Consider responses to the edge:

<table>
<thead>
<tr>
<th></th>
<th>Prewitt</th>
<th>Kirsch</th>
<th>Robinson-3</th>
<th>Robinson-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>-h</td>
<td>-9h</td>
<td>-2h</td>
<td>-3h</td>
</tr>
<tr>
<td>NE</td>
<td>-3h</td>
<td>-9h</td>
<td>-3h</td>
<td>-4h</td>
</tr>
<tr>
<td>N</td>
<td>-h</td>
<td>-9h</td>
<td>-2h</td>
<td>-3h</td>
</tr>
<tr>
<td>NW</td>
<td>+h</td>
<td>-h</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>W</td>
<td>+3h</td>
<td>7h</td>
<td>2h</td>
<td>3h</td>
</tr>
<tr>
<td>SW</td>
<td>+3h</td>
<td>15h</td>
<td>3h</td>
<td>4h</td>
</tr>
<tr>
<td>S</td>
<td>+3h</td>
<td>7h</td>
<td>2h</td>
<td>3h</td>
</tr>
<tr>
<td>SE</td>
<td>+h</td>
<td>-h</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

scale factor: \[ \frac{1}{5} \quad \frac{1}{15} \quad \frac{1}{3} \quad \frac{1}{4} \]

\[
\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & h & h \\ 0 & 0 & h \end{bmatrix} \rightarrow -\frac{9h}{15}
\]

\[
\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \rightarrow -\frac{9h}{15}
\]

\[
\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \rightarrow \frac{15h}{15} = h \quad \text{SW edge}
\]

obtain orientation directly

\[ \Rightarrow \text{largest magnitude} \]

With directional derivatives (compass gradients) we use many filters (usually 8)

\[ G(j,k) = \text{MAX} \{ |G_1(j,k)|, |G_2(j,k)|, \ldots, |G_8(j,k)| \} \]
Edge Detection 2

Threshold Selection

Noise-free images: threshold selected such that all amplitude discontinuities above certain amount are detected as edges.

Noisy images: Trade-off between missing valid edges and creating noise-induced false edges

- Controlled lighting conditions

Known: $P(\text{edge}) \approx 0.05 \quad P(\text{non-edge}) = 0.95$

Take lots of sample measurements, get:

$p(G|\text{NoEdge}) \quad p(G|\text{edge})$
Everything above threshold $t$ gets called an edge:

$$p(\text{pixel is called an edge}) = \int_t^\infty p(G) \, dG$$

$$p(\text{called an edge | is an edge}) = \int_t^\infty p(G | \text{edge}) \, dG = P_p$$

$$p(\text{correct detection}) = \int_t^\infty p(G | \text{edge}) \, dG = P_p$$

$$p(\text{false detection}) = \int_t^\infty p(G | \text{no-edge}) \, dG = P_f$$

Now, take a new image
At a given pixel, compute $G$
Want to assign it to most likely class

Call it an edge if

$$p(\text{edge} | G) \geq p(\text{no-edge} | G)$$

Bayes theorem says:

$$p(\text{edge} | G) = \frac{p(G | \text{edge}) \, p(\text{edge})}{p(G)}$$

Here $p(G) = p(G | \text{edge}) \, p(\text{edge}) + p(G | \text{no-edge}) \, p(\text{no-edge})$

Substituting:

$$p(G | \text{edge}) \, p(\text{edge}) \geq p(G | \text{no-edge}) \, p(\text{no-edge})$$

$$\frac{p(G | \text{edge})}{p(G | \text{no-edge})} \geq \frac{p(\text{no-edge})}{p(\text{edge})}$$

Known as the Maximum Likelihood Ratio Test
The probability of edge misclassification error is

\[ P_E = (1 - P_D) p(\text{edge}) + P_F p(\text{no-edge}) \]

The max likelihood ratio decision rule minimizes this probability.

This rule considers the 2 types of errors to be equally important (false alarms & false dismissals). Suppose they’re not?

