Using Color in Computer Vision Systems

Color Image Processing

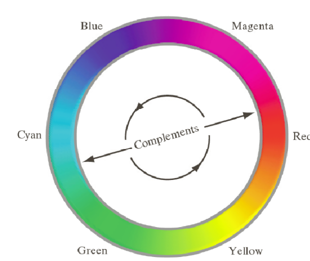
- In gray image, a scalar gray value is assigned to a pixel
- In color image, a color vector is assigned
- Color image processing techniques
  - Monochromatic-based techniques
  - Vector-valued techniques

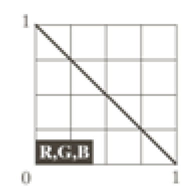
Color Transformation

- Color transformation here deals with processing the components of a color image within the context of *single* color model.
- Given input color image, $f(x,y)$, the transform image is $$g(x,y)=T[f(x,y)]$$ or $$s_i = T_i(r_1, r_2, \ldots, r_n)\ ; \quad i = 1,2,\ldots n$$
- If we work on RGB, $n = 3$ and r_1, r_2, r_3 denote red, green, and blue components of the input image.

Color Complement

- The hues directly opposite one another on the color circle are called complements.





Color Slicing

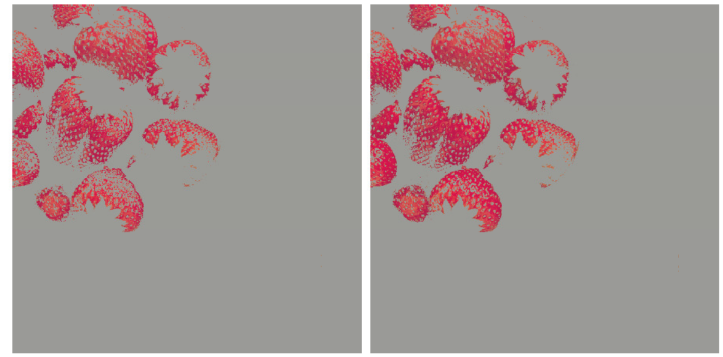
- Highlighting a specific range of colors in an image is useful for separating objects from the background.
- If the colors of interest are enclosed by a cube of width, w, and centered at a color with components $(a_1, a_2, \ldots a_n)$, the transformation is

$$s_i = \begin{cases} 0.5 & \text{if } \left[\left| r_j - a_j \right| > \frac{w}{2} \right]_{\text{any } 1 \le j \le n} \\ r_i & \text{otherwise} \end{cases}$$

- If a sphere with radius R_0 is used, the transformation is

$$s_i = \begin{cases} 0.5 & \text{if } \sum_{j=1}^{n} \left(r_j - a_j \right)^2 > R_0^2 \\ r_i & \text{otherwise} \end{cases}$$

Color Slicing (2)

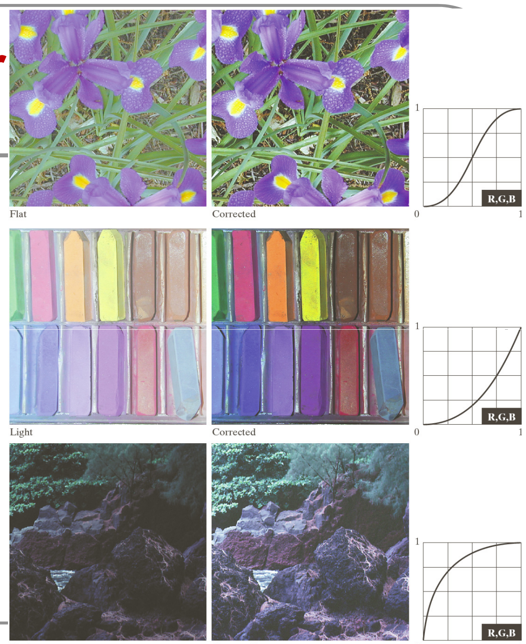


a b

FIGURE 6.34 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).

