Fundamentals of Image Processing

▶ Lecture 6: Edge detection ◀

Master of Computer Science
Sorbonne University

September 2022

Outline

Important definition

- Edge: a pixel at location of a significant (abrupt) intensity change in one direction.
- An edge is not an isolated pixel in the image: at least one of its neighbors is also an edge, otherwise it is noise.
- Contour: a list of edges delimiting the boundary of a region or an object in the image.
- A contour is a closed path while edges are fragments of a contour.
- Edges are low level information: a map of significant intensity changes.
- Contour is higher level information: the frontier of a region.

Introduction

Primary role of edge detection in computer vision

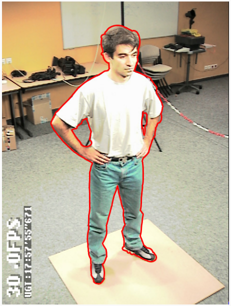
1. Information reduction
   - A lot of image information is represented in the edges.
   - Edges are an important informative part of an image: in the frequency domain, an intensity discontinuity is characterized by an infinite number of frequency components.
2. Edge detection is the first step before the determination of image primitives (straight lines, circles, contours, corners...).
3. Biological knowledge confirms the importance of edges in the mammal vision:
   - the first neuronal layers of the mammal vision system operate tasks of edge detection;
   - the same phenomenon is observed in a CNN performing a computer vision task.

Edge detection: applications

1. Pattern/object recognition, scene classification
2. Matching of key-points: registration, 3D reconstruction, object tracking
3. Compression

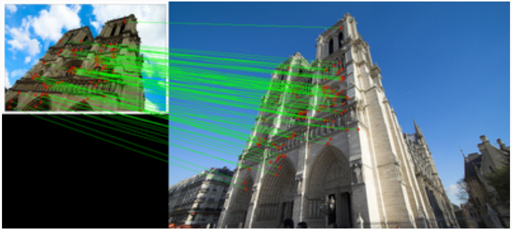
Introduction

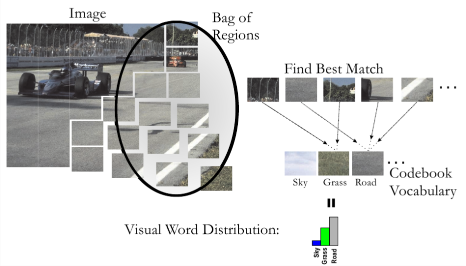
Applications: object segmentation, matching key-points ⇒ reconstruction



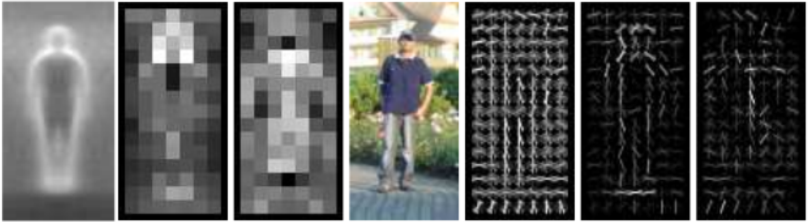
Introduction

Applications: matching key-points $\Rightarrow$ classification





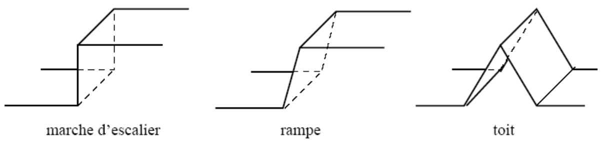
Applications: pattern description $\Rightarrow$ Object recognition

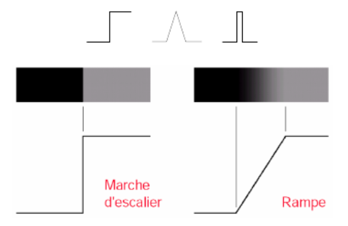


Outline

Edge models

- Several edge models: step, ramp, roof edge

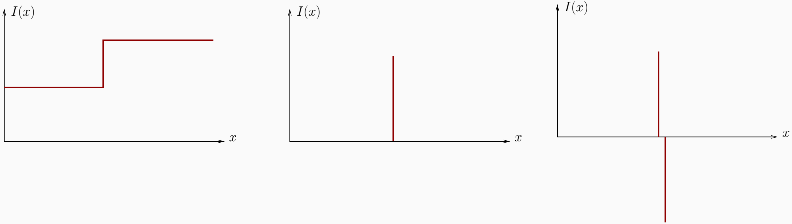




- step edge: ideal model of a region *boundary*
- ramp: a blurred model of a region boundary
- roof edge: model of bright lines

