ELEG404/604: Digital Imaging & Photography

Gonzalo R. Arce

Department of Electrical and Computer Engineering
University of Delaware

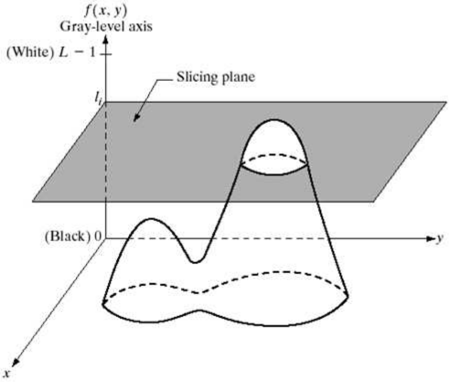
Chapter IX

Pseudocolor Image Processing

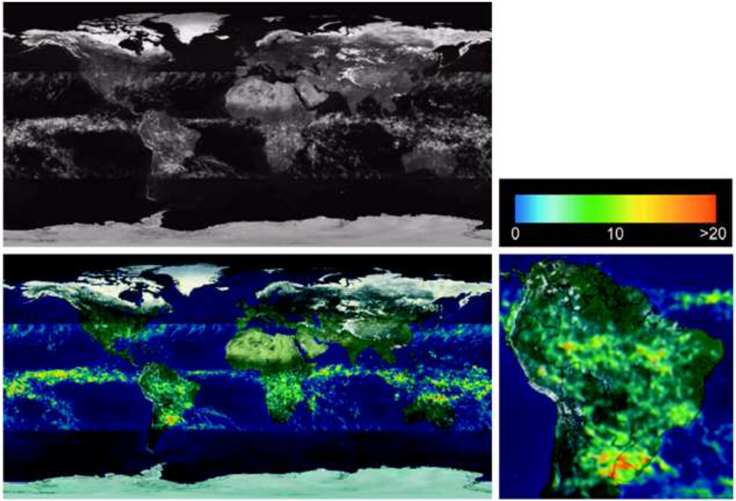
- Assigning colors to gray values yields Pseudocolor (false color) images
- Assignment criteria is application-specific
- Intensity (density) slicing
  - Assign colors based on gray value relation to slicing plane

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

where c_k is the color associated with the kth intensity interval I_k.



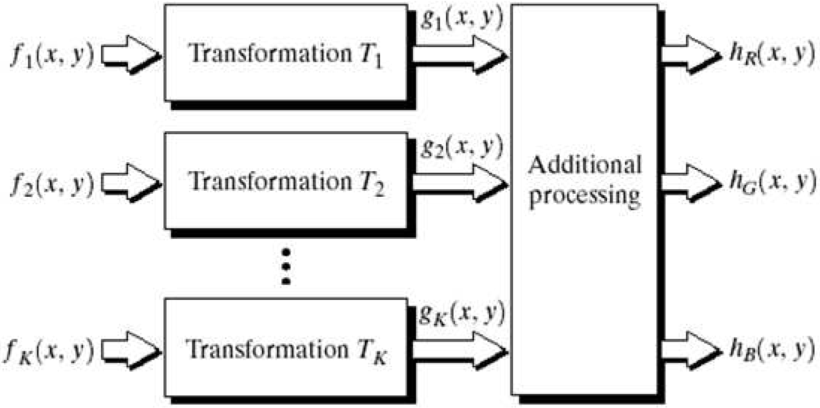
Density Slicing Example



a b
c d

FIGURE (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 America region. (Courtesy of NASA.)

