§5 - Restoration

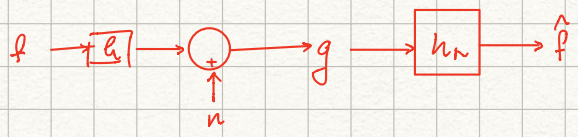
§5.1 Wiener Filtering

§5.1.1 Wiener Filtering for Additive Noise

Image Degradation (Noise & Distortion)

$$g(x,y) = f(x,y) * h(x,y) + n(x,y)$$

(h: Distortion, n: noise)



- Goal: find filter to undo effects of the degradation (Noise & Distortion)
- $h(x,y)$: linear, shift invariant distortion, typically some blur model, (ex: PSF of camera lens)
- $n(x,y)$: Point Noise Model

Point Noise Model

White Gaussian Noise (WGN)

- simple but important example
  - Mean: $E[n(x)] = n(x) = 0$
  - Variance: $E[(n(x) - n(x))^2] = E[n^2(x)] = \sigma_n^2$
  - Autocorrelation: $R_n(x_i, x_j) = E[n(x_i)n(x_j)] = \begin{cases} \sigma_n^2 & \text{when } x_i = x_j \\ 0 & \text{otherwise} \end{cases}$

  Ex: distinct samples of noise are uncorrelated

TERMINOLOGY: Pixel intensities in a WGN image are independent, identically distributed zero mean Gaussian random variables.

Image Restoration

- So, a linear, shift-invariant degradation can be represented by a convolution of the input image with a distortion + additive noise.
- Intuitively, the goal of image restoration in such a case is to find filters that reverse the degradation process.

$$g(x,y) * h_f(x,y) = f'(x,y)$$

s.t. $E[(f(x,y) - \hat{f}(x,y))^2]$ is small

i.e., "minimize the mean squared error (MSE)"

Wiener Filter for Additive Noise

- Long derivation ⇒ find the optimal solution for degradation process for $h(x) = \delta(x)$ (i.e. a solution that only accounts for additive noise).
- Wiener Filter optimally satisfies the MSE constraint for additive noise.

↳ In freq. domain: $H_r(u) = \dfrac{S_f(u)}{(S_f(u) + S_n(u))} = \dfrac{S_f(u)}{S_f(u) + S_n(u)}$

where $S_f(u) = |F(u)|^2$ is power spectrum of signal

& $S_n(u) = \sigma_n^2$ is power spectrum of WGN

↳ INSIGHT:

$$H_r(u) = S_f(u) / (S_f(u) + S_n(u))$$
$$= 1 / (1 + S_n(u)/S_f(u))$$
$$= 1 / (1 + 1/SNR)$$

$H(u)$ approaches 1 for large SNR and 0 for poor SNR, as a function of freq.

white noise.

Signal to Noise Ratio

↓

as SNR ↑, $H(u) \to 1$

as SNR ↓, $H(u) \to 0$

└ How do we estimate $S_f(u)$ & $S_n(u)$?

$S_g(u) = S_f(u) + S_n(u) = S_f(u) + \sigma_n^2$

(S_f: pristine/original; S_n: noise)

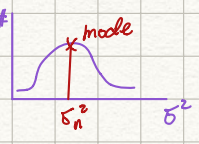
$\therefore S_f(u) = S_g(u) - \sigma_n^2$

STEPS:

1) grab a patch of constant content

2) capture σ^2 using 7×7 window on a pixel by pixel basis.

3) Plot σ^2 histogram



4) Take mode as estimate of σ_n^2, since most commonly occuring

§5.1.2 Methods to estimate degradation

Restoration with Distortion & WGN

- Reintroduce LSI distortion & assume $h(x)$ known
- Again, we'd like to estimate $f(x)$, the original signal, but this time with both $h(x)$ & additive noise known

Estimating Distortion $h(x)$

There are # of ways to estimate the distortion introduced by $h(x)$:

1. Image Observation.
2. Experimentation
3. Modeling

1) Image Observation

① Take subimage with simple structures from image

② Construct estimate of what subimage should be like prior to distortion

③ Determine subimage distortion function h_s based on observed subimage g_s & constructed subimage f_s^{est}

$$H_s(u,v) = G_s(u,v) / F_s^{est}(u,v)$$

④ Reconstruct complete distortion function h based on h_s

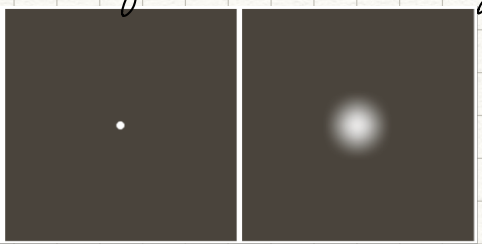
⑤ Works based on the assumption of Shift Invariance.

2) Experimentation

① Useful if equipment similar to equipment used to acquire degraded image available.

② Image an impulse (small dot of light) and adjust settings till the impulse is close to that produced by the degradation.

③ Use the estimated degradation function to restore image.



a b

FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.

3) Modeling

- Useful if physical conditions can be modeled. (camera conditions, environment conditions, etc.)

Ex: Linear Motion Blur

ex: for L=4 & horizontal motion blur : $h(x,y) = [0.25 \;\; 0.25 \;\; 0.25 \;\; 0.25]$

Ex: Assume a model that "fits" images

ex: autocorrelation function : $S_f(u) = 2a\sigma_f^2 / (a^2 + 4\pi^2 u^2) + \mu_f^2 \delta(u)$

§5.1.3 Wiener Deconvolution

Wiener (Least Squares) Deconvolution Filter

Note:

- "Wiener filter for additive Noise": assumes NO Distortion
- "Wiener Deconvolution": compensate both distortion & noise

2 approaches: ① Inverse filter (BAD) (×) (small H → Noise predominates) ② Wiener Deconvolution (✓) (✓)

② $f \to [h] \to (+) \leftarrow n \to g \to [h_1] \to [h_2] \to \hat{f}$ (undo (n), undo (h))

② $H_2(u) = \frac{1}{H(u)}$

① final estimate $f_d = f * h$, ⇒ ② apply inverse of h

① Recall: $S_f(u) = |F(u)|^2$, $S_{fd}(u) = H(u) \cdot F(u)$, $S_{fd}(u) = |H(u)|^2 \cdot S_f(u)$

$\Rightarrow H_1(u) = \frac{S_{fd}(u)}{S_{fd}(u) + S_n(u)} = \frac{|H(u)|^2 S_f(u)}{|H(u)|^2 S_f(u) + S_n(u)}$

$\Rightarrow H_{WD} = H_1(u) \cdot H_2(u) = \frac{H^*(u)}{|H(u)|^2 S_f(u) + S_n(u)}$

if SNR ≫ 1, $H_{WD}(u) = \frac{1}{H(u)}$

SNR ≪ 1, $H_{WD}(u) \to \phi$

$H_{WD} = \frac{H^*(u)}{|H(u)|^2 + 1/SNR}$

Derivation:

① Inverse Filter (BAD).

