



ELEG 404/604 Digital Image Processing

Gonzalo R. Arce

Color Image Processing

Department of Electrical and Computer
Engineering

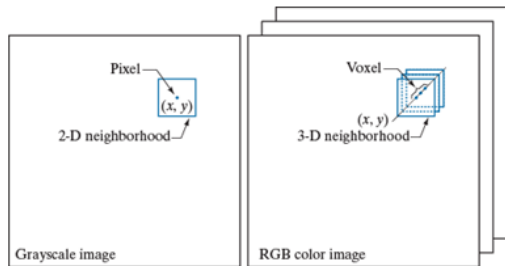
University of Delaware,
Newark, Delaware, 19716

Basics of Full-Color Image Processing

- Per-component-image processing
- Vector-based processing

$$\mathbf{c}(x, y) = \begin{bmatrix} c_R(x, y) \\ c_G(x, y) \\ c_B(x, y) \end{bmatrix} = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix}$$

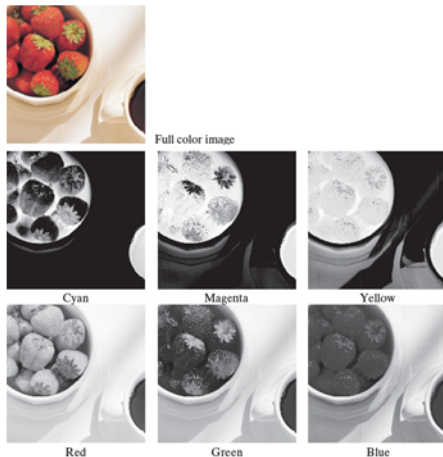
a b
FIGURE 7.27
Spatial neighborhoods for grayscale and RGB color images. Observe in (b) that a *single* pair of spatial coordinates, (x, y) , addresses the same spatial location in all three images.



Color Transformations

$$s_i = T_i(r_i) \quad i = 1, 2, \dots, n$$

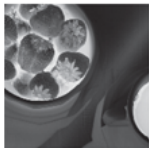
- $n \rightarrow$ number of component images
- $T_i \rightarrow$ set of transformation or color mapping functions
- Performed in any color model
- Some are better suited to specific models



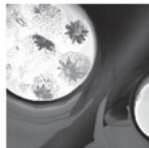
Color Transformations



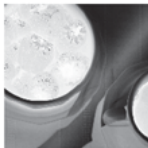
Full color image



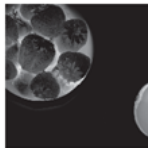
Cyan



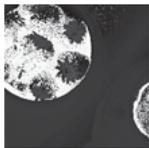
Magenta



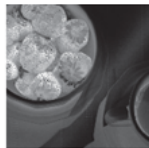
Yellow



Black



Hue



Saturation



Intensity

Modifying the intensity of a full-color image

$$s_i = T_i(r_i) \quad i = 1, 2, \dots, n$$

- HSI color space

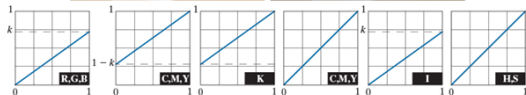
$$s_1 = r_1$$

$$s_2 = r_2$$

$$s_3 = kr_3$$

- RGB color space

$$s_i = kr_i \quad i = 1, 2, 3$$



a b
c d e f g h

FIGURE 7.29 Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c) The required RGB mapping function. (d)–(e) The required CMYK mapping functions. (f) The required CMY mapping function. (g)–(h) The required HSI mapping functions. (Original image courtesy of MedData Interactive.)

