Edge Detection

- What’s an edge
- Image gradients
- Edge operators
Edges seem fundamental to human perception.
They form a compressed version of the image.
From Edges To Cats

Deep-Learning based generative model.

Demo

https://affinelaye.com/pixsrv/
Maps and Overlays
Corridor
Corridor
Edges and Regions

Edges:
- Boundary between bland image regions.

Regions:
- Homogenous areas between edges.

→ Edge/Region Duality.
Discontinuities

- A. Depth discontinuity: Abrupt depth change in the world
- B. Surface normal discontinuity: Change in surface orientation
- C. Illumination discontinuity: Shadows, lighting changes
- D. Reflectance discontinuity: Surface properties, markings

→ Sharply different Gray levels on both sides
REALITY
More Reality

Very noisy signals
→ Prior knowledge is required!!
Optional: Illusory Contours

- No closed contour, but we still perceived an edge.
- This will not be further discussed in this class.
Ideal Step Edge

Rapid change in image => High local gradient

\[ f(x) = \text{step edge} \]

1\textsuperscript{st} Derivative \( f'(x) \)

2\textsuperscript{nd} Derivative \( f''(x) \)

maximum

zero crossing
Edge Properties

Original

Orientation

Contrast
Edge Descriptors

- **Edge Normal:**
  - Unit vector in the direction of maximum intensity change
- **Edge Direction:**
  - Unit vector perpendicular to the edge normal
- **Edge position or center**
  - Image location at which edge is located
- **Edge Strength**
  - Speed of intensity variation across the edge.
Images as 3-D Surfaces
Geometric Interpretation

Since $I(x,y)$ is not a continuous function:
1. Locally fit a smooth surface.
2. Compute its derivatives.
Image Gradient

The gradient of an image

$$\nabla I = \begin{bmatrix} \frac{\delta I}{\delta x}, \frac{\delta I}{\delta y} \end{bmatrix}$$

points in the direction of most rapid change in intensity.
Magnitude And Orientation

\[
\left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)
\]

Measure of contrast: \( G = \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}} \)

Edge orientation: \( \theta = \arctan\left( \frac{\partial I}{\partial y}, \frac{\partial I}{\partial x} \right) \)
Gradient Images

The gradient operator is rotationally invariant ....
Real Images

... but not directly usable in most real-world images.
Edge Operators

- Difference Operators
- Convolution Operators
- Trained Detectors
- Deep Nets
Gradient Methods

\[ F(x) \]

Edge = Sharp variation

Large first derivative
1D Finite Differences

In one dimension:

\[
\frac{df}{dx} \approx \frac{f(x + dx) - f(x)}{dx} \approx \frac{f(x + dx) - f(x - dx)}{2dx}
\]

\[
\frac{d^2 f}{dx^2} \approx \frac{f(x + dx) - 2f(x) + f(x - dx)}{dx^2}
\]
Coding 1D Finite Differences

Line stored as an array:

- for i in range(n-1):
  q[i]=(p[i+1]-p[i])

- for i in range(1,n-1):
  q[i]=(p[i+1]-p[i-1])/2

- q=(p[2:]-p[:-2])/2
2D Finite Differences

\[
\begin{align*}
\frac{\partial f}{\partial x} & \approx \frac{f(x + dx, y) - f(x, y)}{dx} \quad \approx \frac{f(x + dx, y) - f(x - dx, y)}{2dx} \\
\frac{\partial f}{\partial y} & \approx \frac{f(x, y + dy) - f(x, y)}{dy} \quad \approx \frac{f(x, y + dy) - f(x, y - dy)}{2dy}
\end{align*}
\]
Coding 2D Finite Differences

Image stored as a 2D array:

- \( dx = p[1:,\cdot]-p[:-1,\cdot] \)
- \( dy = p[\cdot,1:]-p[\cdot,:-1] \)

- \( dx = (p[2:,\cdot]-p[:-2,\cdot])/2 \)
- \( dy = (p[\cdot,2:]-p[\cdot,:-2])/2 \)

Image stored in raster format:

```c
int i;
for(i=0;i<xdim;i++){
    dx[i] = p[i+1] - p[i];
    dy[i] = p[i+xdim] - p[i];
}
```

- Only 1D array accesses
- No multiplications
  
  \( \rightarrow \) Can be exploited to increase speed.
Noise in 1D

Consider a single row or column of the image:

\[
\frac{d}{dx} f(x)
\]
Fourier Interpretation

Differentiating emphasizes high frequencies and therefore noise!

