Chapter 5

changes in brightness in an image are important

Image gradient

$$\nabla I = \begin{bmatrix} \partial I / \partial x \\ \partial I / \partial y \end{bmatrix}$$

[dx, dy] = gradient (im); quiver (dx, dy);
mag = sqrt (dx.^2 + dy.^2); ori = atan2 (dy, dx);

Derivative of Gaussian filters

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

$$\frac{\partial (G_\sigma ** I)}{\partial x} = \left(\frac{\partial G_\sigma}{\partial x} \right) ** I$$

Sharp brightness changes lie on curves in img called <u>edges</u> : made of edge pts

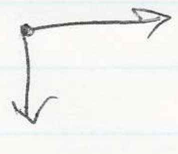
Gradient based edge following

Alg. 5.1

* find high mag gradient points
  e.g., [rows, cols] = find (mag > THRESH1);
  ↑ gradient magnitude image
* keep track of visited pixels
* pick a point
* look in gradient direction (both sides)
  + find max gradient

* pop point
* Expand chain at that point
  + look along ⊥ to gradient (both ways)
  + check for max (at that location) along its gradient direction
  + check max is above THRESH2
  + push point onto stack

Corners:



Give [x, y, θ] pose

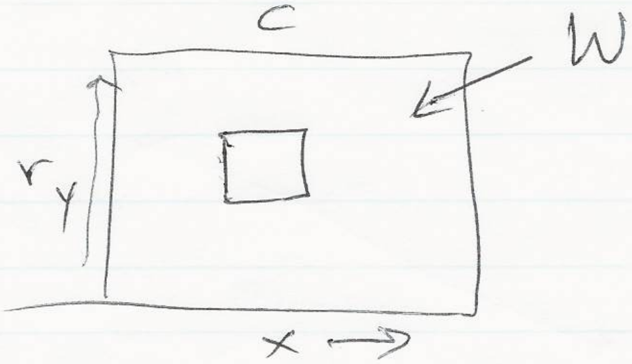
Find corner: Find edge curves, then find corners where direction changes

Harris corner detector

Moravec proposed:

$$E(u,v) = \sum_x \sum_y w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

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



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)

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

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

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

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

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: → just the covariance matrix

So just get all dx, dy "pts" from window

$$pts = \begin{bmatrix} dx_1 & dy_1 \\ dx_2 & dy_2 \\ \vdots & \\ dx_n & dy_n \end{bmatrix}$$

Find covariance matrix:

M = pts' * pt;

H = M (H from text)

corner response, R at each pixel (r, c):

R(r,c) = det (H) − 0.05 * trace (H)^2

<u>Matlab example</u> <u>A5 dir</u>

<u>Section 5.3.2</u> 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 $\underbrace{\text{Laplacian of Gaussian}}_{\text{log}}$ of image

Laplacian

Consider $\nabla^2 f = \dfrac{\partial^2 f}{\partial x^2}$ template $[-1 \ \ 1]$ for $\partial f / \partial x$

Let $\dfrac{\partial f}{\partial x}(r,c) = f(r,c+1) - f(r,c)$

$$\frac{\partial f}{\partial x}(r,c-1) = f(r,c) - f(r,c-1)$$

So, $$\frac{\partial}{\partial x}\left(\frac{\partial f(r,c-1)}{\partial x}\right) = \big(f(r,c+1) - f(r,c)\big) - \big(f(r,c) - f(r,c-1)\big)$$
$$= f(r,c+1) - 2f(r,c) + f(r,c-1)$$

template $[1 \ \ -2 \ \ 1]$

Likewise $$\frac{\partial}{\partial y}\left(\frac{\partial(r-1,c)}{\partial y}\right) = f(r-1,c) - 2f(r,c) + f(r+1,c)$$

template: $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

Can do both at once (add them)

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

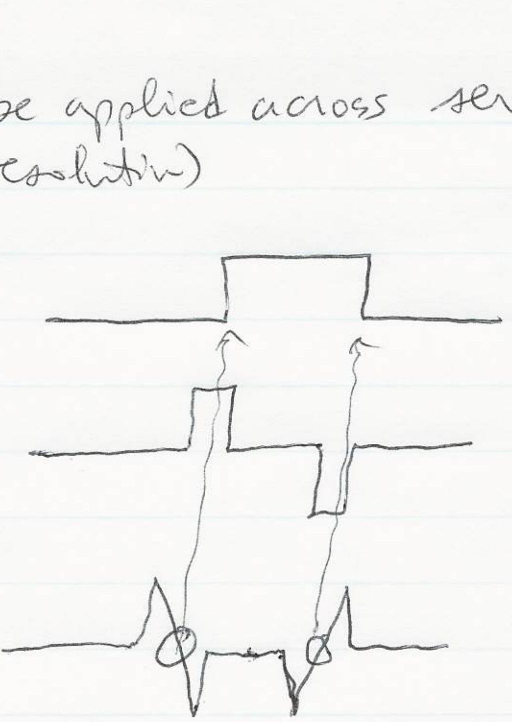
look at fspecial('laplacian', 0.000001);
may use opposite sign

the Laplacian can be applied across several scales (spatial resolution)

Consider f:

$\partial f / \partial x$

$\partial^2 f / \partial x$



zerocrossings

(line 108 in .m file)

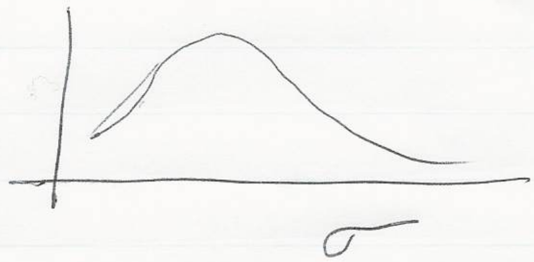
Can use Laplacian across scale to find interesting points:

Figure 5.12 shows corner point and how

$$r(x,y) = \underset{\sigma}{\operatorname{argmax}} \nabla^2_\sigma I(x,y)$$

looks

∇^2_σ



line 116 in A4.m

Can find max r at every pixel, + then pick local maxima spatially

line 13 in A6.m

use a local spatial max (+ maybe a threshold)
to get spatial + scale maxima as
interest points.

A6: Part I: implement Alg. 5.2

Fix scale k
Apply corner detector
initialize list of patches
for each corner detected
(x_c, y_c) is location of corner
$r(x_c, y_c) = \arg\max_\sigma \nabla_\sigma^2 I(x_c, y_c)$ — don't interpolate + find max; use fine σ spacing

$H(\theta)$ = orientation histogram within radius kr 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
$(x_c, y_c, r) \in$ position and scale local extrema
For each triple
$H(\theta) \leftarrow$ histogram gradient orientations radius kr
$\theta_p \leftarrow \arg\max_\theta H(\theta)$

$[x_c, y_c, r, \theta_p] \leftarrow$ output