Chapter 5

Changes in brightness in an image are important.

\[ \nabla I = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \]

\( [dx, dy] \) = gradient (im); 
\( \text{quiver} (dx, dy) \);
\( \text{mag} = \sqrt{dx^2 + dy^2} \);
\( \text{ori} = \text{atan2} (dy, dx) \)

Derivative of Gaussian filters

Can convolve derivative of Gaussian with image rather than smoothing derivative of image

\[ \frac{\partial (G * I)}{\partial x} = (\frac{\partial G}{\partial x}) * I \]

Sharp brightness changes lie on curves in image called edges; made of edge pts.

Gradient-based edge following
Alg. 5.1

* Find high mag gradient point
  e.g. [rows, cols] = find (mag > THRESH1);

* Keep track of visited pixels gradient magnitude image
* Pick a point
* Look in gradient direction (both sides)
  + Find max gradient

look along edge direction for neighboring edge point

grad line

push (set) point

* Expand chain at that point
  + Look along 1 to gradient (both ways)
  + Check for max (at that location)
    + Check max is above THRESH2
  + Push point onto stack

Corners:

\[
\text{Give } [x, y, \theta] \text{ pose}
\]

Find corner:

Find edge curves, then find corners where direction changes.
Harris corner detector

Moreave proposed:

\[ E(u,v) = \sum_{x,y} w(x,y) \left[ I(x+u,y+v) - I(x,y) \right]^2 \]

At every pixel; i.e., there's an \( E_{x_1c}(u,v) \)

Harris showed:

Taylor expansion of \( I(x+u,y+v) = I(x,y) + u f_x(x,y) + v f_y(x,y) \)

Substitute: (omit \( w \) for now)

\[ E(u,v) = \sum_{x,y} \left[ I(x,y) + u f_x(x,y) + v f_y(x,y) - I(x,y) \right]^2 \]

\[ = \sum_{x,y} \left[ u^2 f_x^2(x,y) + 2uv f_x f_y(x,y) + v^2 f_y^2(x,y) \right] \]

\[ = \sum_{x,y} \left[ \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \right] \]

\[ = \begin{bmatrix} u & v \end{bmatrix} \left( \sum_{x,y} \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \]

Let \( M = \sum_{x,y} w(x,y) \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix} \) for gradient:

\( \rightarrow \) just the covariance matrix.
So just get all \(dx, dy\) "pts" from window

\[
\mathbf{pts} = \begin{bmatrix}
dx_1 \\
dy_1 \\
dx_2 \\
dy_2 \\
\vdots \\
ds_m \\
dy_m
\end{bmatrix}
\]

Find covariance matrix:

\[
\mathbf{M} = \mathbf{pts}^T \times \mathbf{pts}^T
\]

\[
\mathbf{H} = \mathbf{M}
\]

\(\mathbf{H}\) from text

Corner response, rat each pixel \((r, c)\):

\[
\mathbf{R}(r, c) = \det(\mathbf{H}) - 0.05 \times \text{trace}(\mathbf{H})^2
\]

Matlab example

\texttt{AS div}

Section 5.3.2 Neighborhood
The weights \( W \) may all be 1 (above) or like a 2D Gaussian.

Laplacian: \((\nabla^2 f)(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\)

Smooth first (why?): Gaussian.

So laplacian of Gaussian of image

\( \log \)
Laplacian

Consider \( \nabla^2 f = \frac{d^2 f}{d x^2} \) template \([-1 1]\)

for \( \frac{df}{dx} \)

let \( \frac{df}{dx}(r, c) = f(r, c+1) - f(r, c) \)

\( \frac{df}{dx}(r, c-1) = f(r, c) - f(r, c-1) \)

So, \( \frac{df}{dx}(r, c-1) = (f(r, c+1) - f(r, c)) - (f(r, c) - f(r, c-1)) \)

\( = f(r, c+1) - 2f(r, c) + f(r, c-1) \)

Template: \([1 -2 1]\)

Likewise \( \frac{df}{dy}(r-1, c) = f(r-1, c) - 2f(r, c) + f(r+1, c) \)

Template: \([1 -2 1]\)

Can do both at once (add them)

\[
\begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]

or

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & -8 & 1 \\
1 & 1 & 1
\end{bmatrix}
\]

look at special ('Laplacian') \(0, 0, 0, 0, 0, 1\)

may use opposite sign
the Laplacian can be applied across several scales (spatial resolution).

Consider:

\[ \frac{\partial f}{\partial x} \]

\[ \frac{\partial^2 f}{\partial x^2} \]

(liner 108 in .m file)

Can use Laplacian across scale to find interesting points.

Figure 5.12 shows corner point and how:

\[ r(x, y) = \text{argmax} \ \nabla^2 I(x, y) \]

\[ \text{line 116 in A4.m} \]

Can find max \( r \) at every pixel, and then pick local maxima spatially.

\[ \text{line 17 in A6.m} \]
use a local spatial max (or maybe a threshold) to get spatial + scale maximum as interest points.

A6: Part I: implement Alg. 5.2

For scale \( k \)

Apply corner detector

Initialize list of patches

for each corner detected

\((x_c, y_c)\) is location of corner

\(r(x_c, y_c) = \arg \max_{r} \nabla^2 I(x_c, y_c)\)

\(H(\theta) = \text{orientation histogram}
\)

within radius \( r \) of \((x_c, y_c)\)

\(\theta_p = \arg \max_{\theta} H(\theta)\)

\([x_c, y_c, r, \theta_p]\) output

A6: Part II: implement Alg. 5.2 + 5.3

Assume fixed scale \( k \)

\((x_c, y_c, r)\) = position and scale local extrema

for each triple

\(H(\theta) = \text{histogram gradient orientation radius for}\)

\(\theta_p = \arg \max_{\theta} H(\theta)\)

\([x_c, y_c, r, \theta_p]\) output