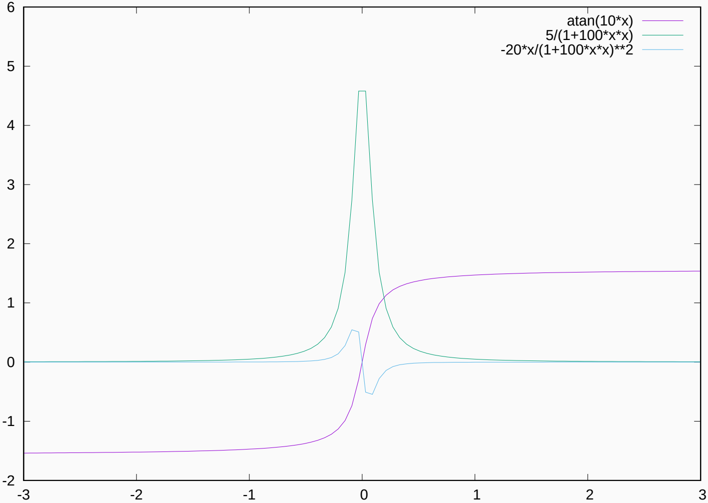
Edge models

- Most frequent model: the step / ramp edge. In 1-D:



  - $I(x)$ Heaviside step function
  - First derivative (in the sense of distributions): Dirac delta $I'(x) = \delta(x)$
    - $I'(x)$ maximal at the edge
  - Second derivative: $I''(x) = \delta'(x)$
    - $\delta'(x)$: Dirac delta derivative
    - $I''(x)$ is null at the edge

Edge models, smooth and continuous version



Edge models

- Image $f(x, y)$. Edges: locations of significant variations of f
- Gradient of f:

$$\overrightarrow{\nabla f} = \overrightarrow{G} = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{pmatrix}^T \qquad (1)$$

- Magnitude (length) of gradient vector $\overrightarrow{\nabla f}$:

$$G = \left\| \overrightarrow{\nabla f} \right\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \qquad (2)$$

- Direction of gradient vector $\overrightarrow{\nabla f}$ (unit vector):

$$\overrightarrow{g} = \frac{\overrightarrow{\nabla f}}{\left\| \overrightarrow{\nabla f} \right\|} \qquad (3)$$

Edge models

Definition (edges)

- Pixels for which the gradient magnitude G is maximal in the gradient direction $\overrightarrow{g}$

$$\frac{\partial f}{\partial \overrightarrow{g}} \text{ maximal or minimal and } \frac{\partial^2 f}{\partial \overrightarrow{g}^2} = 0 \tag{4}$$

$$\text{with } \frac{\partial}{\partial \overrightarrow{g}} = \overrightarrow{g} \cdot \overrightarrow{\nabla}$$

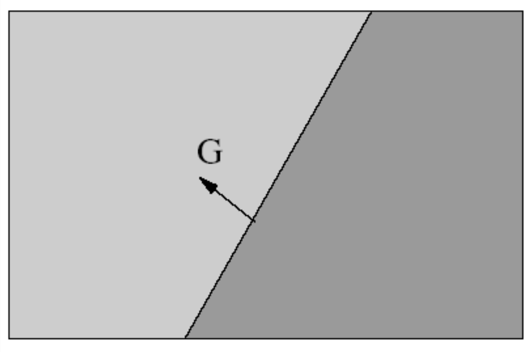
- In Cartesian coordinates, the equation is complex and non linear:

$$\frac{\partial f}{\partial x}\frac{\partial}{\partial x}\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} + \frac{\partial f}{\partial y}\frac{\partial}{\partial y}\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} = 0 \tag{5}$$

Edge models

Definition (edges)

- Location of gradient maxima in the direction of gradient:

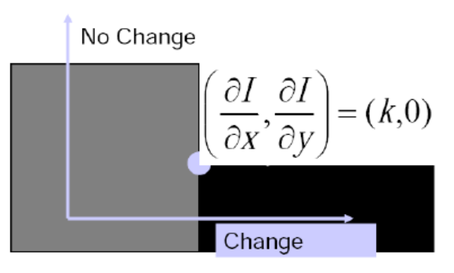


- $\vec{g}$ is orthogonal to the contour that contains the edge: the intensity along the curve describing the contour is supposed constant (isophote: curve of equal intensity)

Edge models

Definition (edges)

