计算机辅助手术讲座（3）

Image Guided Surgery (3)

Basic Image Operations

顾 力栩 (*Lixu Gu*)

上海交通大学 Med-X研究院

2009.11

pixel operation

- 点操作 (Point operation)
- 代数操作 (Algebraic operation)
- 几何操作 (Geometric operation)

Point Operation

- Point opearition:
  - Output pixel's gray level depends only upon the gray level of the corresponding input pixel.
  - Contrast enhancement; contrast stretching; gray scale transformation;
  - pixel-by-pixel copying operations, except that the gray levels are modified according to the transformation function.

$$B(x,y) = f[A(x,y)]$$

Point Operation

- **Linear point operations:**

$$D_B = f(D_A) = aD_A + b$$

a>1: contrast increased; 0<a<1: contrast reduced

a=1 & b≠0: shift gray level; a<0: reverse the contrast.

- **Nonlinear monotonic point operations:**

$$f(x) = x + Cx(D_m - x)$$

D_m: maximum gray level;

C: determine the amount of increase (C>0) or decrease (C<0) in the midlevel gray range.

Application

- Photometric calibration
- Contrast enhancement
- thresholding
- Contour lines
- clipping

Algebraic operation

- Algebraic operation:
  - Produce an output image which is the pixel-by-pixel sum, difference, product, or quotient of two or more images.
  - If one of the input image is a constant, it can be treated as a linear point operation.

Definition

- The four algebraic image-processing operations are expressed mathematically as:

$$C(x,y) = A(x,y) + B(x,y)$$
$$C(x,y) = A(x,y) - B(x,y)$$
$$C(x,y) = A(x,y) \times B(x,y)$$
$$C(x,y) = A(x,y) \div B(x,y)$$

Where, A(x,y) and B(x,y) are the input images

C(x,y) is the output image.

Application

- Image Addition:
  - Averaging for noise reduction
  - *Double-exposure* effect
- Image Subtraction:
  - Background subtraction
  - Motion detection
  - Gradient magnitude
- Multiplication and division

Geometric Operation

- Geometric operation:
  - Change the spatial relationships among the objects in an image.
  - Moving things around within the image.
  - Two algorithms are required:
    1. Spatial transformation
    2. Gray level interpolation

Transformation

- Simple transform:
  - Suppose the coordinate before and after the transform are (x,y) and (x',y')
  - Translation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

  - Rotation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Transformation

- Scale change: $\begin{bmatrix} x' \\ y' \end{bmatrix} = k \begin{bmatrix} x \\ y \end{bmatrix}$
- Order of transform
  - Translation followed by scale change

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = k\left(\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}\right) = \begin{bmatrix} kx + k\Delta x \\ ky + k\Delta y \end{bmatrix}$$

  - Scale change followed by translation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = k \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} kx + \Delta x \\ ky + \Delta y \end{bmatrix}$$

Typical Transformations

- Scale change followed by rotation

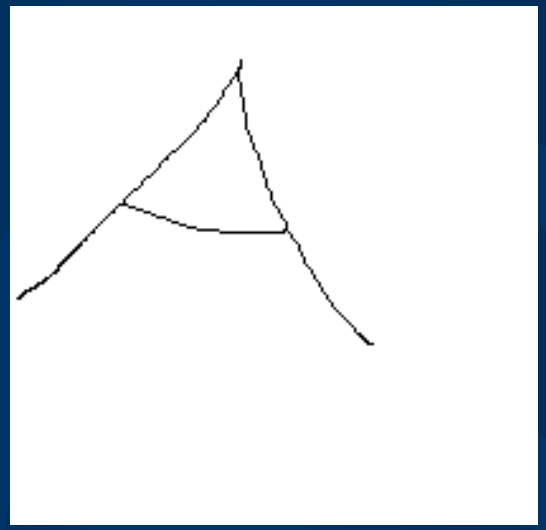
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} k \begin{bmatrix} x \\ y \end{bmatrix} = k \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Rotation followed by scale change