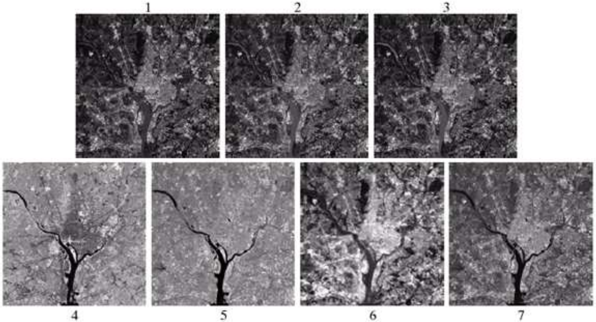
Multispectral Extensions



- Pseudocolor is often used in the visualization of multispectral images
  - Example of satellite images: visible spectrum, infrared, radio waves, etc.
  - Transformations are application and spectral band dependent

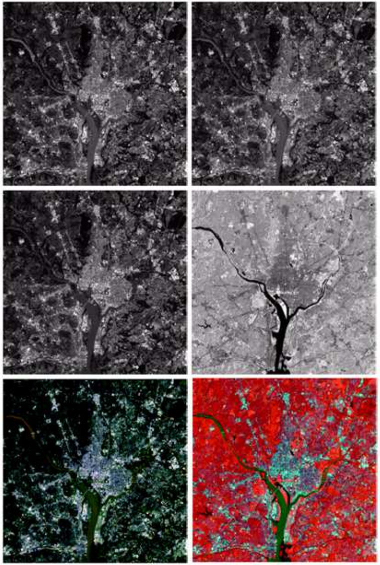
Washington DC LANDSAT Example (I)

Band No.	Name	Wavelength (μm)	Characteristics and Uses
1	Visible blue	0.45–0.52	Maximum water penetration
2	Visible green	0.52–0.60	Good for measuring plant vigor
3	Visible red	0.63–0.69	Vegetation discrimination
4	Near infrared	0.76–0.90	Biomass and shoreline mapping
5	Middle infrared	1.55–1.75	Moisture content of soil and vegetation
6	Thermal infrared	10.4–12.5	Soil moisture; thermal mapping
7	Middle infrared	2.08–2.35	Mineral mapping



Wash. DC LANDSAT Example (I)

- Images in bands 1-4
- Bottom left: color composite image using
  - Band 1 (visible blue) as blue
  - Band 2 (visible green) as green
  - Band 3 (visible red) as red
  - Result is difficult to analyze
- Bottom right: color composite image using
  - Bands 1 and 2 as above
  - Band 4 (near infrared) as red
  - Better distinguishes between biomass (red dominated) and man-made structures



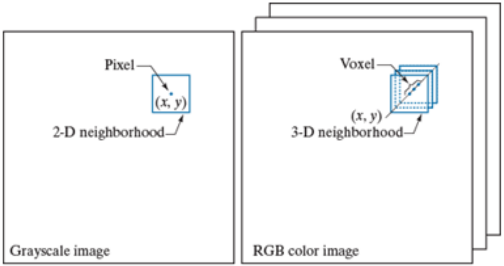
Basics of Full-Color Image Processing

- 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

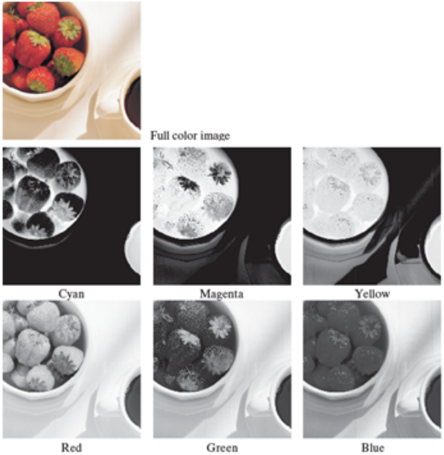
FIGURE 7.27 Spatial neighborhoods for grayscale and RGB color images. Observe in (b) that a *single* pair of spatial coordinates, (x, y), addresses the same spatial location in all three images.



