Image Differentiation

COMPSCI 527 — Computer Vision
Outline

1 The Meaning of Image Differentiation
2 A Conceptual Pipeline
3 Implementation
4 The Derivatives of a 2D Gaussian
5 The Image Gradient
What Does Differentiating an Image Mean?

Values

Derivatives in $x$
What Does Differentiating an Image Mean?

Can we reconstruct the black curve?
Cameras

Can I recover $C(x,y)$ from $I(z,i,y)$?
Somehow reconstruct the continuous sensor irradiance $C$ from the discrete image array $I$

Differentiate $C$ to obtain $D$

Sample the derivatives $D$ back to the pixel grid

Each would be hard to implement

Surprisingly, the cascade turns out to be easy!
From Discrete Array to Sensor Irradiance

What would the transformation from $I$ to $C$ look like formally, if we could find one? Example: Linear interpolation

$$C(x) = \sum_{j=-\infty}^{\infty} I(j') P(x-j)$$

$$P(j) = \delta(j)$$
Linear Interpolation as a Hybrid Convolution

\[ C(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} l(i, j)P(x - j, y - i) \]
Gaussian Instead of Triangle

- Noise $\Rightarrow$: fit rather than interpolating
- Noise $\Rightarrow$: filter with a truncated Gaussian
- $P(x, y) = G(x, y) \propto e^{-\frac{1}{2} \frac{x^2 + y^2}{\sigma^2}}$

\[
C(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) G(x-j, y-i)
\]
Differentiating

\[ I(r, c) \rightarrow C(x, y) \rightarrow \frac{\partial}{\partial x} C(x, y) \rightarrow D(x, y) \rightarrow I_c(r, c) \]

\[
C(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) G(x - j, y - i)
\]

(still don’t know how to do this, just plow ahead)

\[
D(x, y) = \frac{\partial C(x, y)}{\partial x} = \frac{\partial}{\partial x} \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) G(x - j, y - i)
\]

\[
D(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) G_x(x - j, y - i)
\]

• We transferred the differentiation to \( G \),
and we know how to do that!
(still don’t know how to implement a hybrid convolution)
Sampling

\[ D(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) G_x(x - j, y - i) \]

- We are interested in the values of \( D(x, y) \) on the integer grid: \( x \rightarrow c \) and \( y \rightarrow r \)

\[ I_c(r, c) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) G(c - j, r - i) \]

Wait! This is a standard, discrete convolution
We know how to do that!

To differentiate an image array, convolve it (discretely) with the (sampled, truncated) derivative of a Gaussian
The Derivatives of a 2D Gaussian

- The Gaussian function is separable:
  \[ G(x, y) \propto e^{-\frac{1}{2} \frac{x^2 + y^2}{\sigma^2}} = g(x) g(y) \]
  where
  \[ g(x) = e^{-\frac{1}{2} \frac{x^2}{\sigma^2}} \]
  \[ G_x(x, y) = \frac{\partial G}{\partial x} = \frac{\partial g}{\partial x} g(y) = d(x) g(y) \]
  \[ d(x) = \frac{dg}{dx} = -\frac{x}{\sigma^2} g(x) \]
- Similarly, \( G_y(x, y) = g(x) d(y) \)
- Differentiate (smoothly) in one direction, smooth in the other
- \( G_x(x, y) \) and \( G_y(x, y) \) are separable as well
The Derivatives of a 2D Gaussian

\[ G_x(x, y) = d(x)g(y) \quad \text{and} \quad G_y(x, y) = g(x)d(y) \]
Normalization

- Can normalize $d(c)$ and $g(r)$ separately
- For smoothing, constants should not change:
- We want $k \star g = k$ (we saw this before)
- For differentiation, a unit ramp should not change:
  $u(r, c) = c$ is a ramp
- We want $u \star d = u$ (see notes for math)
The Image Gradient

- Image gradient: \( \nabla I(r, c) = \frac{\partial I}{\partial x} = g(r, c) = \begin{bmatrix} l_x(r, c) \\ l_y(r, c) \end{bmatrix} \)
- View 1: Two scalar images \( l_x(r, c) \), \( l_y(r, c) \)
The Image Gradient

- View 2: One vector image $g(r, c)$

- We can now measure changes of image brightness
- *Edges* are of particular interest