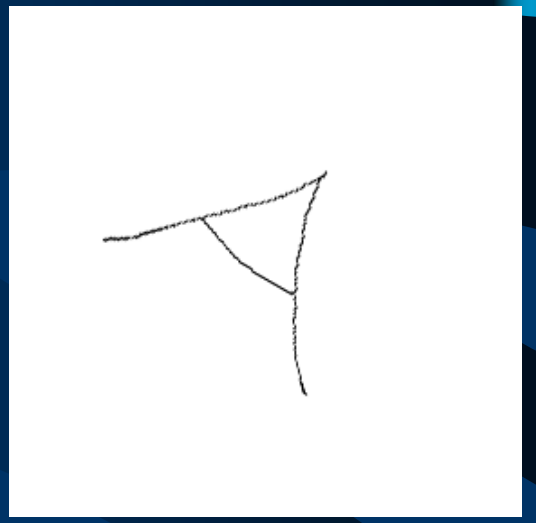
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = k \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = k \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Application

Original



Translation



Rotation (30^0)

Scale change (60%)

Interpolation Algorithm

- Nearest Neighbor Interpolation:
- Bilinear Interpolation (square interpolation):

  see detail at pp.118~119
- Higher Order Interpolation:
  - 样条函数 （B-Spline）
  - 多项式函数 （Polynomials）

Neighbourhood Operators

Neighbourhood Operators

- Pixel Connectivity:

  for a pixel P(x,y)

  - 4-neighbours:

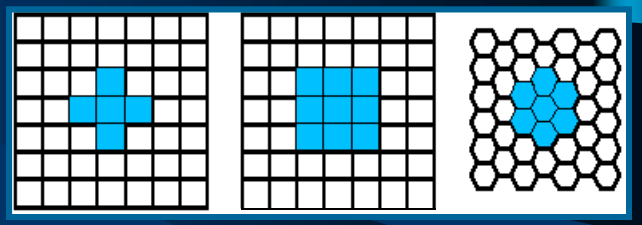
    $N_4(P) = \{(x+1,y), (x-1,y), (x,y+1),(x,y-1)\}$

  - 8-neighbours :

    $N_8(P) = N_4(P) \cup \{(x+1,y+1), (x+1,y-1), (x-1,y+1),(x-1,y-1)\}$

  - 4(8)-connected:

    two pixels within $N_4(P)$ or $N_8(P)$



Connected Component Labeling

- Connected components labeling:
  - groups the pixels in an image into components based on pixel connectivity
  - Labels components with a gray level or a color (color labeling)
- Connected component labeling works by scanning an image, pixel-by-pixel (from top to bottom and left to right) in order to identify connected pixel regions
- Intensity criterion(IC): the same set of intensity values (1 for a binary image; a value range for a gray level image)

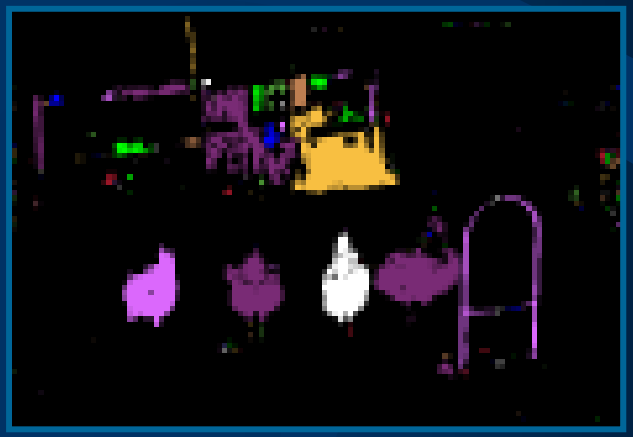
Labeling Algorithm

- Labeling algirithm: for a pixel *P* satisfy IC
  1. Step 1 (First Scan):
     - If all neighbors do not satisfy IC, assign a new label to *P*
     - if only one neighbor satisfy IC, assign its label to *P*
     - if one or more of the neighbors satisfy IC, assign one of the labels to *P* and make a note of the equivalences.
  2. Step2 (Resolve equivalence) : The equivalent label pairs are sorted into equivalence classes by a equivalence resolve algorithm (e.g. Floyd-Warshall algorithm) and a unique label is assigned to each class
  3. Step3 (Second scan) : Each label is replaced by the label assigned to its equivalence classes

Labeling Algorithm

- Examples:

```
%banner S     Labels        %banner C     Labels
 #####          00000         #####          00000
#     #       1     2       #     #       1     2
#             1             #             1
 #####          33333         #             1
      #               4      #             1
#     #       5     4       #     #       1     3
 #####          66666         #####          44444
```



