Lecture 2: Basic Properties of Images:

An image is an array of intensity values:
\[ I(x,y), i = 1 \text{ to } m, j = 1 \text{ to } n \]

What is this image?

Probably you say:
"It is an image of a dark box surrounded by a bright background."

Why?

A simple model of images (~1980's) says that an image is piecewise smooth, or weakly smooth.

Neighboring pixels have similar intensity values (e.g., 20 to 15),
but sometimes there is a big jump (e.g., 13 to 20).

This simple model says that images consist of regions. The intensity is roughly constant within each region. The intensity jumps between regions.

This can be used to segment images into regions.

This makes tasks like object detection easier — i.e., we can segment the image and then recognize objects.

\[
\begin{array}{c}
\text{Image} \\
\text{Segment} \\
\end{array} \rightarrow \begin{array}{c}
\text{Object} \\
\text{Recognition} \\
\end{array}
\]

Note: This model is too simple. Images are more complicated — e.g., they include texture regions.

In particular, we usually cannot segment objects without knowing what they are.
Are images piecewise smooth?

In one-dimension, calculate \( \frac{dI}{dx} \) or \( I_{x+1} - I_x \) on lattice.

Calculate the histogram for identity

\[
H(z) = \frac{1}{N} \sum_{i=1}^{N} I \left( I_{x+1} - I_x = z \right)
\]

The histogram has the standard form.

\[
H(z) \begin{cases} 
    1, & \text{if } a = b \\ 
    0, & \text{otherwise}
\end{cases}
\]

Note: The histogram is an image statistic - i.e., a function on the image.

It is the observed marginal distribution of the image derivative \( p(\Delta I) \). \( \Delta I = I_{x+1} - I_x \)

The histogram represents this distribution by bins

\[
p(\Delta I)
\]

Note: We could also represent \( p(\Delta I) \) by a parameterized probability distribution - but we need to know the form of the distribution (it is not a Gaussian).

The statistics of \( \frac{dI}{dx} \) suggest that images are piecewise smooth - at least as a very simple approximation.

We obtain similar statistics for two-dimension images - for \( \frac{\partial I}{\partial x} \) and \( \frac{\partial I}{\partial y} \). (Also for depth.)

In fact, we get similar histograms for other derivatives, for \( \frac{\partial^2 I}{\partial x^2}, \frac{\partial^2 I}{\partial y^2}, \frac{\partial^2 I}{\partial x \partial y} \). See Mumford & Lee. See Green.

(Note: higher order derivatives are non-local.)
Linear Filter Theory

\[ F * I_i = \sum_j S_{i-j} I_j \]

A derivative filter obeys \( \sum_j F_{i-j} = 1 \), for any \( i \).

\[ F_{i,j} = S_{i+1,j} - S_{i,j} \]

\[ F_{i,j} = S_{i+1,j} - 2S_{i,j} + S_{i-1,j} \]

\[ F_{i,j} = 3I_{i+1} - 2I_i + 3I_{i-1} \]

2nd order derivative.

\[ F * I(x) = \int F(x-y) I(y) \, dy \]

Example: Gaussian smoothing: \( G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \)

This blurs the image and smooths out small structures in the image.

Often combine differentiation and smoothing, \( \rightarrow \) e.g., smooth the image to eliminate small image structures and then differentiate.

Increasing \( \sigma \) in \( G_\sigma \) smooths the image more and removes more image structure.

\[ \frac{dG_{\sigma}}{dx} * I(x) = \int \frac{d}{dx} G_\sigma(x-y) I(y) \, dy \]

Continuous Filter \( \rightarrow \) Discrete Filter

\[ \frac{d}{dx} I(x) \rightarrow I_{i+1} - I_i \]

Note: many ways to discrete derivatives, some give better approximations.
Filter Banks

Sets of Filters:
- Derivatives, Smoothing, Derivatives and Smoothing
- Gabor filters

\[ g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{x^2}{\sigma^2}} \]

Sinusoidal and Gaussian

Cosine Gabors:

