Agenda

• Few words on next week’s lab
  ➢ Curve fitting in Excel

• Intro to Computer Vision, cont.
  ➢ Blob extraction for binary images

Dimensionality Reduction Machine (3D => 2D)

• What have we lost?
  ➢ Angles
  ➢ Distances (lengths)

Funny things happen …

• Parallel lines are not ….

• Distances can’t be trusted …
Large Focal Length Compresses Depth

400 mm 200 mm 100 mm 50 mm 28 mm 17 mm

© 1995-2005 Michael Reichmann

Image Projection

• Modeling (pinhole) projection

• The coordinate system
  ➢ Use the pinhole model as an approximation
  ➢ Put the optical center (COP) at origin
  ➢ Put image plane (Projection Plane, PP) in front of the COP
    ➢ Why?
      ➢ The camera looks down the negative z-axis
        ➢ Required for right-handed coordinates

Projection Equations

• Given $x, y, z$ then $x' = -\frac{d}{z} x$, $y' = -\frac{d}{z} y$
2D Image to 2D Pixel

\[ (c - Oc) = \frac{x'}{s_x} \quad (r - Or) = \frac{y'}{s_y} \]

Images in a Computer

• An image is a 2-D table of numbers or 2D Matrix

\[
\begin{array}{cccc}
10 & 22 & 42 & 133 \\
14 & 30 & 132 & 33 \\
22 & 57 & 130 & 25 \\
17 & 39 & 44 & 84 \\
15 & 30 & 46 & 72 \\
\end{array}
\]

Any 2D matrix can be seen as an image

Binary Image Analysis

• Binary Image = Image w/ only 0 and 1 as entries
• Notation:
  ➢ B – denotes the binary image
  ➢ B[i,j] – denotes a pixel in the image
  ➢ B[0,0] – upper leftmost pixel
• Neighborhoods
  ➢ 4 – Neighborhood
  ➢ 8 – Neighborhood

Applying Masks to Images

• Convolution of the Image w/ another “Signal”
• Masks have origins
  ➢ Symmetric masks – origins are the center pixels

\[
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]
Recursive Labeling Algorithm

- Let \( I \) denote the image.
- Step 1: Convert \( I \) to \( B \).
- Step 2: Let \( B' = -B \), label = 0
  1. for \( i = 1 \) rows
  2. for \( j = 1 \) columns
  3. if \( B'(i,j) == -1 \)
  4. label = label + 1;
  5. Let \( M = \) neighbors of \( P \) whose pixel value = -1;
  6. Add \( M \) onto Stack;
  7. Set \( B'(i,j) = \) label;
  8. while (Stack is not empty)
  9. Remove pixel \( P \) from Stack
  10. Let \( M = \) neighbors of \( P \) whose pixel value = -1;
  11. Add \( M \) onto Stack;
  12. Set \( B'(M) = \) label;
  13. end
  14. end
  15. end
  16. end

Algorithm Pictorially
Recursive Labeling Algorithm

1. Let I denote the image.
2. Step 1: Convert I to B.
3. Step 2: Let B' = -B, label = 0
4. for i = 1: rows
5. for j = 1: columns
6. if B'(i,j) == -1
7. label = label + 1;
8. Add (i,j) onto Stack and set B'(i,j) = label;
9. while (Stack is not empty)
10. Remove pixel P from Stack
11. Let M = neighbors of P whose pixel value = -1;
12. Add M onto Stack;
13. Set B'(M) = label;
14. end

Result

```
0 0 0 0 0 0 0 0
0 1 1 0 0 2 2 0
1 1 1 1 0 2 2 0
1 1 1 1 0 2 2 0
0 1 1 1 0 2 2 0
0 0 0 0 0 0 0 0
```

Morphological Operators

- Structuring Elements (S)
  - BOX(3,5)
  - DISK(5)
  - RING(5)

- Basic Operators
  - Translation \( X_t \) of a set of pixels by a position vector \( t \) is given by
    \[ X_t = \{ x + t \mid x \in X \} \]

Dilation: Minkowski addition \( B \oplus S = \bigcup_{s \in S} B_s \)
Morphological Operators

• Erosion: \( B \ominus S = \{ z \mid (B)_z \subseteq S \} \)

• Opening: \( B \bullet S = (B \ominus S) \oplus S \)

• Closing: \( B \circ S = (B \oplus S) \ominus S \)

Region Properties

• Area – Total # of pixels in the region of interest
• Centroid
• Perimeter
• Circularity
• Mean Radial Distance
• Standard Deviation/Variance of Radial Distance
• Bounding Box & Extremal Points
• Spatial Moments
• Ellipse Properties