➢**Segmentation**

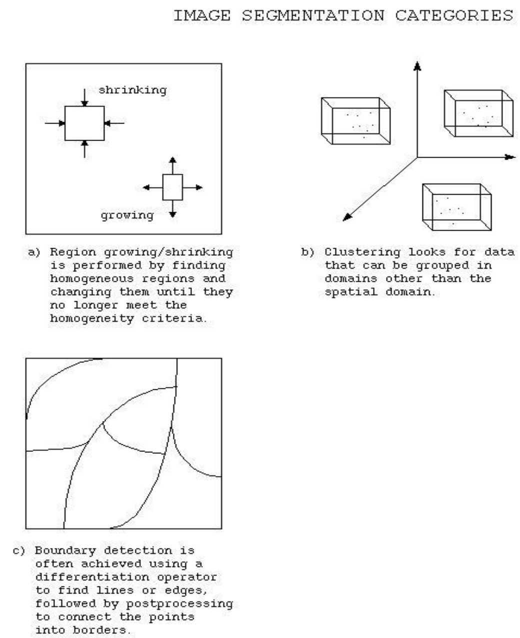
✓The goal of image segmentation is to find regions that represent objects or meaningful parts of objects

✓Image segmentation methods will look for objects that either have some measure of ***homogeneity*** within themselves, or have some measure of ***contrast*** with the objects on their border

✓ The homogeneity and contrast measures can include features such as gray level, color, and texture

✓ Image segmentation techniques can be divided into three main categories:

1. *Region growing and shrinking*
2. *Clustering methods*
3. *Boundary detection*



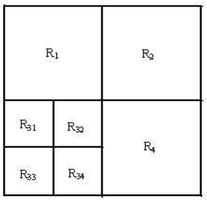
✓**Region Growing and Shrinking**

- Region growing and shrinking methods segment the image into regions by operating principally in the row and column, *(r,c)*, based image space

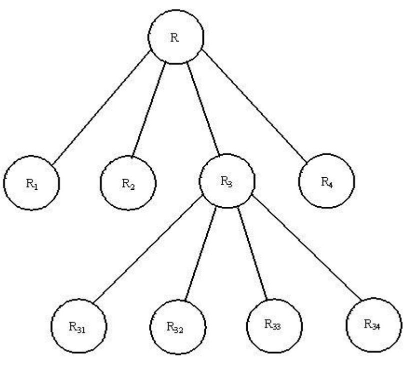
- Methods can be *local*, operating on small neighborhoods, *global*, using the entire image, or a combination of both

- *Split and Merge:*
  - Split and merge methods use graph structures to represent the regions and their boundaries
  - The data structure most commonly used is a *quadtree* – each node can have four children
  - This data structure facilitates the splitting and merging of regions

Quadtree Data Structure. a) A partitioned image where R_i represents different regions, b) The corresponding quadtree data structure



(a)



(b)

- *Split and merge* proceeds as follows:

1. Define a *homogeneity test*:
   Define homogeneity measure; may incorporate (features of interest) brightness, color, texture, or other application- specific information, determine criterion the region must meet to pass the test.

2. Split the image into equal sized regions.

3. Calculate the homogeneity measure for each region.

4. If the homogeneity test is passed for a region, then a merge is attempted with its neighbor(s). If criterion is not met, the region is split.

- There are many variations of the split and merge algorithm; including algorithms based only on region splitting or region merging

- Algorithms based on splitting only are called ***multiresolution*** algorithms

- The results from these approaches will be quite similar, with the differences apparent only in computation time

- Parameter choice, such as the minimum block size allowed for splitting, will heavily influence the computational burden as well as the resolution available in the results

- The user-defined homogeneity test is largely application-dependent, but the general idea is to select features that will be similar within an object and different from the surrounding objects

- We can consider gray level variance as a homogeneity measure and define a homogeneity test that requires the gray level variance within a region to be less than some threshold (Merge)

