Introduction to Image Analysis

Image Processing vs Analysis

- So far we have been studying “low level” tasks which are processing algorithms.
- Image analysis concerns “higher level” tasks which are intended to “understanding” a scene.
- Consider a robot with a camera, navigating a room.
Basics of Segmentation

- Detection of discontinuities in
  - Graylevel
  - Texture
  - Shape
  - Etc.

Graylevel Discontinuities: Edge Detection

- **Edge Detection** is accomplished mainly through the application of derivative-type filters, combined with some type of thresholding.

- Example:

  Sobel Edge Detector:

  \[
  \begin{bmatrix}
  -1 & 0 & 1 \\
  -2 & 0 & 2 \\
  -1 & 0 & 1 \\
  \end{bmatrix}
  \approx \frac{\partial f}{\partial x}
  \begin{bmatrix}
  -1 & -2 & -1 \\
  0 & 0 & 0 \\
  1 & 2 & 1 \\
  \end{bmatrix}
  \approx \frac{\partial f}{\partial y}
  \]

  \[
  |\frac{\partial f}{\partial x}| + |\frac{\partial f}{\partial y}|
  \]

  Compute Magnitudes, Add, Threshold

  Edge Image
Image gradient

- The gradient of an image: \( \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \)

- The gradient points in the direction of most rapid change in intensity

\[
\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ 0 \end{bmatrix} \quad \nabla f = [0, \frac{\partial f}{\partial y}]
\]

- The gradient direction is given by:
  \[
  \theta = \tan^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)
  \]
  - how does this relate to the direction of the edge?

- The *edge strength* is given by the gradient magnitude

\[
\| \nabla f \| = \sqrt{ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 }
\]
Edges From First and Second Order Derivatives

Edges in Noise
LoG Edge Detector

Laplacian: \[ \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \Rightarrow \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{ or } \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \]

- Derivative computation, especially second order, is sensitive to noise.
- To mitigate this sensitivity, we can first "prefilter" using a Gaussian low-pass.

\[ h(x, y) = e^{-\left(\frac{x^2 + y^2}{2\sigma^2}\right)} \]

\[ \nabla^2 (f * h) = \frac{\partial^2 (f * h)}{\partial x^2} + \frac{\partial^2 (f * h)}{\partial y^2} \]

Application of LoG

\[ \nabla^2 (f * h) = (\nabla^2 f) * h = f * (\nabla^2 h) \]

Linear Operations

Laplacian of Gaussian
Effect of LoG on a Step Edge

Model of an ideal digital edge

Convolve with Gaussian

Take First Derivative

Zero-crossing locates the edge

Take Second Derivative

Example

Original Sobel Edge

LoG Image LoG Threshold LoG 0-Cross
Designing an edge detector

- Criteria for an “optimal” edge detector:
  - **Good detection**: the optimal detector must minimize the probability of false positives (detecting spurious edges caused by noise), as well as that of false negatives (missing real edges)
  - **Good localization**: the edges detected must be as close as possible to the true edges
  - **Single response**: the detector must return one point only for each true edge point; that is, minimize the number of local maxima around the true edge

Canny edge detector

- This is probably the most widely used edge detector
- Theoretical model: step-edges corrupted by additive Gaussian noise
- Canny has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of *signal-to-noise ratio* and localization
- MATLAB: edge(image, ‘canny’)

Canny edge detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression
   - Thin multi-pixel wide “ridges” down to single pixel width
4. Linking of edge points

Source: D. Lowe, L. Fei-Fei

Example

- original image (Lena)
Example

norm of the gradient

Example

thresholding
Example

Non-maximum suppression

At q, we have a maximum if the value is larger than those at both p and at r. Interpolate to get these values.

Source: D. Forsyth
Suppression of Non-maxima

- Edge magnitude and direction
- Local maxima in derivative across the edge direction
- Interpolate to give sub-pixel precision
- Remove non-maxima
Edge linking

Assume the marked point is an edge point. Then we construct the tangent to the edge curve (which is normal to the gradient at that point) and use this to predict the next points (here either r or s).

Edge Linking

- Edge detection operators identify pixels individually as belonging to an edge or not
  - Need to connect points thus identified to recover a proper edge.
- Suppose two nearby points are identified as edge pixels.
- Compare gradients.

