Color Image Processing

CS 663, Ajit Rajwade

Pouring in color

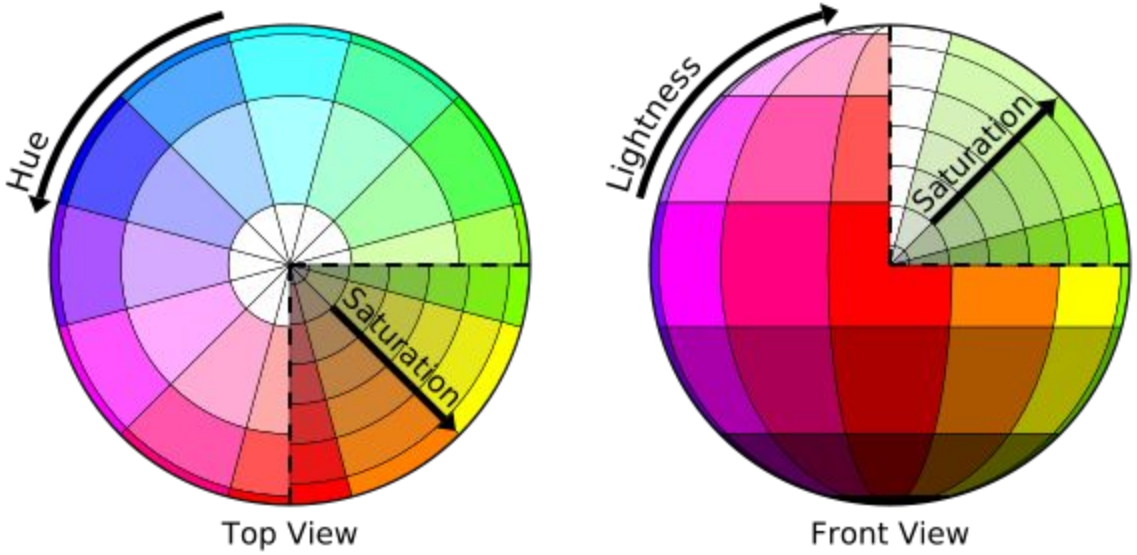
- **Grayscale image:** 2D array of size M x N containing scalar intensity values (graylevels).

- **Color image:** typically represented as a 3D array of size M x N x 3 again containing scalar values. But each pixel location now has three values – called as **R(red)**,**G(green)**, **B(blue)** intensity values.

- Most file formats store color images based on this representation. It is also the default representation for display of color images.

Questions

- What are RGB? How are red, green, blue determined?
- Are there other ways of representing color?
- How do you distinguish between different intensities of the same color (**shades**)? Between varying levels of whiteness in a color (**tints**)?



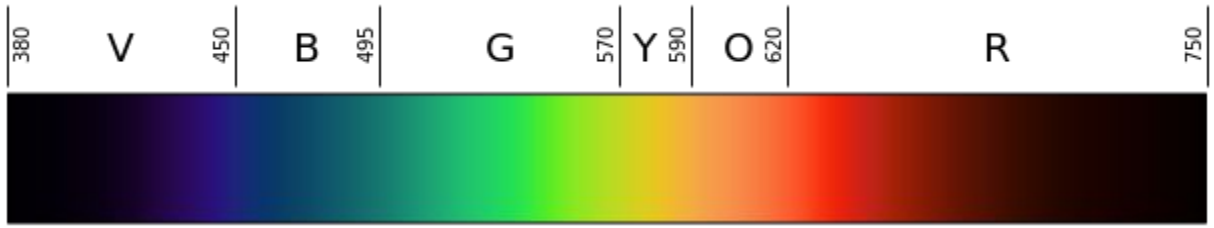
http://en.wikipedia.org/wiki/Tints_and_shades

More questions

- How would you define an edge in a color image?
- How do you smooth color images?
- What are the advantages of color images over grayscale images?
- What do we know about human color perception?
- What do color-based optical illusions teach us?
- Why do we consider only 3 channels (i.e. RGB)? Are there images with more channels? Where are they used?

Color perception and physics

- Human perception of color is not fully understood, but there are some well understood physical principles.
- The color of light is defined by its constituent **wavelengths** (inverse of frequency).
- The visible part of the electromagnetic spectrum lies between **450 nm (violet) to 700 nm (red)**.



Ultraviolet

Infrared

Color	Wavelength
violet	380–450 nm
blue	450–495 nm
green	495–570 nm
yellow	570–590 nm
orange	590–620 nm
red	620–750 nm

Color physics

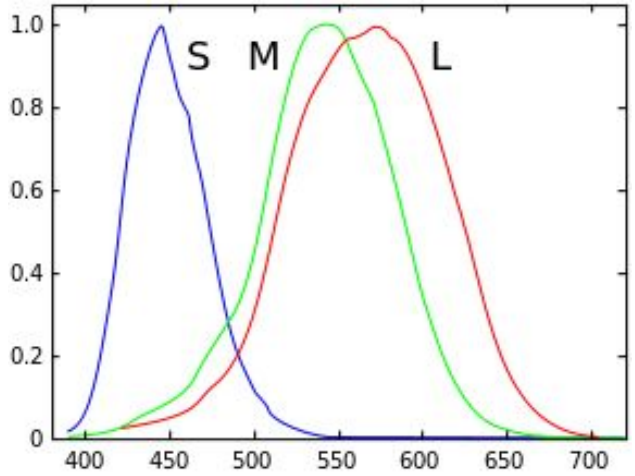
- White light is a blend of several wavelengths of light, which get separated by “dispersive elements” such as prisms.
- Objects which reflect light that is balanced in several visible wavelengths appear “**white**”.
- Objects which reflect light in a narrow range of wavelengths appear “**colored**” (example: green objects reflect light between 500 to 560 nm).
- No color starts or ends abruptly at a particular wavelength – the transitions are smooth.

Human color perception

- The human retina has two types of receptor cells that respond to light – the **rods** and the **cones**.
- The **rods** work in the **low-light** regime and are responsible for **monochromatic vision**.
- The **cones** respond to brighter light, and are responsible for **color** perception.
- There are around 5-7 million cones in a single retina.

Human color perception

- There are 3 types of cones. Each type responds differently to light of different wavelengths: **L** (responsive to long wavelengths, i.e. red), **M** (medium wavelengths, i.e. green) and **S** (short wavelengths, i.e. blue).



Response sensitivity functions for LMS cells

Yellow color: L is stimulated a bit more than M and S is not stimulated
Red: L is stimulated much more than M and S is not stimulated
Violet: S is stimulated, M and L are not

Color-blindness: absence of one or more of the three types of cones

Human color perception

- Consider a beam of light striking the retina. Let its spectral intensity as a function of wavelength λ be given as I(λ).
- The three types of cone cells re-weigh the spectral intensity and produce the following response:

$$a_L = \int I(\lambda) c_L(\lambda) d\lambda \approx \sum_{\lambda} I(\lambda) c_L(\lambda)$$

$$a_M = \int I(\lambda) c_M(\lambda) d\lambda \approx \sum_{\lambda} I(\lambda) c_M(\lambda)$$

$$a_S = \int I(\lambda) c_S(\lambda) d\lambda \approx \sum_{\lambda} I(\lambda) c_S(\lambda)$$

$$\begin{pmatrix} a_L \\ a_L \\ a_L \end{pmatrix} = \begin{pmatrix} c_L \\ c_M \\ c_S \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \\ . \\ . \\ I_{N_\lambda} \end{pmatrix} = CI$$

Vectors of length N_λ

C is a matrix of size 3 x N_λ and I is a vector of N_λ elements

Human color perception

- The colors R,G,B are called primary colors – their corresponding wavelengths are 435, 546, 700 nm respectively.
- These values were standardized by CIE (International Commission on Illumination – Commission Internationale de l'Eclairage).

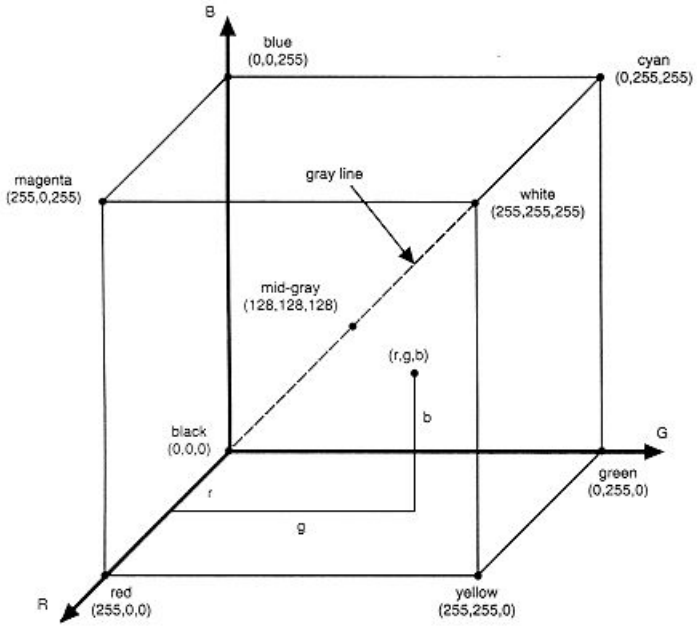
Display systems (CRT/LCD)

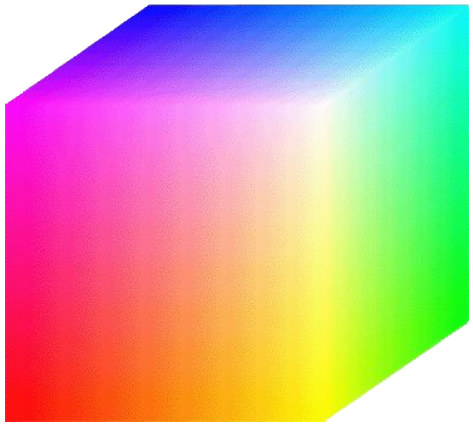
- The interior of a cathode ray tube (CRT) contains an array of triangular dot patterns (triads) containing electron-sensitive phosphor. Each dot in the triad produces light in one of the three primary colors based on the intensity of that primary color.
- Thus the three primary colors get mixed in different proportions by the color sensitive cones of the human eye to perceive different colors.
- Though the electronics of an LCD system is different from CRT, the color display follows similar principles.

Color Models (Color Spaces)

- A purpose of **color model** is to serve as a method of representing color.
- Some color models are oriented towards **hardware** (eg: monitors, printers), others for applications involving **color manipulation**.
- **Monitors**: RGB, **Printers**: CMY, **human perception**: HSI, **efficient compression and transmission**: YCbCr.

RGB color model





- Defines a cartesian coordinate system for colors – in terms of R,G,B axes.

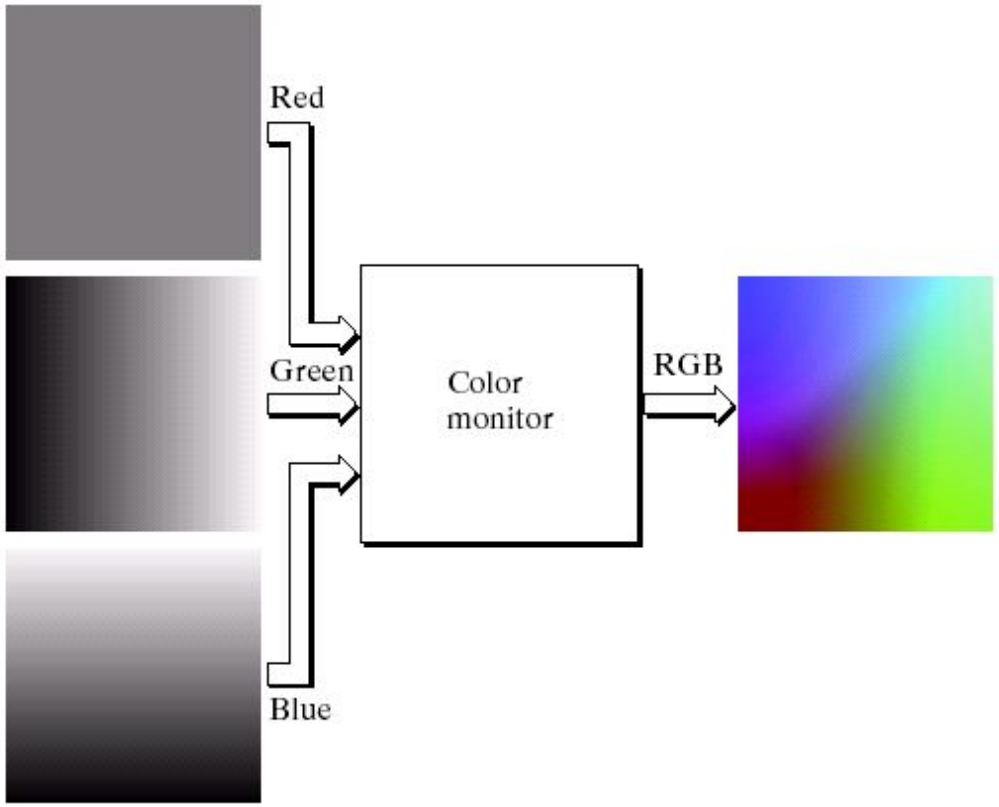
- Images in the RGB color model consist of three component images, one for each primary color.