$f \to [h] \to (+) \leftarrow n \to g \to [h_r] \to \hat{f}$

$\hat{F} = G \cdot H_r \to H_r = \frac{1}{H}$

$\therefore \hat{F} = \frac{G}{H}$, $G = FH + N$

$\therefore \hat{F} = \frac{FH+N}{H} = F + N/H$

- small H, N dominates.
- ⇒ you will see complete distortion
- ⇒ ∴ BAD!!

Wiener Deconvolution Filtering

$$\hat{F} = \left[\frac{H^* S_f}{|H|^2 S_f + S_n} \right] G$$

$$\hat{F} = \left[\frac{1}{H} \frac{|H|^2}{|H|^2 + S_n/S_f} \right] G$$

H^*: Complex conjugate of H

$S_n = |N|^2$

$S_f = |F|^2$

Wiener Deconvolution Filtering

- Power Spectrum of undegraded image often not known.
- Soln: Replace ratio between noise & image variance by specified const. K.

$$\hat{F} = \left[\frac{1}{H} \frac{|H|^2}{|H|^2 + K} \right] G$$

Demo Img:

a b
c d

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

a) Inverse vs. c) Wiener Deconvolution →

a b c

FIGURE 5.28 Comparison of inverse and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

a b c
d e f
g h i

FIGURE 5.29 (a) 8-bit image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.

§ 5.2 Adaptive Processing - Lee Filter

Global Filters : Drawback

- Apply same process to every point in the image
- All filters so far are Global Filters, so far give no regard to the underlying image characteristics.
- This is problematic since images are generally non-stationary in nature
  (ex. the image characteristics vary from one point to another)
- This result in:
  - Oversmoothing of visually important detail (ex: edges). Edge → Retain

- Undersmoothing of noise in smooth regions in high noise cases. Low Contrast → Smooth

Adaptive Spatial Smoothing Filters

- Solution: change the amount of spatial smoothing based on the underlying characteristics of the image
- What do we want to do?
  - Preserve edges & other image detail
    ⇒ Reduce smoothing where such detail exists
  - Reduce noise that is very visible to the observer
    ⇒ Increase smoothing where smooth regions with little detail exists.
- How do we do that?
  - Local statistical measures. (ex: mean & variance) give a good indication of local image characteristics

LLMSE (Local Linear Minimum Mean Squared Error)

minimize LLMSE

- 'Local': window in the image
- 'linear': in this case, a linear function approximation.
- 'minimum': optimization
- 'mean square error': minimize the difference between the original image & the filtered image.

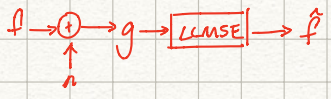
Solution:

- Lee Filter (1980)
  provides LLMMSE solution for the additive noise case.
  - Assume noise added to a signal: (Pristine Img, Noise)

$$g(x,y) = f(x,y) + n(x,y)$$

  - Filter $g(x,y)$ to produce $\hat{f}(x,y)$
  - Assume a linear estimate for $\hat{f}(x,y)$:

(Linear Guess) $\hat{f}(x,y) = \alpha g(x,y) + \beta$, $\alpha, \beta \in \mathbb{R}$



Criterion:

- Use 1D signals (no loss of generality)
- Assume a linear Estimate for $\hat{f}(m)$.

$$\hat{f}(m) = \alpha g(m) + \beta.$$

- We need a criterion & we will resort to the mean square error between the desired image $f(m)$ & the estimate $\hat{f}(m)$
- So, we want to minimize:

$$E\left[\left(\hat{f}(m) - f(m)\right)^2\right] \quad \forall m$$

MSE

Optimization: ⇓

Set $J(\alpha,\beta) = E\left[(\hat{f}(m) - f(m))^2\right]$

→ Partial Derivative

$$J(\alpha,\beta) = E\left[(\alpha g(m) + \beta - f(m))^2\right]$$

$$\frac{\partial J(\alpha,\beta)}{\partial \alpha} = 2E\left[(\alpha g(m) + \beta - f(m))\, g(m)\right] = 0$$

① $\alpha E[g^2(m)] = E[f(m)g(m)] - \beta E[g(m)]$

$\frac{\partial J(\alpha,\beta)}{\partial \beta} = 2E[(\alpha g(m) + \beta - f(m))] = 0$

② $\beta = E[f(m)] - \alpha E[g(m)]$

- Sub ② into ①

$\alpha E[g^2(m)] = E[f(m)g(m)] - (E[f(m)] - \alpha E[g(m)])E[g(m)]$

$\alpha(E[g^2(m)] - E[g(m)]^2) = E[f(m)g(m)] - E[f(m)]E[g(m)]$

① $\alpha = \frac{\overline{f(m)g(m)} - \overline{f(m)}\,\overline{g(m)}}{\overline{g^2(m)} - \overline{g(m)}^2}$ "mean of square minus square of mean"

$= \frac{\sigma^2_{fg(m)}}{\sigma^2_{g(m)}}$ ← ③ $\sigma^2_{g(m)} = \sigma^2_{f(m)} + \sigma^2_n(m)$

Definition of Independence: $E[XY] = E[X]E[Y]$

$\beta = E[f(m)] - \alpha E[g(m)]$

But $E[g(m)] = E[f(m)] + E[n(m)]$ (n(m) mean = 0)