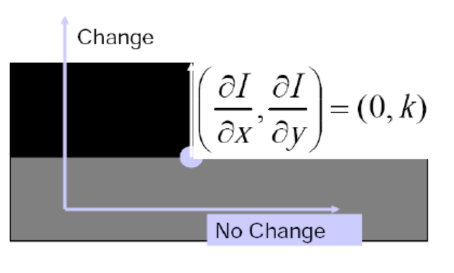
- Location of gradient maxima in the direction of gradient:



Vertical edge

Definition (edges)

- Location of gradient maxima in the direction of gradient:

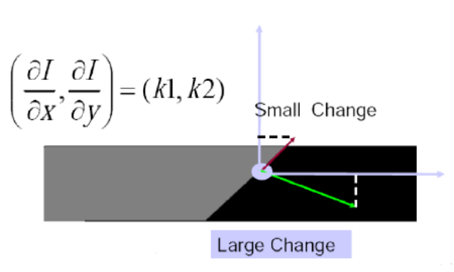


Horizontal edge

Edge models

Definition (edges)

- Location of gradient maxima in the direction of gradient:



Oblique edge

Outline

Edge detection with filtering: principle

Discrete approximation of differential operators: finite difference

- Edge $\Leftrightarrow$ abrupt variation of intensities: significant response of derivative
  - 1^{st} order approach $\Longleftrightarrow$ approximations of gradient
  - 2^{nd} order approach $\Longleftrightarrow$ approximations of Laplacian
- How to approximate differential operators: finite difference techniques, for example:

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

- Finite difference is a convolution!

Discrete approximation of gradient

1. Discrete approximation of gradient:
    - linear filtering (convolution, which mask?)
2. Magnitude (vector gradient norm)
    - Euclidean (or ℓ_2) norm: $\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$,
      Manhattan (or ℓ_1) norm: $|\frac{\partial f}{\partial x}| + |\frac{\partial f}{\partial y}|$, *etc.*
    - local maxima in the gradient direction
3. Suppression of non maxima
4. Threshold
5. Edge linking

Discrete approximation of gradient

Approximation of gradient $\overrightarrow{\nabla f} = \begin{pmatrix} \partial_x f & \partial_y f \end{pmatrix}^T$, $\partial_x = \frac{\partial}{\partial x}$

- $\partial_x f \approx f(x, y) - f(x-1, y)$ (and $\partial_y f \approx f(x,y) - f(x, y-1)$) see Lecture 5, slide 25, to identify mask values
- $\partial_x f \approx f(x+1, y) - f(x-1, y)$
- Roberts (gradient in direction $\frac{\pi}{4}$): $\partial_1 f \approx f(x+1, y+1) - f(x, y)$, $\partial_2 f \approx f(x, y+1) - f(x+1, y)$
- Others masks: Prewitt, Sobel, *etc.*

gradient:

$+1$	-1

Roberts:

$+1$	0
0	-1

Prewitt:

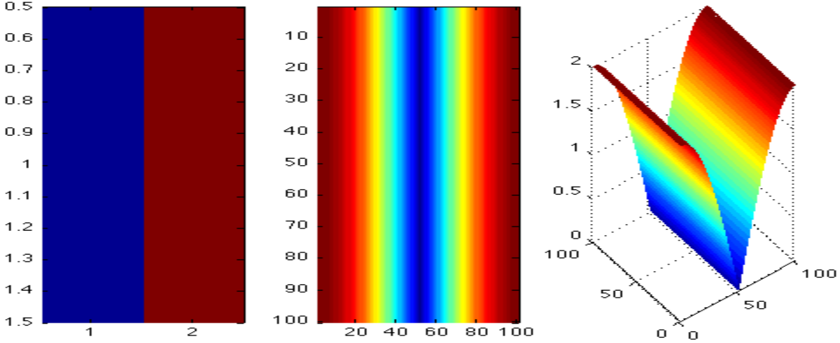
$+1$	0	-1
$+1$	0	-1
$+1$	0	-1

Sobel:

$+1$	0	-1
$+2$	0	-2
$+1$	0	-1

- For gradient and Roberts filters: where is the origin in the mask?

