1

Enabling Technologies for Sports (5XSF0), Module 04

Image Restoration and Freq. Filtering & Color Imaging and Transformations

Peter H.N. de With

(p.h.n.de.with@tue.nl)

slides version 1.0

2

Overview Module 04

- **Noise and filtering**
  - Noise models
  - Spatial filtering and periodic noise and
- **Special filtering techniques**
  - Frequency-Band Filtering
  - Adaptive filtering
- **Color imaging and transformations**
  - Color models and systems
  - Color representations and associated processing

3

Module 04 – Part 1 Image Restoration & Reconstruction

Noise models, spatial noise filters, freq.-domain noise filters, projection imaging and CT

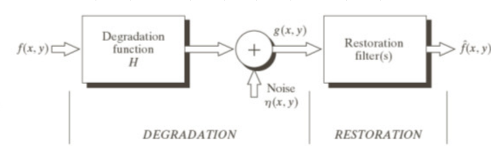
4

Model of Image Degradation/Restoration

1. Model communication imperfections and noise insertion by a channel with degradation and restoration filters.
2. Degradation can be derived by measuring and analysis
3. Model is linear (pos. invariant) 'filter' process & addition of noise, so $g(x,y) = h(x,y) \otimes f(x,y) + n(x,y)$

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

FIGURE 5.1 A model of the image degradation/restoration process.



Noise Sources and Considerations

1. Noise is generated when capturing the image, caused by
   1. **Sensor: intrinsic noise and temperature**
   2. **Light level of the scene**
   3. **Insufficient resolution processing (cost reduction)**
2. Noise has spatial and frequency properties, so it can be analyzed with the DFT and in time domain
3. For analysis purposes (also here), noise is assumed to be spatially and signal independent. This is sometimes invalid (e.g. X-ray and nuclear medicine imaging)

$$g(x,y) = h(x,y) \otimes f(x,y) + n(x,y)$$
$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

Widely Used Noise Models – (1)

1. **Gaussian (normal) noise** $p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(z-\mu)^2/2\sigma^2}$
   1. Mathematically tractable
   2. Note mean μ and st.dev. σ
2. **Rayleigh noise** $p(z) = \frac{2}{b}(z-a)e^{-(z-a)^2/b} \quad (z \geq a)$
   1. Often in comm. problems
   2. Has mean and variance with

$$\bar{z} = \mu = a + \sqrt{\pi b/4}; \quad \sigma^2 = b(4-\pi)/4$$

3. **Erlang (Gamma) noise** $p(z) = \frac{a^b z^{b-1}}{(b-1)!} e^{-az} \quad (z \geq 0)$
   1. Has mean and variance
   2. *a*>0 and *b* positive integer $\quad \bar{z} = \mu = b/a; \quad \sigma^2 = b/a^2$

Widely Used Noise Models – (2)

1. **Exponential noise** $p(z) = ae^{-az} \quad (a, z \geq 0)$
   1. Mathematically tractable
   2. Note mean μ and st.dev. σ $\quad \mu = 1/a; \quad \sigma^2 = 1/a^2$
2. **Uniform noise** $p(z) = \frac{1}{b-a}; \quad (a \leq z \leq b)$
   1. Often in analysis problems
   2. Has mean and variance with

$$\mu = (a+b)/2; \quad \sigma^2 = (b-a)^2/12$$

3. **Impulse (salt and pepper) noise** $p(z) = P_a \quad (z = a)$
   1. Noise pulses can be pos. or negative $\quad p(z) = P_b \quad (z = b)$
   2. *b*>a, intensity *b* will be light dot
   3. If either probability is zero: unipolar noise

