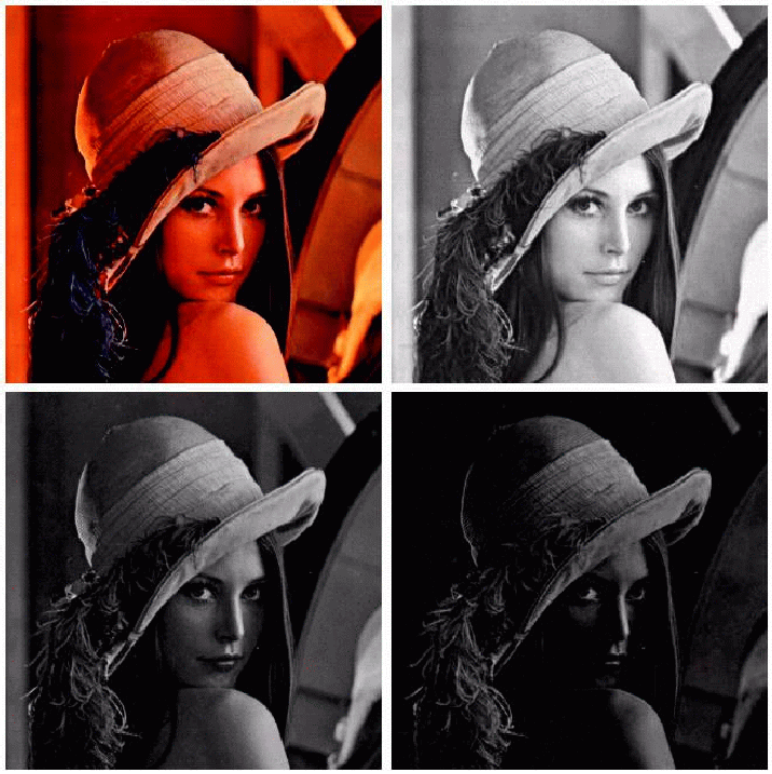
6.6 Smoothing and sharpening (Original images...)



a b
c d

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

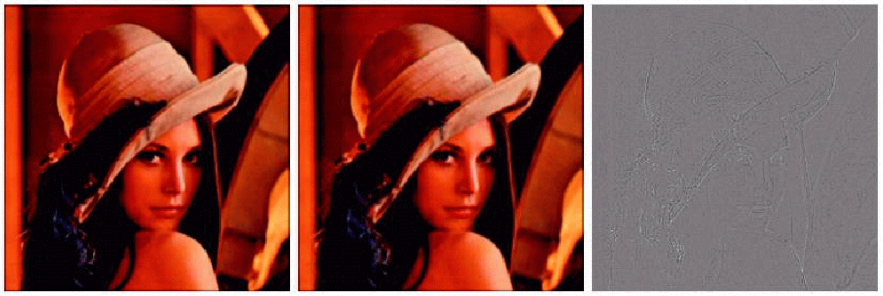
a b c

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

6.6.1 Colour image smoothing: • **Smoothing by neigbourhood averaging can be performed using RGB colour vectors,** $\overline{\mathbf{c}}(x,y) = (1/K) \sum_{(x,y)\in S_{xy}} \mathbf{c}(x,y)$ **or can be carried out on a per-colour-plane basis**

$$\overline{\mathbf{c}}(x,y) = \begin{pmatrix} (1/K) \sum_{(x,y)\in S_{xy}} R(x,y) \\ (1/K) \sum_{(x,y)\in S_{xy}} G(x,y) \\ (1/K) \sum_{(x,y)\in S_{xy}} B(x,y) \end{pmatrix}$$

Example 6.12:



