Lecture 4:
Edge Based Vision

Dr Carole Twining

Thursday 18th March

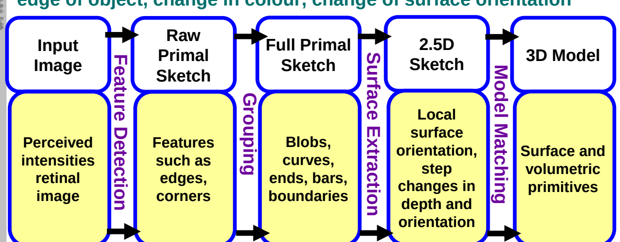
2:00pm – 2:50pm

Overview:

- Marr's Theory of Vision
  - why edges matter
- Edges and Derivatives
  - convolution and filters
- Edges and Scale
  - physical edges persist across scales
- Edge Detection
  - Problem with noise, and accurate edge location
- Edge growing
  - Thresholding with hysteresis
  - Edge relaxation
- Hough Transform
  - Finding lines

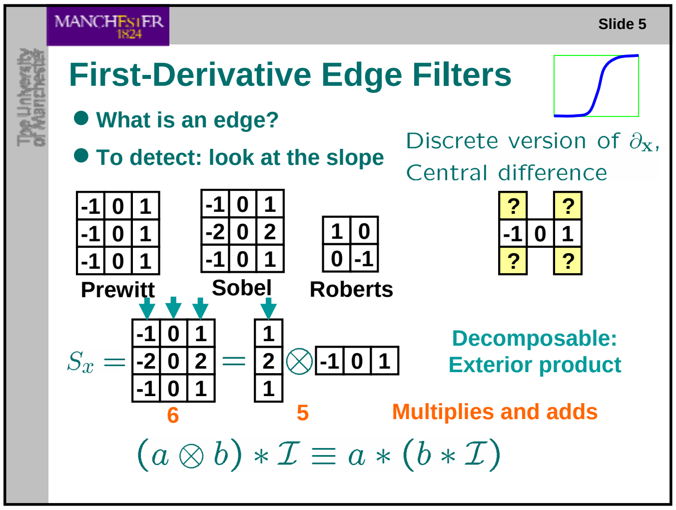
Marr's Theory of Vision:

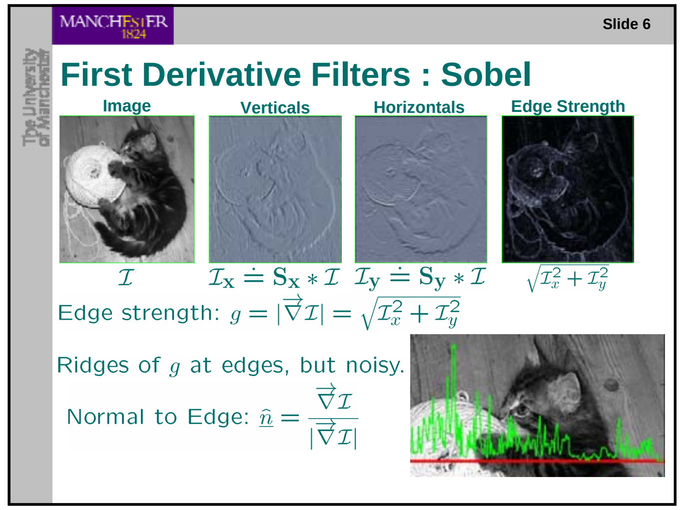
Edges in images correspond to physical events:
edge of object, change in colour, change of surface orientation