- When an RGB image is given as input to a display system, the three images combine to produce the composite image on screen.

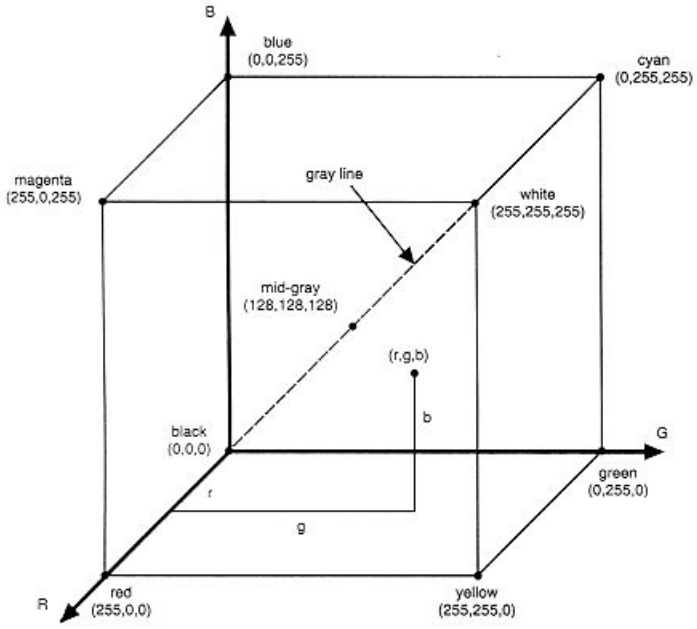
- Typically, an 8 bit integer is used to represent the intensity value in each channel, giving rise to (2^8)^3 = 1.677 x 10^7 colors.

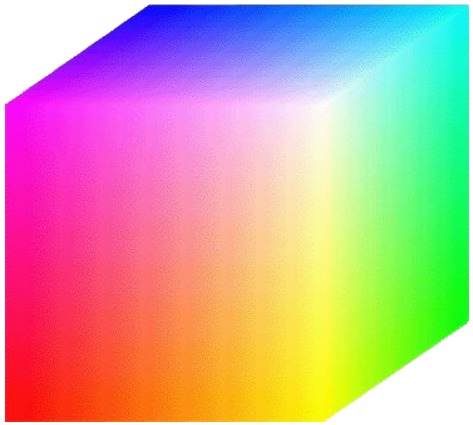
a
b

FIGURE 6.9
(a) Generating the RGB image of the cross-sectional color plane (127, G, B). (b) The three hidden surface planes in the color cube of Fig. 6.8.



CMY(K) color space





•The colors cyan, magenta and yellow are **"opponents"** of red, green and blue respectively, i.e. cyan and red lie on diagonally opposite corners of the RGB cube, i.e. C = 255-R, M = 255-G, Y = 255-B

•Cyan, magenta and yellow are called **secondary colors of light**, or **primary colors of pigments**. A cyan colored surface illuminated with white light will **not allow the reflection of red light**. Likewise for magneta and green, yellow and blue.

•CMY are the colors of ink pigments used in printing industry. Color printing is a subtractive process – the ink subtracts certain color components from white light.

•For purposes of display, white is the full combination of RGB and black is the absence of light. For purposes of printing, **white is the absence of any printing**, and **black is the full combination of CMY**.

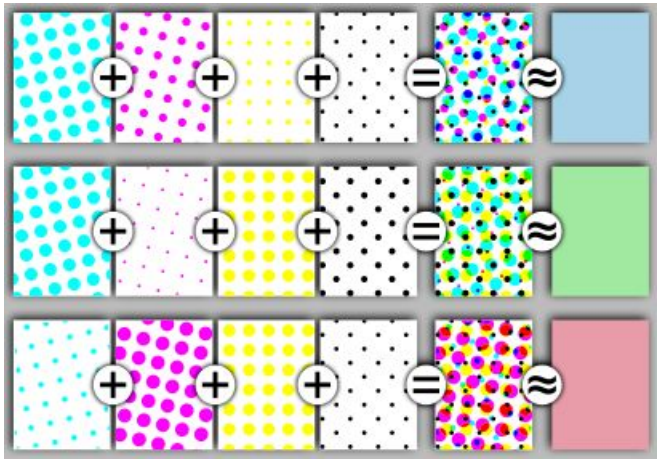
See also:
http://home.comcast.net/~rtruscio/colorsys.htm

CMY(K) color space

- The printer puts down dots (of different **sizes**, **shapes**) of CMY colors with tiny **spacing** (of different **widths**) in between. The spacing is so tiny that our eye perceives them as a single solid color (optical illusion!). This process is called **color half-toning**.
- While black is a full combination of CMY, it is printed on paper using a separate black-color ink (to save costs). This is the ‘K’ of the CMYK model.

Color half-toning

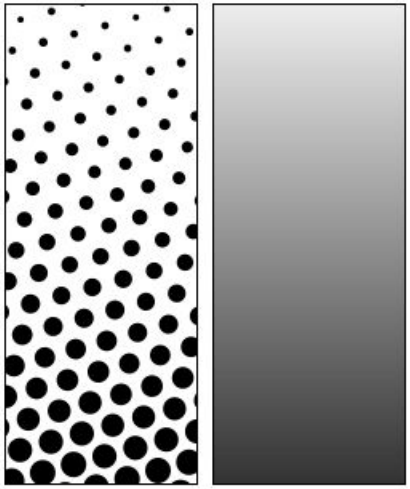
http://en.wikipedia.org/wiki/Halftone



Three examples of color half-toning with CMYK separations. From left to right: The cyan separation, the magenta separation, the yellow separation, the black separation, the combined halftone pattern and finally how the human eye would observe the combined halftone pattern from a sufficient distance.

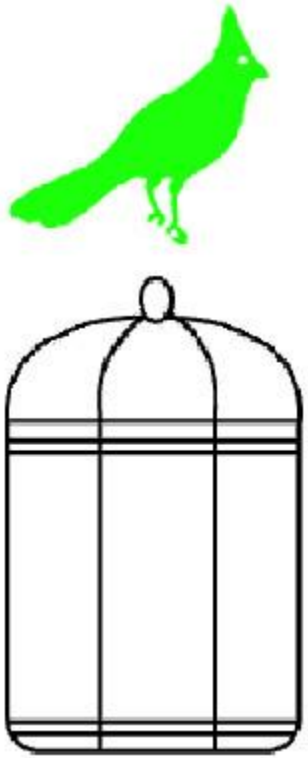
Digression: Gray-scale half-toning

http://en.wikipedia.org/wiki/Halftone



Left: Halftone dots. Right: How the human eye would see this sort of arrangement from a sufficient distance.

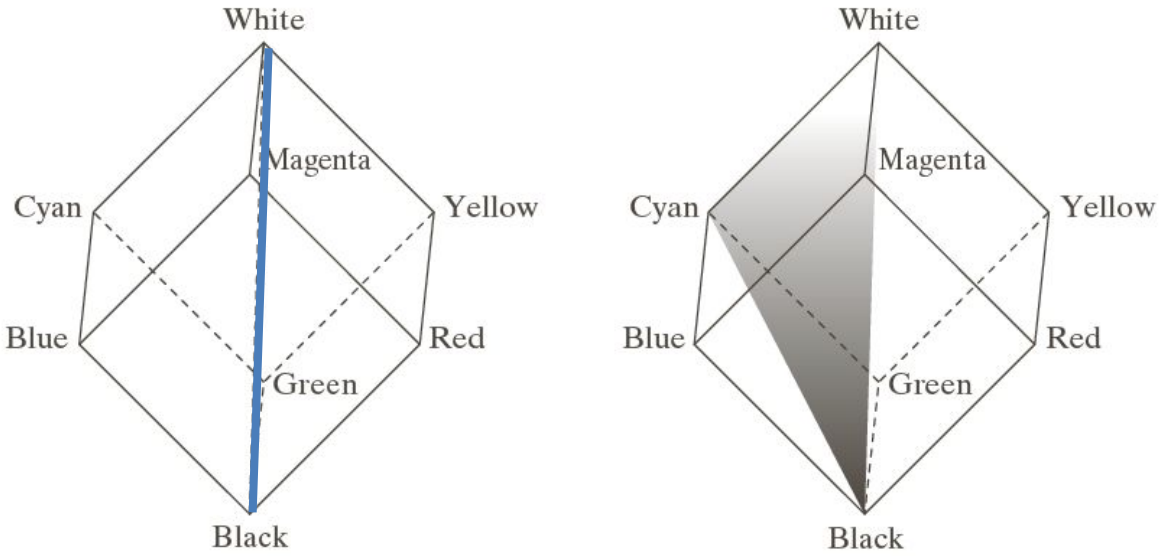
Digression: negative after-images!



http://thebrain.mcgill.ca/flash/a/a_02/a_02_p/a_02_p_vis/a_02_p_vis.html#

HSI color space

- RGB, CMY are not intuitive from the point of view of human perception/description.
- We don’t think naturally of colors in the form of combinations of RGB.
- We tend of think of color as the following components: **hue** (the “inherent/pure” color – red, orange, purple, etc.), **saturation** (the amount of white mixed in the color, i.e. pink versus magenta), **intensity** (the amount of black mixed in the color, i.e. dark red versus bright red).



a b

FIGURE 6.12 Conceptual relationships between the RGB and HSI color models.

• **Intensity increases** as we move from black to white on the intensity line.

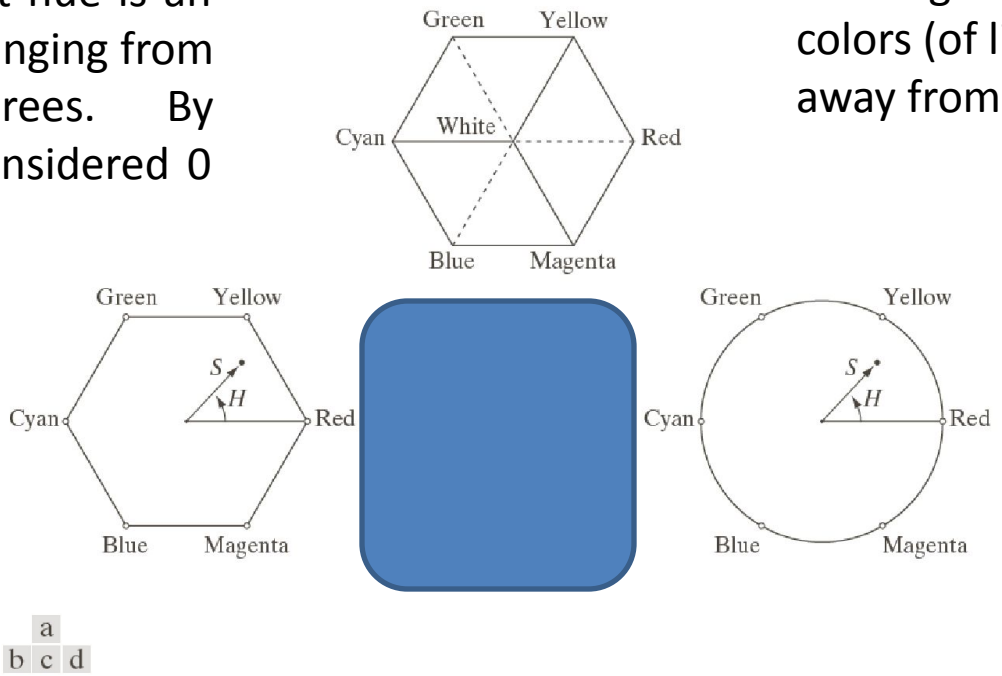
• Consider a plane perpendicular to the intensity line (in 3D). **Saturation of a color increases** as we move on that plane away from the point where the plane and the intensity line intersect.

• How to determine hue? Pick any point (e.g. yellow) in the RGB cube, and draw a triangle connecting that point with the white point and black point. All points inside or on this triangle have the same hue. Any such point would be a color corresponding to a convex combination of yellow, black and white, i.e. of the form **a** x yellow + **b** x black + **c** x white, where **a**, **b**, **c** are non-negative and sum to 1. **By rotating this triangle about the intensity axis**, you will get different hues.

