Imaging processes imply that …

• Image of a scene strongly depends on pose & lighting
• Grey levels by themselves are almost meaningless
• The image itself is meaningful (where & what) but
  extracting the explicit meaning is complex
• Detecting similar scenes by comparing grey levels
  images often fails

Suggestion:
• Find Image features - meaningful image regions
• Use them later for description, comparison, matching

(Visual events)

Image Features

(General) Image Features

Point features
• Color, color invariants

Local features
• Edges
• Corners, interest points

Non-local features
• Lines, circles, ellipses, converging lines
• Texture

Many types of intensity Edges

• Object geometry,
• Occlusion,
• Albedo,
• Shadow

Also (not now)
• Color edges
• Texture edges
• Motion edges

(board: How edges are generated? Edge profiles).

Desirable properties
• Tell something about the world
• Invariant, Repeatable detection
  "Invariance" – insensitivity to non-interesting variables (lighting, pose, face expression, …)
• Reduce image complexity\ dimension (?)
• Easy to compute

Image features: Meaningful types of image regions

Edges

Edge - Significant discontinuity of intensity over a curve

Significant - large difference between grey levels
- occurs over non-zero length smooth curve

Why edges?
• Exist and common in (almost) all images
• Give information of objects in the scene
• Not so sensitive to illumination
• HVS early processing is based on edge analysis.
Detecting edges - Introduction

- The task: at every pixel, decide if it is an edge or not.
- Edge = location, direction, strength, (profile)

Edge - Significant discontinuity of intensity over a curve

- Goal: detecting the meaningful edges
  - Not well specified...
  - Is "meaningful" = boundaries of object?
  - What is "an object"?
  - Detect what people prefer
- Edgel - a term for edge pixel or short curve of edge pix.

Edges are informative

- The HVS perceive the edges and infer the rest
- Line drawings of object boundaries work for us.
  Why?

Detecting edges - I

- We look for a linear operation: \( R = C^T X \)
- Why linear? Simplest, fast, analyzable
- The decision is made by comparing the scalar \( R \) to a threshold.
- Design \( R \) to be large for edge, small for non-edge.
- \( R \) is proportional to \( \|C\| \). Let \( \|C\| = 1 \).
- \( R \) is a random variable distributed either according to \( N(0, \sigma^2) \) (no edge) or according to \( N(C^T S, \sigma^2) \) (edge).
- Which \( C \) is best? Any decision will err less by making the \( C^T S \) largest, which happen for \( C = S / \|S\| \).

Why?

Signal and Noise

A common way to analyze processes/algorithms:
Divide the given data to clean "signal" and "noise"

- Signal
  - Constant, characteristic, piecewise smooth/constant

- Noise
  - Non-characteristic, small scale changes (texture)
  - Correlated

Taking noise into account is essential!

Signal and Noise

A common way to analyze processes/algorithms:
Assumption: the given data = clean "signal" + "noise"

- Signal - a simplified model of the expected data
  - Common clean image signal: piecewise smooth/constant

- Noise - the part that is hard to predict/analyze
  - Noise sources: camera noise ("true"), quantization, small scale changes (texture)
  - Common noise model: white zero-mean (Gaussian) noise

Taking noise into account is essential!

Detecting edges - I

Signal detection approach

- Consider a fixed size region - a vector of grey levels \( X \)
- Consider an exactly known edge model: e.g. a straight edge in a given direction, with step profile between \( I_o, I_1 \)
- Consider two possibilities
  \[
  \text{Edge} : \quad X(j) = S(j) + N(j) \quad j = 1, \ldots, m \\
  \text{No Edge} : \quad X(j) = N(j) \quad j = 1, \ldots, m 
  \]

where \( N(j) \) is white, zero mean, Gaussian noise (b).

- Goal: make a decision between edge and no-edge
Detecting edges - I

\[ R = C^2 \times \]

- Detection is done by applying the inner product on all candidate regions and threshold.
- Name: Matched filter, correlation, template matching
- Comes from digital communication
- Can detect any known signal \( S \) in white noise.
- Theoretically optimal, larger region \( \rightarrow \) better detection (why?)

Problems:
- Non-edge != noise (subtract average grey level first)
- Edge profiles are not always step-like
- Edge parameter (orientation, size, ...) are not known (search, solve, multi-detection, steerability)

Detecting edges - II

The Gradient approach

Motivation: Edge - Significant discontinuity
- Find derivatives/ gradients
- Test gradient against threshold / find local maximum
- A function \( f(x,y) \)
  - The gradient \( \nabla f(x,y) = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \)

Implementation by Discrete Gradients
- Simplest and most local derivatives
- Symmetric derivatives
- Evaluated by inner product / convolution with masks
  \[
  \begin{bmatrix}
  1 & -1 \\
  -1 & 1
  \end{bmatrix}
  \]

Detecting edges using Gradients

- Find Gradient components
- Find magnitude everywhere
- Threshold or find local maximum

Fitting a local plane using Least Squares

The image is approximated locally by a Taylor series - a plane
\[ I(x,y) = I(0,0) + x f_x + y f_y + \ldots + y + c \]

For every point in the neighborhood, with coordinates and gray level \( x_j, y_j, I_j = I(x_j, y_j) \),
\[ I_j = ax_j + by_j + c. \]

Choose a neighborhood with \( N \geq 3 \) points and get an overdetermined set of equations