Illustrated noise models

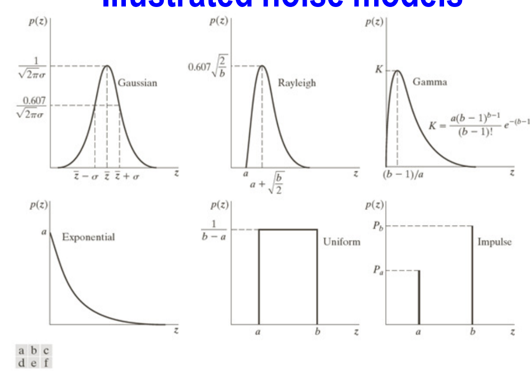
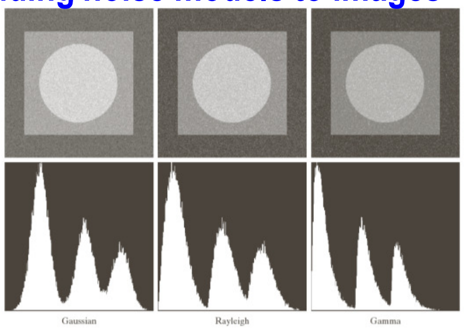


FIGURE 5.2 Some important probability density functions.

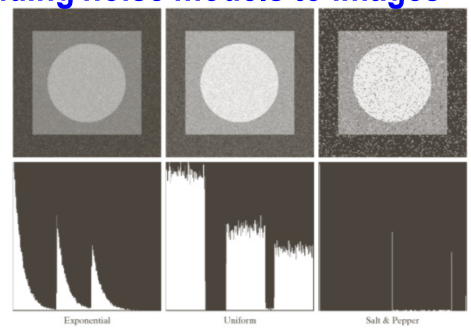
Adding noise models to images – (1)



Gaussian Rayleigh Gamma

FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Adding noise models to images – (2)



Exponential Uniform Impulse (S&P)

FIGURE 5.4 (Continued) Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.

Periodic noise

- Periodic noise is spatially dependent (exception!)
- Very suited for filtering in the frequency domain!
- Model is sinusoid, with

$$r(x,y) = A\sin[2\pi u_0(x+B_x)/M + 2\pi v_0(y+B_y)/N]$$

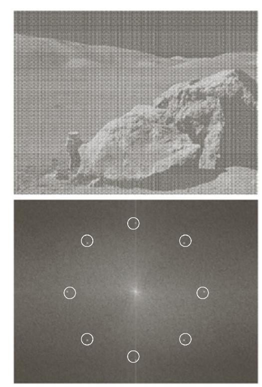
- u and v are frequencies, B's are phase (displacements)
- Derive the frequency domain representation

Periodic noise – Visual example

- Periodic noise is spatially dependent

Note spectrum dots!

Each sine wave gives a conjugate pair in freq. domain



a
b

FIGURE 5.5 (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

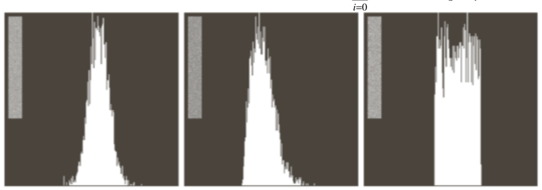
Measuring noise parameters

- Take selected area(s) of image which have no detail and measure the local histogram

$$\mu = \sum_{i=0}^{L-1} z_i p_S(z_i)$$

- Measure e.g. mean and variance

$$\sigma^2 = \sum_{i=0}^{L-1} (z-\mu)^2 p_S(z_i)$$



a b c

FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

Spatial mean filtering / restoration of noise

Covers the case of noise only $g(x,y) = f(x,y) + n(x,y)$

1. **Arithmetic mean filter**
   1. Simple filter, average in area
   2. Rectangular window mxn

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

2. **Geometric mean filter**
   1. Perform similar to mean filter
   2. Tends to loose less details