HSI space

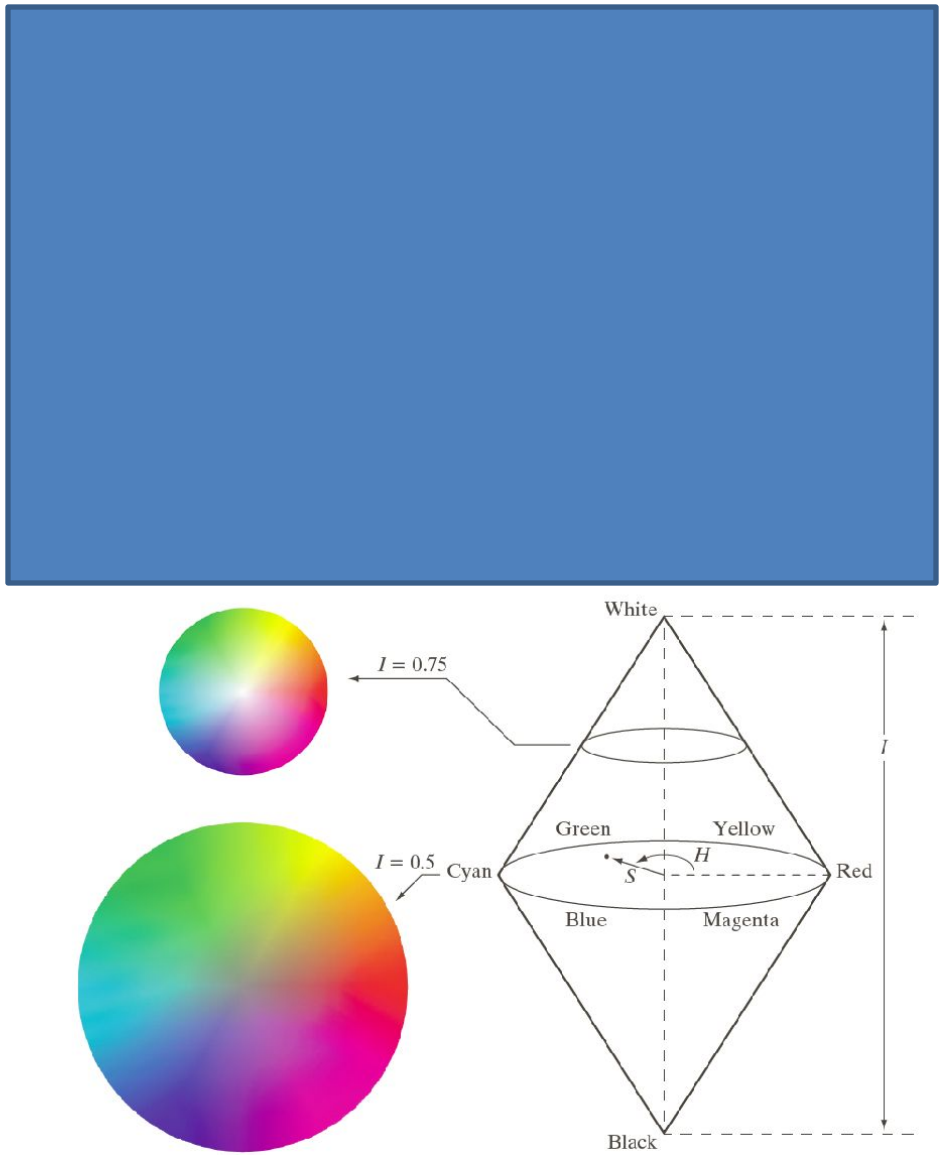
By rotating the triangle about the intensity axis, you will get different hues. In fact hue is an **ANGULAR** quantity ranging from 0 to 360 degrees. By convention, red is considered 0 degrees.

Primary colors are separated by 120 degrees. The secondary colors (of light) are 60 degrees away from the primary colors.



a
b c d

FIGURE 6.13 Hue and saturation in the HSI color model. The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.



a
b

FIGURE 6.14 The HSI color model based on circular color planes. The circles are perpendicular to the vertical intensity axis.

To be very accurate, this HSI spindle is actually hexagonal. But it is approximated as a circular spindle for convenience. This approximation does not alter the notion of hue or intensity and has an insignificant effect on the saturation.

RGB to HSI conversion

- Conversion formulae are obtained by making the preceding geometric intuition more precise:

$$\theta = \cos^{-1}\left(\frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right),$$

$$\text{hue} = h = \theta \text{ if } B \le G$$

$$= 2\pi - \theta \text{ if } B > G$$

$$S = 1 - \frac{3\min(R,G,B)}{R+G+B}$$

$$I = \frac{R+G+B}{3}$$

Refer to textbook for formulae to convert back from HSI to RGB

HSI and RGB

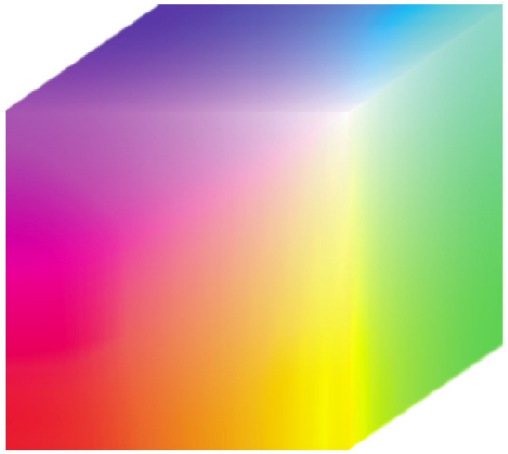
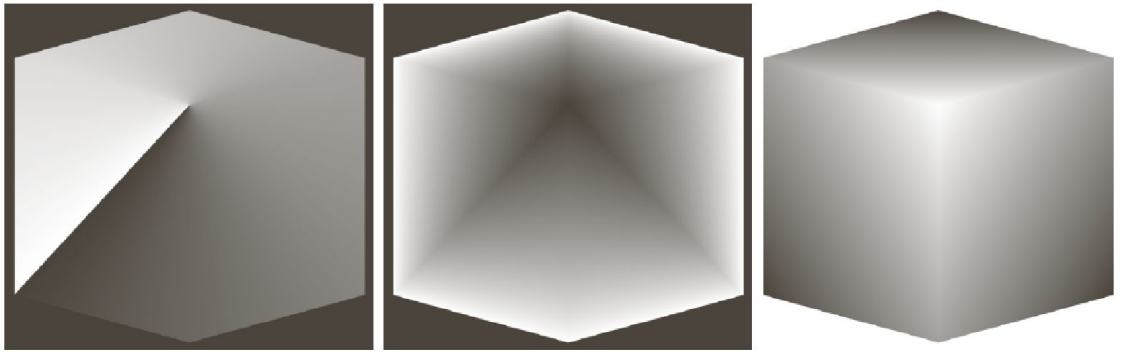
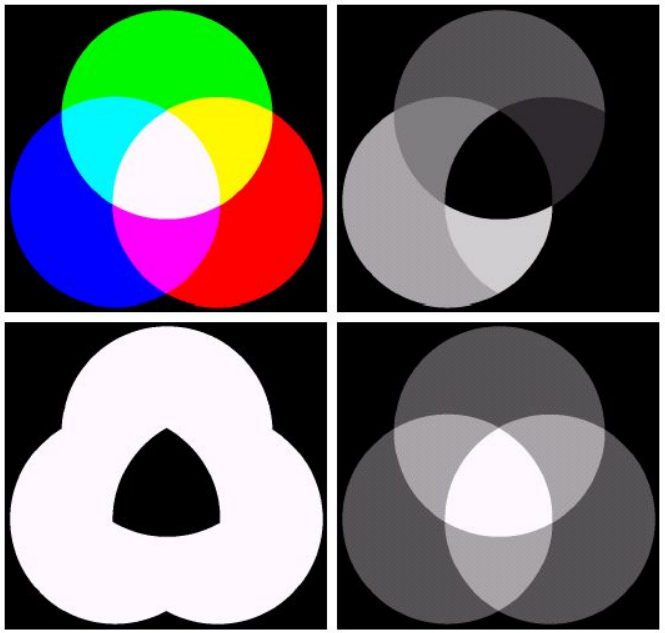


FIGURE 6.8 RGB 24-bit color cube.



a b c

FIGURE 6.15 HSI components of the image in Fig. 6.8. (a) Hue, (b) saturation, and (c) intensity images.



a b
c d

FIGURE 6.16 (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity.

Practical use of hue

$$\theta = \cos^{-1}\left(\frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right),$$

$$\text{hue} = h = \theta \ \text{ if } B \le G$$

$$= 2\pi - \theta \ \text{ if } B > G$$

Hue is invariant to:

- Scaling of R,G,B
- Constant offsets added to R,G,B

What does this mean physically?

Practical use of hue

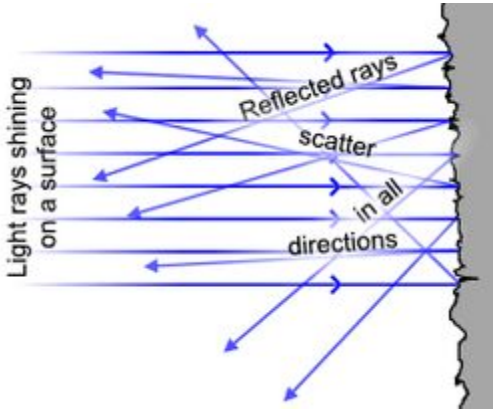
- To understand this, we need to understand a model which tells you the what color is observed at a particular point on a surface of an object illuminated by one or more light sources.
- This color is given by:

$$I^{(C)} = I^{(C)}_{ambient} + I^{(C)}_{diffuse} + I^{(C)}_{specular}, C \in \{R, G, B\}$$

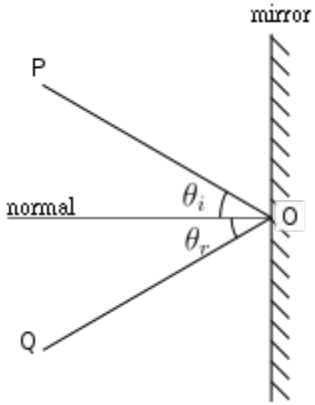
Ambient light (say due to sunlight): constant effect on all points of the object's surface

Diffuse reflection of light from a directed source off a **rough** surfaces: varies from point to point on a surface

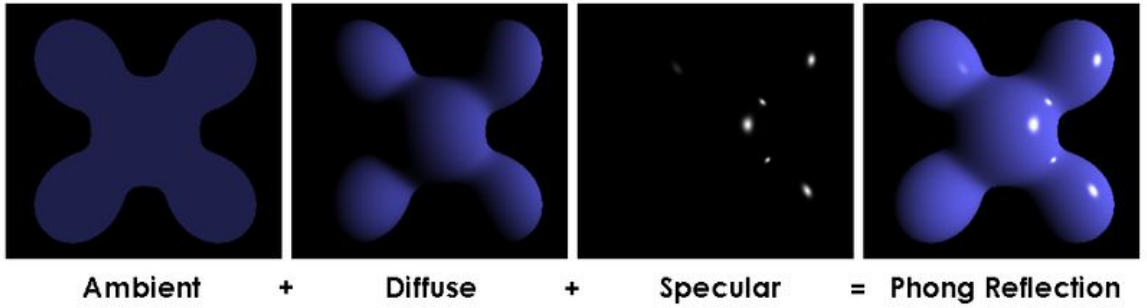
Reflection from **shiny** surface: varies from point to point on a surface



Diffuse reflection from an irregular surface



Specular reflection



- Diffuse reflection from a rough surface: "diffuse" means that incident light is reflected in all directions.
- Specular reflection: part of the surface acts like a mirror, the incident light is reflected only in particular directions

Diffuse reflection Diffuse + specular reflection

$$I^{(C)} = I^{(C)}_{ambient} + I^{(C)}_{diffuse} + I^{(C)}_{specular}, C \in \{R, G, B\}$$

$$= k_a I_a + k_d^{(C)} L(\hat{n} \bullet \hat{s}) + k_s L(\hat{r} \bullet \hat{v})^{\alpha}$$

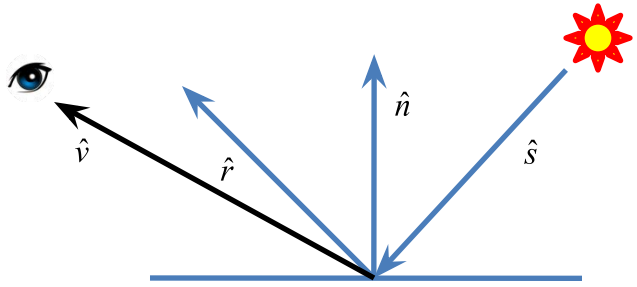
L=Strength of white light source, k_a,k_d,k_s: surface reflectivity (fraction of incident light that is reflected off the surface)

Vector normal to the surface at a point

Lighting direction

Viewing direction

Direction of reflected light



For shiny surfaces, α is large.

Practical use of hue

- The **ambient** and **specular** components are assumed to be the same across RGB (neutral reflection model). So they get subtracted out when computing R-G,G-B,B-R. Hence **hue is invariant to specular reflection**!
- Notice: hue is independent of **strength of lighting** (why?), **lighting direction** (why?) and **viewing direction** (why?).
- This makes hue useful in object detection and object recognition or in applications such as detection of faces/foliage in color images.
- Hue is thus said to be an **"illumination invariant"** feature.

