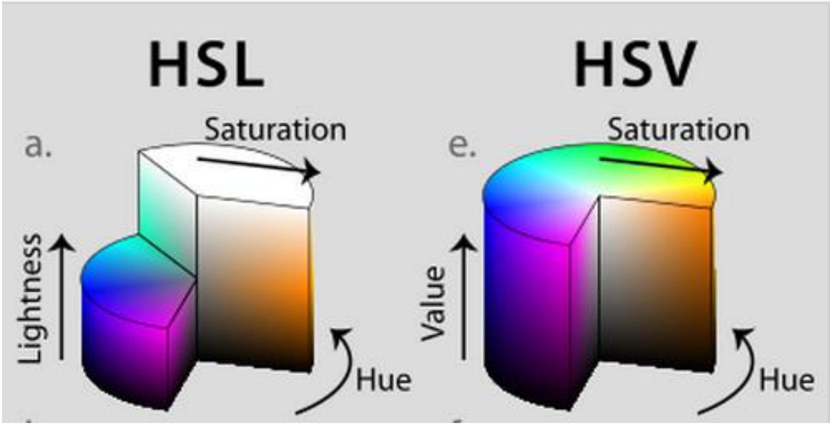
?

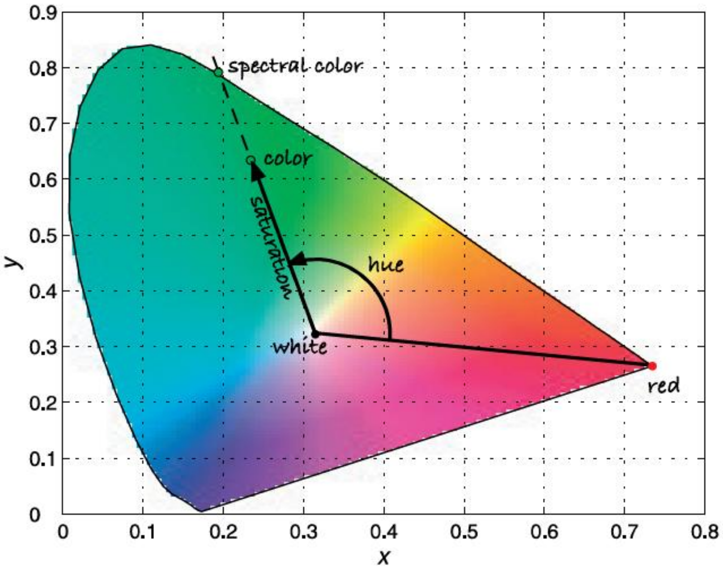
Other Color Spaces

- The definition from ITU Recommendation 709

$$Y^{709} = 0.2126R + 0.7152G + 0.0722B$$

- HSV color space
  - H : Hue
  - S (C): Saturation or Chroma
  - V (L): Value or lightness





RGB-HSV color spaces

colorspace('RGB->HSV', [1, 0, 0]) ans = 0 1 1

colorspace('RGB->HSV', [0, 1, 0]) ans = 120 1 1

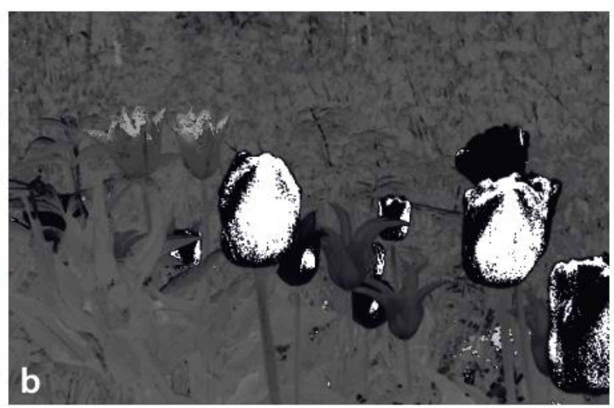
colorspace('RGB->HSV', [0, 0, 1]) ans = 240 1 1