Modifying the intensity of a full-color image

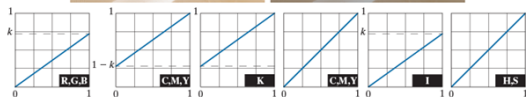
$$s_i = T_i(r_i) \quad i = 1, 2, \dots, n$$

- CMY color space

$$s_i = kr_i + (1 - k) \quad i = 1, 2, 3$$

- CMYK color space

$$s_i = \begin{cases} r_i & i = 1, 2, 3 \\ kr_i + (1 - k) & i = 4 \end{cases}$$



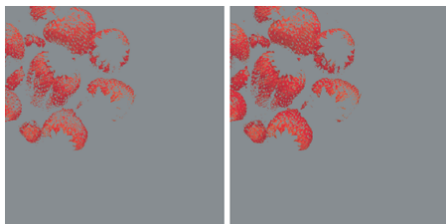
a b
c d e f g h

FIGURE 7.29 Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c) The required RGB mapping function. (d)–(e) The required CMY mapping functions. (f) The required CMY mapping function. (g)–(h) The required HSI mapping functions. (Original image courtesy of MedData Interactive.)

Color Slicing

- Stand out a color of interest from the background.
- Use the region defined as a mask for further processing, e.g. segmentation.
- Vector-based processing

$$s_i = \begin{cases} 0.5 & \text{if } [|r_j - a_j| > \frac{W}{2}]_{\text{any } 1 \leq j \leq n} \\ r_i & \text{otherwise} \end{cases} \quad i = 1, 2, \dots, n$$



a b

FIGURE 7.32 Color-slicing transformations that detect (a) reds within an RGB cube of width $W = 0.2549$ centered at $(0.6863, 0.1608, 0.1922)$, and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color $(0.5, 0.5, 0.5)$.

Tonal Corrections

- Stand out a color of interest from the background.
- Equally distribution of the intensities of a color image is desired.
- Adjust image's brightness and contrast.
- S-shape curve is used for boosting contrast.
- Power-law transformations.

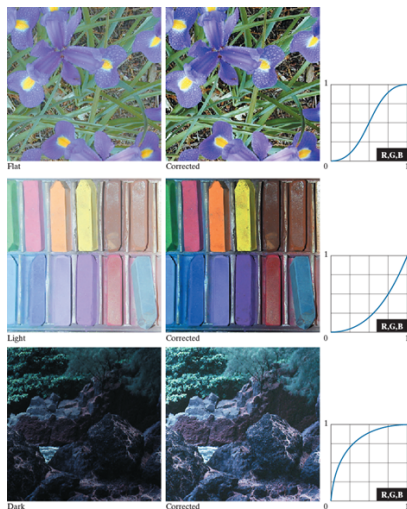


FIGURE 7.33 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not always alter the image hues significantly.

Color Balancing

- White balance is used to correct the effect of light.

$$s_i = \frac{W_s}{W_r} r_i$$

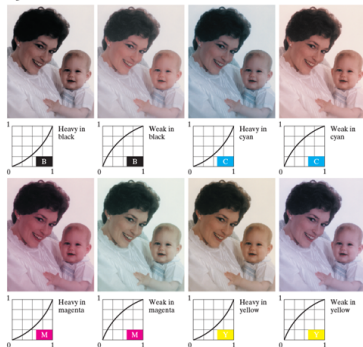
where W_s represents a destination color and W_r represents a source color.

- Skin tones are also used for color balance.



Original/Corrected

FIGURE 7.34 Color balancing a CMYK image.

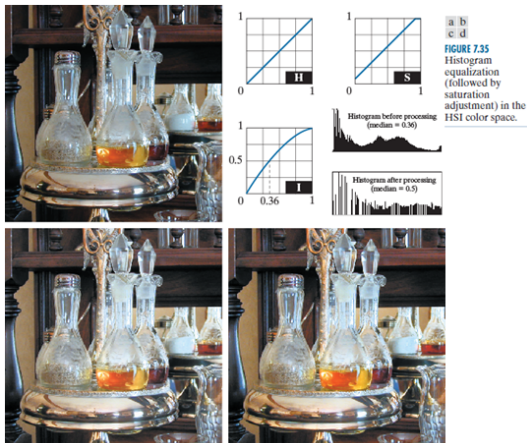


Histogram Processing of Color Images

- Transformation that produces an image with a uniform histogram.

$$s_k = (L - 1) \sum_{j=0}^k p_r(r_j)$$
$$k = 0, 1, 2, \dots, L - 1$$

- Would it be wise to equalize color components independently?
- Alter colors of the image.
- HSI color space is ideal.