Food for thought

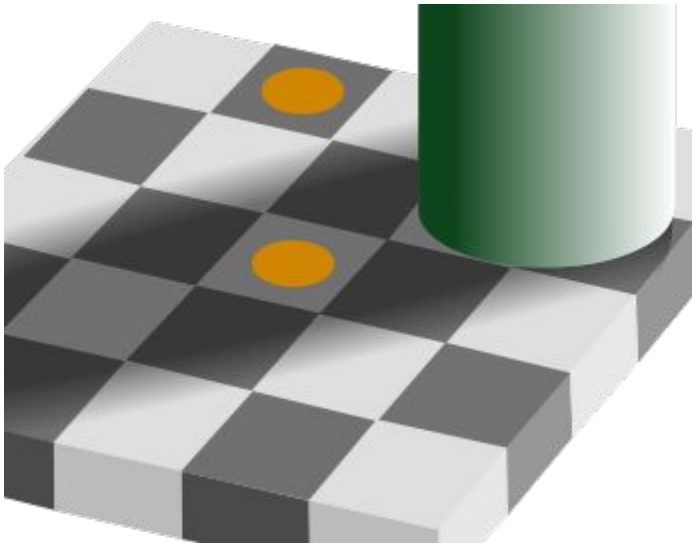
- We've heaped praises on hue, all along. Any ideas on its demerits?
- Suppose we define the following quantities (r,g,b) [the chromaticity vector] derived from RGB:

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

Is the chromaticity vector also an illumination invariant feature? How does it compare to hue?

Digression: Playing with color: seeing is not (!) believing

http://thebrain.mcgill.ca/flash/a/a_02/a_02_p/a_02_p_vis/a_02_p_vis.html#



http://en.wikipedia.org/wiki/Optical_illusion

Operations on color images

- Color image histogram equalization
- Color image filtering
- Color edge detection

Histogram equalization

- Method 1: perform histogram equalization on RGB channels separately.
- Method 2: Convert RGB to HSI, histogram equalize the intensity, convert back to RGB.
- Method 1 may cause alterations in the hue – which is undesirable.
- Method 2 will change only the intensity, leaving hue and saturation unaltered. It is the preferred method.






Top row: original images
Middle row: histogram equalization channel by channel
Bottom row: histogram equalization on intensity (of HSI) and conversion back to RGB

Color image smoothing: bilateral filtering

- Remember the bilateral filter (HW2): an edge-preserving filter for grayscale images.
- It smoothes the image based on local weighted combinations driven by difference between spatial coordinates and intensity values.

$$I(x,y)=\frac{\sum_{(i,j)\in N(x,y)} I(i,j)w(i,j)}{\sum_{(i,j)\in N(x,y)} w(i,j)},\ w(i,j)=\exp\left(-\left[\frac{(i-x)^2+(j-y)^2}{\sigma_s^2}+\frac{(I(i,j)-I(x,y))^2}{\sigma_I^2}\right]\right),$$

$N(x,y)=$ small neighborhood (usually square) centered at *(x,y)*

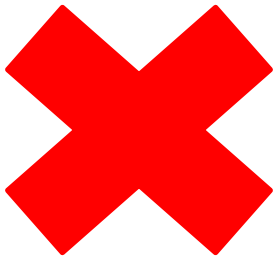
Bilateral filtering for color images

- You can filter each channel separately, i.e.

$$I_C(x,y) = \frac{\sum_{(i,j)\in N(x,y)} I_C(i,j) w_C(i,j)}{\sum_{(i,j)\in N(x,y)} w_C(i,j)}, \quad w_C(i,j) = \exp\left(-\left[\frac{(i-x)^2+(j-y)^2}{\sigma_s^2} + \frac{(I_C(i,j)-I_C(x,y))^2}{\sigma_I^2}\right]\right),$$

$N(x,y)$ = small neighborhood (usually square) centered at *(x,y)*,

$C \in \{R,G,B\}$



Bilateral filtering for color images

- Or you can filter the three channels in a coupled fashion, i.e. the smoothing weights are same for all three channels and they are derived using information from all three channels.

$$I_C(x,y) = \frac{\sum_{(i,j)\in N(x,y)} I_C(i,j)w(i,j)}{\sum_{(i,j)\in N(x,y)} w(i,j)}, w(i,j) = \exp\left(-\left[\frac{(i-x)^2+(j-y)^2}{\sigma_s^2} + \frac{\sum_{C\in\{R,G,B\}}(I_C(i,j)-I_C(x,y))^2}{\sigma_I^2}\right]\right),$$

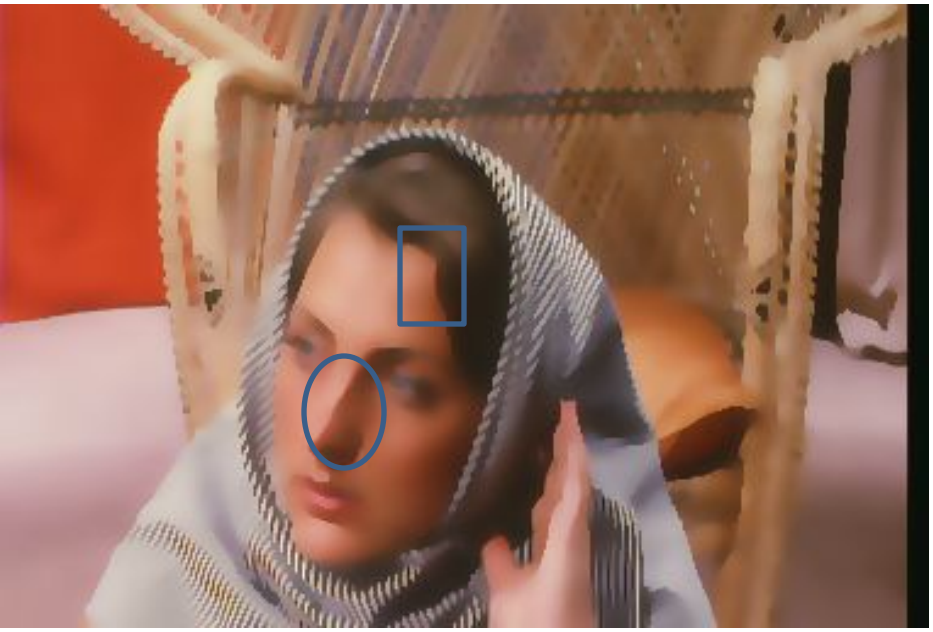
$N(x,y)$ = small neighborhood (usually square) centered at *(x,y)*

What's wrong with separate channel bilateral filtering?

Channel by channel: Color artifacts around edges. RGB channels are highly inter-dependent – you shouldn't treat them as independent.

Separate channel

Coupled

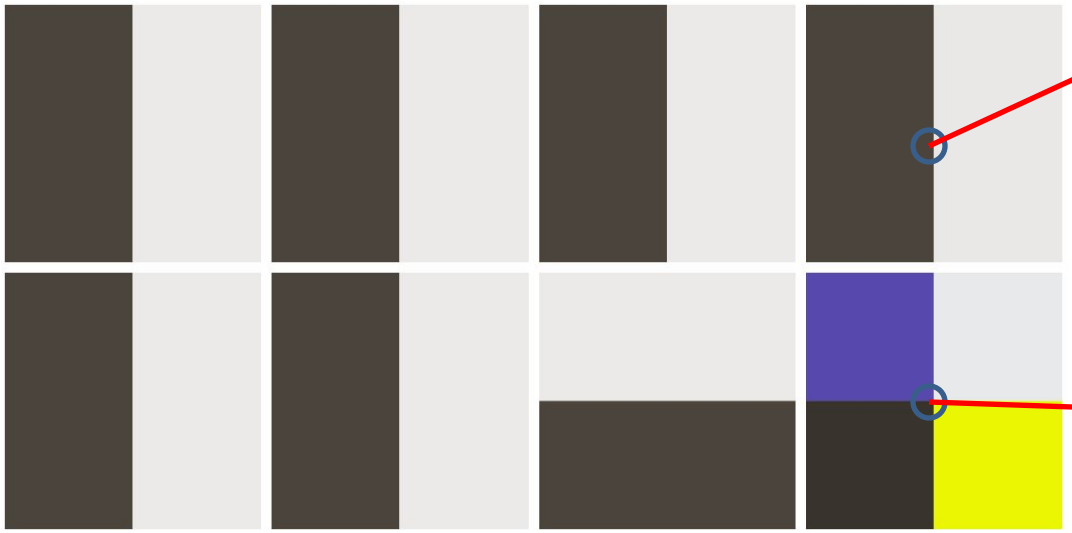


Color Edges

- A color (RGB) image will have three gradient vectors – one for each channel.
- We could compute edges separately for each channel.
- Option: Combine (add) channel-per-channel edges together to get a composite edge image. Not a good one? Why (see next slide)

Color Edges

- Problem: the two circled points have the same edge strength (mathematically), though one *appears* to be a stronger edge.



(Rx,Gx,Bx) = (255,255,255),
(Ry,Gy,By) = (0,0,0),
Rx^2 + Gx^2 +Bx^2 +
Ry^2+Gy^2+By^2 = 3*255^2

(Rx,Gx,Bx) = (255,255,0),
(Ry,Gy,By) = (0,0,255),
Rx^2 + Gx^2 +Bx^2 +
Ry^2+Gy^2+By^2 = 3*255^2

a b c d
e f g h

FIGURE 6.45 (a)–(c) *R*, *G*, and *B* component images and (d) resulting RGB color image. (e)–(g) *R*, *G*, and *B* component images and (h) resulting RGB color image.

Color Gradient/Edge

- To find the color gradient, we want to ask the question: **along which direction in XY space is the total magnitude of change in intensity the maximum?**
- The squared change in intensity in a direction (cos Θ, sin Θ) is given by (square of the directional derivative of the intensity):

$$E(\theta) = (R_x \cos\theta + R_y \sin\theta)^2 + (G_x \cos\theta + G_y \sin\theta)^2 + (B_x \cos\theta + B_y \sin\theta)^2$$

$$= (R_x^2 + G_x^2 + B_x^2)\cos^2\theta + (R_y^2 + G_y^2 + B_y^2)\sin^2\theta + 2\sin\theta\cos\theta(R_x R_y + G_x G_y + B_x B_y)$$

- We want to maximize this w.r.t Θ. Take derivative with respect to Θ and set it to zero.

Color Gradient/Edge

- This gives the color gradient direction which makes an angle Θ w.r.t. the X axis, given by:

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2(R_x R_y + G_x G_y + B_x B_y)}{(R_x^2 + G_x^2 + B_x^2) - (R_y^2 + G_y^2 + B_y^2)}\right)$$

- For a grayscale image, this turns out to be

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2I_x I_y}{I_x^2 - I_y^2}\right) = \frac{1}{2}\tan^{-1}\left(\frac{\frac{2I_y}{I_x}}{1-\left(\frac{I_y}{I_x}\right)^2}\right) = \tan^{-1}\left(\frac{I_y}{I_x}\right)$$

Color Gradient/Edge

- Consider

$$E(\theta) = (R_x^2 + G_x^2 + B_x^2)\cos^2\theta + (R_y^2 + G_y^2 + B_y^2)\sin^2\theta + 2\sin\theta\cos\theta(R_xR_y + G_xG_y + B_xB_y)$$

$$= \begin{pmatrix}\cos\theta & \sin\theta\end{pmatrix}\begin{pmatrix} R_x^2 + G_x^2 + B_x^2 & R_xR_y + G_xG_y + B_xB_y \\ R_xR_y + G_xG_y + B_xB_y & R_y^2 + G_y^2 + B_y^2 \end{pmatrix}\begin{pmatrix}\cos\theta \\ \sin\theta\end{pmatrix}$$

Local color gradient matrix

- It turns out that the Θ (i.e. the **color gradient**) we derived is given by the eigenvector of this matrix corresponding to the **larger eigenvalue**. The direction perpendicular to it (i.e. the eigenvector corresponding to the **smaller eigenvalue**) is the **color edge**.

a b
c d

FIGURE 6.46
(a) RGB image.
(b) Gradient computed in RGB color vector space.
(c) Gradients computed on a per-image basis and then added.
(d) Difference between (b) and (c).

a b c

FIGURE 6.47 Component gradient images of the color image in Fig. 6.46. (a) Red component, (b) green component, and (c) blue component. These three images were added and scaled to produce the image in Fig. 6.46(c).

PCA on RGB values

- Suppose you take N color images and extract RGB values of each pixel (3 x 1 vector at each location).
- Now, suppose you build an eigenspace out of this – you get 3 eigenvectors, each corresponding to 3 different eigenvalues.

PCA on RGB values

- The eigenvectors will look typically as follows:

  0.5952 0.6619 0.4556

  0.6037 0.0059 -0.7972

  0.5303 -0.7496 0.3961

- Exact numbers are not important, but the first eigenvector is like an average of RGB. It is called as the **Luminance Channel** (Y). It is similar to the intensity in the HSI space.

PCA on RGB values

- The second eigenvector is like Y-B, and the third is like Y-G. These are called as the **Chrominance Channels**.
- The Y-Cb-Cr color space is related to this PCA-based space (though there are some details in the relative weightings of RGB to get Luminance and Chrominance – denoted by Cb and Cr).
- The values in the three channels Y, Cb and Cr are decorrelated, similar to the values projected onto the PCA-based channels.

PCA on RGB values

- The luminance channel (Y) carries most information from the point of view of human perception, and the human eye is less sensitive to changes in chrominance.
- This fact can be used to assign coarser quantization levels (i.e. fewer bits) for storing or transmitting Cb and Cr values as compared to the Y channel. This improves the compression rate.
- The JPEG standard for color image compression uses the YCbCr format. For an image of size M x N x 3, it stores Y with full resolution (i.e. as an M x N image), and Cb and Cr with 25% resolution, i.e. as M/2 x N/2 images.