colorspace('RGB->HSV', [0, 0.5, 0]) ans = 120 1 0.5

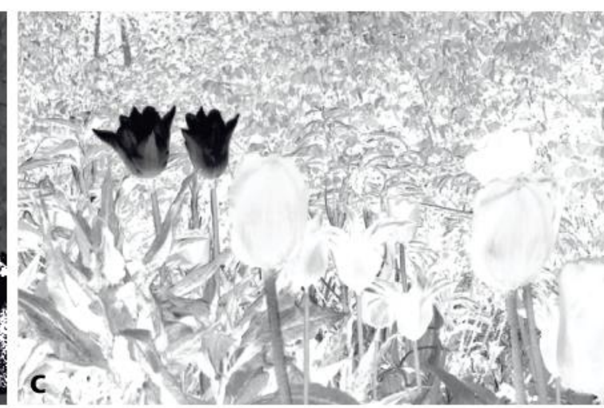
colorspace function can be used for images

Original image

Hue image



Saturation image



Other color spaces

- The colorspace function can convert between 13 different color spaces. For example:
  - YUV,
  - YCbCr
  - L* a* b*
  - L* u* v* .
  - RGB
  - XYZ
  - HSV

Transforming between Different Primaries

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = C \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$\frac{1}{y_w}\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = C \begin{pmatrix} J_R & 0 & 0 \\ 0 & J_G & 0 \\ 0 & 0 & J_B \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

	R_{CIE}	G_{CIE}	B_{CIE}	R_{709}	G_{709}	B_{709}	D_{65}	E
x	0.7347	0.2738	0.1666	0.640	0.300	0.150	0.3127	0.3333
y	0.2653	0.7174	0.0089	0.330	0.600	0.060	0.3290	0.3333
z	0.0000	0.0088	0.8245	0.030	0.100	0.790	0.3582	0.3333

$$\frac{1}{y_w}\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = C \begin{pmatrix} J_R & 0 & 0 \\ 0 & J_G & 0 \\ 0 & 0 & J_B \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

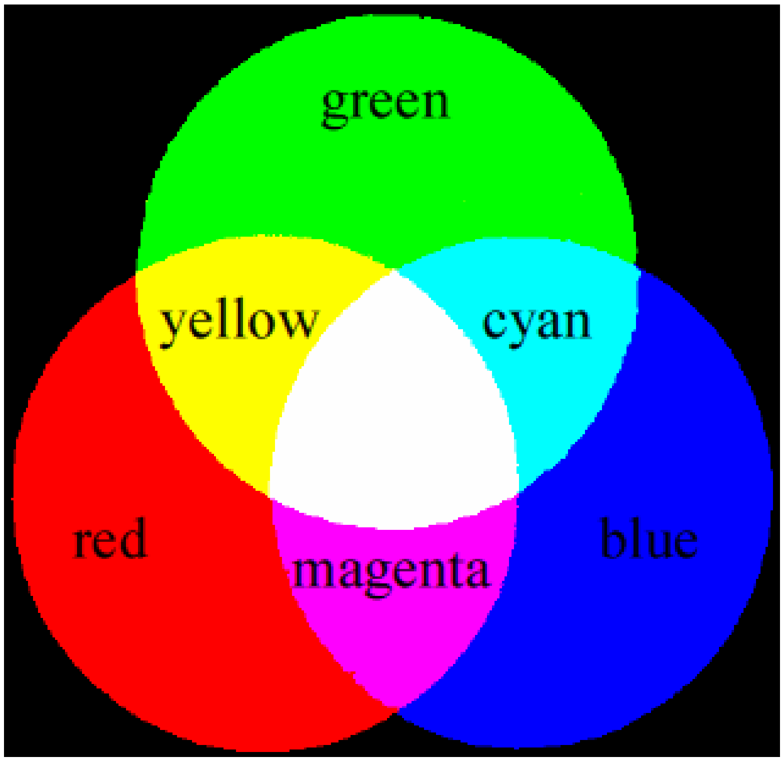
$$\begin{pmatrix} J_R \\ J_G \\ J_B \end{pmatrix} = C^{-1} \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} \frac{1}{y_w}$$