**Magnitude Match**

\[ |\nabla f(x_0, y_0)| - |\nabla f(x_i, y_i)| < T_m \]

**Angular Match**

\[ \angle \nabla f(x_0, y_0) - \angle \nabla f(x_i, y_i) < T_a \]
**Edge Linking II**

- Keep a **running list** of points so linked together.
- For book-keeping, assign a different (fictitious) gray level to each linked set of points to keep account of **different edges**.

![Diagram](image)

**Canny edge detector**

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression
   - Thin multi-pixel wide “ridges” down to single pixel width
4. Linking of edge points
   - Hysteresis thresholding: use a higher threshold to start edge curves and a lower threshold to continue them

Source: D. Lowe, L. Fei-Fei
Hysteresis thresholding

- Use a high threshold to start edge curves and a low threshold to continue them
  - Reduces drop-outs

Source: S. Seitz

Hysteresis thresholding

original image

high threshold (strong edges)  low threshold (weak edges)  hysteresis threshold

Source: L. Fei-Fei
Effect of \( \sigma \) (Gaussian kernel spread/size)

<table>
<thead>
<tr>
<th>Original</th>
<th>Canny with ( \sigma = 1 )</th>
<th>Canny with ( \sigma = 2 )</th>
</tr>
</thead>
</table>

The choice of \( \sigma \) depends on desired behavior
- large \( \sigma \) detects large scale edges
- small \( \sigma \) detects fine features

Source: S. Seitz

Edge detection is just the beginning…

- Berkeley segmentation database:
  [http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/](http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/)
Fitting Curves to Points:

Parameter space representation

- A line in the image corresponds to a point in Hough space

Image space

\[ y = m_0x + b_0 \]

Hough parameter space

Source: S. Seitz
Parameter space representation

- What does a point \((x_0, y_0)\) in the image space map to in the Hough space?

- Answer: the solutions of \(b = -x_0m + y_0\)
- This is a line in Hough space
Parameter space representation

- Where is the line that contains both 
  \((x_0, y_0)\) and \((x_1, y_1)\)?

\[ b = -x_1 m + y_1 \]
Parameter space representation

- Problems with the (m,b) space:
  - Unbounded parameter domain
  - Vertical lines require infinite m

- Alternative: polar representation

\[ x \cos \theta + y \sin \theta = \rho \]

Each point will add a sinusoid in the (\(\theta, \rho\)) parameter space.

Hough Transform

- Simplest case: Lines
  - Fit a straight line to a set of edge pixels.
  - Hough Transform (1962): Pattern Matching
  - Parameterization of a line in the plane:

\[ x \cos \theta + y \sin \theta = \rho \]

Hough Transform Algorithm

- Subdivide the parameter plane into bins
- Accumulate the total number sinusoids that cross each bin
- Threshold the value of the bins in the parameter plane to declare the presence of lines
  - Note the effect of the length of lines on the accumulated values
- Note similarity to the Radon Transform

Hough Transform Examples

Original image

Courtesy: P. Salembier
Hough Transform Examples

Original image

![Hough Transform Examples](image.png)

Courtesy: P. Salembier
Hough Transform Examples

Original IC image (256x256)

Effect of noise

- Peak gets fuzzy and hard to locate
Random points

- Uniform noise can lead to spurious peaks in the array

Dealing with noise

- Choose a good grid / discretization
  - Too coarse: large votes obtained when too many different lines correspond to a single bucket
  - Too fine: miss lines because some points that are not exactly collinear cast votes for different buckets
- Increment neighboring bins (smoothing in accumulator array)
- Try to get rid of irrelevant features
  - Take only edge points with significant gradient magnitude
Incorporating image gradients

• Recall: when we detect an edge point, we also know its gradient direction
• But this means that the line is uniquely determined!

• Modified Hough transform:
  • For each edge point \((x,y)\)
    \(\theta = \text{gradient orientation at } (x,y)\)
    \(\rho = x \cos \theta + y \sin \theta\)
    \(H(\theta, \rho) = H(\theta, \rho) + 1\)
  end

Hough transform for circles

• How many dimensions will the parameter space have?
• Given an oriented edge point, what are all possible bins that it can vote for?
Hough transform for circles

Conceptually equivalent procedure: for each \((x, y, r)\), draw the corresponding circle in the image and compute its “support”.