R channel

B channel

G channel

Image containing eigencoefficient value corresponding to 1st eigenvector (with maximum eigenvalue)

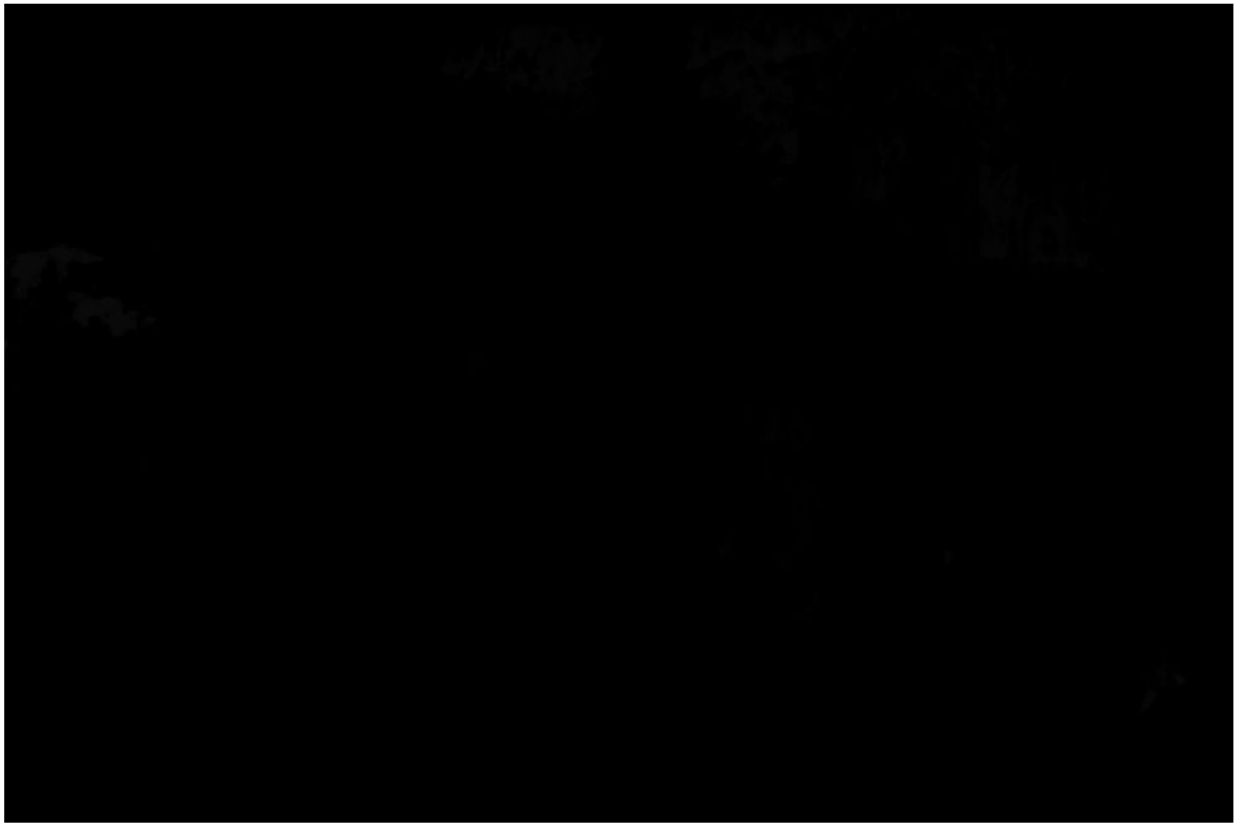
Image containing eigencoefficient value corresponding to 2nd eigenvector (with second largest eigenvalue)

Image containing eigencoefficient value corresponding to 3rd eigenvector (with least eigenvalue)

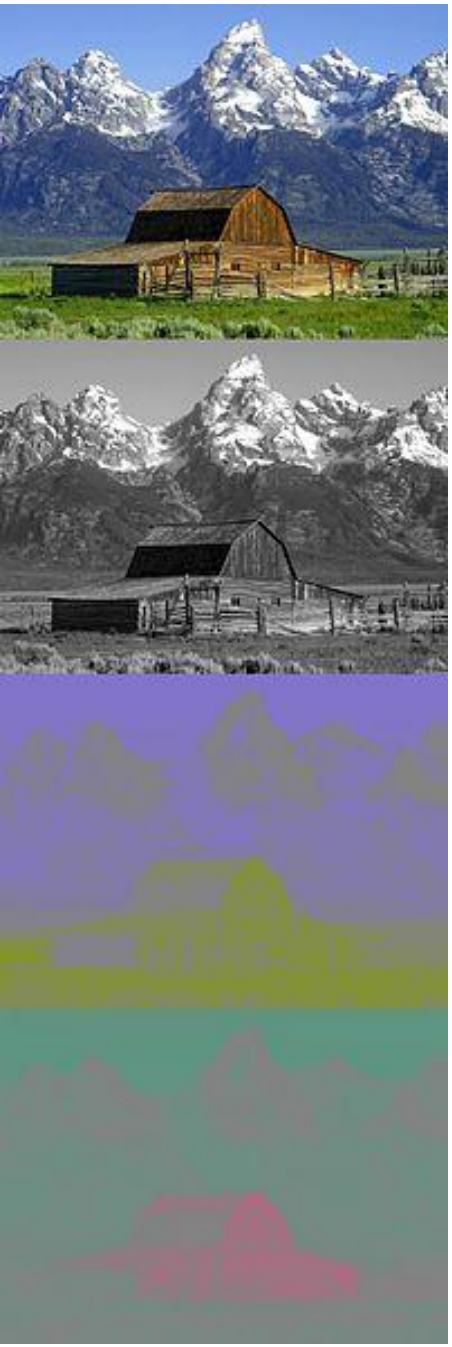
The variances of the three eigen-coefficient values:
8411, 159.1, 71.7

Y channel

Cb channel



Cr channel



RGB and its corresponding Y, Cb, Cr channels

$$Y = 16 + (65.481R + 128.553G + 24.966B)$$

$$C_B = 128 + (-37.79R - 74.203G + 112B)$$

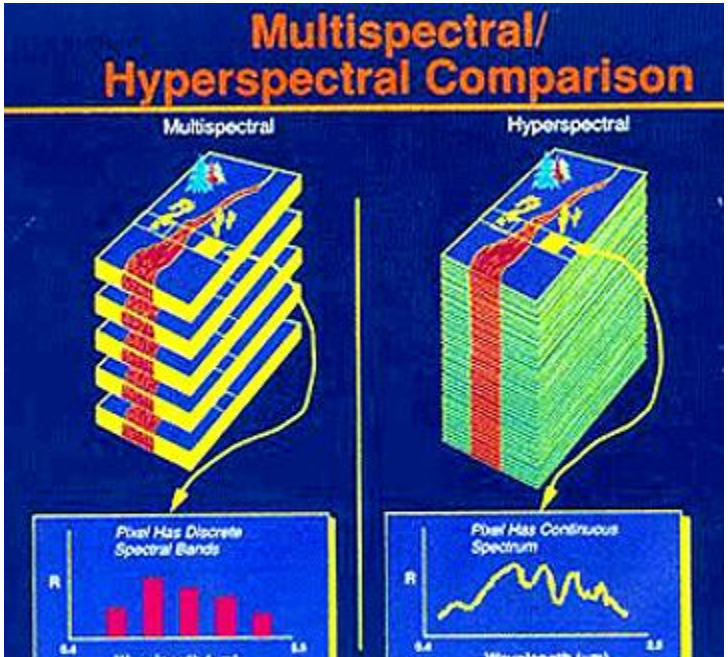
$$C_R = 128 + (112R - 93.786G - 18.214B)$$

Beyond color: Hyperspectral images

- Images of the form M x N x L, where L is the number of channels. L can range from 30 to 30,000 or more.
- Finer division of wavelengths than possible in RGB!
- Can contain wavelengths in the infrared or ultraviolet regime.

Sources of confusion ☺️

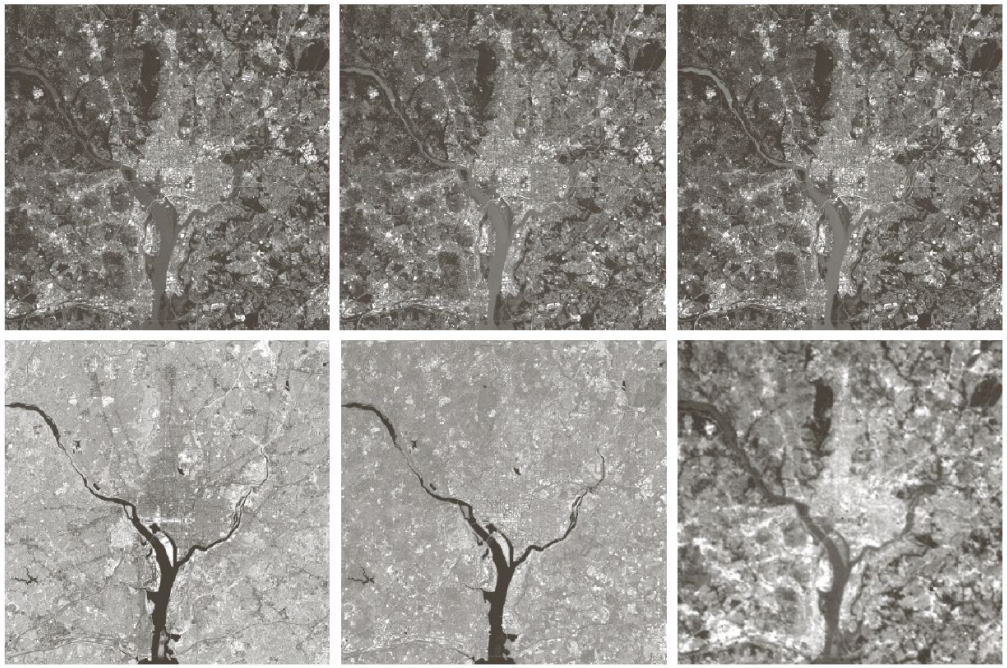
- Hyperspectral images are abbreviated as HSI!
- Hyperspectral images are different from multispectral images. The latter contain few, discrete and discontinuous wavelengths. The former contain many more wavelengths with continuity.



Beyond color: Hyperspectral images

- Widely used in remote sensing (satellite images) – often different materials/geographical entities (soil, water, vegetation, concrete, landmines, mountains, etc.) can be detected/classified by spectral properties.
- Also used in chemistry, pharmaceutical industry and pathology for classification of materials/tissues.

Example multispectral image with 6 bands



a b c
d e f

FIGURE 11.38 Multispectral images in the (a) visible blue, (b) visible green, (c) visible red, (d) near infrared, (e) middle infrared, and (f) thermal infrared bands. (Images courtesy of NASA.)

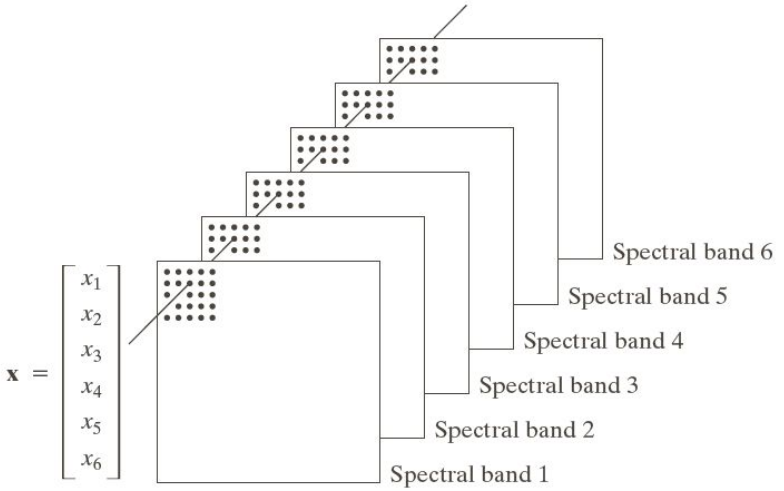
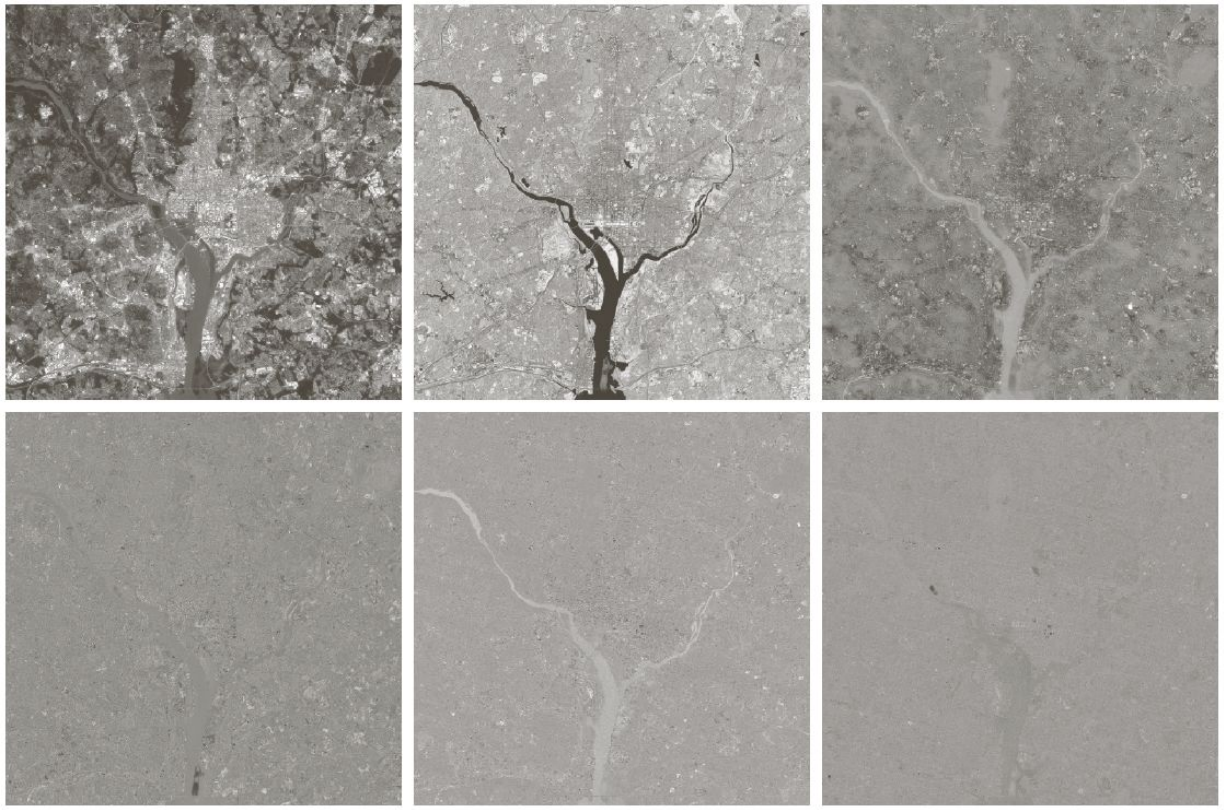