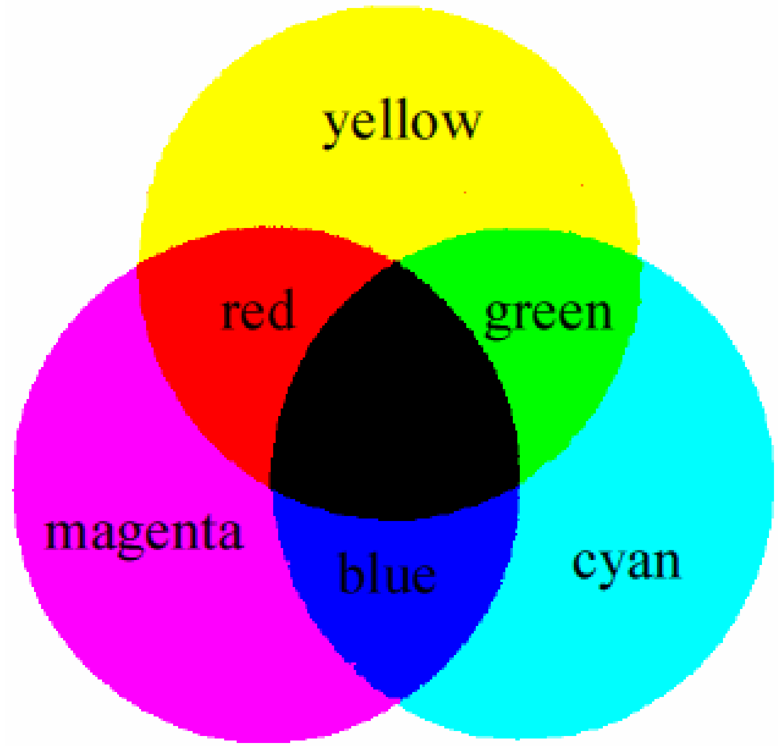
- white_cc = tristim2cc([1 1 1])

What is white

- White is both the absence of color and also the sum of all colors.
- One definition of white is standard daylight which is taken as the mid-day Sun in Western/Northern Europe which has been tabulated by the CIE as illuminate D65.
- d65 = blackbody(lambda, 6500);
- lambda2xy(lambda, d65)
- ans = 0.3136 0.3241

Mixing light and mixing pigment





$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = 1 - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

R+B+G=white (additive) R+G=Y

C+M+Y=black (subtractive) C+M=B etc...

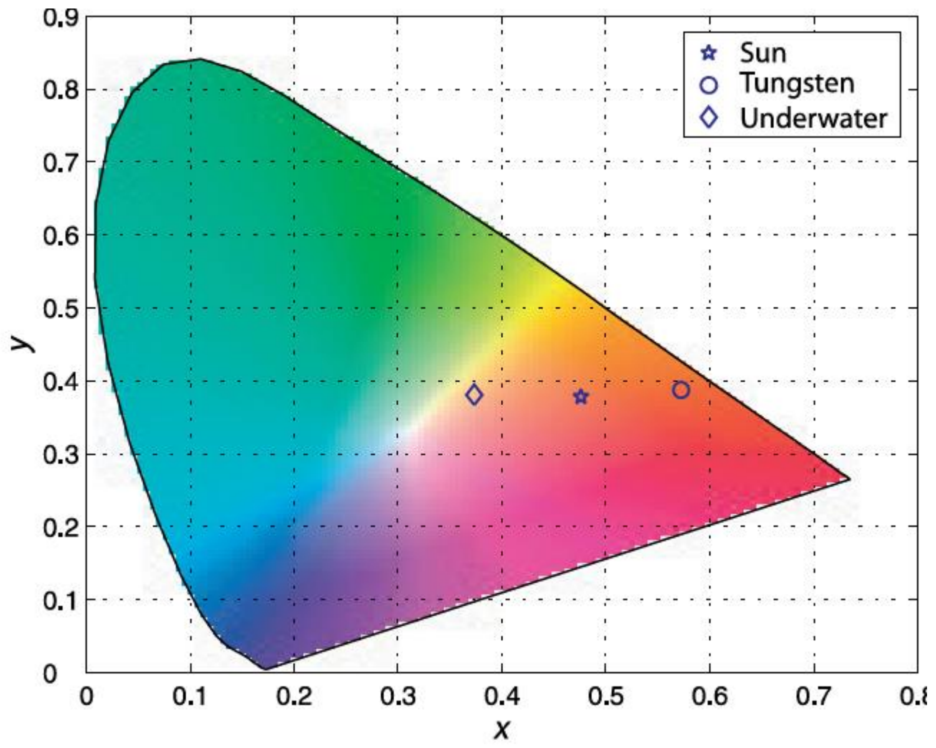
(CMYK common in printing, where K is black pigment)