- Then *gray level variance* is defined as:

$$\text{Gray level variance} = \frac{1}{N-1} \sum_{(r,c)\in REGION} [I(r,c) - \bar{I}\,]^2$$

$$\text{where} \quad \bar{I} = \frac{1}{N} \sum_{(r,c)\in REGION} I(r,c)$$

- The sum is taken over the region of interest and *N* is the number of pixels in the region
- The *variance* is basically a measure of how widely the gray levels within a region vary

- A similar approach involves searching the image for a homogeneous region, a *seed region*, and enlarging it until the homogeneity criteria is no longer met
- At this point, a new region is found that exhibits homogeneity and is grown
- This process continues until the entire image is divided into regions
- The *seed region* selection can heavily influence the resulting segmented image

- CVIPtools has these homogeneity criteria:

1. *Pure uniformity* – constant gray levels
2. *Local mean versus global mean – Local mean is greater than global mean*
3. *Local Standard deviation vs. global mean – local SD is less than10% of global mean*
4. *Variance – min. percentage of pixels are within two SD*
5. *Texture – four quadrants have similar texture*

Split and Merge various homogeneity criteria


a) Original image


b) Local mean vs global mean


c) Local SD vs global mean

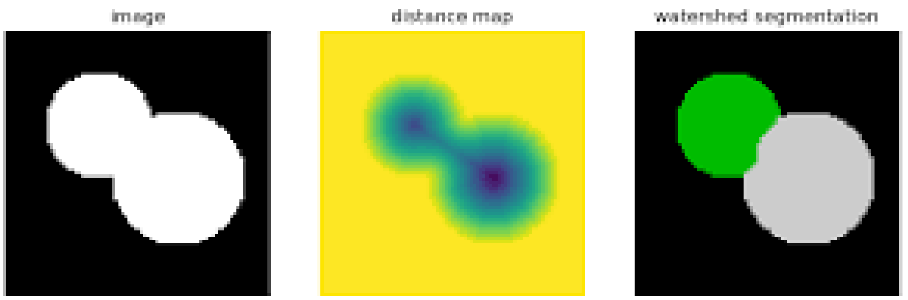

d) Variance with *Threshold* 25, *Percentage* = 0.7 (70%)


e) Weighted gray level distance with *Threshold* = 25


f) Texture homogeneity with *Similarity* = 50, and *Pixel distance* = 2

- Image 256x256, *Entry level* 6

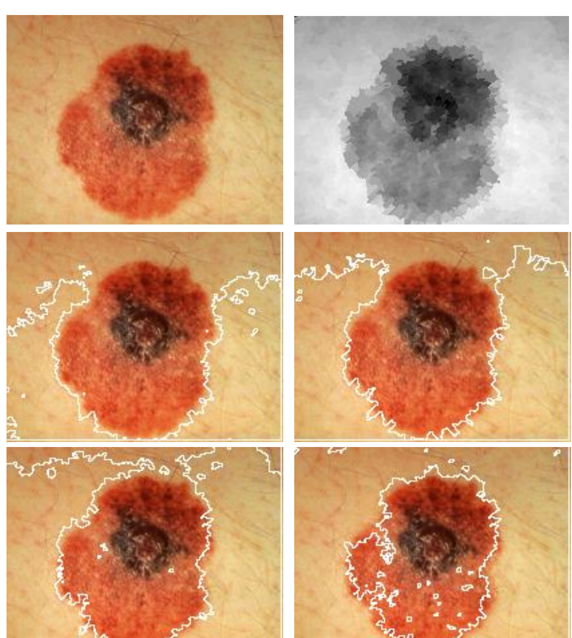
- *Watershed segmentation algorithm:*
  - The watershed algorithm is based on modeling a gray level image as a topographic surface, with higher gray levels corresponding to higher elevations
  - The image is flooded to create pools of water corresponding to segments in the image
  - When rising water reaches a point where two pools will merge, a dam is built to prevent the merging (watershed lines)



- The watershed segmentation algorithm in CVIPtools was initially designed to separate a single object from the background in color images
- It provides the user with two parameters – *merge* and *threshold; merge* checkbox will merge or not
- If merge is selected, the *threshold* parameter determines the amount of merging that will occur