Color Image Smoothing

- Per-component-image processing
- Vector-based processing

$$\bar{\mathbf{c}}(x, y) = \begin{bmatrix} \frac{1}{K} \sum_{(s,t) \in S_{xy}} R(s, t) \\ \frac{1}{K} \sum_{(s,t) \in S_{xy}} G(s, t) \\ \frac{1}{K} \sum_{(s,t) \in S_{xy}} B(s, t) \end{bmatrix}$$



a b c

FIGURE 7.38 Image smoothing with a 5×5 averaging kernel. (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.

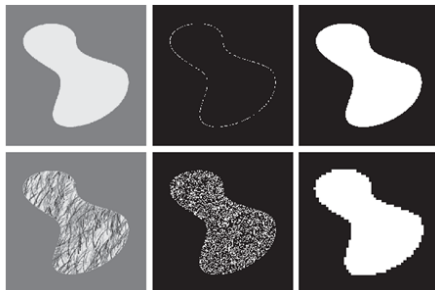
Segmentation

- Partitioning an image into a collection of regions or objects based on:
 - Discontinuities (edge-based).
 - Similarity (predefined criteria).

a b c
d e f

FIGURE 10.1

(a) Image of a constant intensity region.
(b) Boundary based on intensity discontinuities.
(c) Result of segmentation.
(d) Image of a texture region.
(e) Result of intensity discontinuity computations (note the large number of small edges).
(f) Result of segmentation based on region properties.



Segmentation

- R represent the entire region occupied by the image.

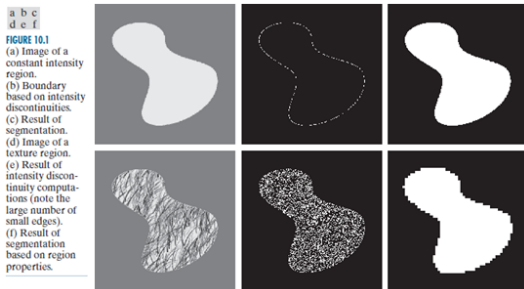
$$\bigcup_{i=1}^n R_i = R$$

R_i is a connected set, for $i = 0, 1, 2, \dots, n$

$R_i \cap R_j = \emptyset$ for all i and $j, i \neq j$

$Q(R_i) = \text{TRUE}$ for $i = 0, 1, 2, \dots, n$

$Q(R_i \cup R_j) = \text{FALSE}$ for any adjacent region R_i and R_j



Segmentation in HSI Color Space

- Saturation is used as a mask.
- Intensity is not used for segmentation of color images.
- Range of hue values of the regions of interest is used as descriptor.
- The product of the mask with the hue is determined.
- The segmented image is obtained by thresholding this product.

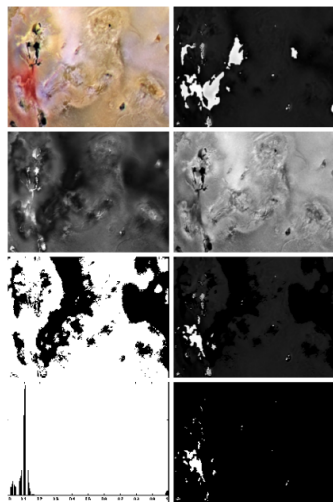
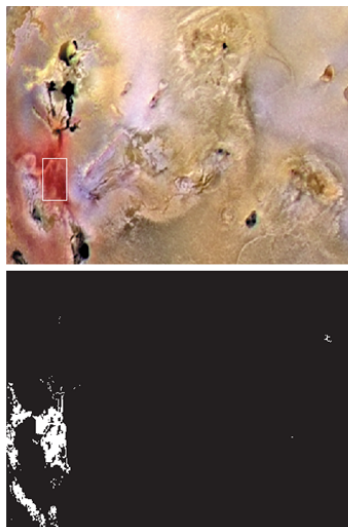


FIGURE 7.40 Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components from (a).

Segmentation in RGB Color Space

- Samples of color of interest (**a**).
- Classify each pixel in the image (**z**).



a
b
FIGURE 7.42
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. 7.40(h).