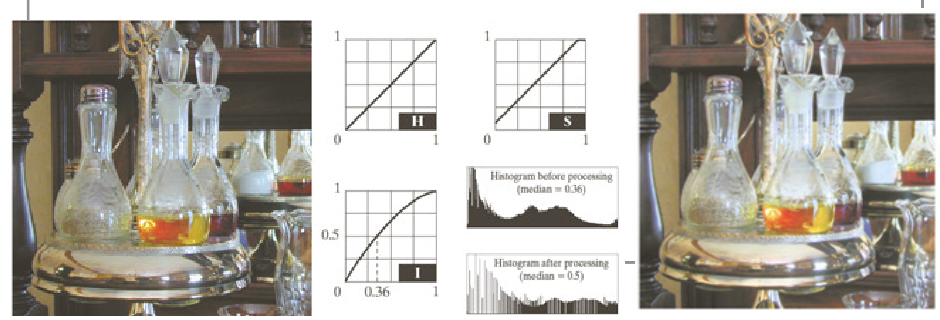
Tone and Color Corrections

- Three common tonal imbalances
  - flat
  - light
  - dark.



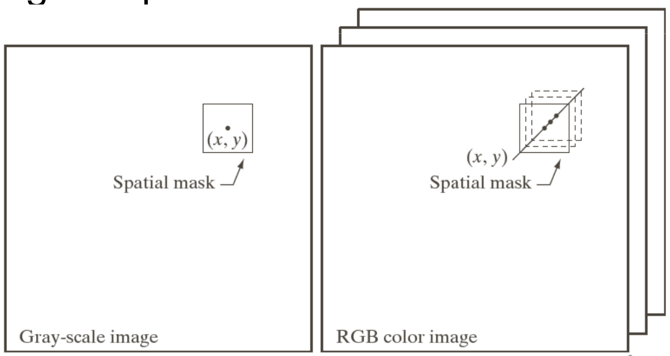
Histogram Processing

- Applying gray-scale image histogram equalization to components of color image yields erroneous color.
- More logical approach is to spread the color intensities uniformly, leaving the colors unchanged.



Smoothing and Sharpening

- Smoothing and sharpening are region-based processing by considering the neighbor pixels.



Color Image Smoothing

- Let S_{xy} denote the set of coordinates defining a neighborhood centered at (x,y) in an RGB color image.
- The average of the RGB component vectors in this neighborhood is

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

Smoothed image

original image

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

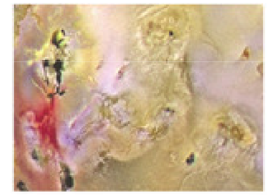
FIGURE 6.40 Image smoothing with a 5 × 5 averaging mask. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results.

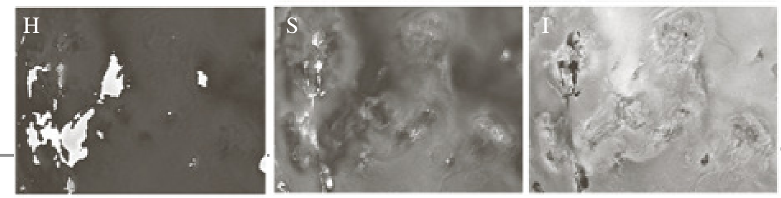
Color Image Segmentation

- Segmentation is a process that partitions an image into regions.
- If we want to segment image base on color, HSI and YUV might be suitable because color is represented in H and U,V separately.

Segmentation in HSI

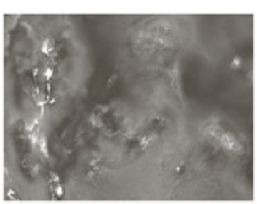
- Suppose that the region of interest is the reddish region in lower left of the image below.

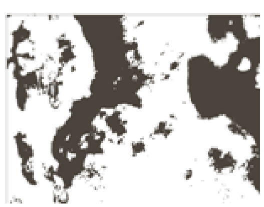




Segmentation in HIS (2)

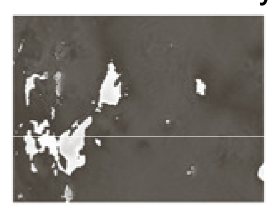
- Binary mask generated by thresholding the S image with threshold 10% of the maximum value in that image. Any pixel value greater than the threshold was set to 1, otherwise it was set to 0.