$\therefore \beta = (1-\alpha)E[g(m)] = (1-\alpha)\overline{g(m)}$

② $\sigma^2_{fg}(m) = \overline{f(m)g(m)} - \overline{f(m)}\,\overline{g(m)}$

$= \overline{f(m)(f(m)+n(m))} - \overline{f(m)}^2 = \overline{f^2(m)} + \overline{f(m)n(m)} - \overline{f(m)}^2 = \overline{f^2(m)} + \overline{f(m)}\,\overline{n(m)} - \overline{f(m)}^2$ ($\overline{n(m)} = 0$)

$= \overline{f^2(m)} - \overline{f(m)}^2$

$= \sigma^2_{f(m)}$

①②③ ⇒

$\hat{f}(m) = \alpha g(m) + \beta$

$= \frac{\sigma^2_f(m)}{\sigma^2_f(m)+\sigma^2_n} g(m) + \frac{\sigma_n^2}{\sigma^2_{f(m)}+\sigma^2_n} \overline{g(m)}$

$SNR \cong \sigma^2_f(m)/\sigma^2_n$ Global Noise: $\sigma^2 = \sigma^2_n(m)$

$\hat{f}(m) = \frac{1}{1+1/SNR} g(m) + \frac{1}{1+SNR} \overline{g(m)}$

{ as SNR ↑, $\hat{f}(m) \to g(m)$
as SNR ↓, $\hat{f}(m) \to \overline{g(m)}$

Lee Filter Observations

- As local image variance ↑ increases and becomes higher relative to noise variance, ⇒ filter returns value closer to g.
  - High local image variance is associated with edges & image detail, which need to be preserved (no smoothing)
- As local image variance ↓ decreases, ⇒ filter returns value close to the mean of pixels
  - Low local image variance is associated with smooth regions, which need to be smoothed to remove.
- These observations meet our requirements!

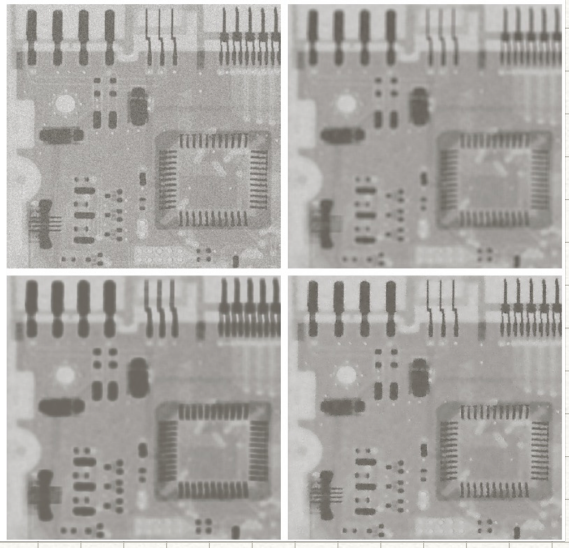
Implementation

Since we don't know f(m), use:

$\hat{f}(m) = \frac{\sigma^2_g(m) - \sigma^2_n}{\sigma^2_g(m)} g(m) + \frac{\sigma_n^2}{\sigma^2_g(m)} \overline{g(m)}$

To implement, calculate local estimates with fixed window size, ex: 7×7

	Outliers	Estimates
Large window	Increase risk of including outliers	Better estimates with more samples
Small window	Decrease risk of outliers affecting statistics	Poorer estimates with fewer samples

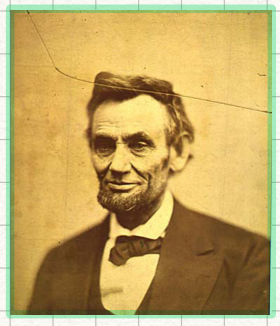


a b
c d

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.

§ 5.3 Digital Inpainting

- **Inpainting:**
  - originated from art restores, who manually fills in parts of a painting, that has cracked off over time.
  - In digital image processing, inpainting refers to the process of automatically restoring missing information in images & videos.

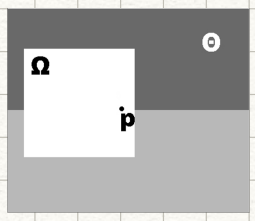


- **Why?** ⇒ ① Remove physical Deterioration: { - cracks, - scratches, - dust

⇒ ② Recover lost blocks in transmission of images & videos

⇒ ③ Remove unwanted image content
  - Power Lines
  - Birds
  - People
  - Text

- **Problem Formulation**
  - Fill in target region Ω, using information from source region Θ ⇒



Inpainting Algorithms

- Digital inpainting algo. generally categorized into two main groups:
  - A) Diffusion-based methods
  - B) Exemplar-based methods.

A) Diffusion-based Methods

- Inspired by the physical Diffusion Process, where molecules spread from areas of high concentration to areas of low concentration to fill a volume
- For digital inpainting, information from source region is "diffused" into the target region to fill in missing info.
- Diffusion in digital image is analogous to repeatedly smoothing (convolution)
- Intuitively, diffusion-based methods repeatedly smooth image content from the source region to the target region until the target region is filled.

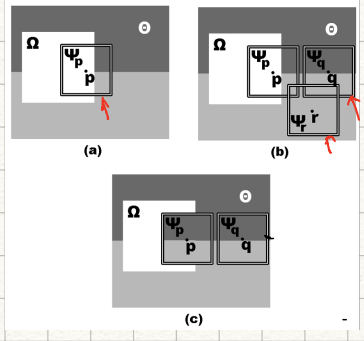
Disadvantage

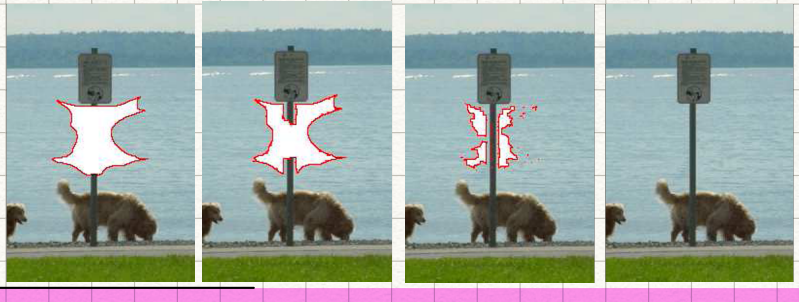
- Appearance of Blurring
  - Very noticeable for large regions & structures.
- Difficult to fill in large regions properly
  - Why?
    - Restricted to using local info.
    - Many situations where local information does not characterize the missing info.