FIGURE 11.39 Formation of a vector from corresponding pixels in six images.

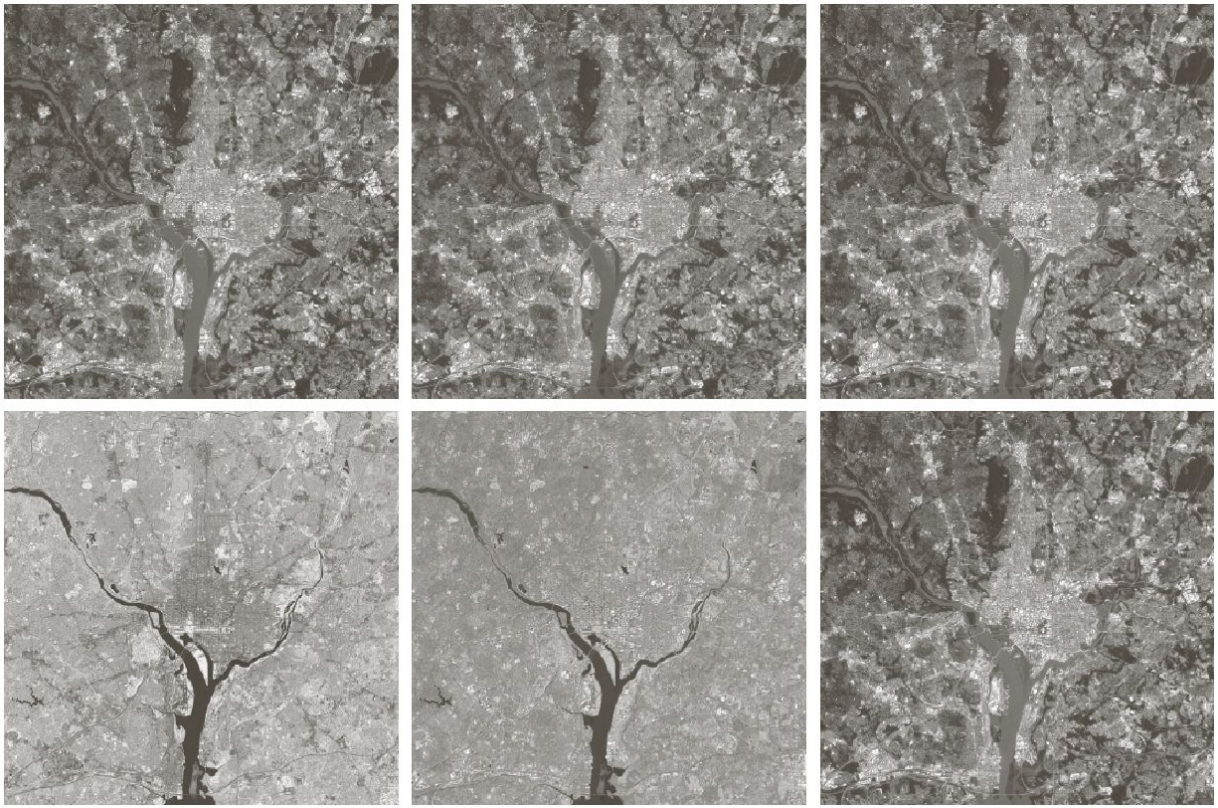


a b c
d e f

FIGURE 11.40 The six principal component images obtained from vectors computed using Eq. (11.4-6). Vectors are converted to images by applying Fig. 11.39 in reverse.

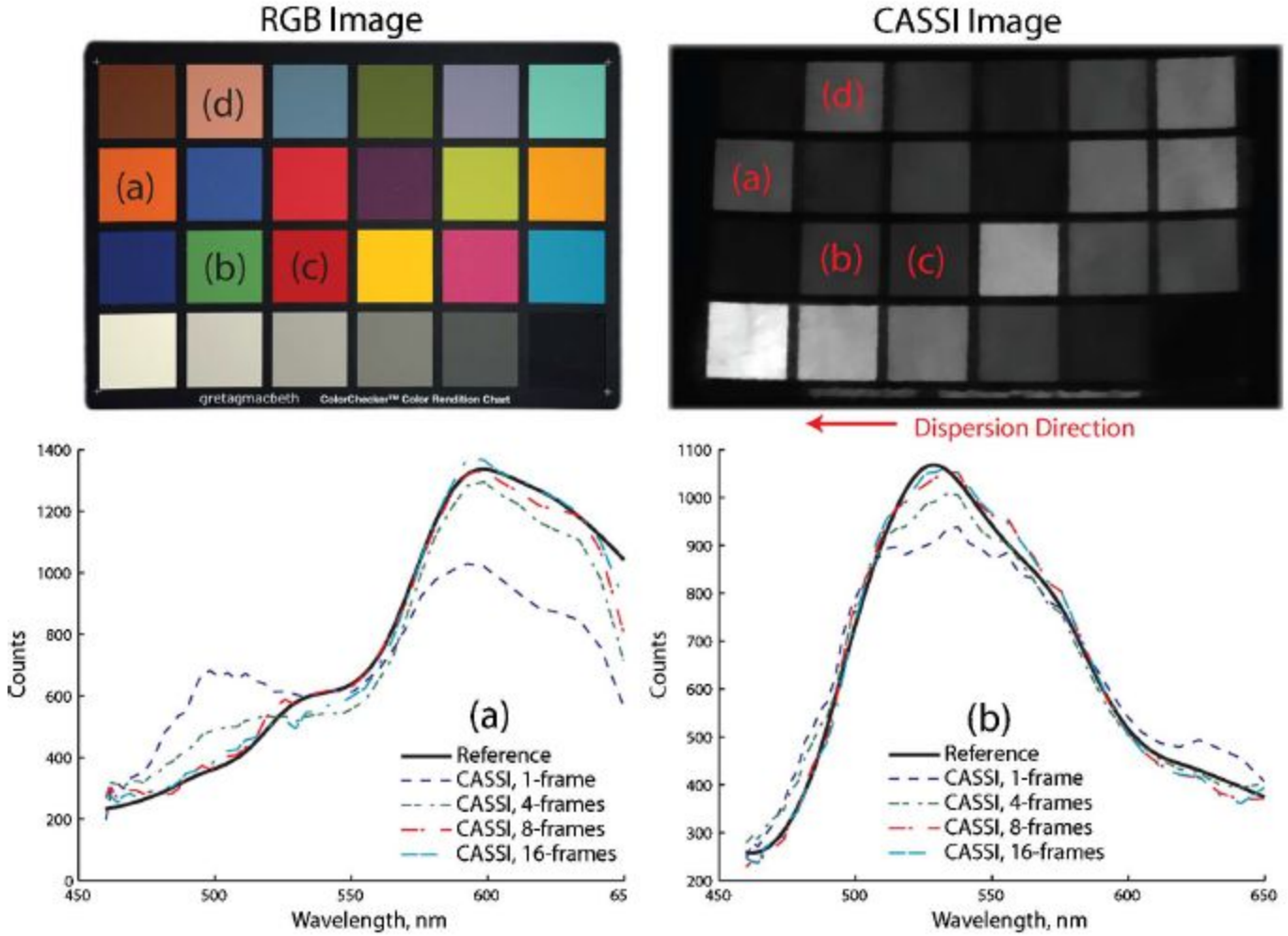
λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
10344	2966	1401	203	94	31

TABLE 11.6
Eigenvalues of the covariance matrices obtained from the images in Fig. 11.38.



a b c
d e f

FIGURE 11.41 Multispectral images reconstructed using only the two principal component images corresponding to the two principal component images with the largest eigenvalues (variance). Compare these images with the originals in Fig. 11.38.


RGB Image
CASSI Image
(d)
(a)
(b)
(c)
gretagmacbeth
Dispersion Direction
Counts
Wavelength, nm
(a)
(b)
Reference
CASSI, 1-frame
CASSI, 4-frames
CASSI, 8-frames
CASSI, 16-frames

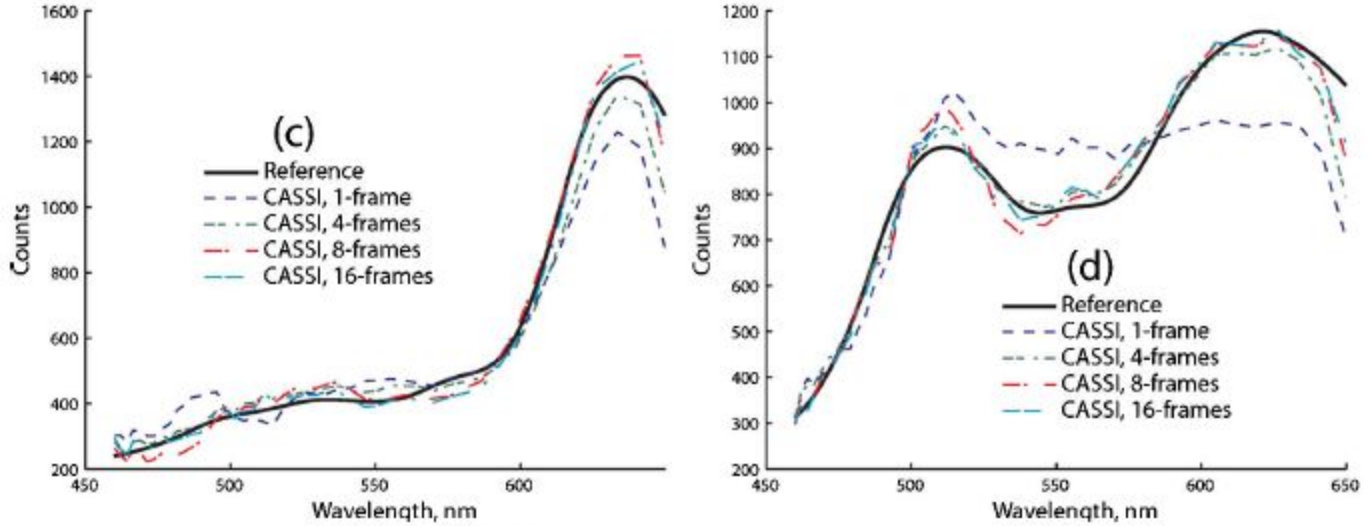


Fig. 5. (Color online) Using the GretagMacbeth ColorChecker as a baseline, comparisons were made with white-light sunlight emulation. (a)–(d) compare snapshot versus 4, 8, and 16 frame reconstructions. Nearby colors can bleed into the spectra, as seen in the RGB image, where the blue just to the right of the orange square was dispersed into the orange. Multiple frames eliminate this problem.