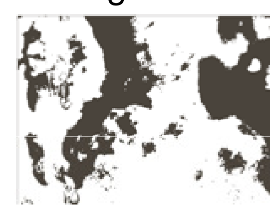

S

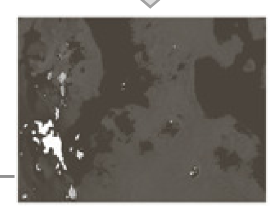

The binary mask generated from thresholding S.

Segmentation in HIS (3)

- Product the binary mask image with the H image.

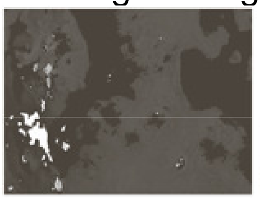

H


Binary mask (B)

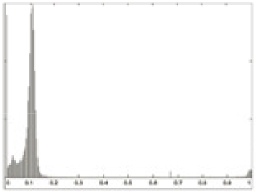

Product of H and B

Segmentation in HSI (4)

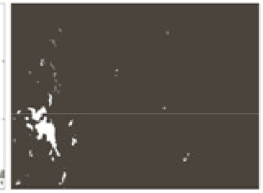
- Segmenting by thresholding



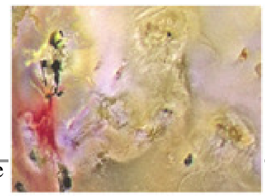
Product of H and B



Histogram of Product of H and B image



Segmented image



Original image

Segmentation in RGB

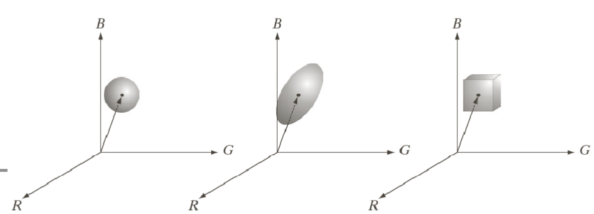
- Given a set of sample color points representative of colors of interest, we obtain an estimate of the "expected" color that we wish to segment.
- Let this "expected" (average) color be denoted by the RGB vector **a**.
- The objective of segmentation is to classify each RGB pixel in a given image as having a color in the specified range or not.

Segmentation in RGB (2)

- Let **z** denote an arbitrary point in RGB space. **z** is similar to **a** if the distance between them is less than a specified threshold D_0.

$$D(\mathrm{z},\mathrm{a}) = \|\mathrm{z}-\mathrm{a}\| = \left[(\mathrm{z}-\mathrm{a})^T(\mathrm{z}-\mathrm{a})\right]^{\frac{1}{2}} = \left[(z_R-a_R)^2+(z_G-a_G)^2+(z_B-a_B)^2\right]^{\frac{1}{2}}$$

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

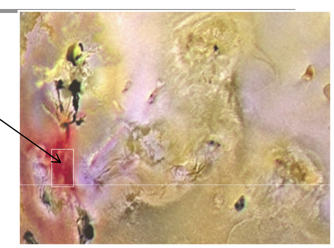


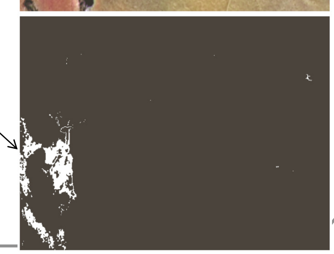
a b c

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

Segmentation in RGB (3)

- Sample reddish colors
- Compute mean => **a**
- Get standard deviation of R,G,B => σ_R, σ_G, σ_B
- Thresholds are +/-1.25*σ
- Result of segmentation