Algo:

- Let Ω be the target region, Θ be the source region.
  - Define boundary $\partial\Theta_1$ in target region Θ
  - Convolve $\partial\Theta_1$ with isotropic smoothing kernel (ex. Gaussian) for a # of iterations
  - Define new boundary $\partial\Theta_2$ in new smaller target region.
  - Repeat process until the entire target region is filled in.


B) Exemplar-based Inpainting





Algo:

- Let Ω be target region
  Θ be source region.
  - Define boundary $\partial\Theta_1$ in target region Θ
  - Find patches with the best match for patches around $\partial\Theta_1$ as exemplars.
    - Similarity between patches can be determined using measures such as mean square error (MSE)
  - Fill patches around $\partial\Theta_1$ with the exemplars
  - Define new boundary $\partial\Theta_2$ in new smaller target region
  - Repeat process till the entire target region is filled in.

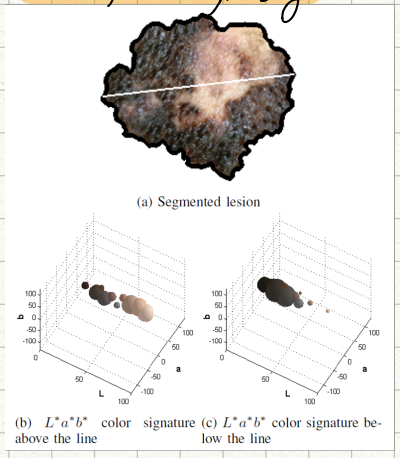
§6 - Color Image Processing

§6.1 Basics

- Understand Light - Object interaction
- More informed image processing

- **Motivation - Biomed. Img Processing**

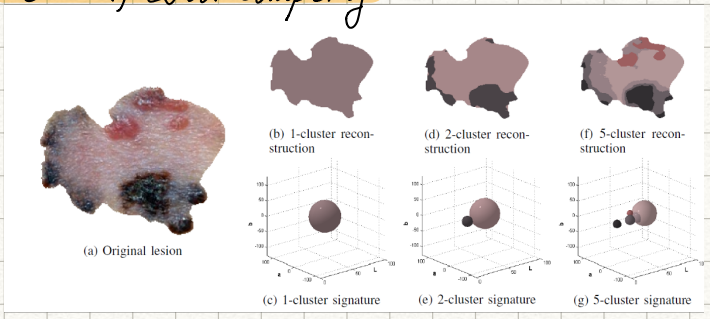
  - **Quantify color asymmetry**



(a) Segmented lesion

(b) $L^*a^*b^*$ color signature above the line (c) $L^*a^*b^*$ color signature below the line

  - **Quantify Colour Complexity**



(a) Original lesion (b) 1-cluster reconstruction (d) 2-cluster reconstruction (f) 5-cluster reconstruction (c) 1-cluster signature (e) 2-cluster signature (g) 5-cluster signature

- **Visible Light**

  - composed of relatively narrow band of freq. in electromagnetic spectrum.
  - Achromatic: grey scale (Blk → White)
  - Chromatic light spans EM spectrum from around 400~700nm

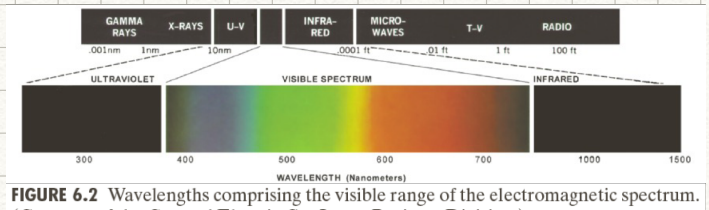


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

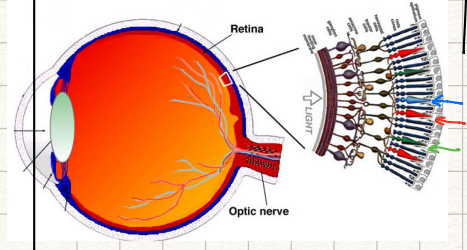
- **Color Perception**

  - Perceived color of an object based on nature of light reflected from object.
  - Ex:
    - If obj. reflects light that is balanced from all visible wavelengths, object is perceived as white
    - If object reflects light with wavelengths mainly in the 575 ~ 625 nm range, object is perceived as red.

- **Cones Revisited**

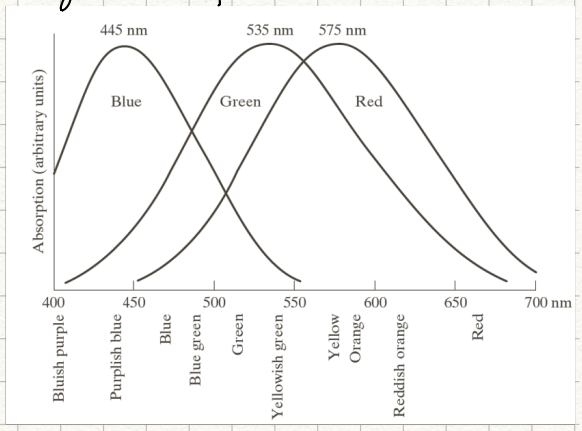
  - 6~7 million cones in the human eyes
  - Divided into 3 main types:
    - L cones (65%) — Max sensitive to long wavelengths (R)
    - M cones (33%) — medium λ (G)
    - S cones (2%) — short λ (B)

⇒ colors can be visualized as weighted combinations of primary RGB.



Cone cells.