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

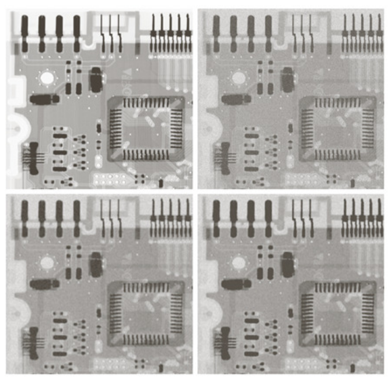
3. **Harmonic mean filter**
   1. Works well for salt noise, not pepper
   2. Performs well also for Gaussian noise

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

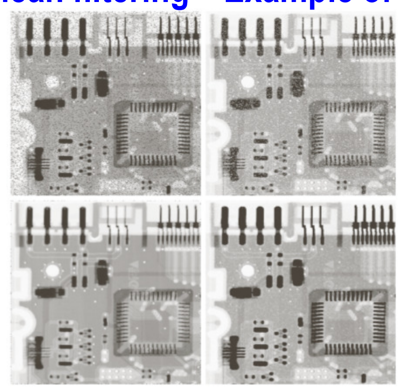
Mean filtering – Example X-ray image – 1

a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



Mean filtering – Example of X-ray image – 2



a b
c d

FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3 × 3 contra-harmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

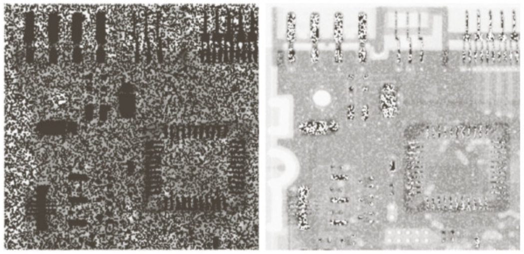
$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Contraharmonic mean filter reduces salt-&-pepper noise

Mean filtering – Example of X-ray image – 3

a b

FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3 × 3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.



In general, arithmetic an geometric mean filters are well suited for random noise (such as Gaussian, uniform). The contraharmonic filter does well for impulse noise, but the dark/light color has to be known.

Order-statistic filters / Median etc.

Covers the case of noise only. Order-statistic filters are spatial filters whose response is based on ordering/ranking the pixels. Is a non-linear process.

1. **Median filter**

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s,t)\}$$

   1. Widely applied
   2. Selects the median of a set of samples
   3. Excellent results for speckle noise, no blur

2. **Max and min filter**

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}}\{g(s,t)\}$$

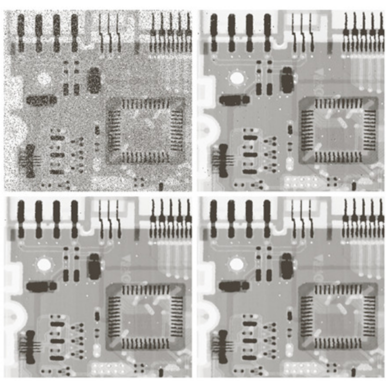
$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}}\{g(s,t)\}$$

   1. Useful for finding bright/dark points
   2. Max filter reduced pepper noise
   3. Min filter reduced salt noise

Example results with 3x3 Median filters

a b
c d

FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3 × 3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.



Median, one iter.

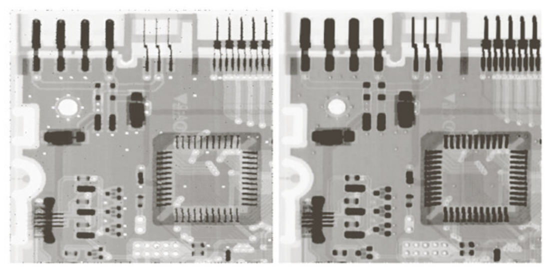
Median, two iter.

Median, three iter.