<table>
<thead>
<tr>
<th>Function</th>
<th>Fourier Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{df}{dx}(x)$</td>
<td>$uF(u)$</td>
</tr>
<tr>
<td>$\frac{df}{\delta x}(x, y)$</td>
<td>$uF(u, v)$</td>
</tr>
<tr>
<td>$\frac{df}{\delta y}(x, y)$</td>
<td>$vF(u, v)$</td>
</tr>
</tbody>
</table>

→ Differentiating emphasizes high frequencies and therefore noise!
\[ f(x) = x^2 \sin\left(\frac{1}{x}\right) \]

Original function

+ Noise

Fourier transform

\[ \frac{df}{dx} \]

\[ F \]

\[ uF \]
Noise in 2D

Ideal step edge

Step edge + noise

Increasing noise level

As the amount of noise increases, the derivatives stop being meaningful.
Removing Noise

Problem:

• High frequencies and differentiation do not mix well.

Solution:

• Suppress high frequencies by
  • using the Discrete Fourier Transform.
Discrete Fourier Transform

\[
F(\mu, \nu) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2i\pi(\mu x/M + \nu y/N)}
\]

\[
f(x, y) = \frac{1}{\sqrt{MN}} \sum_{\mu=0}^{M-1} \sum_{\nu=0}^{N-1} F(\mu, \nu) e^{+2i\pi(\mu x/M + \nu y/N)}
\]

The DFT is the discrete equivalent of the 2D Fourier transform:
• The 2D function \( f \) is written as a sum of sinusoids.
• The DFT of \( f \) convolved with \( g \) is the product of their DFTs.
Fourier Basis Element

Real part of $e^{+2i\pi(ux+vy)}$

where

- $\sqrt{u^2 + v^2}$ represents the frequency,
- $\text{atan}(v, u)$ represents the orientation.
Fourier Basis Element

Real part of \( e^{+2i\pi(ux+vy)} \)

where

- \( \sqrt{u^2 + v^2} \) is larger than before.
Fourier Basis Element

Real part of $e^{2i\pi(ux+vy)}$

where

• $\sqrt{u^2 + v^2}$ is larger still.
Truncated Inverse DFT

\[ F(\mu, \nu) = \frac{1}{\sqrt{M*N}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)e^{-2i\pi(\mu x/M+\nu y/N)} \]

\[ f(x, y) = \frac{1}{\sqrt{M*N}} \sum_{\mu=0}^{M-1} \sum_{\nu=0}^{N-1} F(\mu, \nu)e^{2i\pi(\mu x/M+\nu y/N)} \]

\[ f(x, y) = \frac{1}{\sqrt{M*N}} \sum_{\mu^2+\nu^2<T} F(\mu, \nu)e^{2i\pi(\mu x/M+\nu y/N)} \]

- The sinusoids corresponding to \( \mu^2 + \nu^2 \geq T \) depict high frequencies.
- Removing them amounts to removing high-frequencies.
Smoothing by Truncating the IDFT

Rotated stripes:

• Dominant diagonal structures
• Discretization produces additional harmonics

→ Removing higher frequencies and reconstructing yields a smoothed image.
Removing Noise

**Problem:**
- High frequencies and differentiation do not mix well.

**Solution:**
- Suppress high frequencies by
  - using the Discrete Fourier Transform,
  - convolving with a low-pass filter.
1D Convolution

\[ g \ast f(t) = \int_{\tau} g(t - \tau) f(\tau) d\tau \]
Smooth Before Differentiating

\[ f \]

\[ g \]

\[ g^* f \]

\[ \frac{\partial}{\partial x} (g^* f) \]
Simultaneously Smooth and Differentiate

\[ \frac{\partial}{\partial x} (g \ast f) = \frac{\partial g}{\partial x} \ast f \]

--> Faster because \( \frac{dg}{dx} \) can be precomputed.
Discrete 1D Convolution

Input

\[ 1 \quad 4 \quad -1 \quad 0 \quad 2 \quad -2 \quad 1 \quad 3 \quad 3 \quad 1 \]