Light Absorption of Cones



Quality of a Light Source

↳ 3 Characteristics:

① **Radiance**: total power that flows from / through / to a light source
[Watts per unit area per steradian]

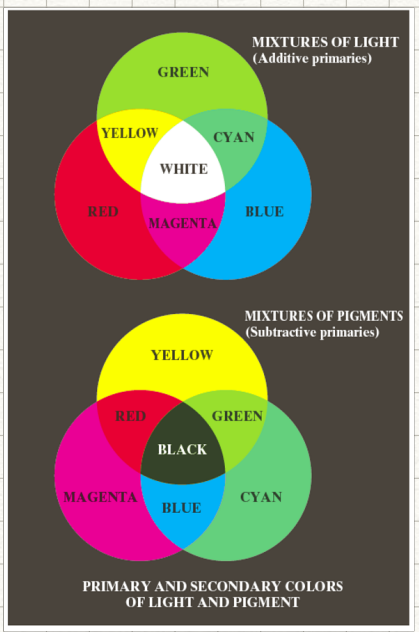
② **Luminance**: Amount of energy observer perceive
[Lumens / lm]
(ex: IR light may have a high radiance, but near zero luminance, since HVS cannot perceive IR)

③ **Brightness**: subjective analogous to achromatic (grayscale) intensity.

Mixture of Lights vs Mixtures of Pigment

↳ Mixture of light primaries (RGB) → Additive

↳ Mixture of pigment primaries → Subtractive.



Color Characteristics → to distinguish one clr from another

① **Brightness**: Intensity (~like grayscale intensity)
② **Hue**: Dominant color perceived by viewer
③ **Saturation**: amount of white mixed with the pure color
(ex: red & white make ⇒ Pink)

Hue + Saturation ⇒ Chromaticity

CIE Chromaticity Diagram

- A method for specifying colors
- Specifies color composition as function of x (~red) and y (~green)
- For any value of x & y, value of z (~blue) can be found as $z = 1-(x+y)$
- The (x, y, z) values of a color specifies percentage of R, G, B needed to form the color
(Trichromatic Coefficients)

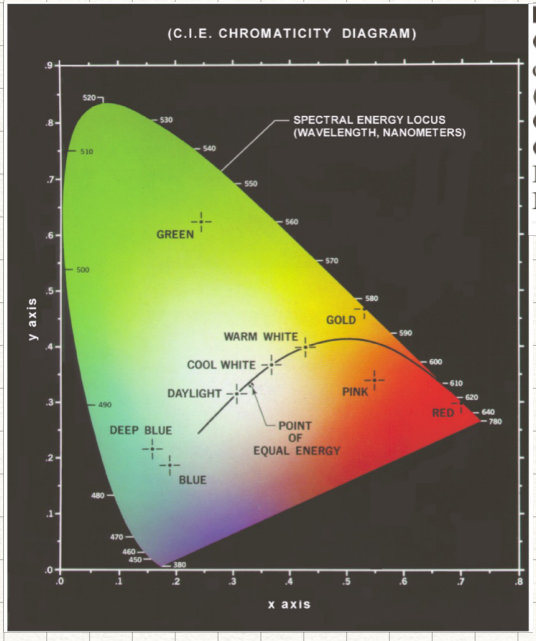


FIGURE 6.5 Chromaticity diagram. (Courtesy of the General Electric Co., Lamp Business Division.)

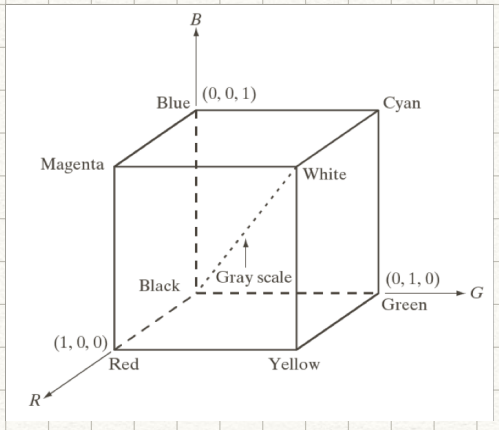
- CIE Chromaticity Diagram Interpretation
  - Pure spectrum colors located around boundary
  - All non-boundary colors are mixture of spectrum colors
  - "Point of equal energy" corresponds to equal fractions of the three primary colors
    - CIE std. for white light
  - Straight line segment joining two points define all colors that can be created by mixing these two colors additively

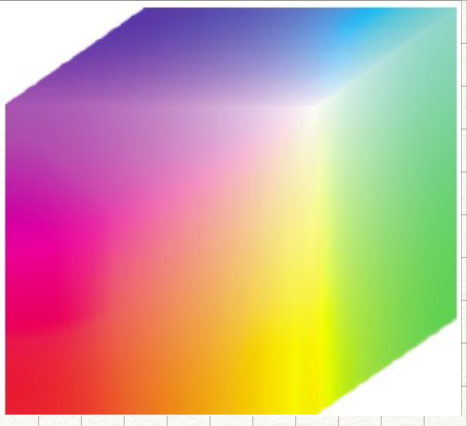
§6.2 Color Models

- Color models (or color spaces or color systems) is a means to standarize colors
- Refers to a coordinate system where each color is represented by a single point

RGB Color Model

- Primarily used for displays and cameras
- Based on Cartesian coordinate system
- Three axis represents intensities of RGB
- Gray scale (points of equal RGB values) extends from black (0, 0, 0) to white (1, 1, 1)
- Ex: 24-bit color (True color)
  - 8-bit (256 lvls) are used to represent each channel
  - Gives a total of $(256)^3 = 16,777,216$ possible colors!





CMY/CMYK Color Models

- Primarily used for printing

- Based on primary colors of pigments
- For CMY, the three axes represent the amount of Cyan, Magenta, & Yellow Pigments to put in to produce a certain color
- Represent as:

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

- In theory, equal amounts of cyan, magenta, & yellow produces black
- In practice, combining them ⇒ results in muddy-looking black

↓

To produce true black in printing, a fourth color (black) is added to produce the CYMK color model

Pros & Cons of RGB:

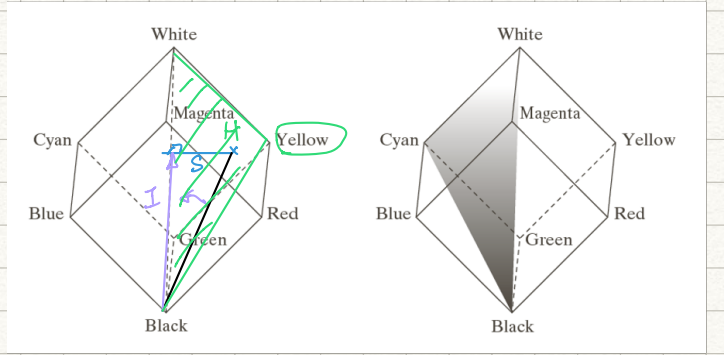
- Pros: - Straightforward (great for hardware implementation)
  - Matches well with HVS's strong response to RGB
- Cons: - Difficult for human description of color.
  (ex: we don't describe color as RGB percentages).
  - Highly redundant & correlated
  (ex: all channels hold Luminance info, ⇒ reduces coding efficiency).
  Compression

