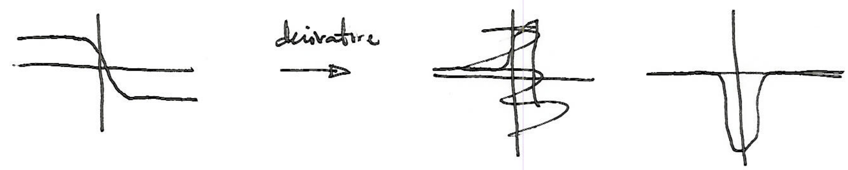
Edge Operators

edge − ~~de~~ large discrete change in image intensities.

→ how do we measure changes in a quantity? derivative!

want to look at differentiated signal

at edge:



⇒ want to find large derivatives (positive or negative).

discretization of signal:

$$f'(t) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$

in vector form,

$$\frac{\partial f}{\partial \underline{x}} \cdot \underline{h} = \lim_{\epsilon \to 0} \frac{f(\underline{x}+\epsilon \underline{h}) - f(x)}{\epsilon}$$ directional derivative

if $\underline{h} = (1,0)$ and $\underline{x} = (x,y)$, then we get

$$\frac{\partial f}{\partial x} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon, y) - f(x,y)}{\epsilon}$$

if $\underline{h} = (0,1)$ and $\underline{x} = (x,y)$, then

$$\frac{\partial f}{\partial y} = \lim_{\epsilon \to 0} \frac{f(x, y+\epsilon) - f(x,y)}{\epsilon}$$

but, image is discrete. What do we do?

Discretize derivative operator!

- This is not unique.

one way is to just use $\epsilon = \Delta x$,

$$\frac{\partial f}{\partial x} \approx \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x}$$

$$\frac{\partial f}{\partial y} \approx \frac{f(x, y+\Delta y) - f(x,y)}{\Delta y}$$

where Δx and Δy are grid spacings.

e.g. if dealing w/ image $\Delta x, \Delta y = 1$ pixel. $\Rightarrow$ $\partial f/\partial x \approx f(x+1, y) - f(x,y)$

if signal is 2D spatial and discretization is interval $[0,1]$ into 20 bins, then $\Delta x = \frac{1}{20}$, $\Delta y = \frac{1}{20}$.

~~So, for images~~

Alternatively, we can choose: $$f'(t) = \lim_{\epsilon \to 0} \frac{f(t) - f(t-\epsilon)}{\epsilon}$$

$$\frac{\partial f}{\partial t} \approx \frac{f(t) - f(t-\Delta t)}{\Delta t}$$

or, can choose: $$f'(t) = \lim_{\epsilon \to 0} \frac{f(t+\frac{1}{2}\epsilon) - f(t-\frac{1}{2}\epsilon)}{\epsilon}$$

$$\frac{\partial f}{\partial t} \approx \frac{f(t+\Delta t) - f(t-\Delta t)}{2\Delta t}$$

In images w/ $\Delta x = \Delta y = 1$, we get

$$\frac{\partial f}{\partial x} \approx \Delta_x^+ f = f(x+1,y) - f(x,y) \qquad \text{forward Euler}$$

$$\frac{\partial f}{\partial x} \approx \Delta_x^- f = f(x,y) - f(x-1,y) \qquad \text{backward Euler}$$

$$\frac{\partial f}{\partial x} \approx \Delta_x^c f = \frac{f(x+1,y) - f(x-1,y)}{2} \qquad \text{central differences}$$

· same holds for $\frac{\partial f}{\partial y}$.

Edge operators as convolution kernels:

1) Euler derivatives

$$\begin{bmatrix} -1 & 1 \end{bmatrix}, \quad \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

* using MATLAB convention for y-coordinate (increases downwards)

2) Roberts cross operator

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

- (1) rotated by 45°.

3) Prewitt edge detectors

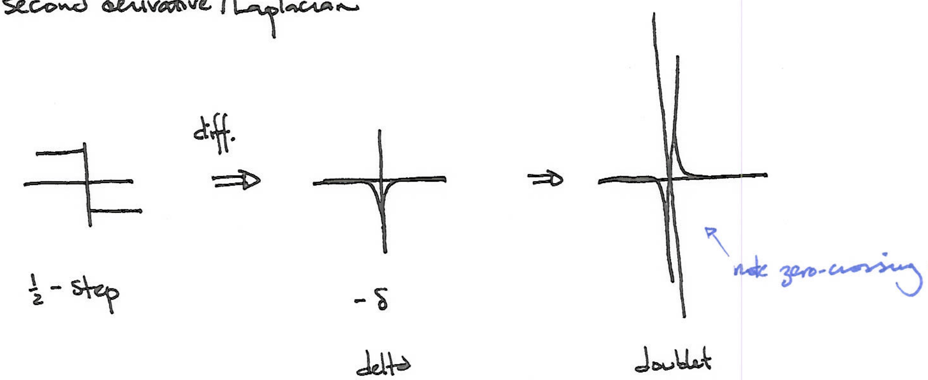
$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

4) Sobel edge detectors

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Second-Derivative Operators

Instead of trying to find peaks of gradient / derivative, why not find zero-crossings of second derivative / Laplacian