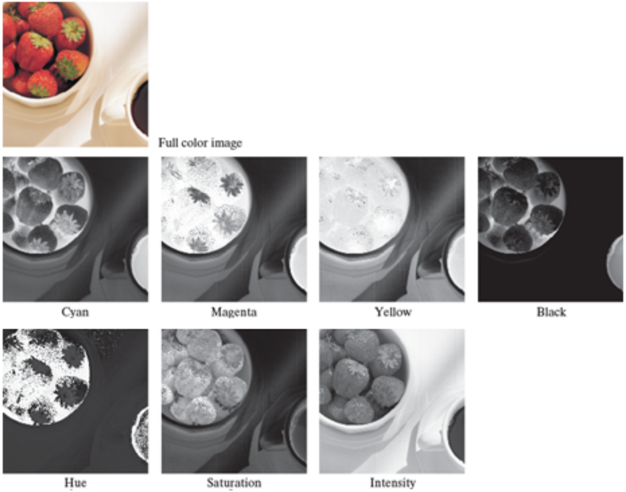
Color Transformations

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

- $n \rightarrow$ 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



Modifying the intensity of a full-color image

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

- HSI color space

$$\begin{aligned} s_1 &= r_1 \\ s_2 &= r_2 \\ s_3 &= kr_3 \end{aligned}$$

- RGB color space

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

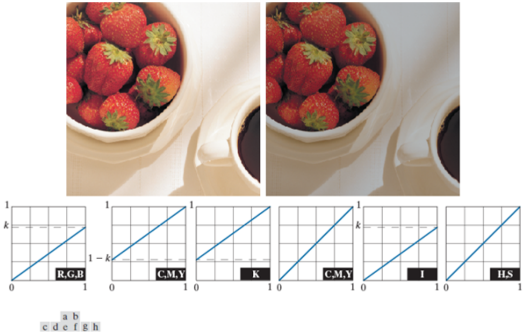


FIGURE 7.29 Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c) The required RGB mapping function. (d)–(e) The required CMYK mapping functions. (f) The required CMY mapping function. (g)–(h)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

Modifying the intensity of a full-color image

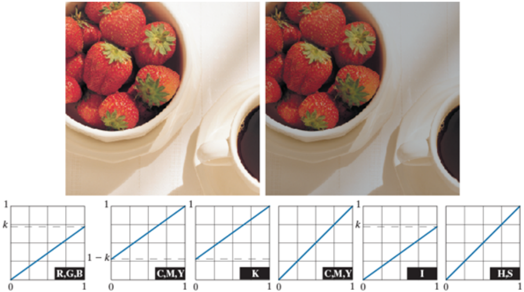
$$s_i = T_i(r_i) \quad i = 1, 2, ..., n$$

- CMY color space

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

- CMYK color space

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



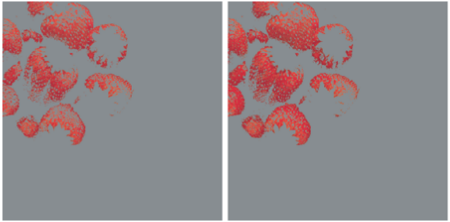
a b
c d e f g h

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 CMYK 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

$$s_i = \begin{cases} 0.5 & \text{IF } \left[|r_j - a_j| > \frac{W}{2} \right]_{\text{ANY } 1 \le j \le n} \\ r_i & \text{OTHERWISE} \end{cases} \qquad i = 1, 2, \ldots, n$$



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.

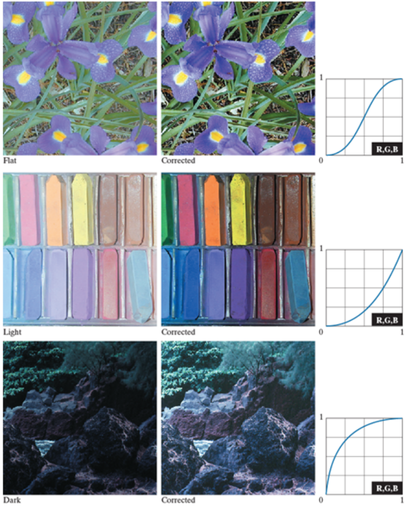


FIGURE 7.33 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not always alter the image hues significantly.

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.

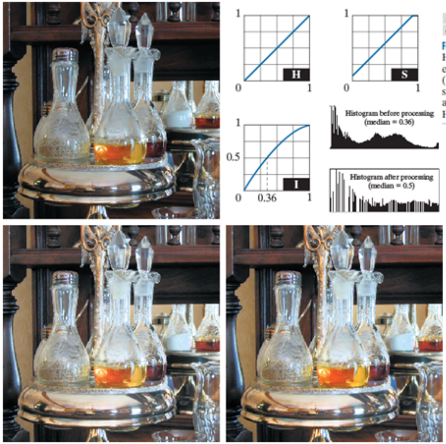
FIGURE 7.34 Color balancing a CMYK image.