a b c

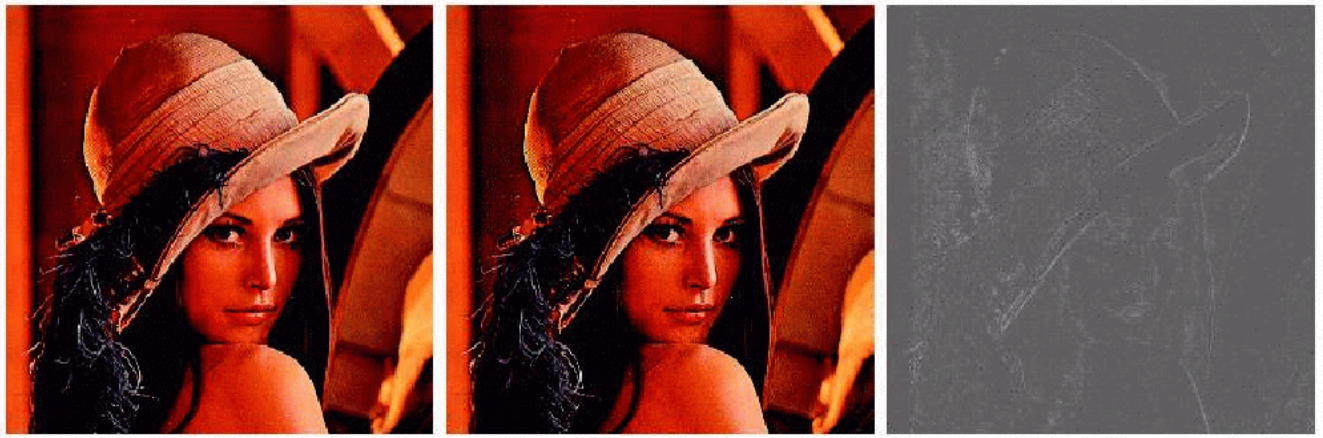
FIGURE 6.40 Image smoothing with a 5 × 5 averaging mask. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results.

6.6.2 Colour image sharpening

Similar to image smoothing, image sharpening can also be performed on a per-colour-plane basis (using the Laplacian),

$$\nabla^2[\mathbf{c}(x,y)] = \begin{pmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{pmatrix}$$

Example 6.13:

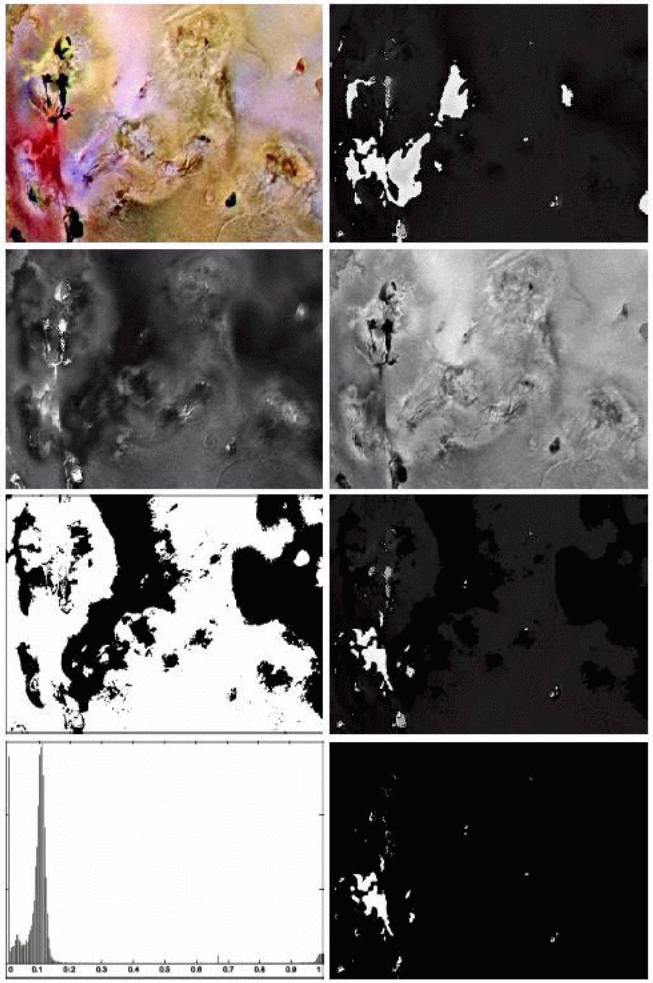


a b c

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

6.7 Colour segmentation

6.7.1 Segmentation in HSI colour space: Example 6.14



a b
c d
e f
g h

FIGURE 6.42 Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components in (a).