A least squares (LS) solution reduces estimation error
\[ (a^*, b^*, c^*) = \arg \min \sum (ax_j + by_j + c - I_j)^2 \]

Gradient of noisy signals

- Noise makes gradient non-accurate
- Derivatives amplify noise!
- Proof: derivatives amplify high frequencies which characterize noise but not piecewise smooth signal, so S/N ratio decreases.
  (example on board)

- In noisy/textured images we get many false edges
- Solution I: accept only strong responses (limited)
- Solution II: Use more than the minimal data - Fit a model
  - Smooth the image
Solving a least Square problem

The minimization

\[ \theta^* = \arg \min C(\theta) = \arg \min [H\theta - I]^T (H\theta - I) \]

\[ \frac{\partial C(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta} (H^T H\theta - 2H^T I + I^T I) \]

\[ = 2H^T H\theta - 2H^T I = 0 \]

where

\[ \frac{\partial C(\theta)}{\partial \theta} = \left( \frac{\partial L}{\partial \theta} \right) \left( \frac{\partial L}{\partial \theta} \right)^T = 2A_{\text{sym}} \frac{\partial L^2}{\partial \theta} = a \]

Solving the normal equations pseudo-inverse

\[ H^T H\theta - H^T I = 0 \Rightarrow \theta^* = \left( H^T H \right)^{-1} H^T I \]

Finding derivatives by fitting a local plane using LS

Consider a 3x3 neighborhood with gray levels

\[ \begin{bmatrix} I_1 & I_2 & I_3 \\ I_4 & I_5 & I_6 \\ I_7 & I_8 & I_9 \end{bmatrix} \]

On a unit step, centered, grid,

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

Solving the LS we get

\[ \theta^* = \left( H^T H \right)^{-1} H^T I \]

Which corresponds to the masks

(Prewitt edge operators)

\[ \begin{bmatrix} P_{\text{w}} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 0 & 1 \end{bmatrix} \]

Combined operators

Prewitt is a combined operator

Sobel is another example

\[ I_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad I_2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \]

- Note: Smoothing and differentiating are contrasting operations, but in 2D we can do both!

- Thresholding the Sobel operator smoothed gradient
- Thresholding produces thick edges

Smoothing and Deriving

- The "alternative" (to model fitting):
  reduce the noise by smoothing the image

- For linear smoothing, which one should come first?
  \[ \left[ I \ast \text{(smooth)} \right] \ast \text{(gradient)} \]
  \[ \left[ I \ast \text{(gradient)} \right] \ast \text{(smooth)} \]

  Does not matter! Why?

- Common:
  \[ I \ast \left[ \text{(gradient)} \ast \text{(smooth)} \right] \]

Gaussian smoothing - the common choice

- Good properties as smoother: symmetric, emphasize center, separable, infinitely deriveable, (semi-group, causality).

\[ G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \]

Gaussian smoothing

\[ I = I_{\text{original}} \ast G(x,y,\sigma) \]

\[ = I_{\text{original}} \ast \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} \]

Smoothing reduces both noise (+) and detail (-)

Micha Lindenbaum
**Gaussian derivatives responses**

1 pixel

3 pixels

7 pixels

**Combined Gaussian derivative**

\[ I \ast [ \text{gradient} \ast \text{smooth} ] \]

\[ G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \]

Board: 1D plot

**Gaussian derivative**

\[ \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \]

**Edge detection with Laplacian**

Basic idea: Finding the maximum of 1st derivative
\[ \sim \text{finding zero crossing of 2nd derivatives} \]

\[ \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \]

Elegant (one filter) but noisy

**Edge det. with Laplacian of Gaussian - LoG**

\[ \nabla^2 (I \ast G(x,y)) = I \ast \nabla^2 G(x,y) \]

Marr-Hildreth: smooth first.

\[ \begin{bmatrix}
1 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 1
\end{bmatrix} \times \begin{bmatrix}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 0
\end{bmatrix} = \begin{bmatrix}
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0
\end{bmatrix} \]

**Canny edge detector**

1. Apply Gaussian smoothing
2. Calculate Gradient magnitude\direction
3. Initialize: a point s.t. GradMag > Th_high
4. Thin: check neighbors in Grad direction and choose maximum
5. Find next: Find pixel neighbor in tangent direction
6. If (GradMag > Th_low) continue to Thin (4), Else continue to initialize (3)

Two thresholds: Hysteresis
Generates more continuous curves

**Canny edge detector**

1. Apply Gaussian smoothing
2. Calculate Gradient magnitude and direction
3. Initialize: Choose a point where GradMag > Th_high
4. Thin (Non maxima suppression):
check neighbors in Grad direction and choose maximum
Using spatial edge support

- In real images, edges are not isolated
- Smooth curves are common
- Isolated edgels are often less important.

- A detected edgel enhances other collinear edgels
- Grouping/saliency approaches use such enhancement

How much smoothing? (scale)

- Small scale: more detail, noise sensitive
- Coarse scale: less noisy, miss details, location inaccurate, smooth over other objects.

Edge detectors - summary

- Sobel (Prewitt, Roberts, ...) - old
- Laplacian - Elegant, perhaps HVS compatible
- Gaussian derivatives, Canny - commonly used
- Edison - uses gradient + a test on step edge profile

- Often, strong smooth edge is not an object boundary, and vice-versa.
- Color, texture edges, multi-scale detection
- Recent trend:
  - Learn from examples how an edge looks like.
  - Optimize performance of manually annotated DB of images