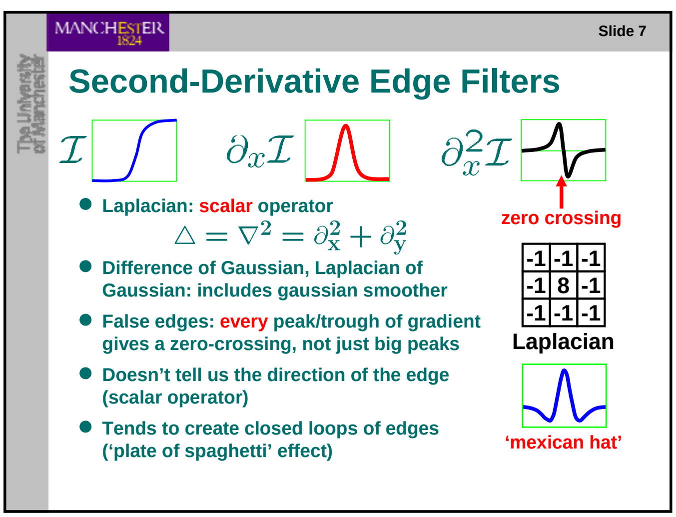


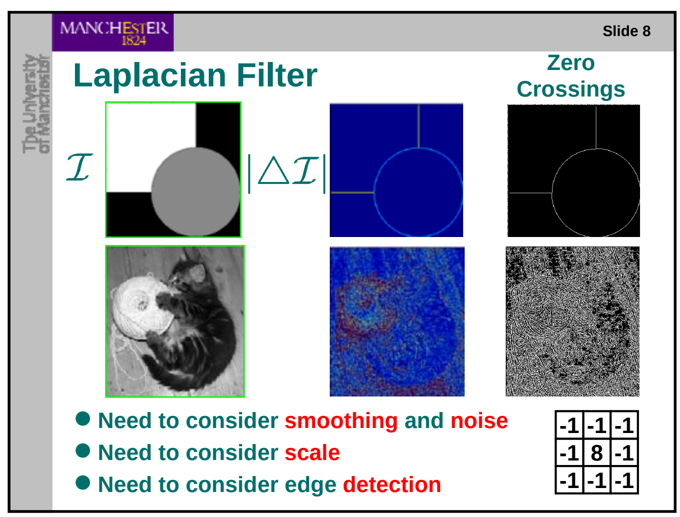
- Agrees with pre-conceptions as to how vision might work
- Not proven possible to build a reliable system in this way
- Still influential
- Pragmatic approach: what do we need to do a specific task?

Edges and Derivatives


MANCHESTER 1824
The University of Manchester
Slide 5
First-Derivative Edge Filters
What is an edge?
To detect: look at the slope
Discrete version of ∂_x, Central difference
Prewitt
Sobel
Roberts
$S_x =$
6
5
Decomposable: Exterior product
Multiplies and adds
$(a \otimes b) * \mathcal{I} \equiv a * (b * \mathcal{I})$

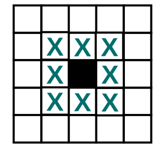

MANCHESTER 1824
The University of Manchester
Slide 6
First Derivative Filters : Sobel
Image
Verticals
Horizontals
Edge Strength
$\mathcal{I}$
$\mathcal{I}_x \doteq S_x * \mathcal{I}$
$\mathcal{I}_y \doteq S_y * \mathcal{I}$
$\sqrt{\mathcal{I}_x^2 + \mathcal{I}_y^2}$
Edge strength: $g = |\vec{\nabla}\mathcal{I}| = \sqrt{\mathcal{I}_x^2 + \mathcal{I}_y^2}$
Ridges of g at edges, but noisy.
Normal to Edge: $\hat{n} = \frac{\vec{\nabla}\mathcal{I}}{|\vec{\nabla}\mathcal{I}|}$


MANCHESTER 1824
The University of Manchester
Slide 7
Second-Derivative Edge Filters
$\mathcal{I}$
$\partial_x \mathcal{I}$
$\partial_x^2 \mathcal{I}$
zero crossing
Laplacian: scalar operator
$\triangle = \nabla^2 = \partial_x^2 + \partial_y^2$
Difference of Gaussian, Laplacian of Gaussian: includes gaussian smoother
False edges: every peak/trough of gradient gives a zero-crossing, not just big peaks
Doesn't tell us the direction of the edge (scalar operator)
Tends to create closed loops of edges ('plate of spaghetti' effect)
Laplacian
'mexican hat'

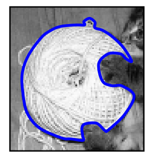

MANCHESTER 1824
The University of Manchester
Slide 8
Laplacian Filter
Zero Crossings
$\mathcal{I}$
$|\triangle \mathcal{I}|$
Need to consider smoothing and noise
Need to consider scale
Need to consider edge detection

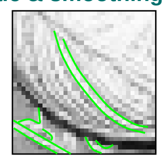
Edges and Scale

Edges and Scale



- Edge filters enhance noise
- What is a 'real' edge and what noise?
- Edges exist at many different scales
- What scales matter depends on application
- Sensible approach: use many different scales
  - Edges persist across scales, allows fusion across scales
- Gaussian gives scale & smoothing separable filter

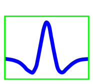




Edges and Scale

Marr-Hildreth:

- Convolve with gaussian $\mathcal{G}$
- Take Laplacian ∇^2 of result:
  - combine into single stage LoG
- Edges at zero-crossings
- Edges move with scale if curved
- No information on direction
- 'Plate of spaghetti' problem



'mexican hat'

Canny:

- Convolve with gaussian $\mathcal{G}$
- Take gradient $\vec{\nabla}$ of result

  $\vec{\nabla}(\mathcal{G} * \mathcal{I}),\ g = |\vec{\nabla}(\mathcal{G} * \mathcal{I})|$
- Find gradient direction:

  $\hat{\underline{n}} = \vec{\nabla}(\mathcal{G} * \mathcal{I})/g$
- Create gaussian-smoothed derivative tuned to this direction
- Take another derivative in that direction to find local maximum, zero-crossing
- Stable across scales