Is this more or less efficient than voting with features?
Generalized Hough Transform: Circles

- Parametric equation for a circle of (fixed) radius $r$

$$\begin{align*}
  x &= r \cos \theta + x_0 \\
  y &= r \sin \theta + y_0
\end{align*}$$

- Identify circle centers.
- Count the number of edge pixels on these circles.
- Change $r$, repeat.

Example of Circle Detection
Segmentation by Thresholding

Global Thresholding

Local Thresholding

Some Statistical Analysis

- Histograms are almost never so nicely separated.

- Suppose each pixel belongs to foreground with probability $a$, and to background with probability $1-a$.

- In classifying any pixel with threshold $T$, we can make two types of errors.

\[
e_1(T) = \int_{-\infty}^{T} p_2(z)dz
\]

Probability of misclassifying a background pixel as foreground

\[
e_2(T) = \int_{T}^{\infty} p_1(z)dz
\]

Probability of misclassifying a foreground pixel as background
Optimal Thresholding

- Pick threshold $T$ to minimize the overall expected probability of error. $e(T) = (1-a)e_1(T) + ae_2(T)$

\[
\frac{\partial e(T)}{\partial T} = (1-a) \frac{\partial}{\partial T} \int_{-\infty}^{T} p_2(z)dz + a \frac{\partial}{\partial T} \int_{T}^{\infty} p_1(z)dz
\]

\[= (1-a)p_2(T) + ap_1(T) = 0\]

\[\frac{p_1(T)}{p_2(T)} = \frac{1-a}{a}\]

Solve for threshold $T$

- Example: Two Gaussians with same variance and different means.

\[T = \frac{m_1 + m_2}{2} + \frac{\sigma^2}{m_1 - m_2} \ln \left( \frac{a}{1-a} \right)\]

Special case: $a=1/2$

\[T = \frac{m_1 + m_2}{2}\]

Morphological Processing

- A binary image can be considered as a set by considering “black” pixels (with image value “1”) as elements in the set and “white” pixels (with value “0”) as outside the set.

- Morphological filters are essentially set operations
Operations on Sets of Points

- Let $x, y, z, \ldots$ represent locations of 2D pixels, e.g. $x = (x_1, x_2)$, $S$ denote the complete set of all pixels in an image, let $A, B, \ldots$ represent subsets of $S$.

- **Union (OR)** $A \cup B = \{x : x \in A \text{ or } x \in B\}$

- **Intersection (AND)** $A \cap B = \{x : x \in A \text{ and } x \in B\}$

Other Set Operations

- **Complement** $\overline{A} = \{x : x \in S \text{ and } x \notin A\}$

- **Translation** $\{z : z = y + x, y \in A\}$

- **Reflection** $\hat{A} = \{y : y = -x, x \in A\}$
Dilation

- **Dilation** of set $F$ with a structuring element $H$ is represented by $F \oplus H$
  \[ F \oplus H = \{ x : (\hat{H})_x \cap F \neq \emptyset \} \]
  where $\emptyset$ represents the empty set.

- $G = F \oplus H$ is composed of all the points that when $\hat{H}$ shifts its origin to these points, at least one point of $\hat{H}$ is included in $F$.

- If the origin of $H$ takes value “1”, $F \subseteq F \oplus H$

**Dilation Example**

- $H$, 3x3, origin at the center
- $H$, 5x3, origin at the center
Another Example of Dilation

![Dilation Example](diagram1)

Erosion (Opposite of Dilation)

![Erosion Example](diagram2)
Another Erosion Example

F
H, 3x3, origin at the center
G

Closing and Opening

- Closing
  \[ F \bullet H = (F \oplus H) \ominus H \]
  - Smooth the contour of an image
  - Fill small gaps and holes

- Opening
  \[ F \circ H = (F \ominus H) \oplus H \]
  - Smooth the contour of an image
  - Eliminate false touching, thin ridges and branches
Closing Example

\[ F \]
\[ F \ast H \]
\[ (F \ast H) \ast H \]

H, 3x3, origin at the center

Opening Example

\[ F \]
\[ F \Theta H \]
\[ (F \Theta H) \Theta H \]

H, 3x3, origin at the center
Morphological Filtering Example: 
Closing, followed by Opening

![Diagram showing morphological operations]

Morphological Boundary Detection

- Morphological gradient \((f \oplus h) - (f \Theta h)\)
  - The difference between the dilated and eroded images,
- Valley detection \(f 
\bullet h - f\)
  - Detect dark text/lines from a gray background
- Boundary detection \(f - f \Theta h\)