⇒

need $\frac{\partial^2 f}{\partial x^2}$, $\frac{\partial^2 f}{\partial y^2}$

$$\frac{\partial^2 f}{\partial x^2} \approx \Delta_x^+ \Delta_x^+ f = \left(\Delta_x^+ f(x+1,y) - \Delta_x^+ f(x,y) \right)$$

$$= f(x+2,y) - f(x+1,y) - f(x+1,y) + f(x,y)$$

$$= f(x+2,y) - 2f(x+1,y) + f(x,y)$$

⇒ shift by one

$$\frac{\partial^2 f}{\partial x^2} \approx \Delta_{x,x} f = f(x+1,y) - 2f(x,y) + f(x-1,y)$$

$$\frac{\partial^2 f}{\partial y^2} \approx \Delta_{y,y} f = \cancel{f(x, y+1)} - \cancel{2f(x,y)} + \cancel{f(x,y-1)}$$

$$= f(x,y+1) - 2f(x,y) + f(x,y-1)$$

⇒

as a convolution kernel

$$\Delta_{x,x} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

$$\Delta_{y,y} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

Example w/ image:

GOAL: Want to derive an energy which cleans up noise.

Q: How do we quantify noise to get an energy?

A: Want to minimize image variation/fluctuation.

$\Rightarrow$ minimize gradient variations/fluctuations

① define an energy to minimize:

$$\mathcal{E} \equiv \iint \left[\left(\partial I/\partial x\right)^2 + \left(\partial I/\partial y\right)^2 \right] dx\,dy$$

$$= \iint \left[I_x^2 + I_y^2 \right] dx\,dy$$

now, to figure out equations to minimize $\mathcal{E}$

② compute the first variation $\frac{\delta \mathcal{E}}{\delta I}$

$$\frac{\partial}{\partial \epsilon}\Big|_{\epsilon=0} \mathcal{E}(I+\epsilon\delta I) = \frac{\partial}{\partial \epsilon}\Big|_{\epsilon=0} \iint \left[(I_x+\epsilon\delta I_x)^2 + (I_y+\epsilon\delta I_y)^2 \right] dx\,dy$$

$$= \iint \left[2(I_x+\epsilon\delta I_x)\,\delta I_x\Big|_{\epsilon=0} + 2(I_y+\epsilon\delta I_y)\,\delta I_y\Big|_{\epsilon=0} \right] dx\,dy$$

variations vanish at boundary as needed.

$$= \iint_D \left[2 I_x \delta I_x \cancel{+ 2 I_y \delta I_y} \right] dx\,dy$$

$\Rightarrow$ int. by parts.

$$= \cancel{2 I_x \delta I\Big|_{\partial D}} + \cancel{2 I_y \delta I\Big|_{\partial D}}$$

$$- 2\iint_D \left[I_{xx}\,\delta I + I_{yy}\,\delta I \right] dx\,dy$$

$$= -2 \iint_D \left[I_{xx} + I_{yy} \right] \delta I \, dx\,dy$$

$\Rightarrow$

$$\frac{\delta}{\delta\epsilon}\Big|_{\epsilon=0} \mathcal{E}(I+\epsilon\,\delta I) = -2\iint_D \Delta I \cdot \delta I \, dx\,dy$$

- if I is already minimal in the sense of $\mathcal{E}$, then ΔI will vanish like in particle case

  if it isn't, then ΔI will be non-zero somewhere.

if we ~~are~~ are not at the minima, then there exists an ϵ and δI such that

$$\mathcal{E}(I+\epsilon\,\delta I) < \mathcal{E}(I)$$

what should δI be?

well, $\frac{\delta}{\delta\epsilon}\Big|_{\epsilon=0} \mathcal{E}(I+\epsilon\,\delta I)$ tells me how $\mathcal{E}(I)$ changes as I vary I by δI

So, if I want a negative derivative, what should δI be?

$$\delta I = \Delta I$$

$\Rightarrow$

$$I(k+1) = I(k) + \epsilon\,\delta I$$
$$= I(k) + \epsilon\,\Delta I$$

- ϵ should be chosen to be small, so update step doesn't overshoot.

- ~~for numerical stability, need $\epsilon \le \frac{1}{2}$.~~

  ~~(try using $\epsilon = 2$ and see what happens!)~~

If we rearrange terms, we get

$$\underbrace{\frac{I(k+1) - I(k)}{\epsilon}}_{\text{Euler derivative!}} = \Delta I$$

$\Rightarrow$

$$\frac{\partial I}{\partial \sigma} = \Delta I$$ heat equation / isotropic diffusion

- see this in thermodynamics; think of pixels as particles w/ thermal energy. want to minimize thermal gradients.

  $\Rightarrow$ if update to much, image is destroyed.

in MATLAB can try:

```
I = I + dt · del2(I)
```

run for desired # iterations and dt.

or may need to discretize equations

$$I(x,y) = I(x,y) + dt \cdot \left(I(x+1,y) + I(x,y+1) + I(x-1,y) + I(x,y-1) - 4I(x,y) \right)$$

↑ this is just one method to discretize equations.
numerical stability requires that $dt \le \frac{1}{2}$. (try $dt = 2$ and see what happens!)

- run for some finite # iterations and small enough dt and image gets smoothed.
- this is equivalent to Gaussian smoothing, where amount of integration is directly related to the standard deviation of the Gaussian filter.