Edge Detection, Corner Detection

Lines

Introduction to Computer Vision
CSE 152
Lecture 10

Announcements

• Assignment 2 due Tuesday, May 3.
• Midterm: Thursday, May 5.

Last Lecture

Edges

1. Object boundaries
2. Surface normal discontinuities
3. Reflectance (albedo) discontinuities
4. Lighting discontinuities (shadow boundaries)

Numerical Derivatives

Take Taylor series expansion of \( f(x) \) about \( x_0 \):
\[
f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2} f''(x_0)(x-x_0)^2 + \cdots
\]
Consider samples taken at increments of \( h \) and first two terms, we have:
\[
f(x_0+h) = f(x_0) + f'(x_0)h + \frac{1}{2} f''(x_0)h^2
\]
\[
f(x_0-h) = f(x_0) - f'(x_0)h + \frac{1}{2} f''(x_0)h^2
\]
Subtracting and adding \( f(x_0+h) \) and \( f(x_0-h) \) respectively yields:
\[
\frac{f'(x_0)}{2h} = \frac{f(x_0+h) - f(x_0-h)}{2h}
\]
\[
\frac{f''(x_0)}{h^2} = \frac{f(x_0+h) - 2f(x_0) + f(x_0-h)}{h^2}
\]
Convolving with First Derivative:
\([-1/2h, 0, 1/2h] \]
Second Derivative:
\([-1/h^2, 2/h^2, -1/h^2] \]
Can often drop \( h \) or \( h^2 \) in denominator Yielding \([1 0 1]\) and \([1 -2 1]\) kernels.
Implementing 1-D Edge Detection

1. Filter out noise: convolve with Gaussian
2. Take a derivative: convolve with [-1 0 1]
   - We can combine 1 and 2.
3. Find the peak of the magnitude of the convolved image: Two issues:
   - Should be a local maximum.
   - Should be sufficiently high.

Canny Edge Detector

1. Smooth image by filtering with a Gaussian
2. Compute gradient at each point in the image.
3. At each point in the image, compute the direction of the gradient and the magnitude of the gradient.
4. Perform non-maximal suppression to identify candidate edgels.
5. Trace edge chains using hysteresis thresholding.

Gradients:
\[ \nabla I = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \]
Is this dI/dx or dI/dy?

Non-maximum suppression

Using gradient direction at q, find two points p and r on adjacent rows (or columns).
If \[ |\nabla I(x)\cdot\nabla I(y)| \]
and then q remains a candidate edgel
p & r are found by interpolation

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).
HysteresisThresholding

- Start tracking an edge chain at pixel location that is local maximum of gradient magnitude where gradient magnitude > $\tau_{\text{high}}$.
- Follow edge in direction orthogonal to gradient.
- Stop when gradient magnitude < $\tau_{\text{low}}$.
  - i.e., use a high threshold to start edge curves and a low threshold to continue them.

Corner Detection

Feature extraction: Corners and blobs

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Step 1: extract features
Step 2: match features
Step 3: align images
Corners contain more info than lines.

- A point on a line is hard to match.

- A corner is easier to match

The Basic Idea

- We should easily recognize the point by looking through a small window
- Shifting a window in any direction should give a large change in intensity

Finding Corners

Intuition:

- Right at corner, gradient is ill-defined.
- Near corner, gradient has two different values.

Distribution of gradients for different image patches

Formula for Finding Corners

We look at matrix:

\[
C(x,y) = \left[ \frac{\sum I_x^2}{\sum I_x I_y} \right]
\]

Matrix is symmetric

\[WHY\ THIS?\]
General Case:

Because $C$ is a symmetric positive definite matrix, it can be factored as follows:

$$C = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

Where $R$ is a 2x2 rotation matrix and $\lambda_i$ is non-negative.

What is region like if:

1. $\lambda_1 = 0$?
2. $\lambda_2 = 0$?
3. $\lambda_1 = 0$ and $\lambda_2 = 0$?
4. $\lambda_1 > 0$ and $\lambda_2 > 0$?

General Case

Since $C$ is symmetric, we have

$$C = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

We can visualize $C$ as an ellipse with axis lengths determined by the eigenvalues and orientation determined by $R$.

Ellipse equation:

$$\begin{bmatrix} u \\ v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$

So, to detect corners

- Filter image with a Gaussian.
- Compute the gradient everywhere.
- Move window over image and construct $C$ over the window.
- Use linear algebra to find $\lambda_1$ and $\lambda_2$.
- If they are both big, we have a corner.

1. Let $e(x,y) = \min(\lambda_1(x,y), \lambda_2(x,y))$
2. $(x,y)$ is a corner if it’s local maximum of $e(x,y)$ and $e(x,y) > \tau$

Parameters: Gaussian std. dev, window size, threshold

What to do with edges?

- Segment linked edge chains into curve features (e.g., line segments).
- Group unlinked or unrelated edges into lines (or curves in general).
- Accurately fitting parametric curves (e.g., lines) to grouped edge points.
Hough Transform
[ Patented 1962 ]

Finding lines in an image
Option 1:
• Search for the line at every possible position/orientation
• What is the cost of this operation?

Option 2:
• Use a voting scheme: Hough transform

Hough transform algorithm
Because vertical lines can't be represented,

Typically use a different parameterization
\[ d = x \cos \theta + y \sin \theta \]
• \( d \) is the perpendicular distance from the line to the origin
• \( \theta \) is the angle this perpendicular makes with the x axis
• Why?

Hough transform algorithm
Basic Hough transform algorithm
1. Initialize \( H(d, \theta) = 0 \); \( H \) is called accumulator array
2. for each edge point \((x,y)\) in the image
   for \( \theta = 0 \) to 180
   \[ l = x \cos \theta + y \sin \theta \]
   \( H(d, \theta) = H(d, \theta) + 1 \)
3. Find the value(s) of \((d, \theta)\) where \(H(d, \theta)\) is maximum
4. The detected line in the image is given by \( l = x \cos \theta + y \sin \theta \)

What's the running time (measured in \# votes)?

Hough Transform: 20 colinear points
• R, \( \theta \) representation of line
• Maximum accumulator value is 20
Hough Transform: “Noisy line”
- $R, \theta$ representation of line
- Maximum accumulator value is 6

Hough Transform: Random points
- $R, \theta$ representation of line
- Maximum accumulator value is 4

Mechanics of the Hough transform
- Difficulties
  - How big should the cells be? (too big, and we cannot distinguish between quite different lines; too small, and noise causes lines to be missed)
- How many lines?
  - Count the peaks in the Hough array
- Which edgels belong to which line?
  - Tag the votes
- Complications, problems with noise and cell size

Number of votes that the real line of 20 points gets with increasing noise