Advanced Topics

- Color consistency
- White balancing
- Color change due to absorption
- Gamma
- Application: color image

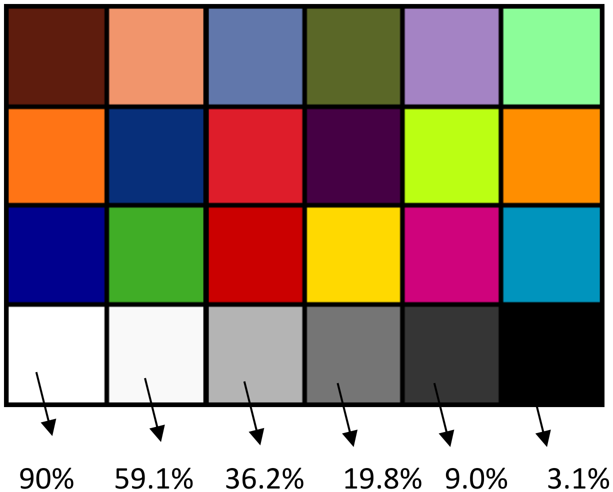
Color Constancy

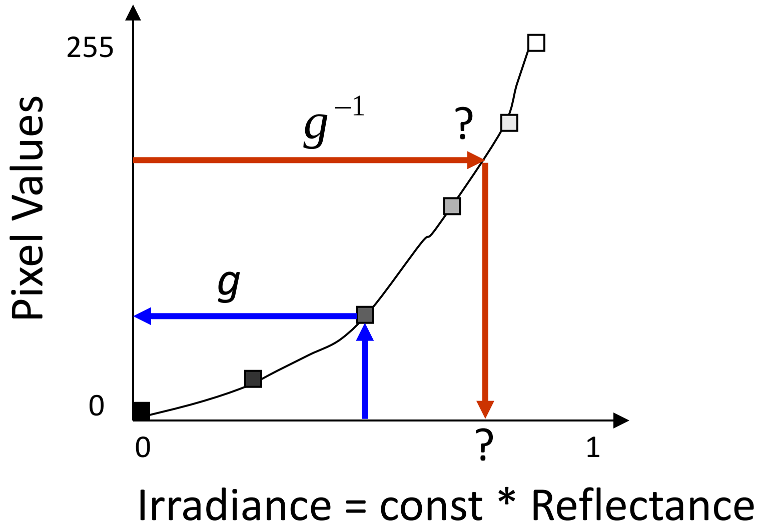
- Redbrick color is changed for different light sources



Color Chart Calibration

- Important preprocessing step for many vision and graphics algorithms
- Use a color chart with precisely known reflectances.





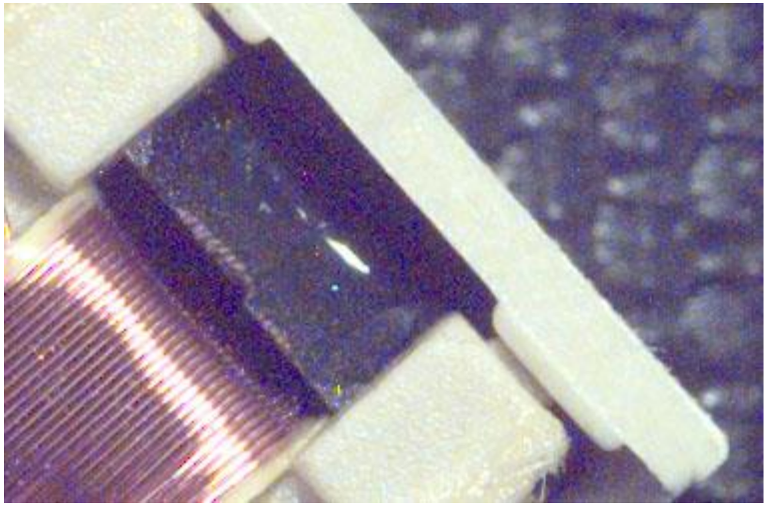
- Use more camera exposures to fill up the curve.
- Method assumes constant lighting on all patches and works best when source is far away (example sunlight).