Mask

\[ 1 \quad 2 \quad 0 \quad -1 \]
Discrete 1D Convolution

Input

\[ W - w + 1 \]

Output

\[ W - w + 1 \]
Discrete 1D Convolution

Input

Output

\[ W - w + 1 \]
Discrete 1D Convolution

Input:

\[
\begin{array}{cccccccc}
1 & 4 & -1 & 0 & 2 & -2 & 1 & 3 & 3 & 1 \\
\end{array}
\]

Output:

\[
\begin{array}{c}
9 & 0 & 1 \\
\end{array}
\]
Discrete 1D Convolution

Input

\[
\begin{array}{cccccccc}
1 & 4 & -1 & 0 & 2 & -2 & 1 & 3 & 3 & 1 \\
\end{array}
\]

\(w\)

Output

\[
\begin{array}{cccc}
9 & 0 & 1 & 3 \\
\end{array}
\]

\(W - w + 1\)
1D Convolution

Input

\[
\begin{array}{ccccccccc}
1 & 4 & -1 & 0 & 2 & -2 & 1 & 3 & 3 & 1
\end{array}
\]

Output

\[
\begin{array}{cccc}
9 & 0 & 1 & 3 & -5
\end{array}
\]

\[W - w + 1\]
Discrete 1D Convolution

Input

\[
\begin{array}{cccccccc}
1 & 4 & -1 & 0 & 2 & -2 & 1 & 3 & 3 & 1
\end{array}
\]

Output

\[
\begin{array}{cccc}
9 & 0 & 1 & 3 & -5 & -3
\end{array}
\]
Discrete 1D Convolution

Input

\[
\begin{array}{cccccc}
1 & 4 & -1 & 0 & 2 & -2 \\
& & & & & 1 & 3 & 3 & 1
\end{array}
\]

\[W\]

Output

\[
\begin{array}{cccccccc}
9 & 0 & 1 & 3 & -5 & -3 & 6
\end{array}
\]

\[W - w + 1\]
Discrete 1D Convolution

\[ m \ast f(x) = \sum_{i=0}^{W} m(i)f(x - i) \]
Discrete 2D Convolution

Convolution mask \( m \), also known as a \textit{kernel}.

\[
\begin{bmatrix}
m_{11} & \cdots & m_{1w} \\
\vdots & \ddots & \vdots \\
m_{w1} & \cdots & m_{ww}
\end{bmatrix}
\]

\[
m**f(x, y) = \sum_{i=0}^{w} \sum_{j=0}^{w} m(i, j)f(x - i, y - j)
\]
Differentiation As Convolution

\[
[-1,1] \rightarrow \frac{\partial f}{\partial x} \approx \frac{f(x + dx, y) - f(x, y)}{dx}
\]

\[
[-0.5,0,0.5] \rightarrow \frac{\partial f}{\partial x} \approx \frac{f(x + dx, y) - f(x - dx, y)}{2dx}
\]

\[
[-1] \rightarrow \frac{\partial f}{\partial y} \approx \frac{f(x, y + dy) - f(x, y)}{dy}
\]

\[
[-0.5] \rightarrow \frac{\partial f}{\partial y} \approx \frac{f(x, y + dy) - f(x, y - dy)}{2dy}
\]

→ Use wider masks to add some smoothing
Smoothing and Differentiating

Compute the difference of averages on either side of the central pixel.
3x3 Masks

x derivative

\[
\begin{bmatrix}
-1 & 0 & 1 \\
-1 & 0 & 1 \\
-1 & 0 & 1
\end{bmatrix}
\quad \text{and} \quad
\begin{bmatrix}
1 & 1 & 1 \\
0 & 0 & 0 \\
-2 & 0 & 2
\end{bmatrix}
\]

Prewitt operator

y derivative

\[
\begin{bmatrix}
-1 & -1 & -1 \\
0 & 0 & 0 \\
1 & 1 & 1
\end{bmatrix}
\quad \text{and} \quad
\begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1
\end{bmatrix}
\]

Sobel operator
Prewitt Example

Santa Fe Mission

Gradient Image
Sobel Example
Gaussian Smoothing

- More principled way to eliminate high frequency noise.
- Is fast because the kernel is
  - small,
  - separable.
Gaussian Functions

\[ g_2(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right) \]