Example results with max/min filters

a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

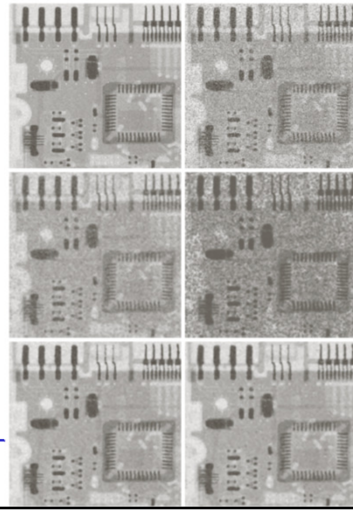


Note the brightening effect of the max filter and the darkening effect of the min filter!

Sample results with mean & median filters

a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image (b) filtered with a 5 × 5; (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.



Adaptive filters

Adaptive filters

- Perform better than static filters as presented earlier
- They have a higher complexity, since measurements are required
- Operate within a window and then adapt to the image contents

1. Adaptive local noise reduction filter, operating in area S, depending on 4 quantities

1. $g(x,y)$ the value of the noisy image
2. σ^2_η, the variance of the noise corrupting $f(x,y)$
3. the local mean m_L of the pixels in the window
4. σ^2_L, the local variance of the pixels in the window

Local noise reduction filter

- **The desired behavior of the filter should be that**
  1. If σ^2_η=0 the filter should return simply the value of $g(x,y)$
  2. If the local variance is high relative to σ^2_η, the filter should return a value close to $g(x,y)$. Edges should be preserved.
  3. If the two variances are equal, the filter should return the arithmetic mean value of the pixels in S. This means that the local area has the same properties as the overall image and local noise then averaged.

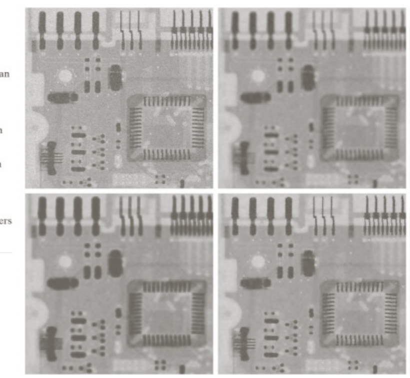
The filter could be as follows

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right]$$

Local noise reduction filter - Result

a b
c d

FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7 × 7.



Adaptive filters are always better!

Adaptive median filter – (1)

- Normal median filter can handle low-density noise
- Varying density noise requires other filtering
- An adaptive median filter may change the filter window S
- The filter still replaces one pixel with the filter output
- Consider that z_{min}= min of window, z_{max} = max of window, z_{med} = median of window, S_{max} = max window size

Adaptive median filter – (2) / Algorithm

Stage A

- $A1 = z_{med} - z_{min}$; $A2 = z_{med} - z_{max}$.
- If $A1 > 0$ and $A2 < 0$, go to stage B
- Else increase window size
- If window size <= S_{max} repeat stage A
- Else output z_{med}

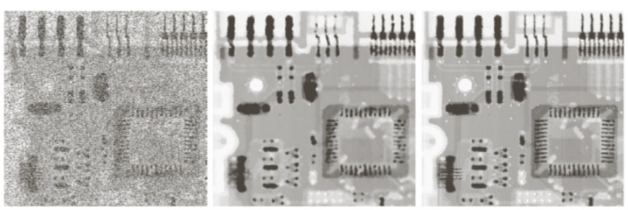
Determine in this stage, whether output z_{med} is impulse or not

Stage B

- $B1 = z_{xy} - z_{min}$; $B2 = z_{xy} - z_{max}$.
- If $B1 > 0$ and $B2 < 0$, output z_{xy}
- Else output z_{med}

Test whether data is impulse or not, if so, then filter

Adaptive median filter – (3) / Results



a b c

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7 × 7 median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

Static filter shows a significant loss of detail (broken conn.)!