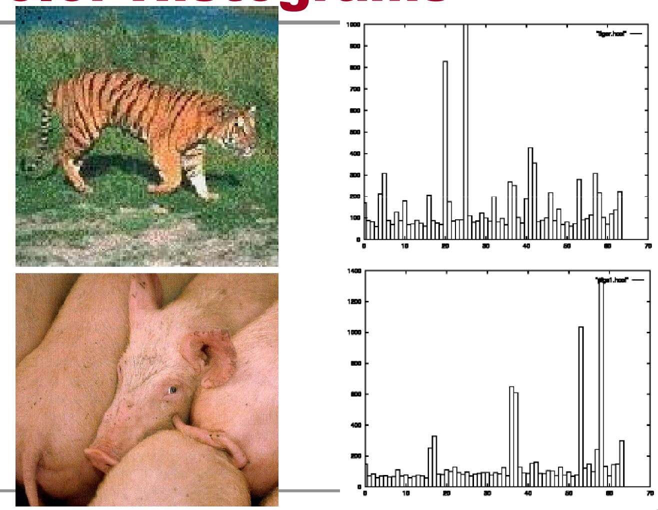




Color Histograms

- Histogram counts the number of pixels of each kind
- Simple methods:
  - Concatenate the higher order two bits of each RGB -> histogram will have $2^6 = 64$ bins
  - Compute 3 separate RGB histograms, then concatenate them into one (Jain and Vailaya 1996)
- Histogram is fast and easy to compute.

Color Histograms



Source: Shapiro and Stockman's colorplate

Color Histograms

- Color histogram is an representation that is relatively invariant to translation, rotation about the imaging axis
- Color histograms can be used for database query or classification by matching the image histogram $h(I)$ and model histogram $h(M)$

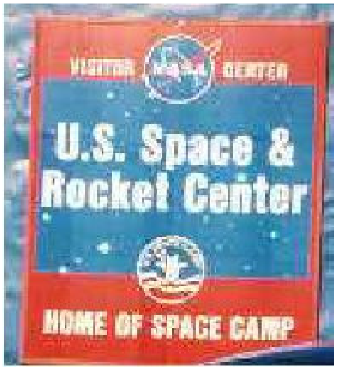
Color Histograms

- $$Match\ (h(I),h(M)) = \frac{\Sigma_{j=1}^{K} \min\{h(I)[j],h(M)[j]\}}{\Sigma_{j=1}^{K} h(M)[j]}$$
- Size can easily be normalized so that different image histograms can be compared
- Then use the Eucledian distance to compare two images

Practical Use of Color Image Analysis

- Image segmentation into regions by clustering on color values and pixel locations
- Color detection by training an algorithm to look for certain colored regions, e.g., white color, skin color

Extracting White Regions

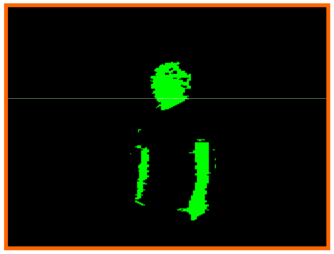


Input image

Output image after labeling CC extracted white pixels

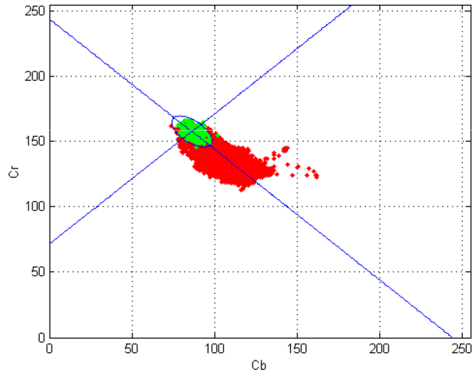
Source: Shapiro and Stockman's colorplate

