Edges

Edge detection: local operation to determine intensity changes

Good edge detector:
Understand the difference between:
1. Variance caused by image noise
2. Variance caused by textures
3. Variance caused by true edges

Edge detection
1. Differential methods: gradient, laplacian
2. Template methods: roberts, prewit, sobel, kirsch
3. Optimisation methods: canny

Edges and derivatives
1st derivative: [-1 1]

$$f'(x) = \lim_{h->0} \frac{f(x+h) - f(x)}{h} \approx f(x+1) - f(x) \quad (h=1)$$

mask: $[-1 \quad 1]$

Backward difference, forward difference, central difference

2st derivative: [1 -2 1]

$$f''(x) = \lim_{h->0} \frac{f'(x+h) - f'(x)}{h} \approx f'(x+1) - f'(x) =$$

$$f(x+2) - 2f(x+1) + f(x) \quad (h=1)$$

- This approximation is **centered** about $x + 1$; by replacing $x + 1$ by x we obtain:

$$f''(x) \approx f(x+1) - 2f(x) + f(x-1)$$

mask: $[1 \quad -2 \quad 1]$

1st derivative: edge is well detected, but location is imprecise,
- ramp edge yield a weak response
- Impulse response is a "whip"

S_3			12	12	12	12	15	18	21	24	24	24
S_3	$\otimes$	M	0	0	0	3	6	6	6	3	0	0

(c) S_3 is an upward ramp

2nd derivative: "double whip" amplifies and locates the step edge

Derivative based edge detectors are sensitive to noise

Gradient filters

- The magnitude of gradient provides information about the strength of the edge
- The direction of gradient is perpendicular to the direction of the edge

$$\nabla f(x,y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix}$$

$$\frac{\partial f}{\partial x} = \frac{f(x+h_x,y)-f(x,y)}{h_x} = f(x+1,y) - f(x,y), \quad (h_x = 1)$$
$$\frac{\partial f}{\partial y} = \frac{f(x,y+h_y)-f(x,y)}{h_y} = f(x,y+1) - f(x,y), \quad (h_y = 1)$$

Roberts: gradient across diagonals, fast, do not have a <u>clear center</u>

-1	0
0	1

0	-1
1	0

Prewitt: simpler to implement

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Sobel: a weight of 2 for smoothing, superior noise-suppression than prewitt

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Laplacian filters

Isotropic, very sensitive to noise

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

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

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

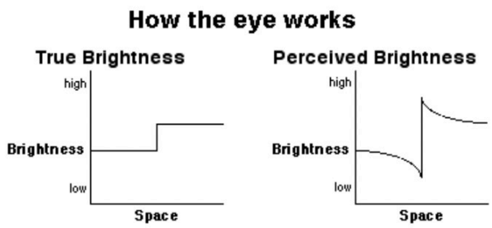
$$\nabla^2 f = [f(x+1),y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

0	1	0
1	-4	1
0	1	0

→ Low-pass filtering → $\nabla^2(.)$ → Zero crossing →

Properties of Laplacian
1. Isotropic
2. Cheaper to implement
3. Does not provide information about edge direction
4. Sensitive to noise

Unsharp masking
Mach bands — a phenomenon in human vision, the true brightness on the left, perceived on the right



Unsharp masking create artificial mach band

Unsharp masking (过程):
1. Blurring the original
2. Subtracting the blurred image from the original
3. Adding the mask to the original

Is — blurring, I-Is — details, I+(I-Is) — improve image details

Laplacian of Gaussian (LoG)
De-noise should be done first, gaussian smooth it
Convolution is associative —> the sequence of laplacian and Gaussian does not matter

The laplacian of Gaussian

$$\nabla^2 G(r) = \left[\frac{r^2 - s^2}{s^4} \right] e^{-\frac{r^2}{2s^2}}$$

Marr-Hildreth method: different variance of gaussian —> zero-crossing

Approximation of the LoG operator: The LoG = DoG — difference of gaussians
DoG: difference of two differently sized gaussians

LoG response — zero crossings

LoG response	Result
0	At a long distance from the edge
>0	One side of the edge
<0	Another side of the edge
0	On the edge itself

Canny edge detector
Canny — the first derivate of the Gaussian closely approximates the operator that optimizes the product of SNR and localization

Flow Chart
1. Smooth by gaussian convolution
2. Differential operators along x and y axis
3. Non-maximum suppression — find peaks in the image gradient
4. Hysteresis thresholding locates edge strings

Non-maxima suppression
Check if gradient magnitude at pixel location (i, j) is local maximum along gradient direction
作用: thinning —> narrowing the edge

Hysteresis thresholding: two threshold tl, th

$$\begin{array}{rll} \|\nabla f(x,y)\| \geq t_h & & \text{definitely an edge} \\ t_l \leq \|\nabla f(x,y)\| < t_h & & \text{maybe an edge, depends on context} \\ \|\nabla f(x,y)\| < t_l & & \text{definitely not an edge} \end{array}$$

- High threshold — start edge curves
- Low threshold — continue them

Multi-scale processing
Multi-scale processing: determine which structures (edges) are most significant by considering the range of scales over which they occur
Interesting scale — scales at which important structures are present

The choice of σ depends on desired behavior
- large σ detects large scale edges
- small σ detects fine features

Image segmentation approaches
• Thresholding
• Region based segmentation

Thresholding
• Global Thresholding
• Adaptive Thresholding — subdivision

Region based segmentation
• Region growing
• Region splitting

Region growing:
Start from some pixels (seeds) representing distinct image regions and to form them, until they cover the entire image. Stop until no more pixels satisfy the criteria for inclusion in that region.