Receiver Operating Characteristic (ROC) curves

Number of pixels:
- Total: 10,000
- True Binary Edge Map: 1000 true edge pixels
- True non-edge pixels: 9000
- Candidate Binary Edge Map: Found 1200 points
  - 800 correct
  - 400 not

True Positive Rate (TPR)

\[ TPR = \Pr(\text{called edge} \mid \text{is edge}) = \frac{\Pr(\text{called edge and is edge})}{\Pr(\text{edge})} \]

\[ \Pr(\text{called edge and is edge}) = \frac{\# \text{ candidate edge pix which are true edge}}{\text{total } \# \text{ pixels in image}} \]

\[ \Pr(\text{edge}) = \frac{\# \text{ true edge pixels}}{\text{total } \# \text{ pixels in image}} = \frac{1000}{10,000} \]

\[ \Rightarrow TPR = \frac{\# \text{ candidate edge pixels which are true edge pix}}{\# \text{ true edge pixels}} \]

\[ = \frac{800}{1000} \]
**TPR** is the same thing as **P₀** defined earlier

\[ P₀ = \int_{t}^{\infty} \rho (G \mid \text{edge}) \, dG \]

Similarly, False Positive Rate (FPR) is **Pₙ**

\[ \text{FPR} = \frac{\Pr (\text{called edge} \mid \text{not an edge})}{\Pr (\text{called an edge and not an edge})} \]

\[ = \frac{\text{# candidate edge pixels which are NOT true edge pixels}}{\text{total # true non-edge pixels}} \]

\[ = \frac{400}{9000} \]

We can plot TPR vs. FPR:

- **TPR**
- **FPR**

Suppose \( t \to \infty \)

- Nothing gets declared an edge pixel
- \# candidate edge pixels \( \to 0 \)
- \( TPR \to 0 \)
- \( FPR \to 0 \)

Suppose \( t \to 0 \)

- All pixels declared edge pixels
- \# correct candidate edge pix \( \to \) \# true edge pix \( \Rightarrow \) \( TPR \to 1 \)
- \# incorrect " " " \( \to \) \# true non-edge pix \( \Rightarrow \) \( FPR \to 1 \)
Suppose we have:

- Curves don't cross.
  - For a fixed acceptable FPR, can get higher (better) TPR using Sobel.
  - For a fixed acceptable TPR, can get lower (better) FPR using Sobel.

→ Sobel is better, regardless of threshold.

Consider instead: curves cross.

Medical imaging — want high TPR, highlight everything for the radiologist.

Want to examine very few pixels (only a few widgets, only need a few starting seeds to find edges).

Curves cross, can't say one is universally better. Depends where you need to operate.
Second derivative operators:

\[ \int f(x) \quad \text{In 2 dimensions:} \]

\[ \nabla \int f' \quad \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2} \quad \text{isotropic} \]

\[ f'' \quad \text{zero-crossing} \quad \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad \text{isotropic} \]

\[ \therefore \text{Laplacian operator} \]

Discrete approximation?

\[
\begin{array}{c|c|c|c|c}
 & D & C & B & A \\
\hline
k-2 & k-1 & k & k+1 \\
\hline
\end{array}
\]

first difference \( (k) = F(k) - F(k-1) \)

\[ = B - C \]

What happens if we define 2nd diff as being the 1st diff of 1st diff?

\(2\text{nd diff} \ (k) = \frac{1}{2}\text{nd diff} \ (k) - \frac{1}{2}\text{nd diff} \ (k-1)\)

\[ = F(k) - F(k-1) - \left[ F(k-1) - F(k-2) \right] \]

\[ = F(k) - F(k-2) - 2F(k-1) \]

\[ = B + D - 2C \]

This is centered on C! Wrong place!

So we define:

\[2\text{nd diff} \ (k) = \frac{1}{2}\text{nd diff} \ (k+1) - \frac{1}{2}\text{nd diff} \ (k)\]

\[ = F(k+1) - F(k) - \left[ F(k) - F(k-1) \right] \]

\[ = F(k+1) + F(k-1) - 2F(k) \]

\[ = A + C - 2B \]
filter \( [1 \ -2 \ 1] \) or \( [-1 \ 2 \ -1] \)

2nd derivative operators

In 2 dimensions:

\[
H = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}
\]

digital Laplacian

Often normalized to provide unit gain average of positive and negative

\[
H = \frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \leftarrow \text{zero response to constant value and to ramp}
\]

This array is 8-neighbor Laplacian:

\[
H_2 = \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \leftarrow \text{not separable}
\]

A separable 8-neighbor Laplacian is

\[
H_3 = \frac{1}{8} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} \quad \text{comes from} \quad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}
\]