Gabor's detect local frequency structure in images

Color Images:
\[ \langle R_{ij}, B_{ij}, G_{ij} \rangle \]
\[ \text{red, blue, green} \]
\[ i, j \in \text{pixel locations} \]

Normalized Color:
\[ \frac{R_{ij}}{R_{ij} + B_{ij} + G_{ij}} \]
Statistical Edge Detection

Edge Detection - Canny Model

Step Edge Model:

\[ \text{Step Edge} + \text{Gaussian Noise} = \text{Noisy Edge} \]

Solution:
- Apply a smoothed derivative filter.
- Threshold response
- Non-Maximal Suppression

E.G.
One-Dimensional
\[ \frac{dG}{dx} * I_o \geq T, \text{ Threshold } T. \]

\[ \text{Gaussian if multiple responses above threshold} \]
\[ I(x) \]
\[ \frac{d(G*I(x))}{dx} \]
\[ T \]
\[ \text{Select maximum response and remove the rest.} \]

In Two-Dimensions
Take derivatives in many directions - e.g.
Select maximum response.

Note: this method assumes a model of an edge.
It does not try to learn an edge model from data.

Statistical Edge Detection:

Get dataset of images \( \{ I^\mu(x) : \mu \in \Lambda \} \)
Pixels are labeled \( \omega^\mu(x) = 1, \text{ pixel } x \in \mu \text{ image is an edge} \)
\( \omega^\mu(x) = 0 \) is not an edge.

Learn conditional distribution:
\[ P( F^\mu | I^\mu(x) | \omega(x) = 1) \]
\[ P( F^\mu | I^\mu(x) | \omega(x) = 0) \]
\[ F = (F_1, \ldots, F_m) \] derivative filters
Example: Filter \( |\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \)

represent the distributions by histograms.

To detect edges: Bayes decision
\[
P(\omega | |\nabla I|) = \frac{P(\nabla I | \omega) P(\omega)}{P(\nabla I)}
\]

Label a pixel as edge \( \omega = 1 \), \( \frac{P(\omega = 1 | \nabla I)}{P(\omega = 0 | \nabla I)} \), otherwise 0.

This requires knowing \( P(\omega) \).

Estimate from data \( P(\omega = 1) \approx 0.03 \), \( P(\omega = 0) \approx 0.97 \).

Because \( \log \frac{P(\nabla I | \omega = 1)}{P(\nabla I | \omega = 0)} \) is monotonic in \( |\nabla I| \),
the Bayes decision is equivalent to thresholding \( |\nabla I| \),
which is similar to Canny.

But, suppose we apply several different filters,
\[
P(\nabla I, \nabla 6_{-1} I, \nabla 6_{-2} I | \omega).
\]

Learn these joint distributions from training examples.

Then apply Bayes.

This does much better than Canny when evaluated on difficult datasets.

Statistical edge detection can combine edge cues from multiple filters in an optimal manner.
Statistical edge detection can combine information from multiple scales, or from multiple orientations.

Problem: If we represent $P(E|w)$ by histograms, then we can need $O(k^m)$ data, where $k$ is no. of histogram bins and $m$ is the number of filters. Requires too much data, and becomes impractical for large $m$. ($m \gg 10$).

Note: less data is needed if we know a parameterized form for the distributions (typically polynomial no. of parameters). But we do not know the parameterized form of the distribution.

Learning: Learn on Training Dataset, evaluate on Test Dataset, Cross-Validation.

Bayes is a special case of Bayes Decision Theory. This includes a loss function so that we can penalize the expected loss.

Note: that the statistical approach described above is a generative model on filters response, but is not a generative model on images.

Alternative related approach (Martin, Fowlkes, Malik)
Extract features $\Phi_1(I(x)), \Phi_2(I(x)), \ldots, \Phi_m(I(x))$
Learn distribution $P(w | \Phi_1, \Phi_m) = \frac{\exp \lambda \Phi(I(x))}{\exp \lambda \Phi(I(x)) + \sum \exp \lambda \Phi(I(x))}$

with $w \in \{0,1\}$
Regression model - learn parameters $\lambda_i$. 

\[ e^{\lambda \Phi(I(x))} \]