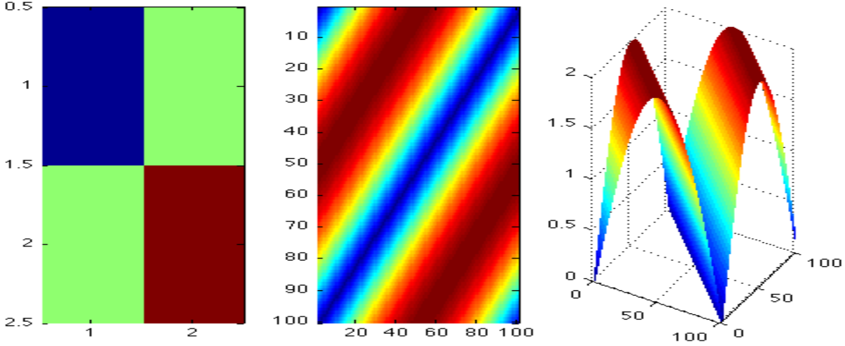
Transfer function of 1st order filters



Gradient filter (impulse response and transfer function)

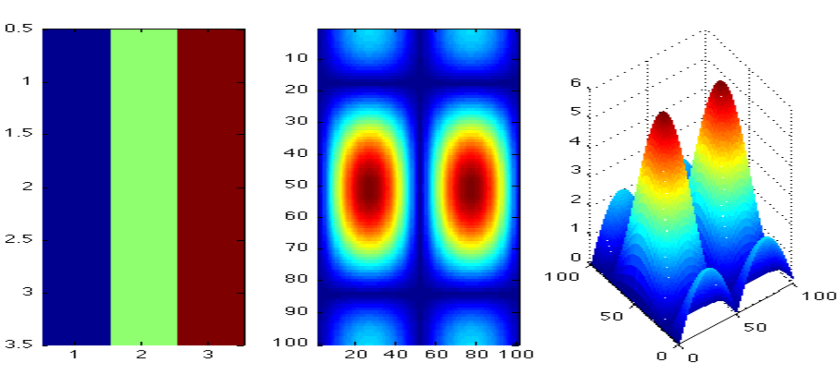
Discrete approximation of gradient

Transfer function of 1st order filters



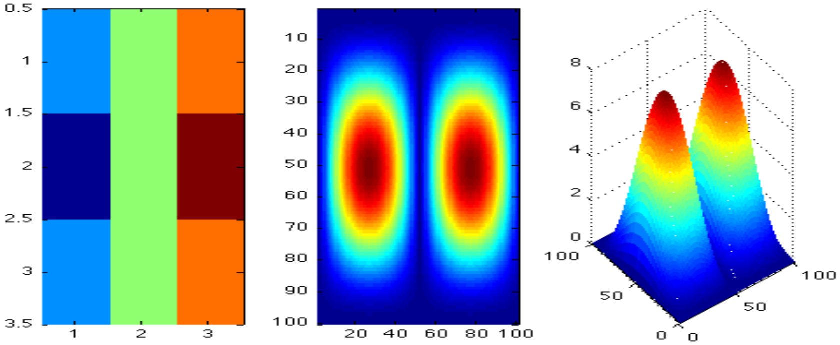
Roberts filter (impulse response and transfer function)

Transfer function of 1st order filters



Prewitt filter (impulse response and transfer function)

Transfer function of 1st order filters



Sobel filter (impulse response and transfer function)

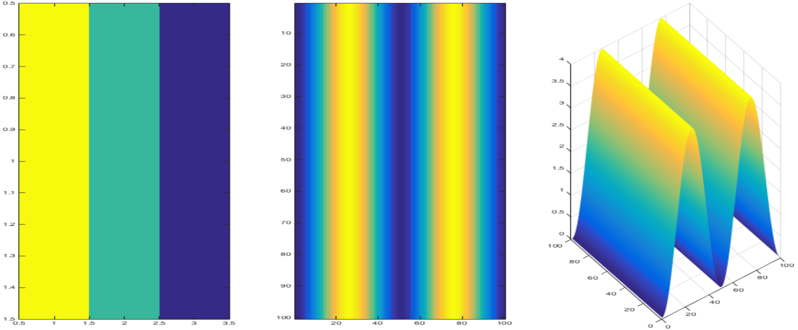
Discrete approximation of gradient

Some remarks on Sobel and Prewitt

- $\begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|}\hline 1 \\ 2 \\ 1 \\ \hline \end{array} \times \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline \end{array}$
- $\begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|}\hline 1 \\ 1 \\ 1 \\ \hline \end{array} \times \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline \end{array}$
- $\begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline \end{array}$ derivative filter similar to Roberts, horizontally oriented
- $\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline \end{array}^T$ is the averaging filter
- $\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline \end{array}^T$ is the best approximation of Gaussian sampled on 3 points ($\sigma \approx 0.86$)