- The threshold parameter creates a histogram using the average gray value(s) within each watershed segment
- Next, it finds the maximum value in the histogram and <u>merges</u> this group with adjacent lower and higher gray levels until the threshold is reached
- The threshold is the *percent of total area in the image*

Watershed Segmentation



a) Original image

b) Watershed segmentation without merging

c) Borders after merging with a threshold of 0.4

d) Merge with a threshold of 0.5

e) Merge with a threshold of 0.6

f) Merge with a threshold of 0.7

Watershed Segmentation

a) Original image

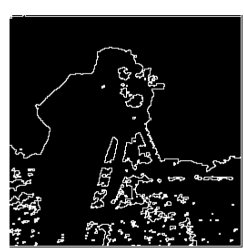
b) Result of watershed segmentation without merging

c) Image with borders after merging with a threshold of 0.3

d) Borders only with threshold of 0.3

Watershed Segmentation (contd)

e) Image with borders after merging with a threshold of 0.6



f) Borders only with threshold of 0.6

g) Image with borders after merging with a threshold of 0.8

h) Borders only with threshold of 0.8

✓Clustering Techniques

- Clustering techniques are image segment-ation methods by which individual elements are placed into groups based on a *similarity* measure

- Differs from region growing and shrinking methods as the mathematical space includes dimensions beyond the *(r,c)* image space

- Mathematical spaces used for clustering may include, color spaces, histogram spaces or complex feature spaces

---

- Simplest clustering method is to divide the space of interest into regions by selecting the center or median along each dimension and splitting it there, used in the *SCT/Center* and *PCT/Median*

- Only *effective* if the space and algorithm are designed intelligently as split alone may not find good clusters

- Methods such as histogram thresholding are also used to decide where to divide the space – e.g. binary thresholding

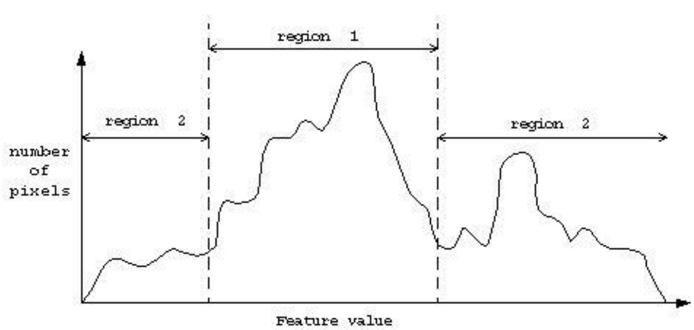
- *Recursive region splitting*

  - A standard clustering method, which uses a thresholding of histograms technique to segment the image

  - A set of histograms is calculated for a specific set of features, each of these histograms is searched for distinct peaks

  - The best peak is selected and the image is <u>split</u> into regions based on this thresholding of the histogram

HISTOGRAM PEAK FINDING



Two thresholds are selected, one on each side of the best peak. The image is then split into two regions. Region 1 corresponds to those pixels with feature values between the selected thresholds, known as those in the peak. Region 2 consists of those pixels with feature values outside the threshold.

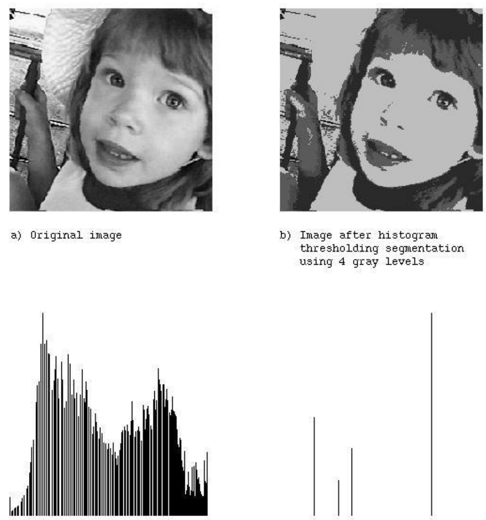
- An example of this type of algorithm:

1. Consider the entire image as one region and compute histograms for each component of interest (for example red, green and blue for a color image)
2. Apply a peak finding test to each histogram
3. Select the best peak and put thresholds on either side of the peak

4. Segment the image into two regions based on this peak
5. Smooth the binary thresholded image so only a single connected subregion is left
6. Repeat steps 1-5 for each region until no new subregions can be created, that is, no histograms have significant peaks

- Many of the parameters of this algorithm are application-specific
- In CVIPtools we have two histogram thresholding based segmentation methods, called *histogram thresholding* and *fuzzy c-means*

HISTOGRAM THRESHOLDING SEGMENTATION



a) Original image

b) Image after histogram thresholding segmentation using 4 gray levels

c) Histogram of image (a)

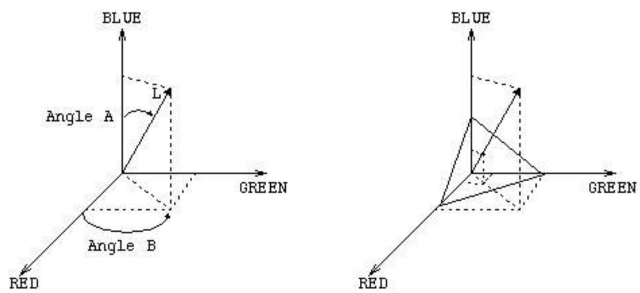
d) Histogram of image (b)

- The **SCT/Center** color segmentation algorithm was initially developed for the *identification of variegated coloring* in skin tumor images

- It *decouples* the color information from the brightness information.

- The brightness levels may vary with changing lighting conditions, so by using the two-dimensional color subspace defined by two angles we have a more robust algorithm

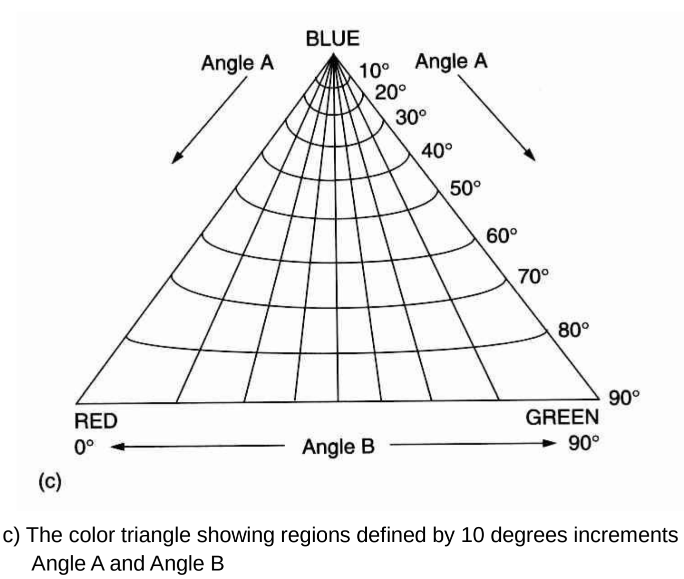
: SCT/CENTER AND COLOR TRIANGLES



a) The spherical coordinate transform separates the red, green, and blue information into a 2-D color space defined by angles A and B, and a 1-D brightness space defined by L.

b) The color triangle

SCT/Center and color triangles (contd)



c) The color triangle showing regions defined by 10 degrees increments on Angle A and Angle B

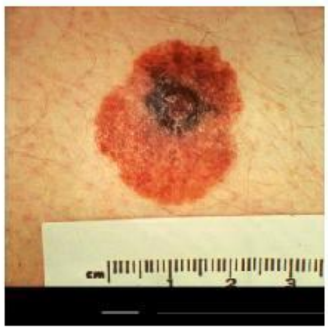
- The SCT/Center segmentation algorithm is as follows:

1. Convert the *(R,G,B)* triple into spherical coordinates – *(L, angle A, angle B)*
2. Find the minima and maxima of *angle A* and *angle B*
3. Divide the subspace, defined by the maxima and minima, into equal-sized blocks (based on angles)