- The integral is always 1.0
- The larger \( \sigma \), the broader the Gaussian is.
- As \( \sigma \) approaches 0, the Gaussian approximates a Dirac function.
DFT of a Gaussian

• The DFT of a Gaussian is a Gaussian.
• It has finite support.
• Its width is inversely proportional to that of the original Gaussian.
Gaussians as Low-Pass Filters

• The Fourier transform of a convolution is the product of their Fourier transforms: $\mathcal{F}(g * f) = \mathcal{F}(g)\mathcal{F}(f)$.
• If $g$ is a Gaussian, so is $\mathcal{F}(g)$.
• Furthermore if $g$ is broad, the support of $\mathcal{F}(g)$ is small.
• So is the support of $\mathcal{F}(g * f)$.
• There are no more high-frequencies in $g * f$.

$\implies$ Convolving with a Gaussian suppresses the high frequencies.
Gaussian Smoothed Images

\[ \sigma = 1 \]

\[ \sigma = 2 \]

\[ \sigma = 4 \]

Original image | Blurred image

Blur \( \sigma = 1 \) | Blur \( \sigma = 2 \) | Blur \( \sigma = 4 \)
Scale Space

Increasing scale ($\sigma$) removes high frequencies (details) but never adds artifacts.
Separability

\[ g_1(x) = \frac{1}{\sqrt{\pi \sigma}} \exp(-x^2 / \sigma^2) \]

\[ g_2(x, y) = g_1(x)g_1(y) \]

\[ \int_u \int_v g_2(u, v)f(x-u, y-v)du dv = \int_u g_1(u)(\int_v g_1(v)f(x-u, y-v)dv)du \]

\[ = \int_v g_1(v)(\int_u g_1(u)f(x-u, y-v)du)dv \]

\[ \rightarrow \text{2D convolutions are never required. Smoothing can be achieved by successive 1D convolutions, which is faster.} \]
Continuous Gaussian Derivatives

Image derivatives computed by convolving with the derivative of a Gaussian:

\[
\frac{\partial}{\partial x} \int_u \int_v g_2(u,v) f(x-u, y-v) dv du = \int_u g_1(u) \left( \int_v g_1(v) f(x-u, y-v) dv \right) du
\]

\[
\frac{\partial}{\partial y} \int_u \int_v g_2(u,v) f(x-u, y-v) dv du = \int_v g_1(v) \left( \int_u g_1(u) f(x-u, y-v) du \right) dv
\]
Discrete Gaussian Derivatives

Sigma=1:

\[
g : 0.000070 \quad 0.010332 \quad 0.207532 \quad 0.564131 \quad 0.207532 \quad 0.010332 \quad 0.000070 \\
g' : 0.000418 \quad 0.041330 \quad 0.415065 \quad 0.000000 \quad -0.415065 \quad -0.041330 \quad -0.000418
\]

Sigma=2:

\[
g : 0.005167 \quad 0.029735 \quad 0.103784 \quad 0.219712 \quad 0.282115 \quad 0.219712 \quad 0.103784 \quad 0.029735 \quad 0.005167 \\
g' : 0.010334 \quad 0.044602 \quad 0.103784 \quad 0.109856 \quad 0.000000 \quad -0.109856 \quad -0.103784 \quad -0.044602 \quad -0.010334
\]

\[\rightarrow\text{ Only requires 1D convolutions with relatively small masks.}\]
Increasing Sigma

Input Images

No Noise

Noise Added

Gradient Images

\( \sigma = 1 \)

\( \sigma = 2 \)

\( \sigma = 4 \)

No Noise

Noise Added

\[ \rightarrow \] Larger sigma values improve robustness but degrade precision.
Derivative Images

\[ \partial I / \partial x \]

\[ \partial I / \partial y \]

\[ \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \]
Derivative Images

\[ I \]

\[ \frac{\partial I}{\partial x} \]

\[ \frac{\partial I}{\partial y} \]

\[ \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}} \]
Gradient-Based Tracking

Maximize edge-strength along projection of the 3-D wireframe.
Gradient Maximization
Real-Time Tracking
Canny Edge Detector

\[ I = \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}} \text{ Thinned gradient image} \]
Canny Edge Detector

Convolution
  - Gradient strength
  - Gradient direction

Thresholding
  Non Maxima Suppression
  Hysteresis Thresholding
Non-Maxima Suppression

Check if pixel is local **maximum** along gradient direction, which requires checking interpolated pixels p and r.
Hysteresis Thresholding

• Algorithm takes two thresholds: high & low
  • A pixel with edge strength above high threshold is an edge.
  • Any pixel with edge strength below low threshold is not.
  • Any pixel above the low threshold and next to an edge is an edge.