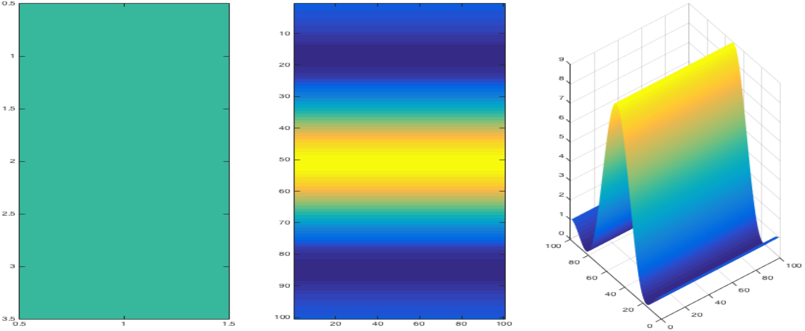
Discrete approximation of gradient

Some remarks on Sobel and Prewitt



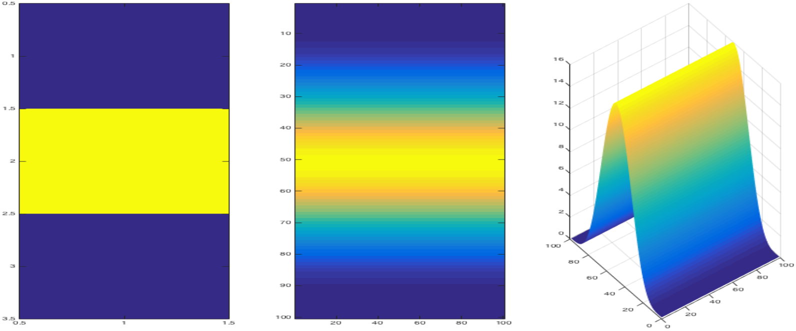
Derivative filter $\begin{pmatrix} 1 & 0 & -1 \end{pmatrix}$ – Impulse response and transfer function

Some remarks on Sobel and Prewitt



Averaging filter $\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ – Impulse response and transfer function

Some remarks on Sobel and Prewitt



Gaussian filter $\begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$ – Impulse response and transfer function

Discrete approximation of gradient

Many others filters: Kirsch

$$H_0 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad H_1 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad \bullet\bullet\bullet\bullet \quad H_7 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

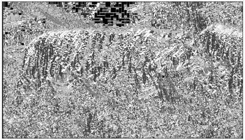
- Select the highest response: $\max_{i \in \{0 \dots 7\}} \{|H_i \star I|\}$
- Selected orientation: $\frac{\pi}{4} arg\ max_{i \in \{0 \dots 7\}} \{|H_i \star I|\}$

Example with Sobel (a frequently used 3×3 filter)

1. Gradient with Sobel
2. $G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \star I \qquad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \star I$
3. Gradient magnitude: $\|G\| = \sqrt{G_x^2 + G_y^2}$
4. Gradient orientation: $\theta = \arctan\left(\frac{G_y}{G_x}\right)$

Original image

$\|G\|$



θ

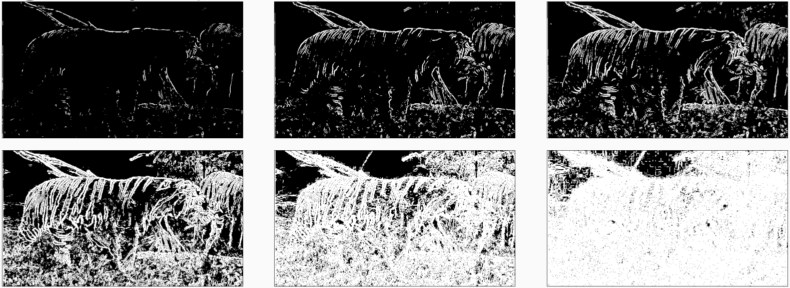
Discrete approximation of gradient

Binary map of edges

- ℓ_2 norm: $\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$ or ℓ_1: $|\frac{\partial f}{\partial x}| + |\frac{\partial f}{\partial y}|$
- Post-processing
    - Local maxima in gradient direction
    - Thresholding
    - ...