- Unique inverse exists because g is monotonic and smooth for all cameras.

Dark Current Noise Subtraction

- Dark current noise is high for long exposure shots
- To remove (some) of it:
    - Calibrate the camera (make response linear)
    - Capture the image of the scene as usual
    - Cover the lens with the lens cap and take another picture
    - Subtract the second image from the first image

Dark Current Noise Subtraction



Original image + Dark Current Noise

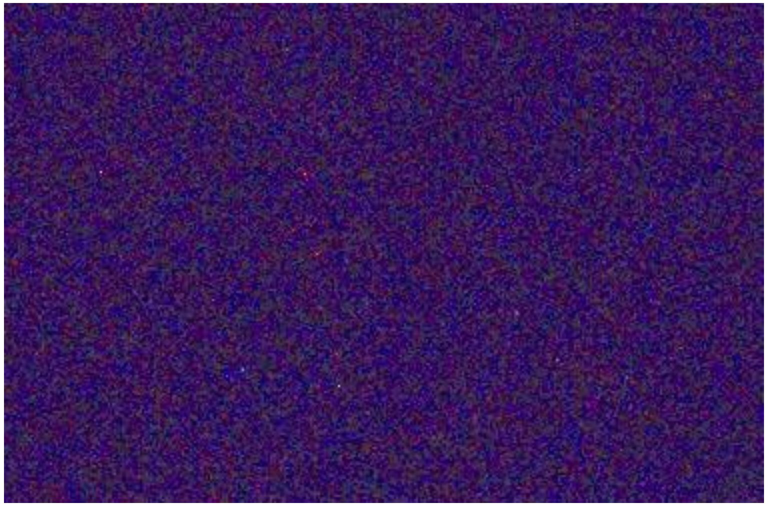
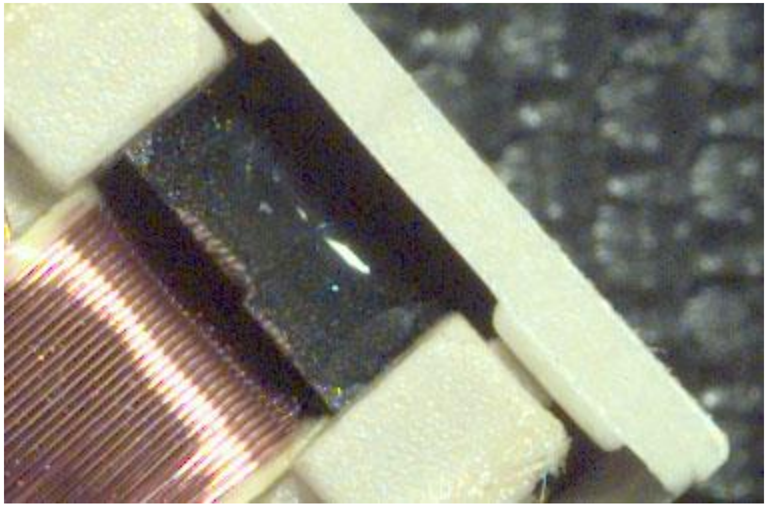