Detecting Skin Color



Skin Regions

Skin color in YCbCr space

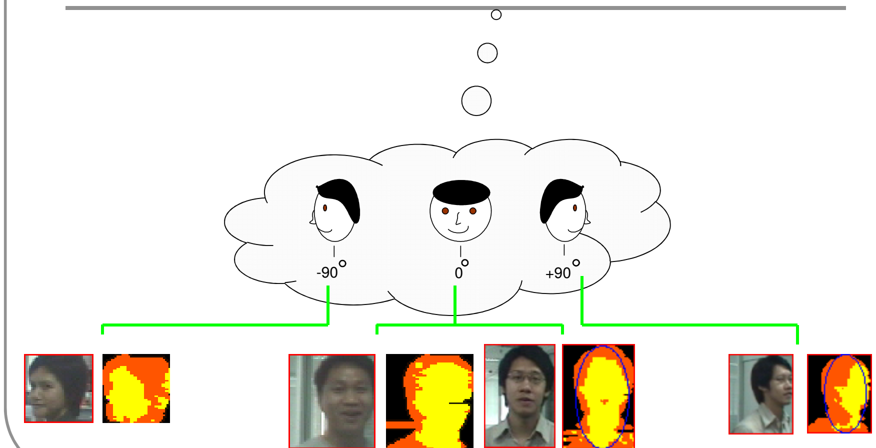


Detecting Human Face

Input image | Labeled image | Boundary of extracted face region

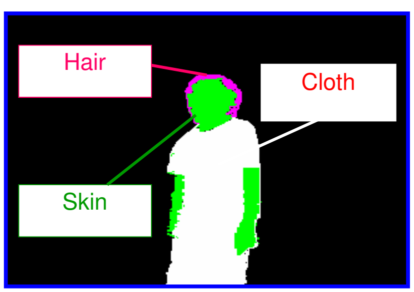
Source: Shapiro and Stockman's colorplate

Face Pose Estimation

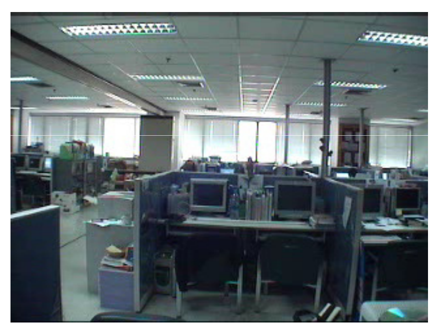


Tracking Persons

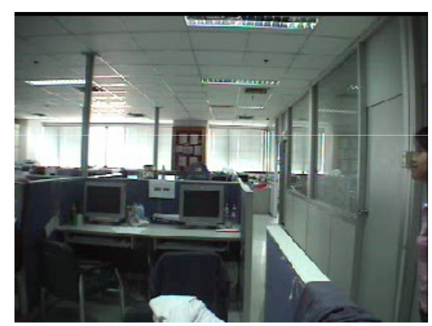
- To track the target person across two cooperative cameras, we obtain the color characteristic to identify the target.



Cooperative person tracking



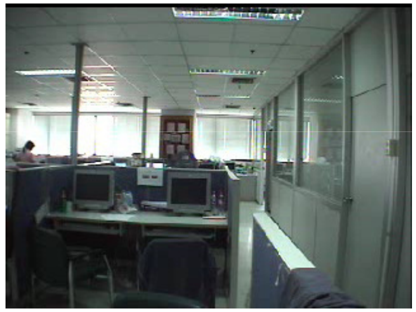
CAM2



CAM1

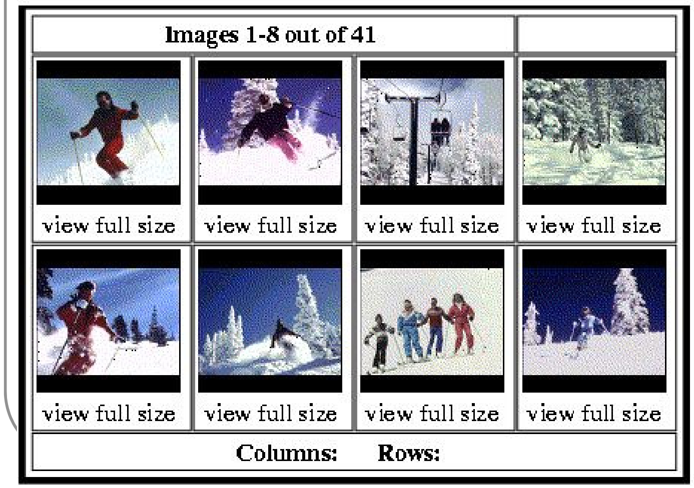
Cooperative person tracking

CAM2



CAM1

Retrieval from image database



Top left image is query image. The others are retrieved by having similar color histogram

Detecting Moving Objects

- Background subtraction is to subtract or difference the current image from a reference background model.