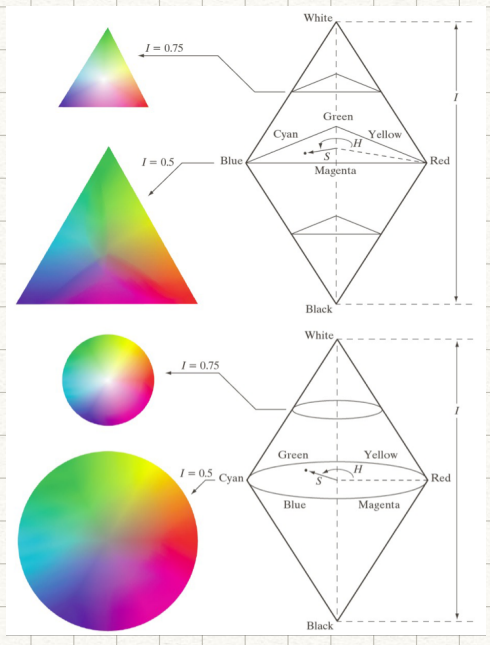
HSI Color Model

- 3 axes: 1) Hue : describes pure/dominant color perceived by observer (ex: pure yellow, red)
  - ↳ All colors on plane identified by White, Black
  - ↳ a pure color corner point have same hue

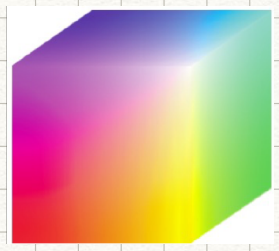
  2) Saturation: (Purity of color): amount of white light mixed with hue
  - ↳ High Saturation ≡ High Purity ≡ Little white ligh mixed with hue.
  - ↳ Distance of associated pure color to intensity axis

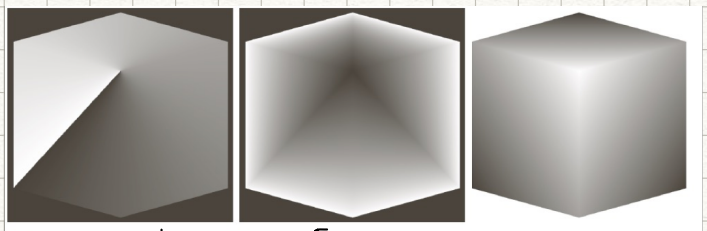
  3) Intensity: Brightness
  - ↳ Projection to gray scale line
- Pro: Useful for human color interpretation.
- Hue + Saturation = color carrying components





Decomposing Img ⇒ HSI (EX):

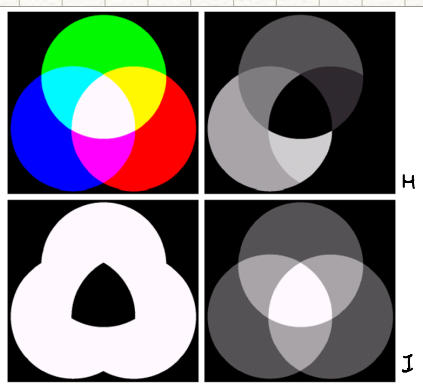




a b c

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

RGB vs. HSI

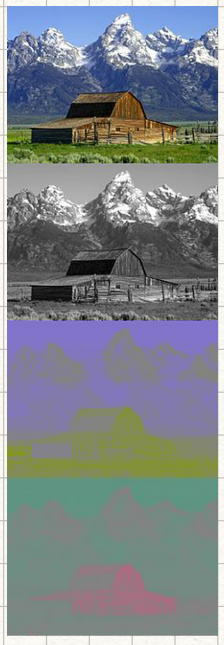


a b
c d

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

YCbCr Color Model

- Useful for image & video compression (ex: JPEG, MPEG)
- 3 Axes: - Y : Luma
  - Cb : Blue diff. = (Blue - Luma)
  - Cr : Red diff. = (Red - Luma)
- Separates Luma from Chroma channels
  ⇒ they can be treated separately
- More closely related to HVS, Recall: Luminance vs. Chromatic Sensitivity



original

Luma ≈ Near grayscale image with most visual information retained

Cb

Cr

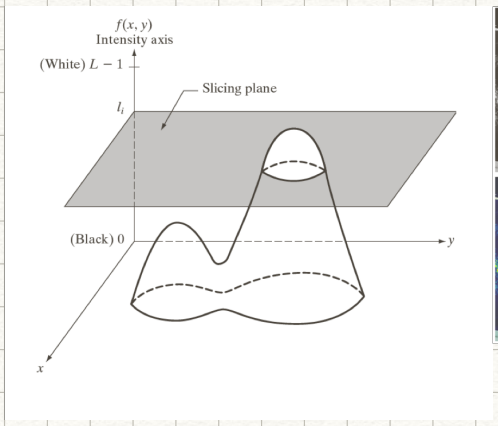
§6.3 Applications

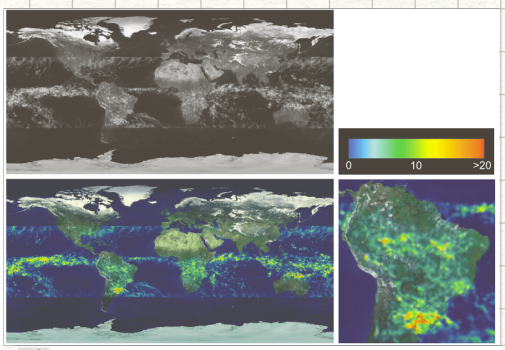
Pseudocolor Image Processing

- Goal: - Assign color to gray levels to convert grayscale image into color image.
- Why? - Improve visualization of image information.
- Motivation: - Humans can discern thousands of color shades but only two dozen or so gray shades.