Histogram Processing of Color Images

- Transformation that produces an image with a uniform histogram.

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

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



a b
c d

FIGURE 7.35 Histogram equalization (followed by saturation adjustment) in the HSI color space.

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.

Color Image Sharpening

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

- Laplacian of each component image

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

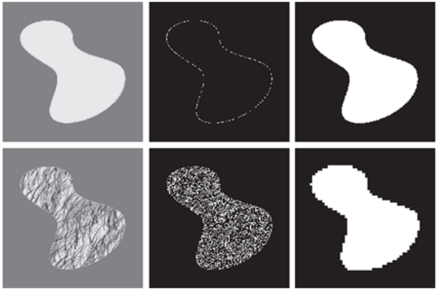
$$s_i(x,y) = r_i(x,y) + \nabla^2[r_i(x,y)]$$

a b c

FIGURE 7.39 Image sharpening using 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.

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.

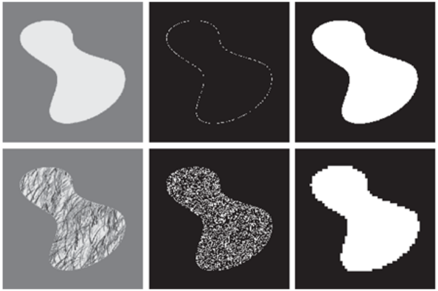
Segmentation

- R represent the entire region occupied by the image.

$$\bigcup_{i=1}^{n} R_i = R$$
$$R_i \text{ IS A CONNECTED SET, FOR } i = 0,1,2,\ldots,n$$
$$R_i \cap R_j = \emptyset \text{ FOR ALL } i \text{ AND } j, i \neq j$$
$$Q(R_i) = \text{TRUE FOR } i = 0,1,2,\ldots,n$$
$$Q(R_i \cup R_j) = \text{ FALSE FOR ANY ADJACENT REGION } R_i \text{ AND } R_j$$

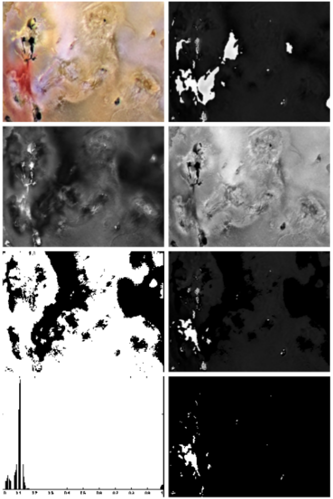
a b c
d e f

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 in HSI Color Space

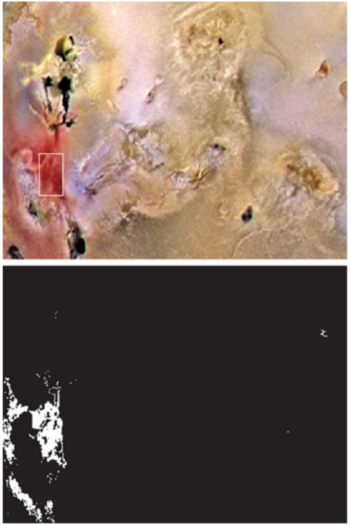
- Saturation is used as a mask.
- Intensity is not used for segmentation of color images.
- Range of hue values of the regions of interest is used as descriptor.
- The product of the mask with the hue is determined.
- The segmented image is obtained by thresholding this product.



a b c d e f g h 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

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

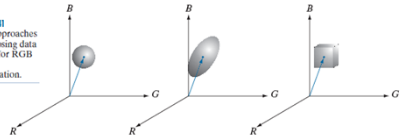
$$D(\mathbf{z},\mathbf{a}) = [(\mathbf{z}-\mathbf{a})^T(\mathbf{z}-\mathbf{a})]^{1/2}$$