Discrete approximation of gradient

Edge binary map: example with Sobel



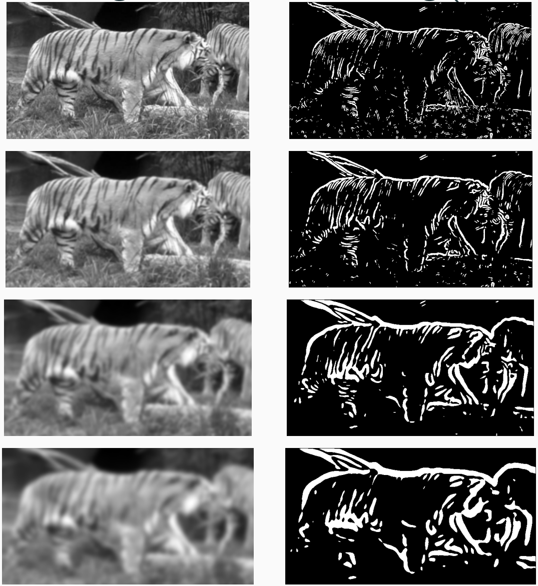
- Low threshold: relevant edges detected but many false positives (noise)
    - ⊖ Thick edges
- High threshold: little noise but too many missed detections
- Solution?

Combine smoothing and derivative filters

- Derivative filters are sensitive to noise, for instance: false positives in the textured areas
- Basic idea: smoothing **before** derivation $\Rightarrow$ local variations are filtered but the dominant variations (edges) remain
    - Linear filters for noises with a null mean. Ex: Gaussian filtering for Gaussian noise
    - Nonlinear filtering for impulse noise. Ex: median filter for Pepper and Salt noise

Discrete approximation of gradient

Gaussian smoothing and derivative filtering (with Sobel)



$\sigma = 1.0$

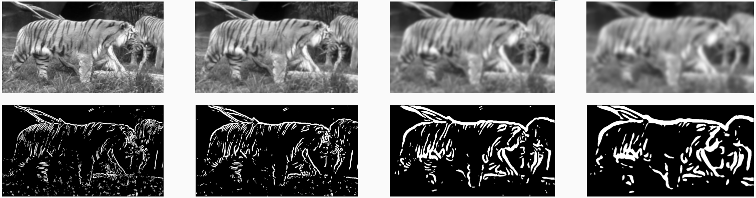
$\sigma = 2.0$

$\sigma = 4.0$

$\sigma = 6.0$

Discrete approximation of gradient

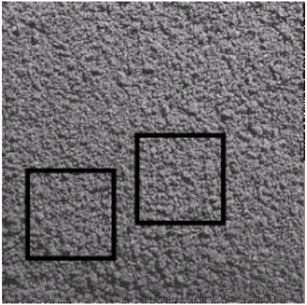
Gaussian smoothing and derivative filtering



- High smoothing: robustness to noise, but thick edges (imprecise localization)
- Low smoothing: sensitivity to noise, but accurate localization
- Threshold value: probably not possible to have an optimal value for the whole image
  - illumination and contrast variations
  - non maxima suppression (see Section Post-processing)

What is texture?

- Possible definition
  - A texture presents, at a given scale, the same aspect whatever the observed area.
    - Spatial repetition of a same pattern whatever the direction.
- More formally:
  - A texture is a realization of a stochastic and stationary process.
    - The pixel and its neighbors have the same intensity distribution.



Digression: texture in the images

What is texture?

- Some examples ...

  - fur
  - walls
  - sponge
  - tablecloth
  - various minerals...

Digression: texture in the images

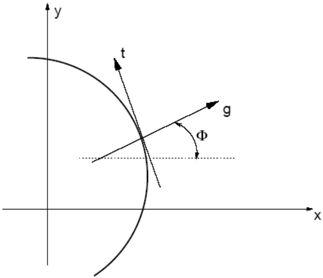
Texture and edge detection

- Texture is a repetition of pattern at small scale: quick intensity variation $\Rightarrow$ high frequency components.
- High pass filters, designed for edge detection, have a high response on textures.
- In general, an edge belongs to a contour (frontier of an object), texture is a false positive.

$\Rightarrow$ smoothing the image before edge detection reduces high frequencies from texture while preserving the main edges: it remains an issue.

Outline

Edge detection: second order approaches



$$\Phi = \mathrm{Arctan}\left(\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}}\right)$$

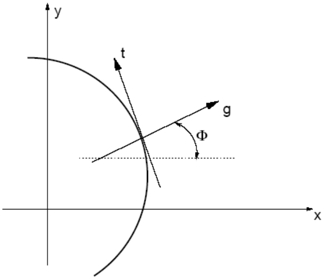
$$\frac{\partial f}{\partial \vec{g}} = \frac{\partial f}{\partial x}\cos\Phi + \frac{\partial f}{\partial y}\sin\Phi$$