Intensity Slicing

- One of the simplest methods for pseudocolor image processing
- Grayscale Image can be viewed as 3D function (x, y, and intensity)
- Suppose we define P planes ⊥ intensity axis
- Each plane 'i' is associated with a color 'Ci'
- Pixels with intensities lying along a particular plane 'i' is assigned the color 'Ci' corresponding to the plane.





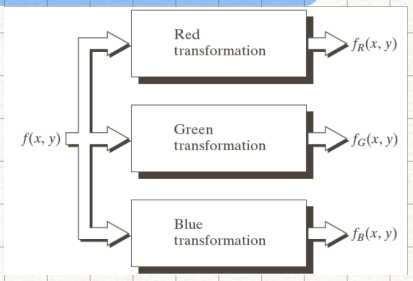
a b
c d

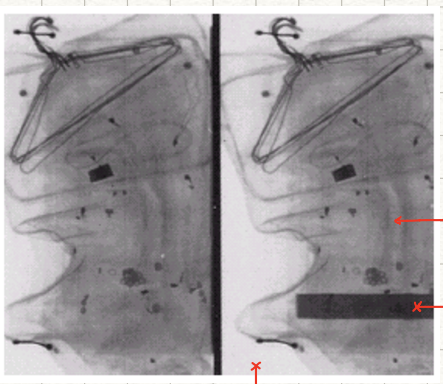
FIGURE 6.22 (a) Gray-scale image in which intensity (in the lighter horizontal band shown) corresponds to average monthly rainfall. (b) Colors assigned to intensity values. (c) Color-coded image. (d) Zoom of the South American region. (Courtesy of NASA.)

Gray Level to Color Transformations

- Intensity slicing limits range of pseudocolor enhancement results
  - Fixed one-to-one relationship between intensity and specified colors
- Alternative Solution:
  - Process grayscale image using independent transformations
  - The results of the transformations are combined to create one composite color image.

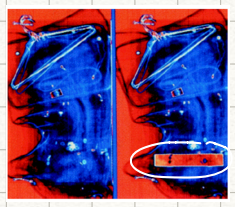
Ex: USE 3 TRANSFORMATIONS

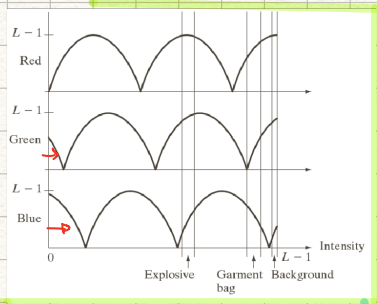




① Garment bag mapped differently than explosive ✓

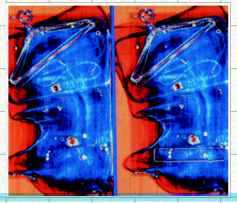
⇒ Easy to spot explosive

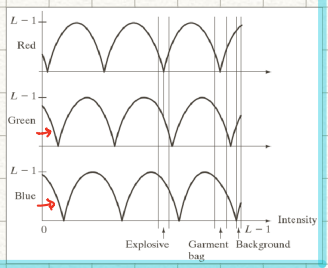




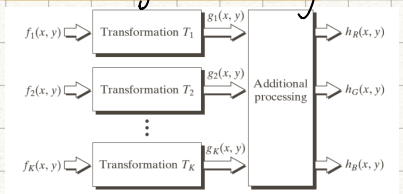
② Garment bag mapped similar than explosive ✗

⇒ Hard to spot explosive





Multi-Image Pseudocoloring



Ex: Multispectral Image Visualization

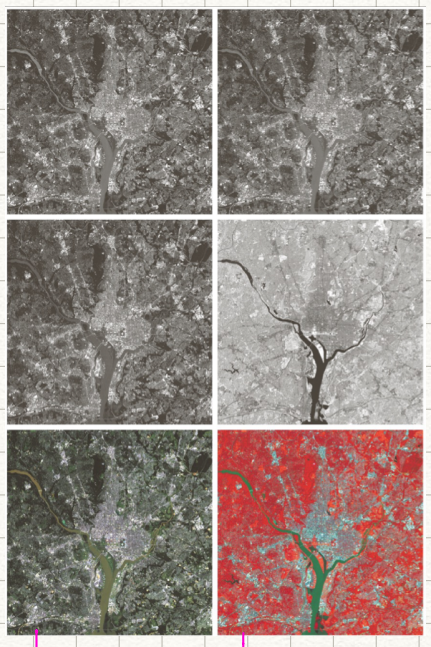


FIGURE 6.27 (a)–(d) Images in bands 1–4 in Fig. 1.10 (see Table 1.1). (e) Color composite image obtained by treating (a), (b), and (c) as the red, green, blue components of an RGB image. (f) Image obtained in the same manner, but using in the red channel the near-infrared image in (d). (Original multispectral images courtesy of NASA.)

combining RGB

combining IR-Green-Blue.

Point Operations In Color Image Processing

- Similar to point processing for grayscale images

$$s_i = T_i(z_1, z_2, \ldots, z_n), \quad i = 1, 2, \ldots, n$$

- Ex: RGB color model
  - $n = 3$
  - z_1, z_2, z_3 denotes R, G, B components of the input image.
- What are Color Complements?
  - Hues opposite one another on the color circle
  - Analogous to grayscale inverse
  - Useful for enhancing details in dark regions of image

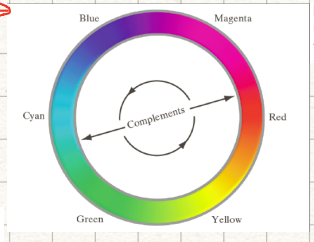


FIGURE 6.32 Complements on the color circle.

Ex

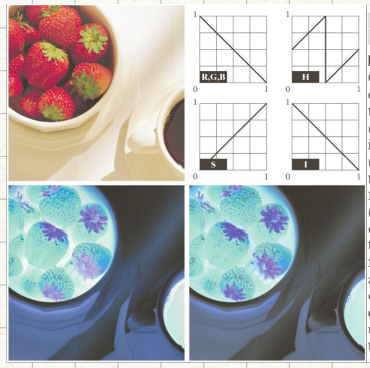


FIGURE 6.33 Color complement transformations. (a) Original image. (b) Complement transformation functions. (c) Complement of (a) based on the RGB mapping functions. (d) An approximation of the RGB complement using HSI transformations.