6.7.2 Segmentation in RGB vector space

Given a set of representative colour points, calculate average colour ($\mathbf{a}$) and (if necessary) covariance matrix ($\mathbf{C}$)

Let $\mathbf{z}$ denote an arbitrary point in RGB space

Three similarity measures:

(1) Euclidean distance

$$\begin{aligned} D(\mathbf{z},\mathbf{a}) &= ||\mathbf{z}-\mathbf{a}|| \\ &= [\,(\mathbf{z}-\mathbf{a})^T(\mathbf{z}-\mathbf{a})\,]^{\frac{1}{2}} \\ &= [\,(z_R-a_R)^2+(z_G-a_G)^2+(z_B-a_B)^2\,]^{\frac{1}{2}} \end{aligned}$$

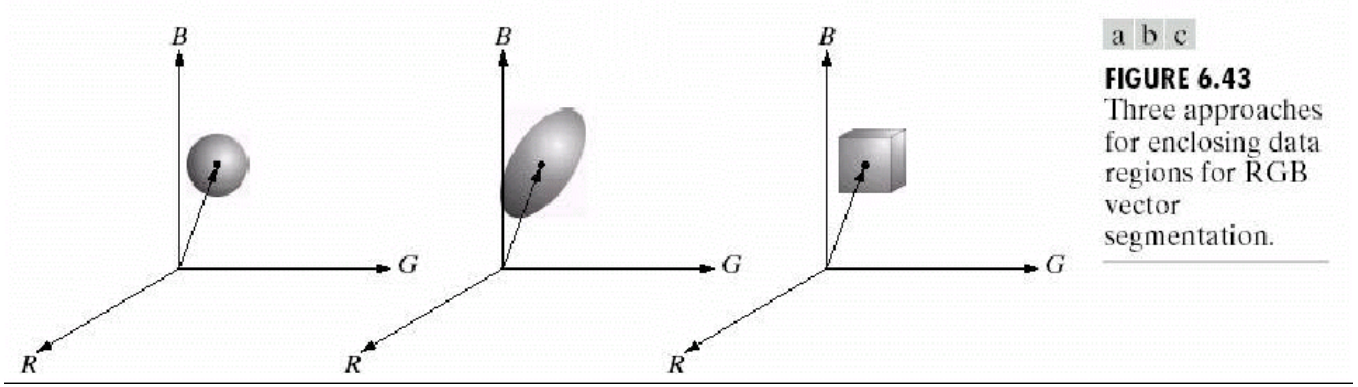
$D(\mathbf{z},\mathbf{a}) \leq D_0 \Rightarrow$ **points within solid spherical region**

(2) "Mahalanobis"-like distance

$$D(\mathbf{z},\mathbf{a}) = [\,(\mathbf{z}-\mathbf{a})^T\mathbf{C}^{-1}(\mathbf{z}-\mathbf{a})\,]^{\frac{1}{2}}$$

$D(\mathbf{z},\mathbf{a}) \leq D_0 \Rightarrow$ **points within solid ellipsoidal region**

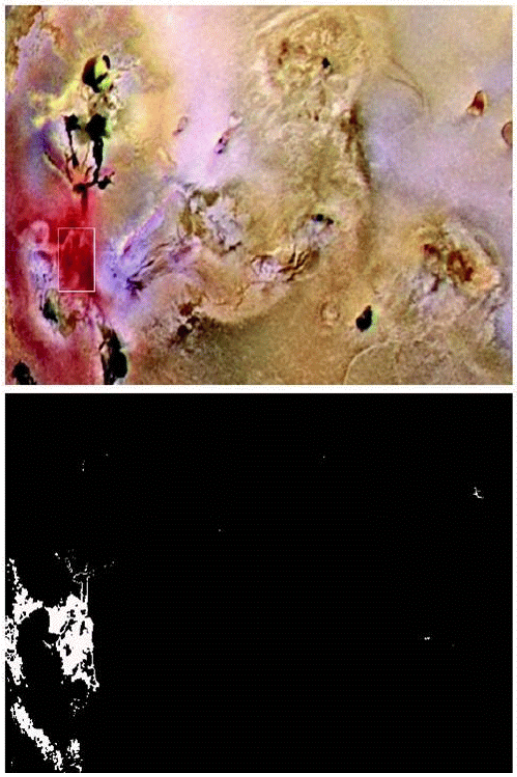
(3) Bounding box (computationally less expensive)
Centered at $\mathbf{a}$, dimensions proportional to standard deviations (σ_R, σ_G, σ_B) along colour axes



a b c

FIGURE 6.43 Three approaches for enclosing data regions for RGB vector segmentation.