Table 2. RMSE of CASSI Spectra versus Reference Spectrometer

	Center				Edge			
Frames	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
1	9.6%	6.9%	4.1%	17.4%	14.0%	11.7%	8.4%	34.3%
4	2.7%	3.6%	2.4%	4.5%	5.3%	8.1%	2.7%	10.2%
8	3.7%	3.3%	2.6%	5.4%	2.7%	3.6%	2.3%	8.9%
16	1.9%	3.4%	2.8%	5.0%	2.9%	4.1%	2.7%	6.8%

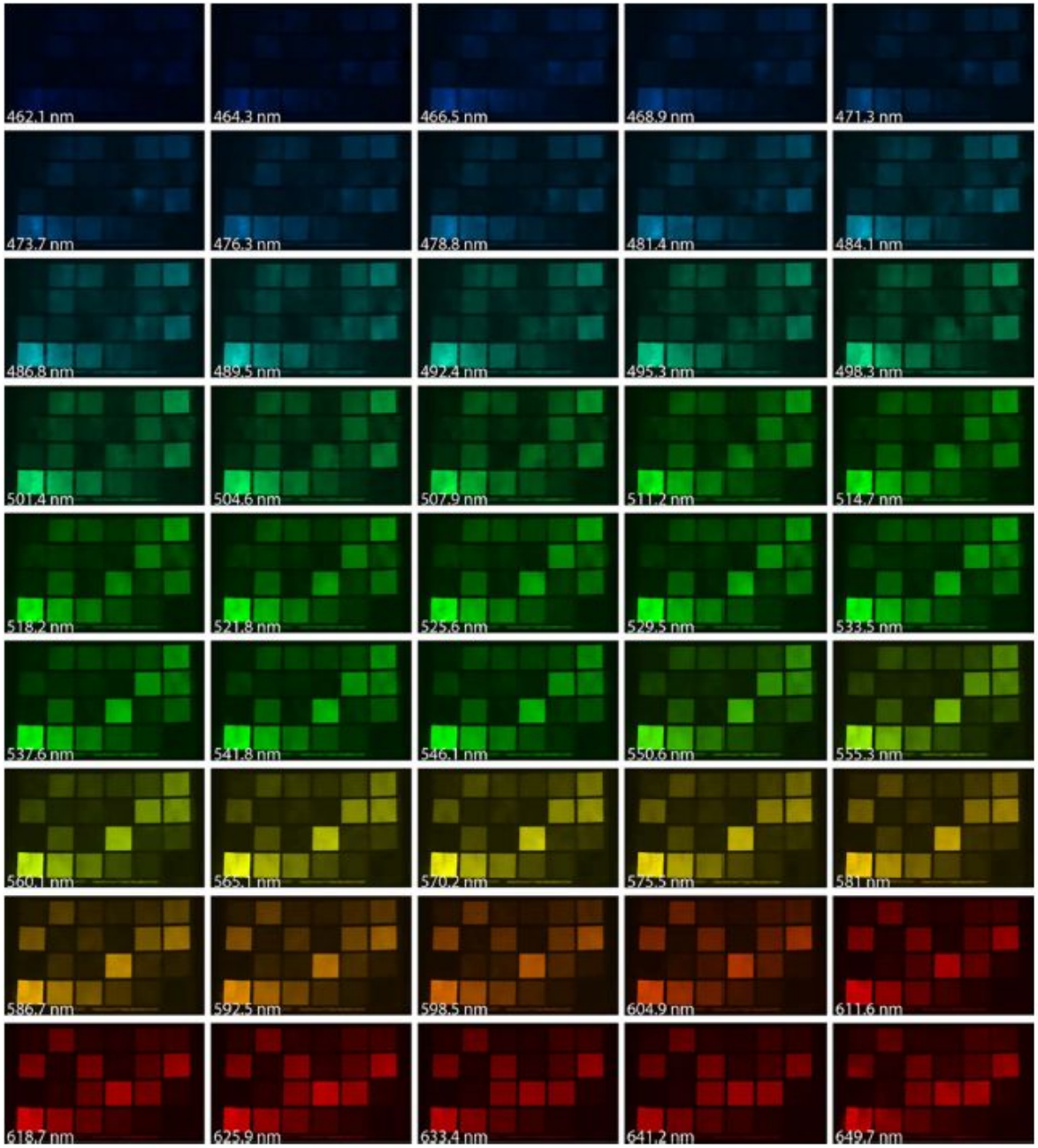


Fig. 6. (Color online) Multiframe CASSI reconstruction of the GretagMacbeth ColorChecker, showing all the channels from 460–650 nm.

Reference color image

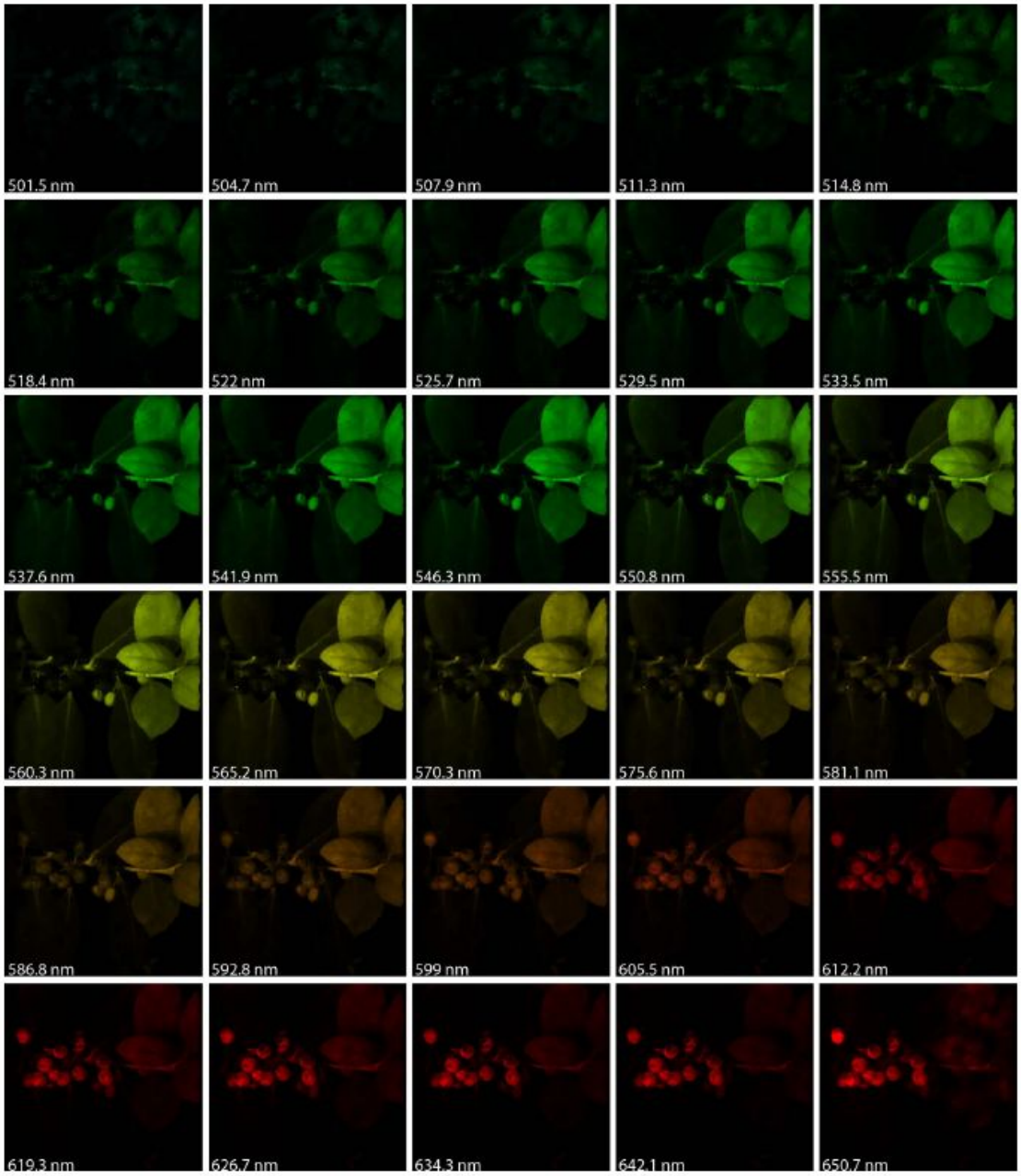


Fig. 8. (Color online) All channels from 500–650 nm from a 24-frame reconstruction of the data cube.

Color image Demosaicing

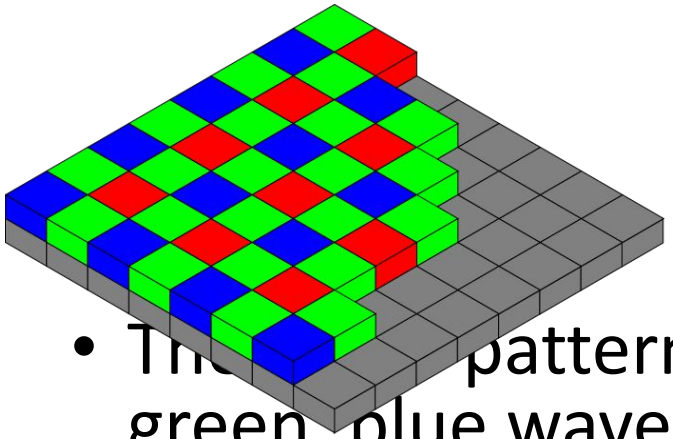
CS 663, Ajit Rajwade

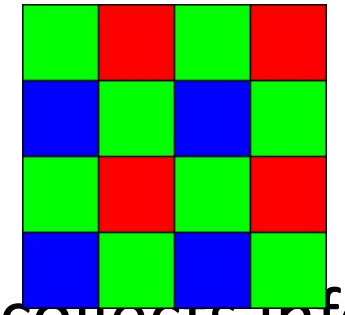
Color Filter Arrays

- It is an array of tiny color filters placed before the image sensor array of a camera.
- The resolution of this array is the same as that of the image sensor array.
- Each color filter may allow a different wavelength of light to pass – this is pre-determined during the camera design.

Color Filter Arrays

- The most common type of CFA is the Bayer pattern which is shown below:





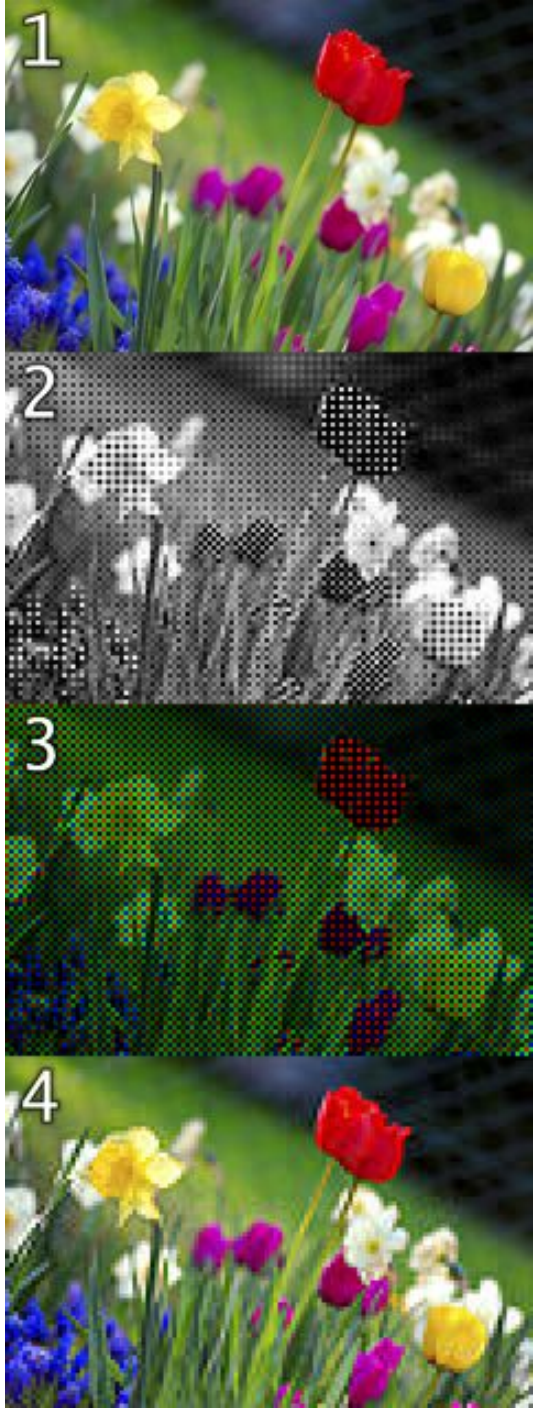
https://en.wikipedia.org/wiki/Color_filter_array

- The Bayer pattern collects information at red, green, blue wavelengths only as shown above.

*The word “mosaic” or “mosaiced” is not to be confused with image panorama generation which is also called image mosaicing.

Color Filter Arrays

- The Bayer pattern uses twice the number of green elements as compared to red or blue elements.
- This is because both the M and L cone cells of the retina are sensitive to green light.
- The **raw** (uncompressed) output of the Bayer pattern is called as the **Bayer pattern image** or the **mosaiced** (*) **image**.
- The mosaiced image needs to be converted to a normal RGB image by a process called color image **demosaicing**.



https://en.wikipedia.org/wiki/Bayer_filter

“original scene”

Mosaiced image

Mosaiced image – just coded with the Bayer filter colors

“Demosaiced” image – obtained by interpolating the missing color values at all the pixels

A Demosaicing Algorithm

- There exist a plethora of demosaicing algorithms.
- We will study one that is implemented in the **“demosaic”** function of MATLAB.
- The algorithm implemented by this function was published in 2004.