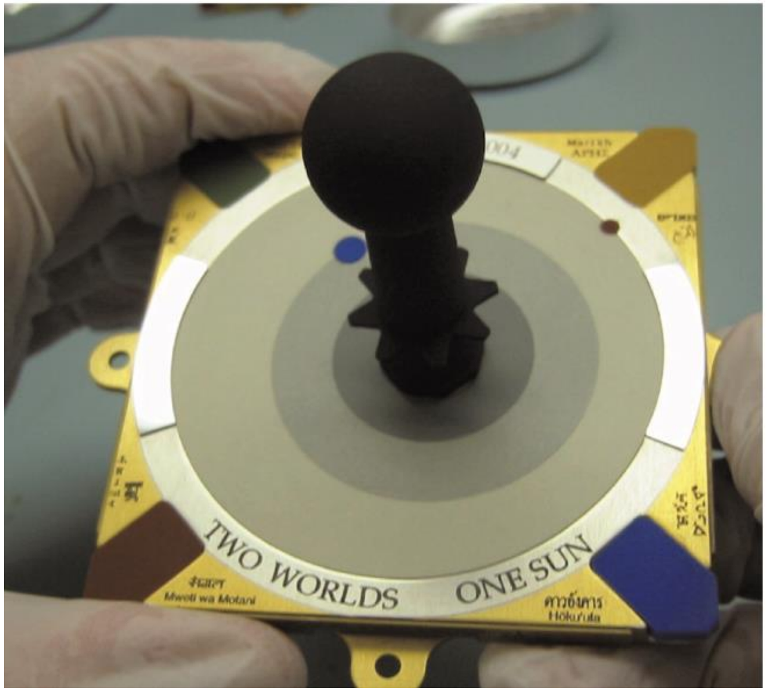
Image with lens cap on



Result of subtraction

White balancing

- Compared to daylight an incandescent lamp appears more yellow, to solve this problem a white balancing process should be done
  - Use a blue filter
  - Adjust the blue value in RGB
  - Use a calibration target



Color Change Due to Absorption

- [R,lambda] = loadspectrum([400:5:700]*1e-9, 'redbrick.dat');
- sun = loadspectrum(lambda, 'solar.dat');
- A = loadspectrum(lambda, 'water.dat');
- d = 2
- T = 10 .^ (-d*A);
- L = sun .* R .* T;
- xy_water = lambda2xy(lambda, L)

Gamma

- In an old fashioned CRT monitor the luminance produced at the face of the display is non-linearly related to the control voltage V according to

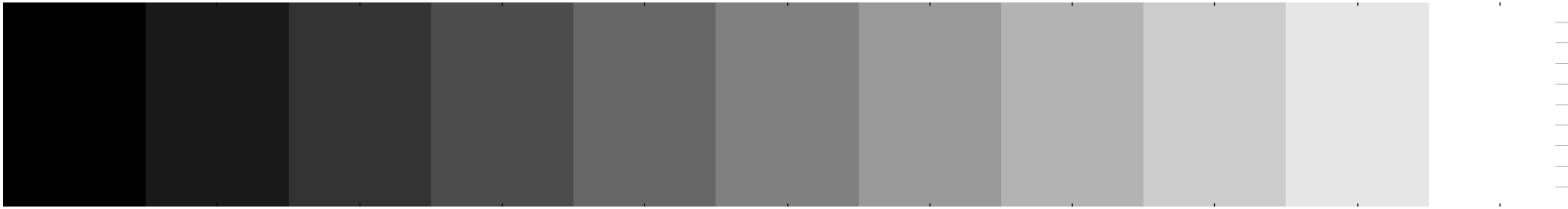
$$L = V^{\gamma}$$ $\gamma \approx 2.2$

- To make a linear relation between camera and monitor, we do a gamma correction on the input signal