$(i = f)$

- Second order: second derivative in the direction of gradient $\frac{\partial^2 f}{\partial \vec{g}^2}$
- Laplacian operator: $\triangle f = \frac{\partial^2 f}{\partial \vec{g}^2} + \frac{\partial^2 f}{\partial \vec{t}^2} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

$$\frac{\partial^2 f}{\partial \vec{g}^2} = 0 \Longleftrightarrow \frac{\partial^2 f}{\partial x^2}\cos^2\Phi + \frac{\partial^2 f}{\partial y^2}\sin^2\Phi + 2\frac{\partial^2 f}{\partial x \partial y}\cos\Phi\sin\Phi = 0$$

Edge detection: second order approaches



$$\Phi = \mathrm{Arctan}\left(\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}}\right)$$

$$\frac{\partial f}{\partial \vec{g}} = \frac{\partial f}{\partial x}\cos\Phi + \frac{\partial f}{\partial y}\sin\Phi$$

$(i = f)$

- Edges: points where the sign of Laplacian $\frac{\partial^2 f}{\partial \vec{g}^2}$ changes
  $\hookrightarrow$ zero-crossing
- Laplacian: $\triangle f = \frac{\partial^2 f}{\partial \vec{g}^2} + \frac{\partial^2 f}{\partial \vec{t}^2} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$
- Approximation: $\triangle f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \approx \frac{\partial^2 f}{\partial \vec{g}^2}$
  - The tangential component is neglected ($\frac{\partial^2 f}{\partial \vec{t}^2} \approx 0$): intensity along an edge is supposed having slow variation
  - Valid for edges with low curvature

Edge detection: second order approaches

Discrete approximation of Laplacian (finite difference)

$$\begin{aligned}\triangle f &\approx [f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)]-4f(x,y)\\ &\approx f\star\triangle_{\text{dis}}\end{aligned}$$

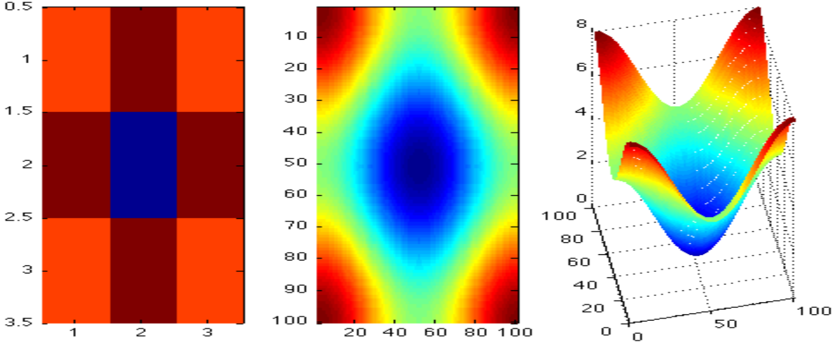
or

$$\begin{aligned}\triangle f \approx\ & [f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)+\\ & f(x+1,y+1)+f(x-1,y-1)+f(x-1,y+1)+\\ & f(x+1,y-1)]-8f(x,y)\approx f\star\triangle'_{\text{dis}}\end{aligned}$$

with:

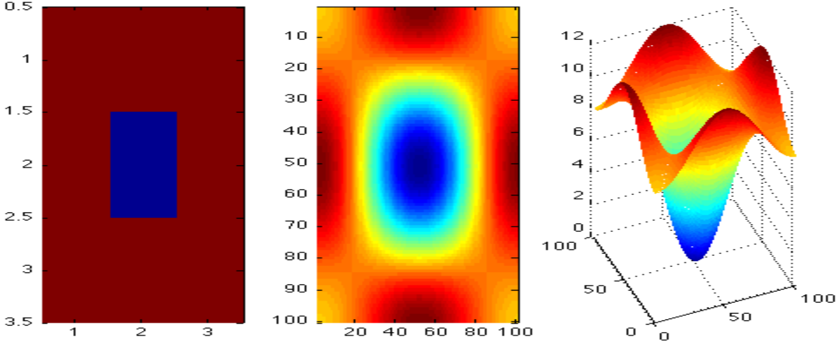
$$\triangle_{\text{dis}}=\begin{array}{|c|c|c|}\hline 0 & 1 & 0\\ \hline 1 & -4 & 1\\ \hline 0 & 1 & 0\\ \hline\end{array}\quad \triangle'_{\text{dis}}=\begin{array}{|c|c|c|}\hline 1 & 1 & 1\\ \hline 1 & -8 & 1\\ \hline 1 & 1 & 1\\ \hline\end{array}$$

Transfer function of second order filters



Laplacian 1 (impulse response and transfer function)

