Edges, Contours and Lines

CS 6384 Computer Vision
Professor Yu Xiang
The University of Texas at Dallas

Some slides of this lecture are courtesy Robert Collins (PSU)
Keypoint Features vs. Edge Points

Keypoints
• Good for feature matching
• Less or no semantic meaning

Edges
• Not robust for feature matching
• With semantic meanings (object boundaries, occlusion boundaries, shadows, etc.)
David Marr’s Theory of Vision (Neuroscientist)

Edges

- Edges occur at boundaries between regions of different color, intensity or texture.

Step Edge, Ramp Edge

Ridge Edge

Roof Edge
Step Edge and Ramp Edge

Step Edge, Ramp Edge

Figure 2
Ridge Edge

Figure 2

Ridge Edge
Ridge Edge
Roof Edge
Image Gradients

• Use image gradients

Central difference

\[ f'(x) = \lim_{h \to 0} \frac{f(x + 0.5h) - f(x - 0.5h)}{h} \]

\[
\begin{bmatrix}
-1 & 0 & 1
\end{bmatrix}
\]

\( X \) derivative
Image Gradients

Gradient Vector: \( \nabla I = \left[ \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]^T \)

Magnitude:
\[
|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}
\]

Orientation:
\[
\theta = \arctan2\left(\frac{\partial I}{\partial y}, \frac{\partial I}{\partial x}\right)
\]
Edge Normal and Edge Direction

• Edge normal
  • Unit vector in the direction of maximum intensity change
  • Gradient direction

• Edge direction
  • Unit vector along edge (perpendicular to edge normal)
Edge Detection

- A simple edge detector using gradient magnitude

  1. Compute gradient vector at each pixel by convolving image with horizontal and vertical derivative filters

  2. Compute gradient magnitude at each pixel

  3. If magnitude at a pixel exceeds a threshold, report a possible edge point
Edge Detection

- $I(x,y)$
- $I_x$
- $I_y$

Magnitude of gradients

Threshold
Mag > 30
Edge Detection

\[ \frac{\partial}{\partial x} (h \ast f) = \left( \frac{\partial}{\partial x} h \right) \ast f \]

Smoothing

Gaussian:

\[ h_{\sigma}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}} \]

\[ \frac{\partial}{\partial x} h_{\sigma}(u, v) \]

Derivative of Gaussian (x)

Image with Edge

Edge Location

Image + Noise

Derivatives detect edge and noise

Smoothed derivative removes noise, but blurs edge

2/16/2022

Yu Xiang
Edge Detection

• Thinning

Along a 1D intensity slice normal to the curve (non-maximum suppression)
  • Direction of gradient
Edge Detection

• How to chose the threshold?
Edge Detection

• How to chose the threshold?

• Hysteresis thresholding
  • Keep a high threshed H and a low threshold L
  • Any edge with strength < L is discarded
  • Any edge with strength > H is kept
  • An edge P with strength between L and H is kept only if there is a path of edges with strength > L connecting P to an edge of strength > H

J. Canny A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, Nov 1986
Contour Detection

• Link edge points into contours
  • Check neighboring pixels

• How to store contours?
  • A list of edgels (edge points)
  • (x, y) coordinates
Contour Detection

• How to store contours?
• Chain code
  • Initial coordinates
  • 8 directions (N, NE, E, SE, S, SW, W, NW)
  • 3 bits (can further be compressed)
  
010765

• Not suitable for further processing
Contour Detection

• How to store contours?
• Arc-length parameterization $\mathbf{x}(s)$

- Start point $(1.0, 0.5)$, $s = 0$
- Next point $(2.0, 0.5)$, $s = 1$
- Next point $(2.5, 1.0)$, $s = 1.7071$

- Can be resampled
- Fourier transform by treating $(x, y)$ as a complex number (contour matching)
mPb Contour Detector

- Oriented gradient of histograms \( G(x, y, \theta) \)

Gradient magnitude:

\[
\chi^2 = \text{distance between the two histograms}
\]

\[
\chi^2(g, h) = \frac{1}{2} \sum_i \frac{(g(i) - h(i))^2}{g(i) + h(i)}
\]