Marr-Hildreth vs Canny

- Both involve pre-smoothing with gaussian
- Both involve second-derivative BUT:

Marr-Hildreth:

- No information on direction
- By adding second-derivative in other direction, increases effect of noise

Canny:

- Create tuned derivative given estimated gradient direction
- Only compute second derivative in gradient direction
- Check that it really is local maximum of edge strength in that direction (see non-maximum suppression)

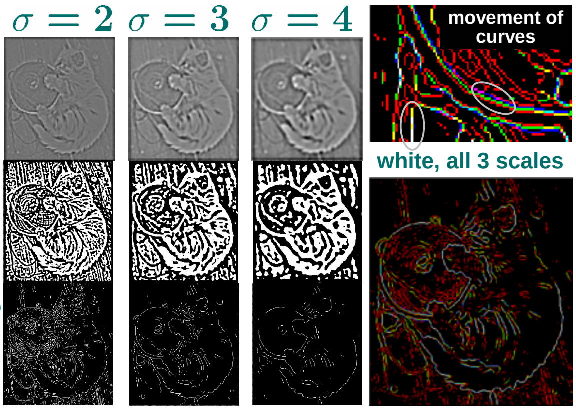
Marr-Hildreth Edge Detection

$\sigma = 2$ $\sigma = 3$ $\sigma = 4$

LoG

LoG>0

zero crossings



Marr-Hildreth Edge Detection

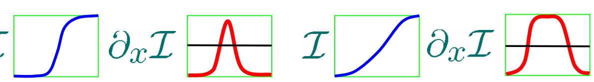
$\sigma = 10$

- Some edges not well localized
- ‘Plate of spaghetti’ effects

Edge Detection

Edge Detection: First Derivatives

$\mathcal{I}$ $\partial_x \mathcal{I}$ $\mathcal{I}$ $\partial_x \mathcal{I}$



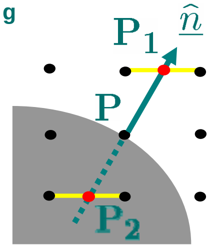
- Position of maximum can be difficult to locate:
  - second-derivative, zero crossing more precise
- Simple threshold:
  - thick edges, need to apply thinning
  - missed edges, streaking (see thresholding with hysteresis)

Edge Detection: Second Derivative

- Zero-crossing more precisely located than maximum
- Thresholding in Marr-Hildreth (LoG):
  - Doesn't use directional information
- 'Plate of spaghetti':
  - continuity =>
    closed loops or meets boundary
- Noise, false edges, double response
- Thinning, edge growing & edge relaxation
  - incorporate neighbourhood information

Non-Maximum Suppression

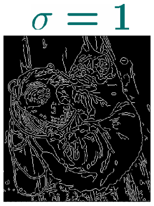
- Start from edge-strength signal g
- Locate possible edge point P
- Identify gradient direction $\widehat{n}$
- Interpolate g at P_1 and $\mathbf{P_2}$
- P is local maximum provided:
  g(P) > g$(\mathbf{P_1})$ & g(P) > g$(\mathbf{P_2})$
- Only accepts as edge if proper maximum, rejects if not
- In practise, only allow a set of discrete possible directions



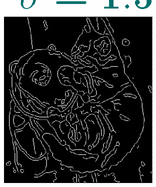
Object & pixel positions

Canny Edge Detector

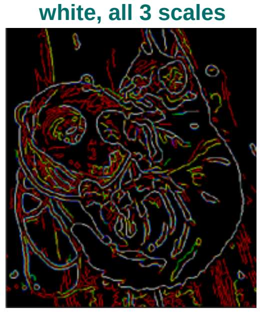
$\sigma = 1$



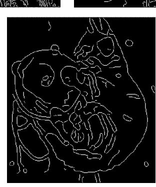
$\sigma = 1.5$



white, all 3 scales



$\sigma = 2$



Canny Edge Detector:

$\sigma = 10$

$\sigma = 20$

From Edge Pixels to Edges

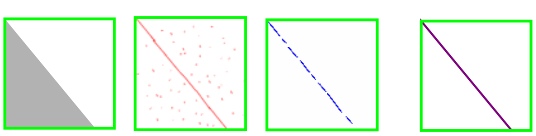
- Have candidate edge pixels
- Have information on edge direction and strength
- Want connected edges:

  Edge growing
- Going from individual edge pixels, to entire, connected edges – curves that are boundaries of objects

Edge Growing

Edge Thresholding with Hysteresis

- Edge strength image, two thresholds T_H & T_L
- Only edges have points g> T_H
- Edges have all points g> T_L
- Start at point g>T_H, and trace connected points with g>T_L