Segmentation in RGB Space

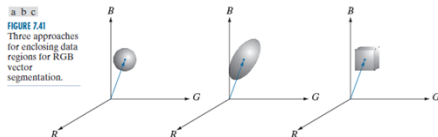
- Euclidean distance to measure of similarity

$$D(\mathbf{z}, \mathbf{a}) = \|\mathbf{z} - \mathbf{a}\|$$

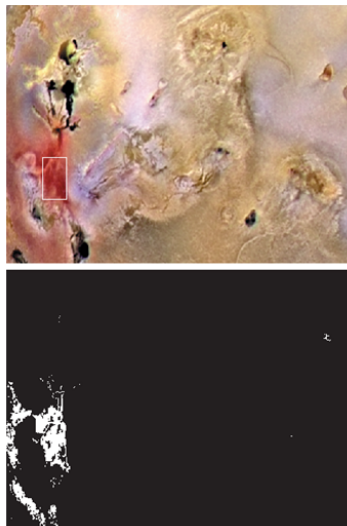
$$D(\mathbf{z}, \mathbf{a}) = [(\mathbf{z} - \mathbf{a})^T (\mathbf{z} - \mathbf{a})]^{1/2}$$

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

C is the covariance matrix of the samples.



Segmentation in RGB Color Space



a
b

FIGURE 7.42

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. 7.40(h).



Segmentation in Lab Color Space

fabric



- How many colors can you distinguish from the background?

Segmentation in Lab Color Space

fabric



- How many colors can you distinguish from the background?
- Six: red, green, purple, yellow, and magenta

Segmentation in Lab Color Space - Nearest Neighbor Classification

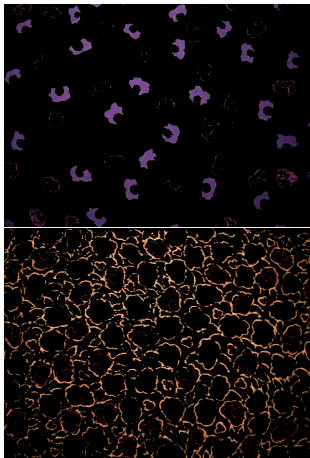
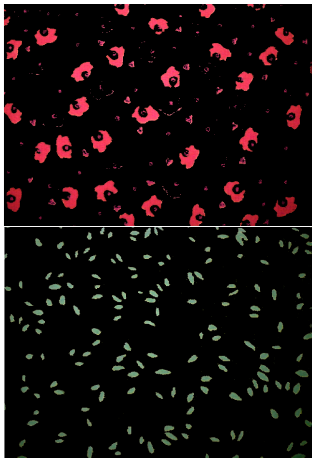
- Uniform changes of components in the Lab color space correspond to uniform changes in perceived color.
- Perceived color differences are measured by Euclidean distances.
- Segmentation can be performed by means of clustering.

There are K clusters with sample mean \mathbf{a}_k .

$$D_k(\mathbf{z}, \mathbf{a}) = \|\mathbf{z} - \mathbf{a}_k\| \quad k = 1, 2, \dots, N$$

Every pixel is assigned to the class that minimizes the color difference.

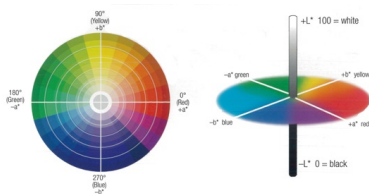
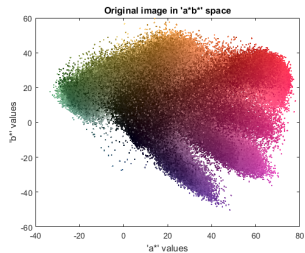
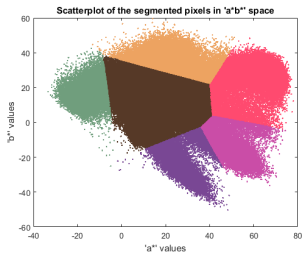
Segmentation in Lab Color Space



<https://www.mathworks.com/help/images/examples/color-based-segmentation-using-the-l-a-b-color-space.html>