$$V = L^{1/\gamma}$$

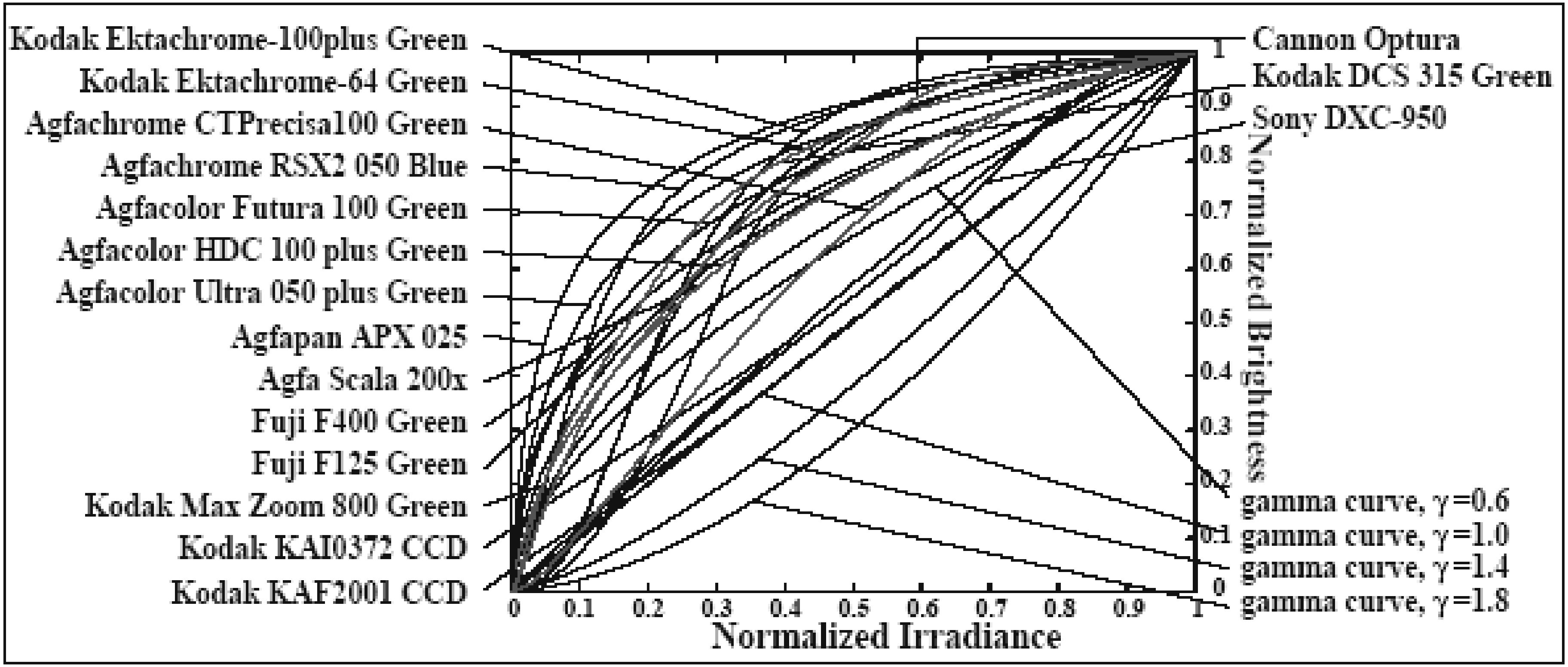
Gamma correction

- wedge = [0:0.1:1];
- idisp(wedge)



- idisp(wedge .^ (1/2.2))