Frequency domain filters – Bandreject

Sometimes, the spectral noise location is known, e.g. with periodic noise. Then define specific spectral noise filters

a b c

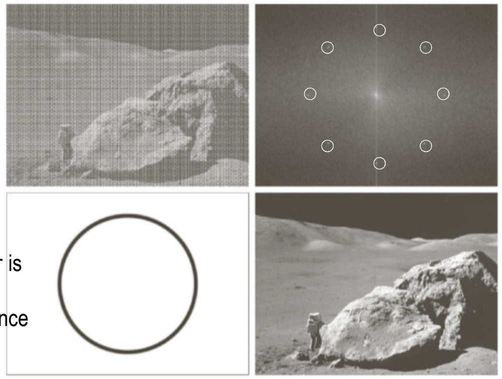
FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Additive periodic noise can be approximated by 2-d sinusoidal functions. Remember the DFT of sine is two conjugate impulses!

Frequency domain filters – Bandreject

a b
c d

FIGURE 5.16
(a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)



Bandpass filter is opposite of bandreject, hence

$H_{BP}=1-H_{BR}$

Frequency domain filters – Bandpass

Bandpass filter is opposite of bandreject, hence

$H_{BP}(u,v)= 1 - H_{BR}(u,v)$

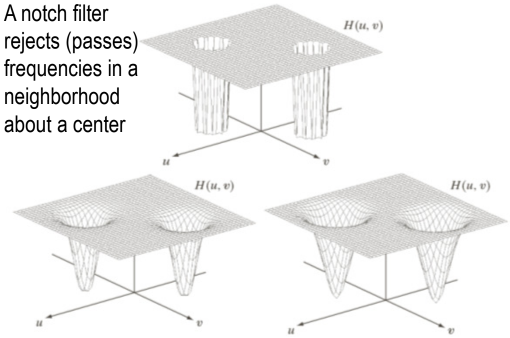
FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.

Frequency domain filters – Notch filters

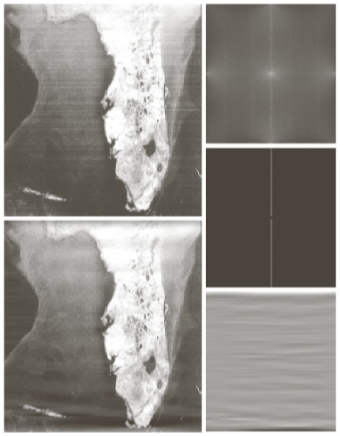
A notch filter rejects (passes) frequencies in a neighborhood about a center

a
b c

FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.



Freq.-domain filters – Notch filter results



a b
c
e d

FIGURE 5.19
(a) Satellite image of Florida and the Gulf of Mexico showing horizontal scan lines. (b) Spectrum. (c) Notch pass filter superimposed on (b). (d) Spatial noise pattern. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)

Estimation of the degradation function

Obtain the degradation by

- Observation
  - Function is not known
  - Measure in rectangles in foreground and background
- Experimentation
  - Perform experiments with equipment to emulate conditions
  - Impose image impulse, measure impulse response, $H=G(u,v)/A$
- Mathematical modeling
  - Test with modeled filters, such as Gaussian, Laplacian etc.

Example: blur caused by linear motion - 1

- Consider the shift $g(x,y)=\int_0^T f\left[(x-x_0(t), y-y_0(t)\right]dt$
- Integration constant with time T and g the blurred image
- Taking the Fourier transform and reversing the integration over T with that of Fourier gives

$$G(u,v)=\int_0^T F(u,v)e^{-j2\pi[ux0(t)+vy0(t)]}dt = F(u,v)H(u,v)$$

- Hence, H is the integral over de exponential
- If the motion variables, are known, then H is computed

Example: blur caused by linear motion - 2

- Consider the function $x_0(t)=at/T$
- Taking the Fourier transform and deriving further

$$H(u,v)=\int_0^T e^{-j2\pi ux_0(t)}dt=\int_0^T e^{-j2\pi uat/T}dt=\frac{T}{\pi ua}\sin(\pi ua)e^{-j\pi ua}$$