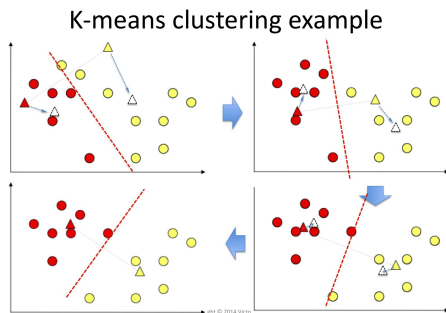


Segmentation in Lab Color Space



Segmentation using k-means clustering

- Partition of a set Q , of observations into a specified number, k , of clusters.
- Assign to the cluster with the nearest mean.
- Iterative procedure.



Segmentation using k-means clustering

- Let \mathbf{Z} be the color pixel dataset of the form

$$\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_Q\}$$

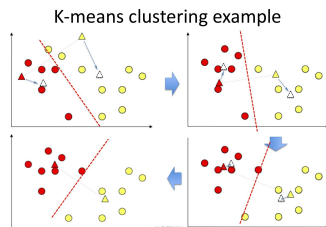
where $\mathbf{z} \in R^n$

- We want to classify the data into k disjoint sets of the form

$$C = \{C_1, C_2, \dots, C_k\}$$

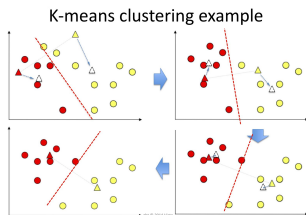
such that the criterion of optimality is satisfied

$$\arg \min_C = \left(\sum_{i=1}^k \sum_{\mathbf{z} \in C_i} \|\mathbf{z} - \mathbf{m}_i\|^2 \right)$$



k-means Algorithm

- Initialize the algorithm:
 $\mathbf{m}_i(1), i = 1, 2, \dots, k$



- Assign samples to clusters whose mean is the closest.

$$\mathbf{z}_q \rightarrow C_i \quad \text{if } \|\mathbf{z}_q - \mathbf{m}_i\|^2$$

$$j = 1, 2, \dots, k (j \neq i); \quad q = 1, 2, \dots, Q$$

- Update the clusters' means

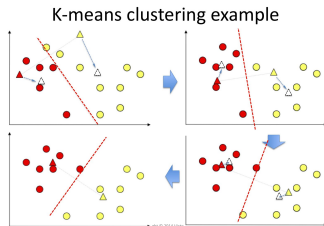
$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{\mathbf{z} \in C_i} \mathbf{z} \quad i = 1, 2, \dots, k$$

where $|C_i|$ is the number of samples in cluster set C_i .



k-means Algorithm

- Compute residual error, E , as the sum of the k Euclidean norms of the differences between the mean vectors in the current and previous steps. Stop if $E \leq T$, where T is a specified threshold.



Segmentation using k-means clustering in different Color Spaces



Figure 5.7 (a) Original color image. Results of segmentation using the k -means clustering algorithm in three different color spaces: (b) $sRGB$, (c) HSV , and (d) $L^*a^*b^*$. In each case, the image has been segmented into six regions. The color assigned to each region is the final centroid of the color values in the corresponding region in the original image after application of the k -means algorithm.

Noise in Color Images

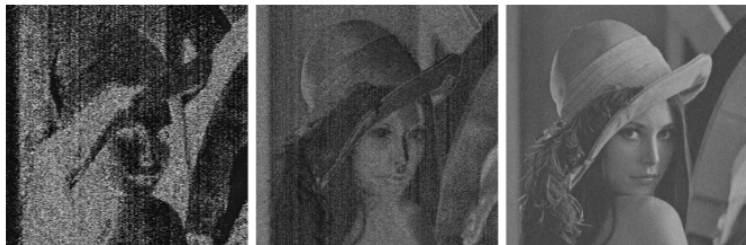


FIGURE 7.47 HSI components of the noisy color image in Fig. 7.46(d). (a) Hue. (b) Saturation. (c) Intensity.

Noise in Color Images



a b
c d

FIGURE 7.48

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