$\mathcal{I}$ $g > T_L$ $g > T_H$ $Result$

Edge Relaxation

- Use context to resolve ambiguity (as in segmentation)

$g(i)$: Edge strength at pixel i

$\underline{e}(i)$: Edge direction at pixel i

Normalise edge strengths $g(i) \Rightarrow P(\underline{e}, i) \leq 1$

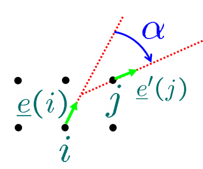
- Compatibility

Pixels i and j,

edge directions $\underline{e}$ and $\underline{e}'$:

$c_{i,j}(\underline{e}, \underline{e}') = 0$ **not neighbours**

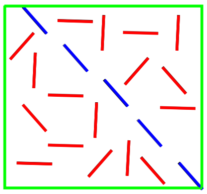
$c_{i,j}(\underline{e}, \underline{e}') = |\cos(\alpha)|$



- As before, update probabilities based on support

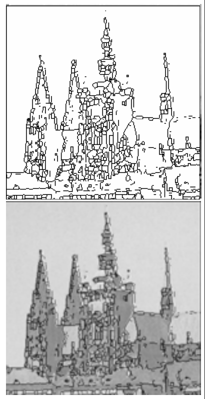
Slide 25

Edge Relaxation



weak and **strong** edges

- Many refinements and alternatives in the literature, but all applying same basic ideas



Hough Transform

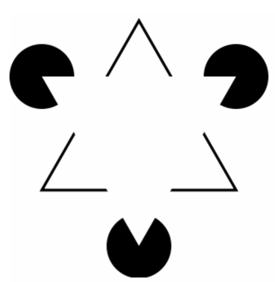
Slide 27

Hough Transform (1)

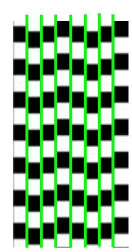
- Have some set of points, parts of edges etc
- Want to put them together into continuous lines
- Strategy:
  - Transform to parameter space
  - Let points vote for lines that could pass through them
  - Look for clusters
- Finding the right parameter space
- Can be extended if you can find such a space for shape of interest

Slide 28

Aside: Lines in human vision



See lines where we have only minimal information



Actually straight, but we don't see them as that!

Slide 29

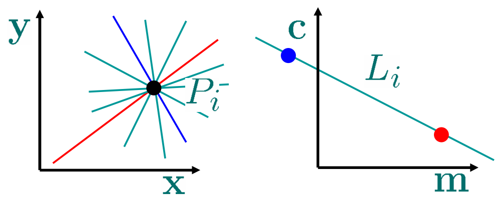
Hough Transform (2)

Set of points $\{P_i = (x_i, y_i)\}$ in image plane.

Any and all straight lines thro' P_i:

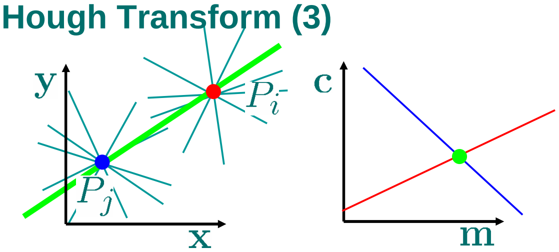
$y_i = mx_i + c$ $\Rightarrow$ $c = -x_i m + y_i$

L_i: line in (c, m) plane, intercept y_i, gradient $-x_i$



Slide 30

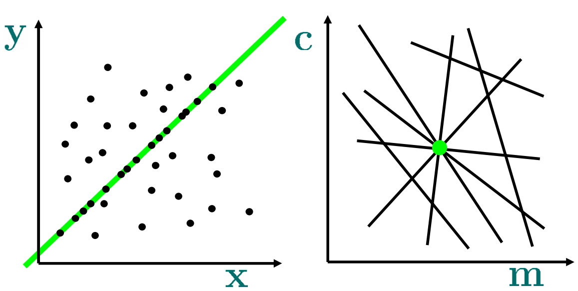
Hough Transform (3)



- Repeat for all points $\{P_i = (x_i, y_i)\}$ in image plane
- Look for points in (c,m) plane where lots of lines cross
- Lines which pass thro' lots of points in image plane

Slide 31

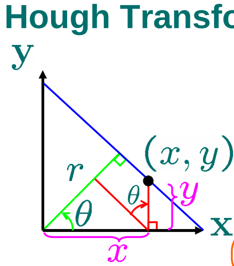
Hough Transform (4)



- Verticals, m is infinite! Need better parameter space

Slide 32

Hough Transform (5)



$$y = mx + c$$

$$(c, m) \Rightarrow (r, \theta)$$

$$r = x\cos\theta + y\sin\theta$$

- Single point $P_i = (x_i, y_i)$
- All possible θ : allowed values of r, sinusoid curve
- Extend to other than lines, generalised Hough transform