- This result can be extended in 2D with $y_0(t)=bt/T$

Example: blur caused by linear motion - 3

a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

Inverse filtering

Suppose we have degradation function $H(u,v)$

* **Simplest approach: inverse filtering**
  - $F^h(u,v) = G(u,v) / H(u,v)$, then with add. noise, we find
  - $F^h(u,v) = F(u,v) + N(u,v) / H(u,v)$
  - Note that $N(u,v)$ is not known!
  - Also: when $H(u,v)$ is zero, then the ratio explodes…., this happens frequently
  - One way around: limit filter frequencies to values close to origin where H is high typically

* **This slide is intentionally left blank**

* **This slide is intentionally left blank**

* **This slide is intentionally left blank**

Module 04 – Part 2
Color image processing

Color fundamentals, triangle, color models, color conversion

Color processing

- **Full-color processing**
  - Involves TV, broadcast and consumer cameras, etc
- **Pseudo color processing**
  - Often used in medical domain: generate color to show…

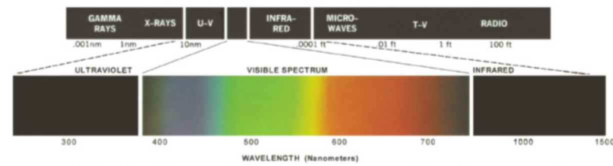


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

Color *light* wavelengths

- **Human eye cones measure light**
  - 65% red, 33% green, 2% blue
  - All cones not equal!
- **RGB are thus primary colors**
  - Blue=435nm
  - Green= 546nm
  - Red=700nm

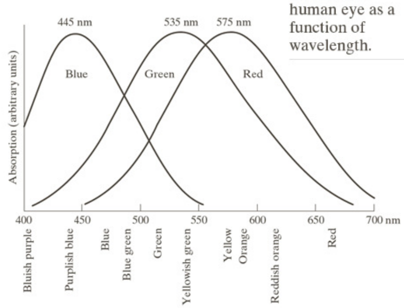
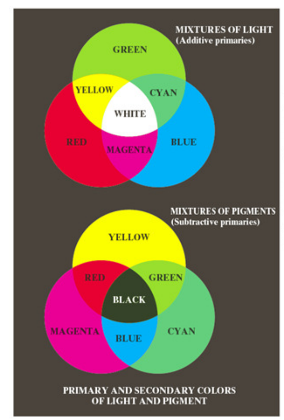


FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

Color mixtures

- **Primary colors can be mixed to secondary colors**
  - Magenta=R+B
  - Cyan=G+B
  - Yellow=R+G
- **Mixing secondary color with an opposite primary (or 3 prim.) gives white**



a
b

FIGURE 6.4 Primary and secondary colors of light and pigments. (Courtesy of the General Electric Co., Lamp Business Division.)

Color triangle

- Colors have been specified by CIE
  - In a triangle
  - x (red) and y (green)
  - z (blue) = $1-(x+y)$
- Any point on boundary is fully saturated

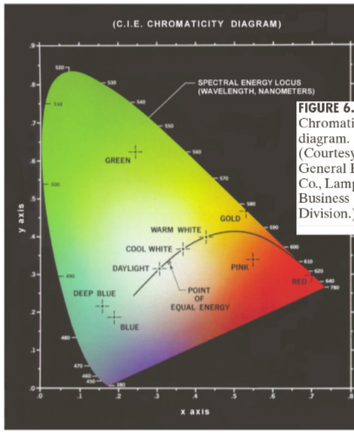


FIGURE 6.5 Chromaticity diagram. (Courtesy of the General Electric Co., Lamp Business Division.)

Typical color models

- Color is specified in models
  - RGB, Red Green Blue, for TV & cameras
  - CMYK, Cyan Magenta Yellow Black, for printing
  - The color black in CMYK is extra
  - HSI, Hue Saturation Intensity, for grayscale printing, old TV, etc

RGB Color Model

- RGB is based on Cartesian coordinates, RGB on axes, CMY on corners, Black at origin

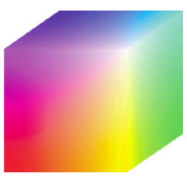


FIGURE 6.8 RGB 24-bit color cube.