Example 6.15: Colour image segmentation in RGB space

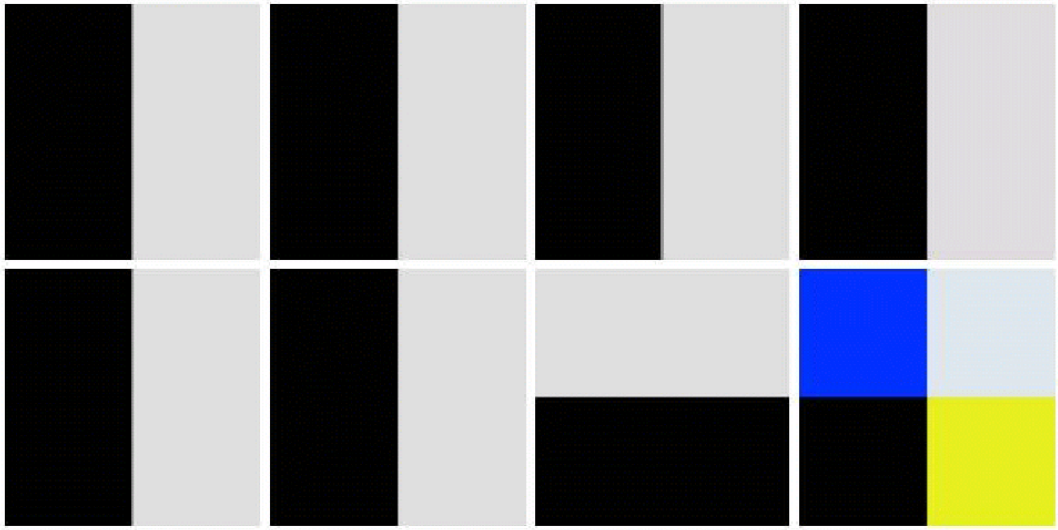


a
b

FIGURE 6.44 Segmentation in RGB space. (a) Original image with colors of interest shown enclosed by a rectangle. (b) Result of segmentation in RGB vector space. Compare with Fig. 6.42(h).

6.7.3 Colour edge detection

Computing the gradient on individual (R,G,B) images (e.g. Sobel operators) and then using the results to form a colour image lead to erroneous results! Example...



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. (f)–(g) R, G, and B component images and (h) resulting RGB color image.

Only consider gradient in middle of images. When we calculate gradient of respective component images and add results to form two RGB gradient images, the value of RGB gradient (in middle) is the same. This is clearly not the case here!

Edge detection in vector (e.g. RGB) space

Sobel operators can not be used. Di Zenzo's method:

$$\mathbf{u} = \frac{\partial R}{\partial x}\mathbf{r} + \frac{\partial G}{\partial x}\mathbf{g} + \frac{\partial B}{\partial x}\mathbf{b}, \quad \mathbf{v} = \frac{\partial R}{\partial y}\mathbf{r} + \frac{\partial G}{\partial y}\mathbf{g} + \frac{\partial B}{\partial y}\mathbf{b}$$

$$g_{xx} = \mathbf{u}\cdot\mathbf{u} = \mathbf{u}^T\mathbf{u} = \left|\frac{\partial R}{\partial x}\right|^2 + \left|\frac{\partial G}{\partial x}\right|^2 + \left|\frac{\partial B}{\partial x}\right|^2$$

$$g_{yy} = \mathbf{v}\cdot\mathbf{v} = \mathbf{v}^T\mathbf{v} = \left|\frac{\partial R}{\partial y}\right|^2 + \left|\frac{\partial G}{\partial y}\right|^2 + \left|\frac{\partial B}{\partial y}\right|^2$$

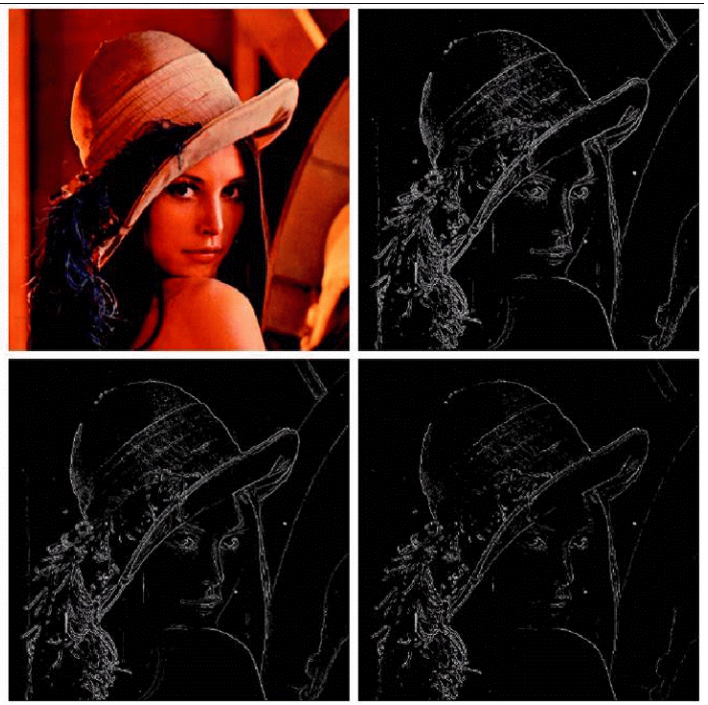
$$g_{xy} = \mathbf{u}\cdot\mathbf{v} = \mathbf{u}^T\mathbf{v} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial y}$$