Measured Response Curves of Cameras



[Grossberg, Nayar

a	b
c	d

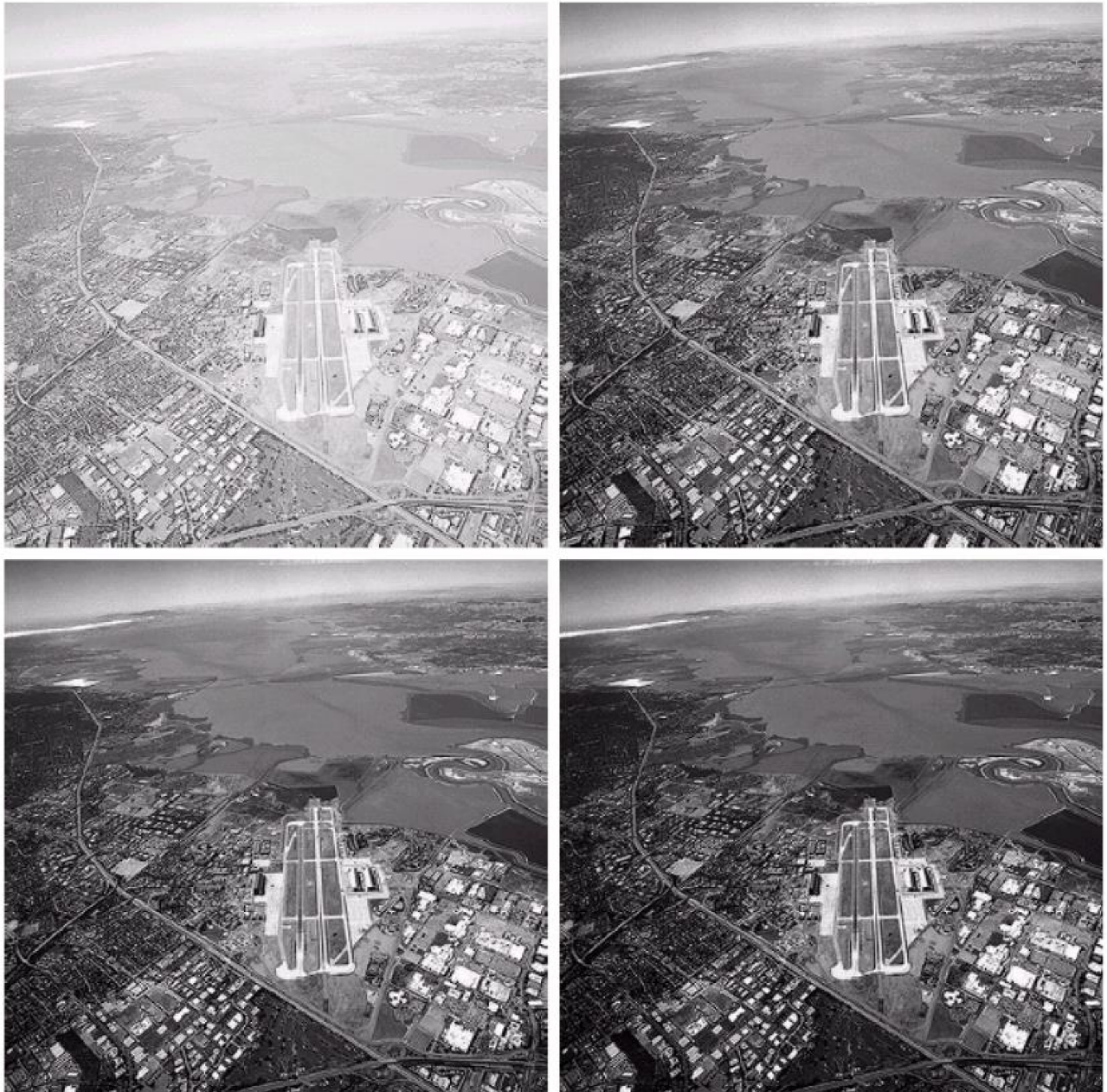


FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

Application: color image

```
flowers = iread('flowers4.png', 'double', 'gamma', 'sRGB');
hsv = colorspace('RGB->HSV', flowers);
idisp( hsv(:,:,1) )
idisp( hsv(:,:,2) )
XYZ = colorspace('RGB->XYZ', flowers);
[x,y] = tristim2cc(XYZ);
xbins = [0 0.01 100]; ybins = [0 0.01 100];
[h,vx,vy] = hist2d(x, y, xbins, ybins);
xycolorspace
hold on
contour(vx, vy, h)
[cls, cxy] = colorkmeans(flowers, 7);
idisp(cls, 'colormap', 'jet', 'nogui')
idisplabel(flowers, cls, 3)
xycolorspace
plot_point(cxy, '*', 'sequence', 'textsize', 10, 'textcolor', 'b')
```

Summary

- The light we observe is a mixture of frequencies, a continuous spectrum, which is modified by reflectance and absorption.
- The spectrum elicits a response from the eye which we interpret as color – for humans the response is a tristumulus, a 3-vector that represents the outputs of the three different types of cones in our eye.
- A digital color camera is functionally equivalent.
- The tristimulus can be considered as a 1-dimensional brightness coordinate and a 2-dimensional chromaticity coordinate which allows colors to be plotted on a plane.
- The spectral colors form a locus on this plane and all real colors lie within this locus.

Summary

- The three primary colors form a triangle on this plane which is the gamut of those primaries.
- Any color within the triangle can be matched by an appropriate mixture of the primaries.
- No set of primaries can define a gamut that contains all colors.
- An alternative set of imaginary primaries, the CIE XYZ system, does contain all real colors and is the standard way to describe colors.
- Tristimulus values can be transformed using linear transformations to account for different sets of primaries.
- Non-linear transformations can be used to describe tristimulus values in terms of human-centric qualities such as hue and saturation.