16M colors!

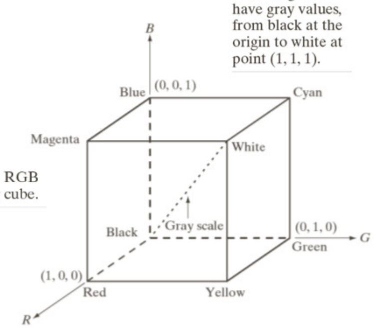


FIGURE 6.7 Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).

RGB safe color cube for coloring images

- RGB safe color cube: only colors at the surface of the cube, total 216 colors
- Guarantee colors at every 8-bit computer/display

• Each surface has 6x6 colors
• All 8-bit gray levels are included
• supports web-based & pseudo-color imaging

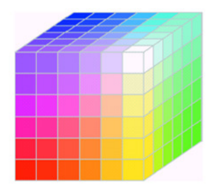


FIGURE 6.11 The RGB safe-color cube.

CMY and CMYK Color Models

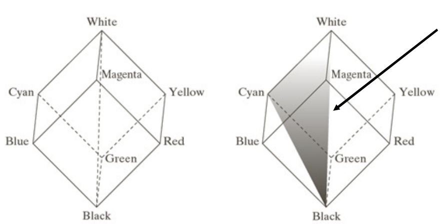
- CMY are secondary colors, or **primary colors of pigments**
- Printing popularity comes from: cyan does not reflect red from white light illumination
- Most devices using colored pigments require CMY input
- Conversion from RGB is simple, and results from
  - $(C, M, Y)^T=(1, 1, 1)^T - (R, G, B)^T$
- Based on assumption that all colors have been normalized to unit interval
- Note that first component equation reflects absence of red, magenta reflects no green, and yellow no blue.
- Likewise, RGB can be easliy obtained from CMY

HSI Color Model

- HSI originates from the wish to describe images in
  - **Intensity** (black & white or gray level image)
  - **Saturation** (strength of the pure color component)
  - **Hue** (color temperature, simply the type of color)
- HSI is much more suited for describing color images, rather than RGB, that is more suited for color image generation
- Convert from RGB cube, by tilting it such that black is at the bottom and white at the top….

HSI Color understanding the RGB cube

- Convert from RGB cube, by tilting it such that black is at the bottom and white at the top….
- Intensity is covered by passing a plane perpendicular to the intensity axis, and hue is in the shaded plane



Shaded plane contains both BW and cyan

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

HSI – Color and Saturation – (1)

- Convert from RGB cube, cross planes are triangular or hexagonal
- Hue is angle leftwise from red orientation
- Saturation is the length of the vector
- Origin is at intersection with intensity plane

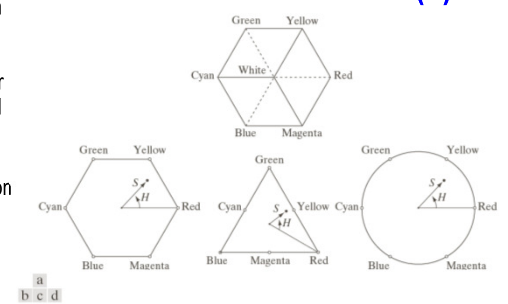


FIGURE 6.13 Hue and saturation in the HSI color model. The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.