Direction of maximum rate of change of $\mathbf{c}(x,y)$: $\theta = \frac{1}{2}\tan^{-1}\left\{\frac{2\,g_{xy}}{(g_{xx}-g_{yy})}\right\}$

Value of rate of change at (x,y) in direction of θ:

$$F(\theta) = \left\{\frac{1}{2}[\,(g_{xx}+g_{yy}) + (g_{xx}-g_{yy})\cos(2\,\theta) + 2g_{xy}\sin(2\,\theta)\,]\right\}^{\frac{1}{2}}$$

Example 6.16: Edge detection in vector (RGB) space



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).

6.8 Noise in colour images

The noise content usually has the same characteristics in each colour channel, but channels can be affected differently when, e.g., the electronics of a particular channel malfunctions

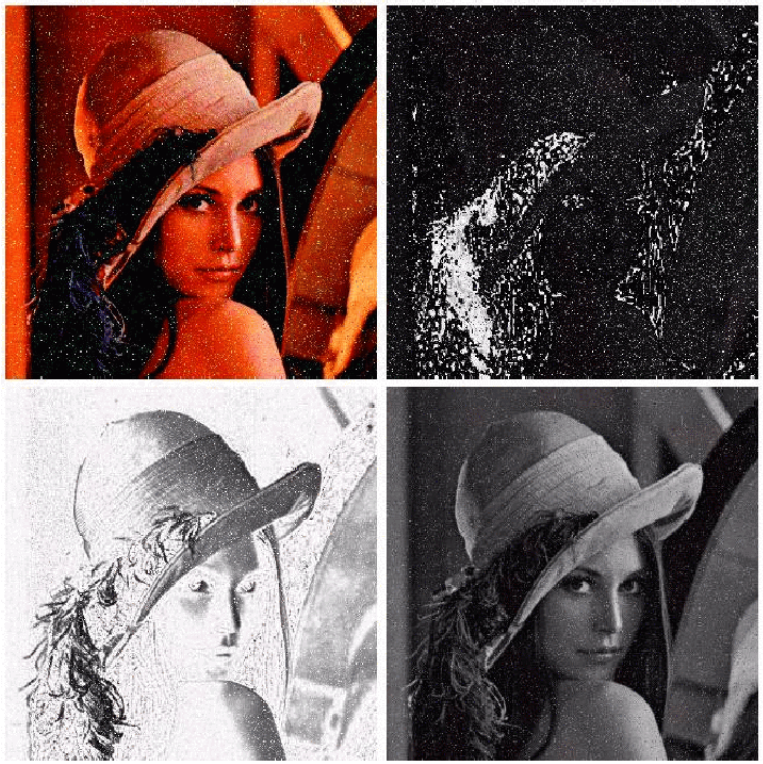
Example 6.17: Converting noisy RGB images to HSI

a b
c d

FIGURE 6.48
(a)–(c) Red, green, and blue component images corrupted by additive Gaussian noise of mean 0 and variance 800. (d) Resulting RGB image. [Compare (d) with Fig. 6.46(a).]

a b c

FIGURE 6.49 HSI components of the noisy color image in Fig. 6.48(d). (a) Hue. (b) Saturation. (c) Intensity.



a b
c d

FIGURE 6.50
(a) RGB image with green plane corrupted by salt-and-pepper noise. (b) Hue component of HSI image. (c) Saturation component. (d) Intensity component.

Noise reduction using an averaging filter (linear) gives the same results in vector space as it does if the component images are processed independently. This is not the case for other filters, e.g. order statistics filters (vector ordering is beyond the scope of this course)

6.9 Colour image compression

Example 6.18: JPEG 2000 compression ($C_R = 230 : 1$)

a b
c d

FIGURE 6.51 Color image compression. (a) Original RGB image. (b) Result of compressing and decompressing the image in (a).

Homework

Due: next week

a1-colorprocessing