P/O for Tone Correction

- Tonal Range: general distribution of color intensities
  - Similar to intensity contrast in grayscale images
- High-key Images
  - Colors concentrated at high intensities
- Low-key Images
  - Colors concentrated at low intensities
- As with grayscale images, it is desirable to distribute color intensities evenly

- Adjust tone (brightness and contrast) to improve appearance of image
- Since colors are not changed, all color channels are transformed using the same transformation for color models where intensity information is spread across multiple channels (ex: RGB, CMY)
- For HSI color model, only I channel is modified
- Operations are similar to intensity contrast adjustment for grayscale images

- Tone Correction for Common Tonal Imbalances
  - × Flat images:
    - Use an s-curve transformation to boost contrast
      - lighten highlight areas
      - darken shadow areas
    - Light & Dark Images
      - Similar to power-law transformations.
      - Stretch light regions and compress dark regions for light images (high gamma)
      - Stretch dark regions and compress light regions for dark images (light gamma)

EX

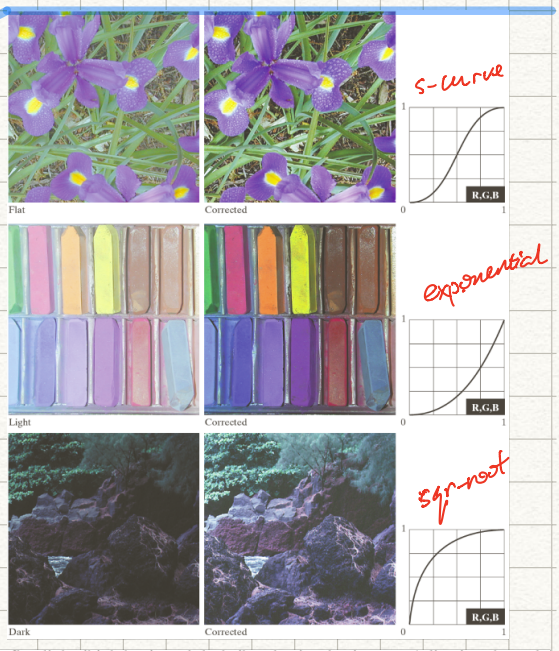


FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not always alter the image hues significantly.

- P/O for Color Correction
  - Various ways to correct color imbalances
  - Perception of a color affected by surrounding colors
  - Proportion of any color (ex: magenta) can be reduced by
    1. Increasing its complementary color (ex: green)
    2. Decreasing portion of the two immediately adjacent colors (ex: red & blue)

EX

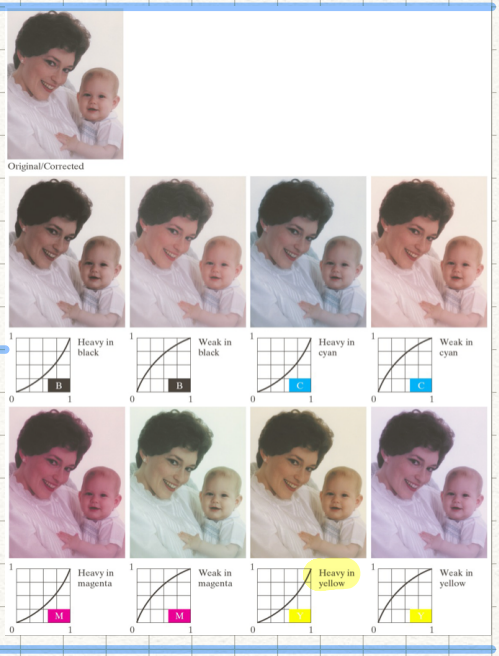


FIGURE 6.36 Color balancing corrections for CMYK color images.

- Color Image Processing
  - Given 3 planes, how to smooth?
    - option A) Smooth each RGB image independently
    - option B) Convert to HSI & smooth the Intensity component only.
  - Preferable to use the HSI approach since will only affect the intensity and not the colors themselves

A) RGB Components

a b
c d

FIGURE 6.38 (a) RGB image. (b) Red component image. (c) Green component. (d) Blue component.

Smooth RGB Channels Independently

B) Intensity Component (HSI)

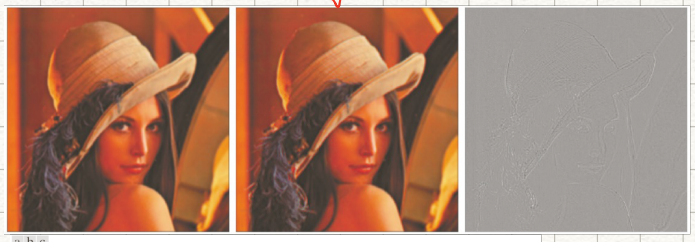
a b c

FIGURE 6.39 HSI components of the RGB color image in Fig. 6.38(a). (a) Hue. (b) Saturation. (c) Intensity.

(Smoothing/Edge) Process Only the I component
(Does not affect Color Component)

A) vs. B) Smoothing Results

- No Difference with 5×5 Avg Mask

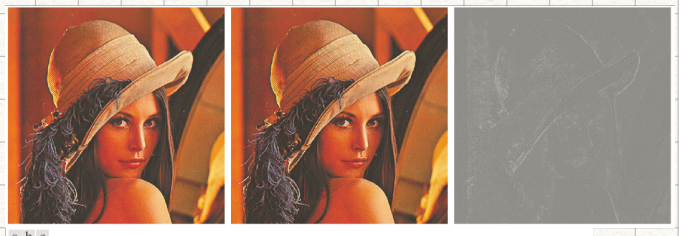


a b c

FIGURE 6.40 Image smoothing with a 5 × 5 averaging mask. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results.

A) vs. B) Image Sharpening

⇒ Some diff, But not noticeable



a b c

FIGURE 6.41 Image sharpening with the Laplacian. (a) Result of processing each RGB channel. (b) Result of processing the HSI intensity component and converting to RGB. (c) Difference between the two results.