HSI – Color and Saturation – (2)

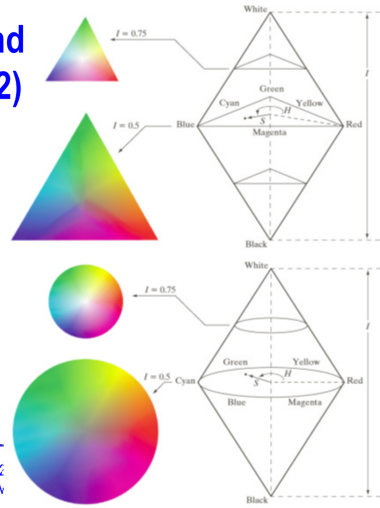


FIGURE 6.14 The HSI color model based on (a) triangular and (b) circular color planes. The triangles and circles are perpendicular to the vertical intensity axis.

Converting color from RGB to HSI – (1)

* From each RGB pixel, the H value $H = \theta \quad B \le G$, $H = 360 - \theta \quad B > G$
* The angle comes from a.o. goniometry analysis

$$\theta = \arccos\{\frac{1/2[(R-G)+R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}}\}$$

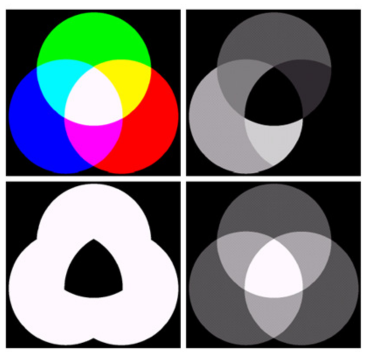
* The saturation and intensity are given by

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

Converting color from RGB to HSI – (2)

Note the following properties

- RGB colors in (a) have fixed saturation
- Intensity grows when more color is added



Hue

Int.

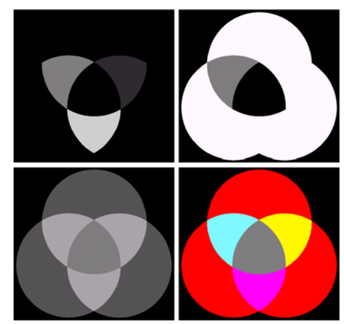
Sat.

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.

Pseudo color - Intensity slicing

- Let the gray scale vary between 0 (black) and L-1 (white)
- Assume P planes perpendicular to the intensity axis
- P+1 partitions of gray scale are made
- Each value intensity l_k converts to color c_k



a b
c d

FIGURE 6.17 (a)–(c) Modified HSI component images. (d) Resulting RGB image. (See Fig. 6.16 for the original HSI images.)

Geometric interpretation of intensity slicing

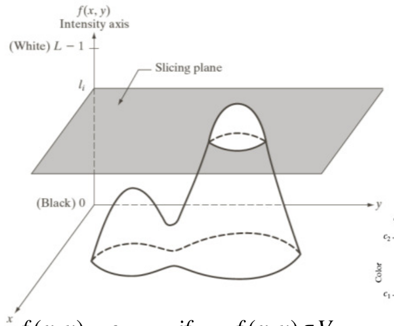


FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

$$f(x,y) = c_k \quad \text{if} \quad f(x,y) \in V_k$$

Alternative view

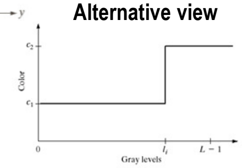
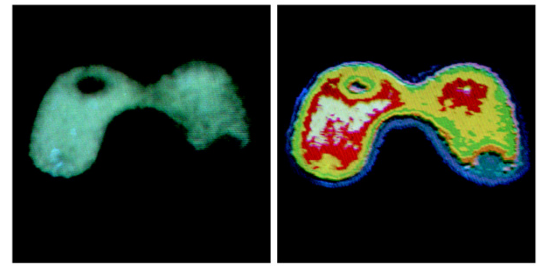


FIGURE 6.19 An alternative representation of the intensity-slicing technique.

Color / Example of intensity slicing



a b

FIGURE 6.20 (a) Monochrome image of the Picker Thyroid Phantom. (b) Result of density slicing into eight colors. (Courtesy of Dr. J. L. Blankenship, Instrumentation and Controls Division, Oak Ridge National Laboratory.)