Malvar, H.S., L. He, and R. Cutler, *High quality linear interpolation for demosaicing of Bayer-patterned color images*. ICASPP, Volume 34, Issue 11, pp. 2274-2282, May 2004.

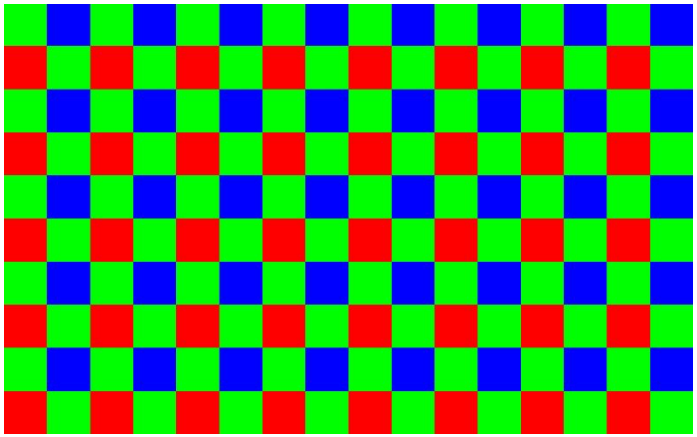
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Demosaicing_ICASSP04.pdf

Demosaicing Algorithm

- Demosaicing involves interpolation of missing color values from nearby pixels.
- The easiest way is to perform linear interpolation – given the structure of the Bayer pattern.

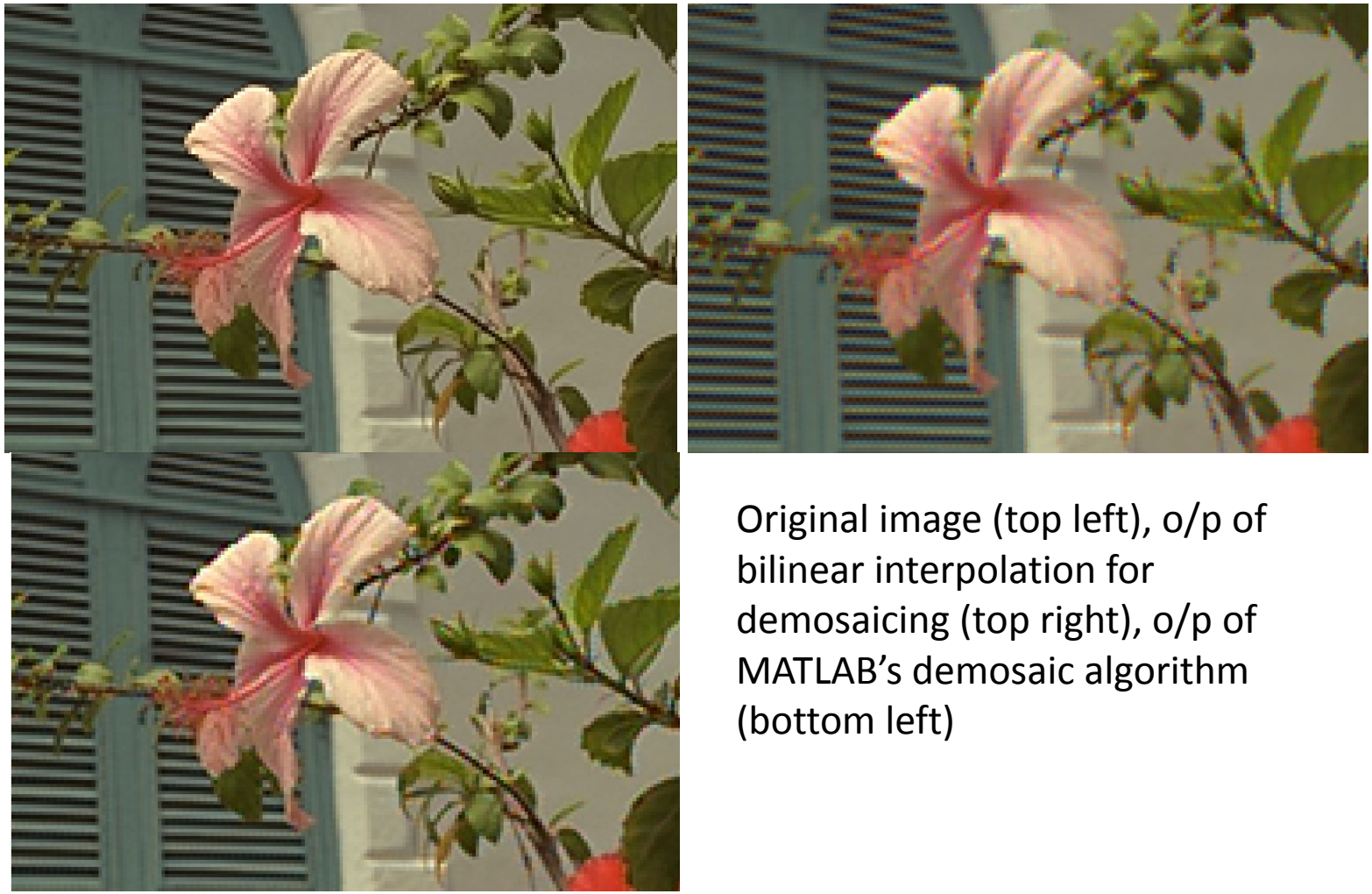
$$\hat{g}(i,j) = \frac{1}{4} \sum g(i+m, j+n)$$

$$(m,n) = \{(0,-1), (0,1), (-1,0), (1,0)\}$$



Demosaicing Algorithm

- But such an algorithm gives highly sub-optimal results at **edges** – as seen in the simulation below.



Original image (top left), o/p of bilinear interpolation for demosaicing (top right), o/p of MATLAB's demosaic algorithm (bottom left)

Demosaicing algorithm

- Make use of the correlation between R,G,B color values for a more edge-aware interpolation!
- Edges in natural images have stronger luminance changes than chrominance changes.
- Consider the case of finding G at an R or a B pixel.
- The R-gradient can be useful information for determining the G value.

Demosaicing algorithm

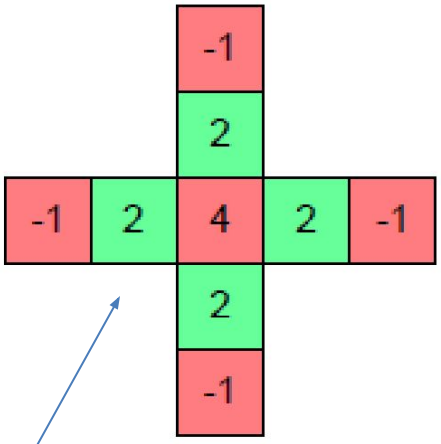
- Consider the case of finding G at an R or a B pixel (x,y).
- Obtain an estimate of the R value at pixel (x,y) by bilinear interpolation.
- If the actual R value at (x,y) differs considerably from the bilinearly interpolated R value at (x,y), it means that there is a sharp luminance change at that pixel.
- The corrected value of G is given as follows:

$$\hat{g}(i,j) = \hat{g}_B(i,j) + \alpha\,\Delta_R(i,j) \qquad \Delta_R(i,j) \triangleq r(i,j) - \frac{1}{4}\sum r(i+m, j+n)$$

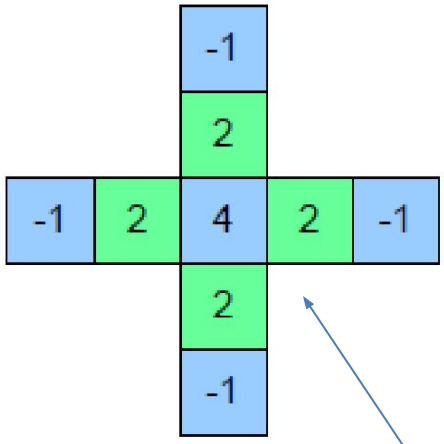
$$(m,n) = \{(0,-2), (0,2), (-2,0), (2,0)\}$$

Bilinearly interp. value

Gain factor



G at R locations



G at B locations

$$\hat{g}(i,j) = \hat{g}_B(i,j) + \alpha\,\Delta_R(i,j)$$

$$\Delta_R(i,j) \triangleq r(i,j) - \frac{1}{4}\sum r(i+m, j+n)$$

$$(m,n)=\{(0,-2),\,(0,2),\,(-2,0),\,(2,0)\}$$

$$\hat{\hat{g}}(i,j) = \hat{\hat{g}}_B(i,j) + \gamma\Delta_B(i,j)$$

$$\Delta_B(i,j) = b(i,j) - \frac{1}{4}\sum_{(m,n)=\{(0,-2),(-2,0),\,(2,0),(0,2)\}} b(i+m, j+n)$$

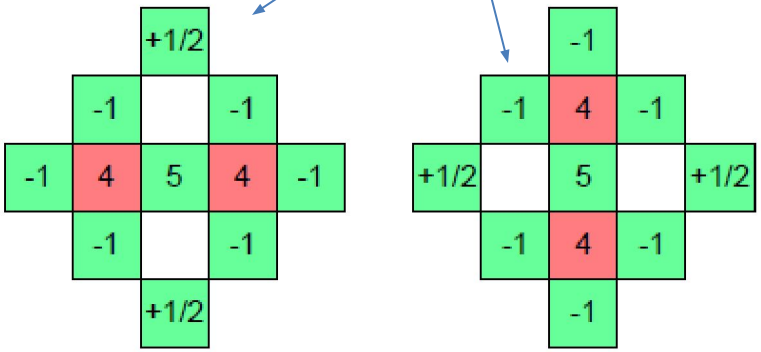
Demosaicing algorithm

- We have seen how to obtain G at an R or a B pixel.

- To obtain the R value at a G pixel, the corresponding formula is

Bilinear interp. value

$$\hat{r}(i,j) = \hat{r}_B(i,j) + \beta\,\Delta_G(i,j)$$

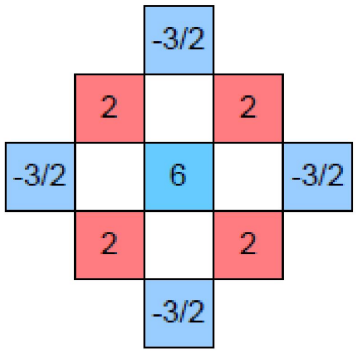


Demosaicing algorithm

- To obtain a R value at a B pixel, the corresponding formula is

Bilinear interp. value

$$\hat{r}(i,j) = \hat{r}_B(i,j) + \gamma\,\Delta_B(i,j)$$



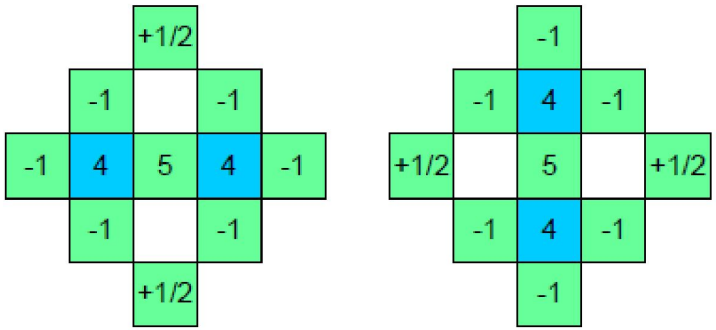
R at blue in
B row, B column

Demosaicing algorithm

- To obtain a B value at a G pixel, the corresponding formula is

Bilinear interp. value

$$\hat{b}(i,j) = \hat{b}_B(i,j) + \beta\Delta_G(i,j)$$



B at green in
B row, R column

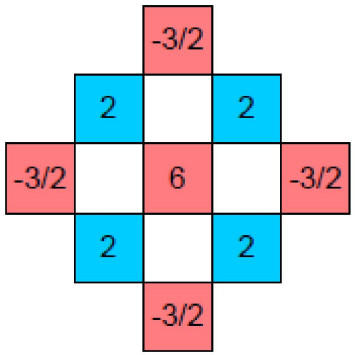
B at green in
R row, B column

Demosaicing algorithm

- To obtain a B value at a R pixel, the corresponding formula is

Bilinear interp. value

$$\hat{b}(i,j) = \hat{b}_B(i,j) + \gamma \Delta_R(i,j)$$



B at red in
R row, R column

Gain factors

- The values α, β, γ are gain factors for the correction due to gradients in the R,G,B channels respectively.

- How are they estimated? In a training phase of the algorithm – performed offline.

- The gain factors were designed to optimize a mean square error criterion.

Demosaicing: when does it happen?

- Your camera acquires images in a raw format, with 12 bits per pixel.
- Remember: at each pixel, only one of the R,G,B values is measured.
- That is, the camera measures just the CFA image.
- The camera then runs a demosaicing algorithm internally to generate the full RGB image.
- This image then goes through various intensity transformations after which it is JPEG-compressed and stored in the camera memory card.
- The demosaicing algorithm described earlier does not perform any noise removal – which can lead to noisy artifacts in the final image!