Region splitting
1. First there is a large region (the entire image)
2. A predicate is used to determine if the region is uniform
3. If not, then the method requires the region be split into two regions
4. Then each of these two regions is independently tested the predicate
5. This procedure continues until all regions are uniform

Summary

Good edge detector: Variance created by noise, texture, true edge

1st derivative, 2rd derivative

Roberts, Prewitt, Sobel (slightly noise-suppression)

Laplacian:
1. Isotropic
2. Easy to implement
3. No information for edge direction
4. Sensitive to noise

Unsharp masking
Machband

1. Blurring image 2. Subtract the blurring image from the original image create this mask
3. Add the mask to the original image to improve details

LoG
0 — far distance from edge, on the edge itself
>0 <0 — edge (one side)
DoG = difference of gaussian

Canny edge detection
1. Apply the gaussian to smooth the image
2. Differential operators apply one and y axis
3. Non-maximum suppression
4. Hysteresis thresholding (high: create the edge, low: continue them)

Multi-scale processing
Interesting scale: the scale that offers most significant edges

Segmentation
- Thresholding: global, adaptive
- Region-based segmentation: region growing, region splitting

Region growing: start from several pixels (seeds), the choice of seeds will influence the result. Then grow from this seeds, after each grow, judge whether the growth of the seeds should be continue by criteria. It will stop until it covers the entire image.

Region splitting
1. First start from a large region (entire image)
2. A predicate is used to determine whether the region is uniformed
3. If not, then split the region into two new regions
4. Then each of these two regions should be tested by the predicate
5. Stop when all the regions are uniform

Practice

a) This question is about **Edge Detection**.

[12 marks]

i) Give the mask for the first-order derivative approximation of an image $f(x, y)$, $G_y = \frac{\partial}{\partial y} f(x, y)$.

(3 marks)

ii) For a piece of image [3,4,4,3,10,10,11,11,12], show the result of the convolution with the mask and indicate where the edges would be detected and why. Use replicate border extension for the convolution.

(5 marks)

iii) Given an input image, describe four steps to perform edge detection in the Canny edge detector.

(4 marks)

(1)

$$G_y = \frac{\partial f(x,y)}{\partial y} = \lim_{h \to 0} \frac{f(x, y+h) - f(x,y)}{h} \approx f(x, y+1) - f(x,y)$$

Assume the horizontal axis is y axis, the mask should be [-1, 1]

(2)

[3 4 4 3 10 10 11 11 12]

↓ after replica border

[3 3 4 4 3 10 10 11 11 12 12]

Apply the mask [-1, 1] on the sequeme, the result is:

[0 1 0 -1 7 0 1 0 1 0]

According the result, we find that there is a value 7 which is a large number and can be detected as the edge. Look back to the original sequeme, we find that the corresponding postion is between 3 and 10.

(3)

- First, smoothing the image by convolution with the Gaussian filter
- Second, apply the differential operators along x and y axis to the image, add them up to create |Gx| + |Gy|
- Third, use non-maximum suppression to narrow the edges. Check whether it is the local maximum along gradient
- Fourth, use Hysteresis thresholding locates edge strings

a) This question is about **Edge Detection**.

[10 marks]

i) Derive the mask for the second-order derivative approximation of an image $f(x, y)$, $G_{xx} = \frac{\partial^2}{\partial x^2} f(x, y)$. Note that x and y are row index and column index, respectively.

(3 marks)

ii) For a piece of image [0,1,0,1,8,7,6,7,7], show the result of the convolution with the mask of the first-order derivative and indicate where the edges would be detected and why. Use replicate border extension for the convolution.

(5 marks)

iii) Having Laplacian operator, why do we still need the operators of Laplacian of Gaussian and Difference of Gaussians? Explain in your own words.

(2 marks)

(1)

$$G_{xx} = \frac{\partial^2}{\partial x^2} f(x,y) = \lim_{x \to h} \frac{\frac{\partial}{\partial x} f(x+h, y) - \frac{\partial}{\partial x} f(x,y)}{h}$$

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

$$\approx f(x+2, y) - f(x+1, y) - (f(x+1, y) - f(x, y))$$

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

Therefore the mask should be $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

(2)

[0 1 0 1 8 7 6 7 7]

$\downarrow$ replica border extension

[0 0 1 0 1 8 7 6 7 7 7]

$\downarrow$ [-1, 1] ⇒ first order derivative mask

[0 1 -1 1 7 -1 -1 1 0 0]

According to the result, we notice that a large value 7 represent the edge occurs. Look back to the original sequence the corresponding position is between 1 and 8.

(3)
Laplacian filter is very sensitive to noise. To make it more robust to noise, we can use Gaussian filter to smooth the image, so we create the combination of Gaussian and laplacian which is LoG operator. DoG is the difference of Gaussian which is an approximation of LoG.