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

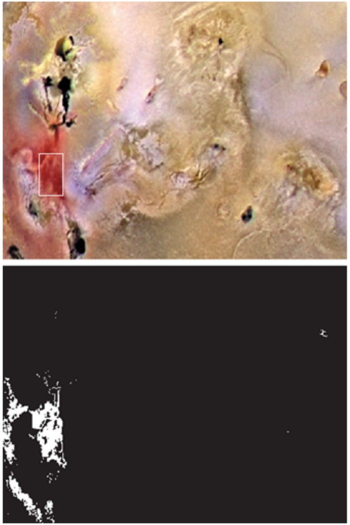
C is the covariance matrix of the samples.

a b c

FIGURE 7.41 Three approaches for enclosing data regions for RGB vector segmentation.



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

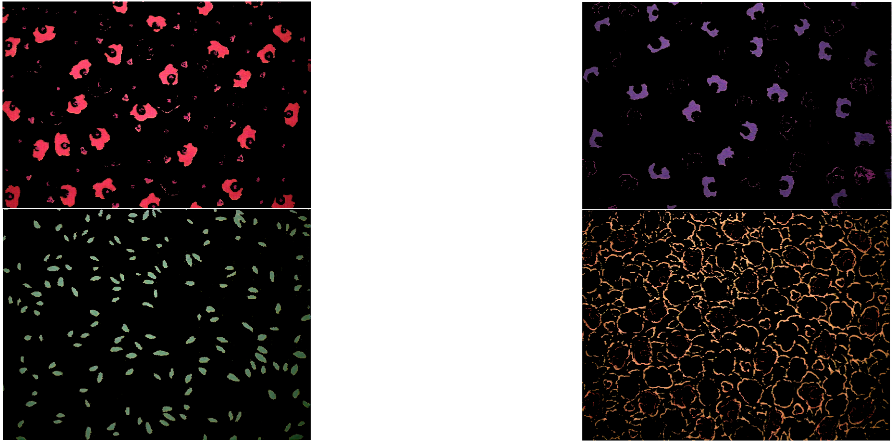
- Uniform changes of components in the Lab color space correspond to uniform changes in perceived color.
- Perceived color differences are measured by Euclidean distances.
- Segmentation can be performed by means of clustering.

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

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

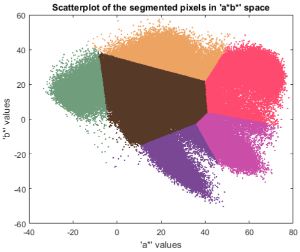
Every pixel is assigned to the class that minimizes the color difference.

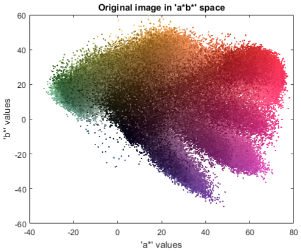
Segmentation in Lab Color Space

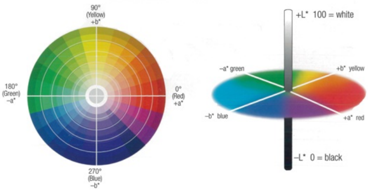


https://www.mathworks.com/help/images/examples/color-based-segmentation-using-the-l-a-b-color-space.html

Segmentation in Lab Color Space



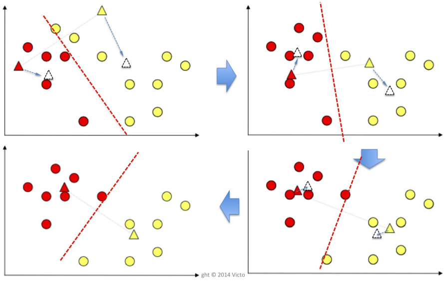




Segmentation using k-means clustering

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

K-means clustering example



Segmentation using k-means clustering

- Let **Z** be the color pixel dataset of the form

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

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

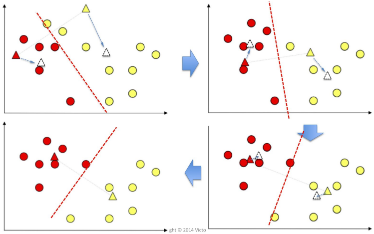
- We want to classify the data into k disjoint sets of the form

$$C = \{C_1, C2, ..., C_k\}$$

such that the criterion of optimality is satisfied

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

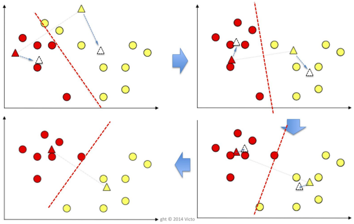
K-means clustering example



k-means Algorithm

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

K-means clustering example



- Assign samples to clusters whose mean is the closest.