Histogram of intensity  
Radius: 5 pixels \( \theta = \frac{\pi}{4} \)

Pablo Arbeláez, Charless Fowlkes, Jitendra Malik. Contour Detection and Hierarchical Image Segmentation. TPAMI’10
mPb Contour Detector

- Brightness, color, texture gradients
  - L*a*b color space: brightness, color a and color b
  - Texture: assign a pixel to a texton id

- 17 Gaussian derivative filters
  - 17D feature vector for each pixel

- K-means clustering, K = 32, textons
- Texture image: pixels with integer [1, K]

https://cran.r-project.org/web/packages/colordist ance/vignettes/color-spaces.html
mPb Contour Detector

Consider multiple scales \([\frac{\sigma}{2}, \sigma, 2\sigma]\)

- \(\sigma = 5\) pixels for brightness
- \(\sigma = 10\) pixels for color and texture

\[
mPb(x, y, \theta) = \sum_s \sum_i \alpha_{i,s} G_{i,s}(x, y, \theta)
\]

Scale Channel (brightness, color a, color b, texture)

\[
mPb(x, y) = \max_\theta \{mPb(x, y, \theta)\}
\]

maximum response over eight orientations in \([0; \pi]\)
Lines

• Lines are common in the human-made world

Lines

- 2D lines

\[ ax + by + c = 0 \]
\[ l = (a, b, c) \]

Normalize by \( \sqrt{a^2 + b^2} \)
\[ l = (\hat{n}_x, \hat{n}_y, d) = (\hat{n}, d) \]

\[ \hat{n} = (\hat{n}_x, \hat{n}_y) = (\cos \theta, \sin \theta) \]

polar coordinates \((\theta, d)\)

\[ x \cos \theta + y \sin \theta + d = 0 \]
Line Detection

- Hough transform
  - Observations vote for model parameters
  - Observations? $(x_i, y_i)$  
  - Model parameters? $(r, \theta)$

\[ r_i(\theta) = x_i \cos \theta + y_i \sin \theta \]

Parameter space (discretized in implementation)

\[ a \cos x + b \sin x = c \cos(x + \varphi) \]
where \(c\) and \(\varphi\) are defined as so:

\[ c = \text{sgn}(a) \sqrt{a^2 + b^2}, \]
\[ \varphi = \arctan\left(-\frac{b}{a}\right), \]
given that \(a \neq 0\).

Line Detection

- Oriented Hough Transform
  - Use gradient orientation as theta

\[ r_i(\theta) = x_i \cos \theta + y_i \sin \theta \]

\[ \theta = \text{atan2}(\frac{\partial I}{\partial y}, \frac{\partial I}{\partial x}) \]

Orientation

\[ \hat{n}_i = (\cos \theta_i, \sin \theta_i) \]

\[ r_i = \hat{n}_i \cdot x_i \]
Line Detection

• Random Sample Consensus (RANSAC)

RANSAC Algorithm {
1. Selects $N$ data items as random
2. Estimates parameter $\hat{x}$
3. Finds how many data items (of $M$) fit the model with parameter vector $\hat{x}$ within a user given tolerance. Call this $K$.
4. If $K$ is big enough, accept fit and exit with success.
5. Repeat step 1 until 4 (as $L$ times)
6. Algorithm will be exit with fail
}

• Sample two edge points
• Estimate the line parameter $$(\theta, d)$$
$$x \cos \theta + y \sin \theta + d = 0$$
• Find how many edgels obey it
Vanishing Points

- Parallel lines in 3D converge in 2D images due to perspective projection

Recovering the Spatial Layout of Cluttered Rooms. Hedau et al., ICCV’09
Application: AprilTag Detection

AprilTag: A robust and flexible visual fiducial system. Edwin Olson. ICRA, 2011
Further Reading

• Section 7.2, 7.4, Computer Vision, Richard Szeliski


• Pablo Arbelaez, Charless Fowlkes, Jitendra Malik. Contour Detection and Hierarchical Image Segmentation. TPAMI’10