• Iteratively label edges
  • Edges grow out from ‘strong edges’
  • Iterate until no change in image.
Canny Results

$\sigma = 1, \ T2 = 255, \ T1 = 1$

‘Y’ or ‘T’ junction problem with Canny operator
Canny Results

σ=1, T2=255, T1=220

σ=1, T2=128, T1=1

σ=2, T2=128, T1=1

Heath et al., PAMI’97
Scale Space Revisited

Increasing scale (\(\sigma\)) removes details but never adds new ones:

- Edge position may shift.
- Two edges may merge.
- An edge may **not** split into two.
Multiple Scales

$\sigma = 1$  
$\sigma = 2$  
$\sigma = 4$

→ Choosing the right scale is a difficult semantic problem.
Scale vs Threshold

Fine scale
High threshold

Coarse scale
High threshold

Coarse scale
Low threshold
In industrial environments where the Canny parameters can be properly adjusted:

- It is fast.
- Does not require training data.

→ A useful tool in our toolbox.
Given an initial pose estimate:

- Find the occluding contours.
- Find closest edge points in the normal direction.
- Re-estimate pose to minimize sum of square distances.
- Iterate until convergence.
Visual Servoing
Space Cleaning

Capturing and deorbiting a dead satellite.

- A more sophisticated version of this old algorithm will blast off in 2025!
- ESA does not yet trust neural nets for such a mission.

https://clearspace.today/
Limitations of the Canny Algorithm

There is no ideal value of $\sigma$!
Steep Smooth Shading

- Rapidly varying gray levels.
- Large gradients.

→ Shading can produce spurious edges.
Texture Boundaries

- Not all image contours are characterized by strong contrast.
- Sometimes, textural changes are just as significant.
Different Boundary Types

Non-boundaries

Boundaries

Intensity

Brightness

Color

Texture

Martin et al., PAMI’04
Training Database

1000 images with 5 to 10 segmentations each.
Learn the probability of being a boundary pixel on the basis of a set of features.
Comparative Results

Image | Canny | 2MM | BG+CG+TG | Human
--- | --- | --- | --- | ---
![Image](image1.png) | ![Canny](can1.png) | ![2MM](2mm1.png) | ![BG+CG+TG](bg1.png) | ![Human](human1.png)

![Image](image2.png) | ![Canny](can2.png) | ![2MM](2mm2.png) | ![BG+CG+TG](bg2.png) | ![Human](human2.png)

Martin et al., PAMI’04
Classification vs Regression

Yes!
Deep Learning

Xie and Tu, ICCV’15
Deep Learning Vs Canny

(a) original image  (b) ground truth  (c) HED: output

(d) HED: side output 2  (e) HED: side output 3  (f) HED: side output 4

(g) Canny: $\sigma = 2$  (h) Canny: $\sigma = 4$  (i) Canny: $\sigma = 8$
Deeper Learning

Liu et al., CVPR’17
Convolutional Neural Network

- Succession of convolutional and pooling layers.
- Fully connected layers at the end.

—> Will be discussed in more detail in the next lecture.
A Partial Explanation?

First and second layer features of a Convolutional Neural Net:
- They can be understood as performing multiscale filtering.
- The weights and thresholds are chosen by the optimization procedure.
50 Years Of Edge Detection

- Convolution operators respond to steep smooth shading.
- Parametric matchers tend to reject non ideal edges.
- Arbitrary thresholds and scale sizes are required.
- Learning-based methods need exhaustive databases.
- There still is work to go from contours to objects.

Canny, PAMI’86 —> Sironi et al. PAMI’15  
Sironi et al. PAMI’15 —> Liu et al. , CVPR’17

Let us talk about deep networks.