$$\mathbf{z}_q \to C_i \qquad \text{IF } \|\mathbf{z}_q - \mathbf{m}_i\|^2$$

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

- Update the clusters' means

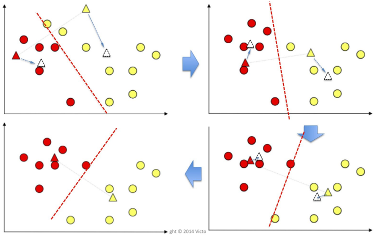
$$\mathbf{m}_i = \frac{1}{C_i} \sum_{\mathbf{z} \in C_i} \mathbf{z} \quad i = 1, 2, ..., k$$

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.

K-means clustering example



Segmentation using k-means clustering in different Color Spaces

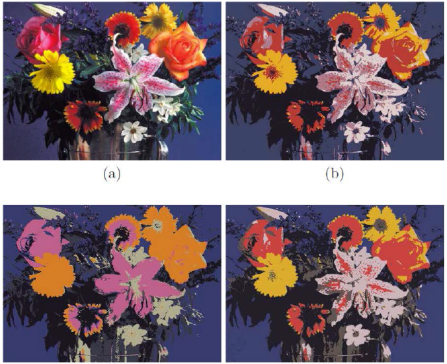

(a) (b) (c) (d)

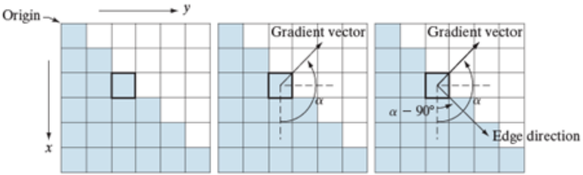
Figure 5.7 (a) Original color image. Results of segmentation using the k-means clustering algorithm in three different color spaces: (b) $sRGB$, (c) HSV, and (d) $L^*a^*b^*$. In each case, the image has been segmented into six regions. The color assigned to each region is the final centroid of the color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

Component-based Color Edge Detection

$$\nabla f(x,y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \qquad M(x,y) = \sqrt{g_x^2 + g_y^2} \qquad \alpha(x,y) = \tan^{-1} = \begin{bmatrix} \frac{g_y}{g_x} \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Prewitt Operators



a b c

FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge direction is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square represents one pixel. (Recall from Fig. 2.19 that the origin of our coordinate system is at the top, left.)

Component-based Color Edge Detection

a b c d
e f g h

FIGURE 7.43 (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 Edge Detection

- One of the ways that the concept of a gradient can be extend to vector functions.

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

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

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

a
b c
d e
f g

FIGURE 10.14 A 3×3 region of an image (the z's are intensity values), and various kernels used to compute the gradient at the point labeled z_5.

Color Edge Detection

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

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

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

The direction and value of maximum rate of change of $\mathbf{c}(x, y)$ is given by

$$\theta(x, y) = \frac{1}{2}\tan^{-1}\left[\frac{2g_{xy}}{g_{xx} - g_{yy}}\right]$$

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

Color Edge Detection

a b
c d

FIGURE 7.44
(a) RGB image.
(b) Gradient computed in RGB color vector space.
(c) Gradient image formed by the elementwise sum of three individual gradient images, each computed using the Sobel operators.
(d) Difference between (b) and (c).

Noise in Color Images

a b
c d

FIGURE 7.46 (a)–(c) Red, green, and blue 8-bit component images corrupted by additive Gaussian noise of mean 0 and standard deviation of 28 intensity levels. (d) Resulting RGB image. [Compare (d) with Fig. 7.44(a).]

Noise in Color Images

a b c

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.