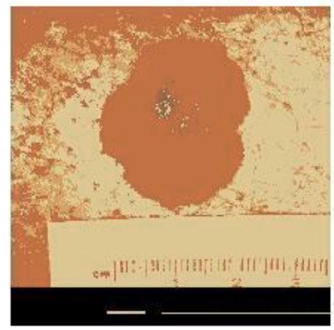
4. Calculate the *RGB* means for the pixel values in each block
5. Replace the original pixel values with the corresponding *RGB* means

- For the identification of variegated coloring in the skin tumor application it was determined that segmenting the image into four colors was optimal

SCT/CENTER SEGMENTATION ALGORITHM



a) Original image



b) SCT/CENTER segmentation of skin tumor using 4 colors.

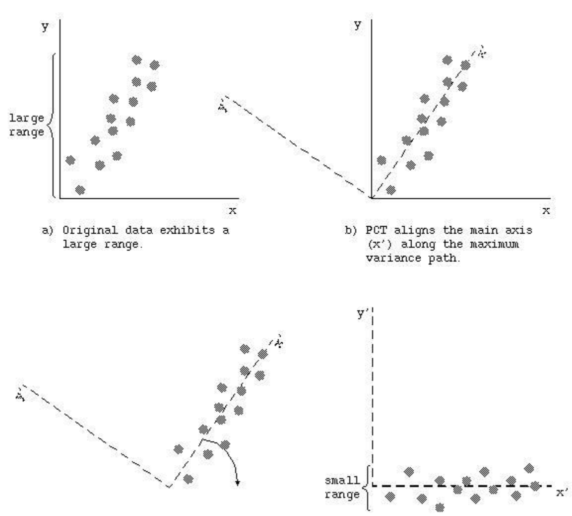
- The **PCT/Median** algorithm was developed because, with features other than variegated coloring, the results provided by the SCT/Center were not satisfactory

- The principal components transform is based on statistical properties of the image, and can be applied to any K-dimensional mathematical space

- The median split is based on an algorithm developed for color compression to map 24-bits per pixel color images into images requiring an average of 2-bits per pixel

- It was believed that the PCT used in conjunction with the median split algorithm would provide a satisfactory color image segmentation, since the PCT aligns the main axis along the maximum variance path in the data set

PRINCIPAL COMPONENTS TRANSFORM



a) Original data exhibits a large range.

b) PCT aligns the main axis (x') along the maximum variance path.

c) The new axes are rotated.

d) Transformed data now has a small range. Most of the variance, or information, is along the x' axis, in one dimension rather than two, as in (a).

- The PCT/Median segmentation algorithm

1. Find the PCT for the RGB image. Transform the RGB data using the PCT.

2. Perform the median split algorithm:

   - Find the axis that has the maximal range (initially it will be the PCT axis)

   - Divide the data along this axis so that there are equal numbers of points on either side of the split – the median point

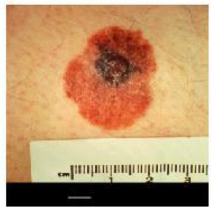
   - Continue splitting at the median along the maximum range segment until the desired number of colors is reached

3. Calculate averages (of colors) for all the pixels falling within a single parallelepiped (box)

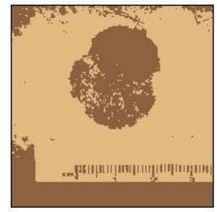
4. Map each pixel to the closest average color values, based on a Euclidean distance measure

- For the skin tumor application it was determined that the optimum number of colors was dependent upon the feature of interest

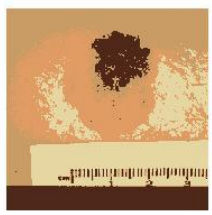
PCT/MEDIAN SEGMENTATION ALGORITHM



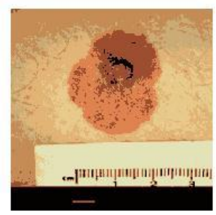
a) Original image



b) PCT/Median segmented image with 2 colors



c) PCT/Median segmented image with 4 colors



d) PCT/Median segmented image with 8 colors