Region Property

- Region Properties are important features for the region analysis (or measurement) after the regions have been labeled (segmented)
- Region Properties:
  - Perimeter and Area
  - Center, Radius and Diameter
  - Centroid Moments and Orientation
  - Extreme Points and Curvature
  - Intensity Properties

Region Property

- Perimeter and Area:
  - Perimeter: The length of the contour of a connected component (region).
    - calculated from the chain-code of the contour,or
    - estimated by the number of pixels on the contour.
  - Area: The number of unit squares contained.
    - Pick's formula: $A(P) = n_I + n_B/2-1$

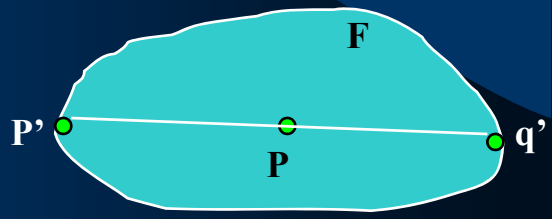
      n_I , n_B : number of interior points or the points lie on borders, respectively.

Region Property

- Center, Radius and Diameter :
  - Eccentricity of a point *P* in *F* is the maximum of distance *d(p,q)* for all points $q \in F$:

$$ecc(p) = max\ d(p,q) \mid q \in F$$

  - Center: The set of points *P* of *least eccentricity*
  - Radius: The value of the *least eccentricity* *d(p,p')*
  - Diameter: The value of the *greatest eccentricity* *d(p',q')*



Region Property

- Centroid Moments and Orientation:
  - Centroid: Given *F*, a set of *n* connected pixels (x_i,y_i), we can define a centroid *c* as:

$$x_c = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad y_c = \frac{1}{n}\sum_{i=1}^{n} y_i$$

  - Moments: The discrete (k,l)-order central moment is defined as:

$$\mu_{k,l} = \sum_{i=1}^{n} (x_i - x_c)^k (y_i - y_c)^l$$

  - Orientation: Orientation is defined here as an angle θ :

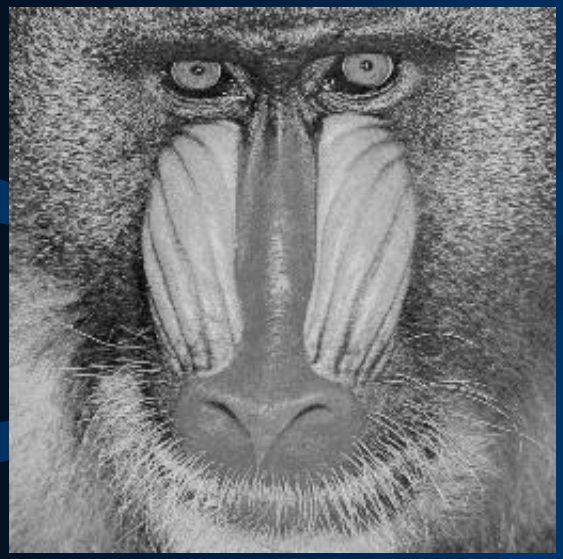
$$\theta = \frac{1}{2}\arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right)$$

Projects

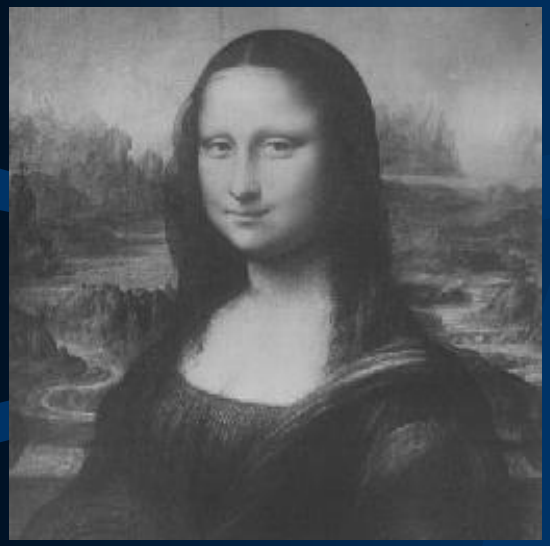
Project -1

- Histogram and threshold:
  - Requirement:
    - Program to realize Histogram analysis and threshold operation
    - Design UI and function buttons
    - Threshold operation can be manual or automatic (Otsu and Entropy)
  - Choose your favorite language

Classic Image Samples



Classic Image Samples



Discussion