Transfer function of second order filters



Laplacian 2: $h(n,m)$ inverted rectangular function, $H(f,g)$ inverted sinc function

Discrete approximation of Laplacian

Edge detection with Laplacian

1. Apply Laplacian on image: $I_L = \triangle I \approx I \star \triangle_{\text{dis}}$
2. Detect the zero-crossings:
    - Apply a 3×3-window centered on (i, j), compute $\max(I_L)$ and $\min(I_L)$ in this window
    - A zero-crossing occurs if $\max(I_L) > 0$, $\min(I_L) < 0$ and $\max(I_L) - \min(I_L) > S$

Discrete approximation of Laplacian

Laplacian: example on a natural image

Original image

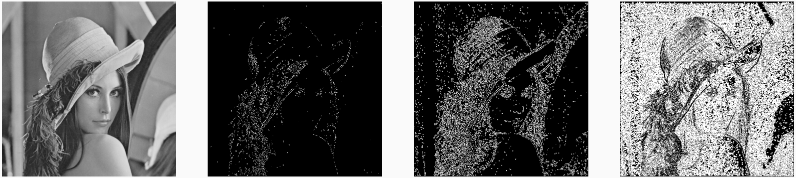
Laplacian

- Image filtered with Laplacian: edges $\Longleftrightarrow$ transitions between positive values (white) and negative values (black)
  - with a threshold S on the amplitude

Discrete approximation of Laplacian

Laplacian: example with several threshold values

Image | low | intermediary | high



- zero-crossing of Laplacian: threshold on amplitude of variation

Pro and cons of Laplacian against gradient

- $\oplus$ more closed path
- $\oplus$ less sensitive to threshold value
- $\ominus$ no edge orientation *v.s* $\oplus$ rotation invariant
- $\ominus$ bad localization (thick edges)
- $\ominus$ high sensitivity to noise (second order derivative)
  $\Rightarrow$ A lowpass filtering (smoothing) is required as pre-processing

Second order approaches

Pre-processing: smoothing lowpass filter, $I \star g$

1. Apply Laplacian on the smoothed image: $\triangle(I \star g)$
2. Convolution and derivatives are linear operators and commute:

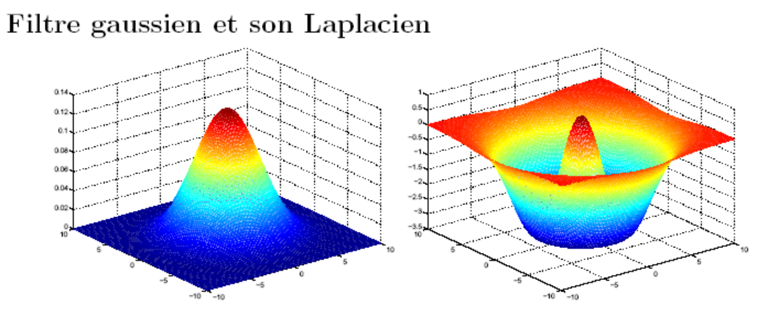
$$\triangle(I \star g) = (\triangle I) \star g = I \star (\triangle g) \tag{6}$$

3. Case of Gaussian smoothing: $g(x, y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}}$.
   Laplacian of g is easy to determine:
   $$\triangle g(x, y) = \frac{4}{\sqrt{2\pi\sigma}} (\frac{x^2+y^2}{2\sigma^2} - 1) e^{-\frac{x^2+y^2}{2\sigma^2}}$$
4. Conclusion: convolution of I with the sampled Laplacian of Gaussian (LoG) $\triangle g$

Second order approaches

Gaussian smoothing + Laplacian: $I \star \triangle g$

$\triangle g = \frac{4}{\sqrt{2\pi\sigma}}(\frac{x^2+y^2}{2\sigma^2} - 1)e^{-\frac{x^2+y^2}{2\sigma^2}}$

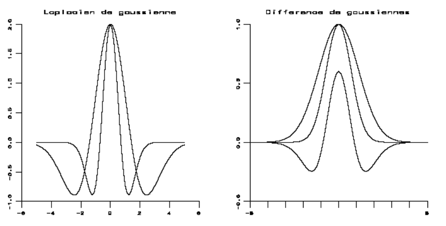


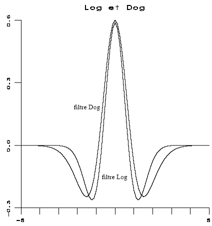
Second order approaches

Laplacian of Gaussian (LoG)

- LoG is similar to the human vision
- LoG can be approximated by a difference of Gaussians ($\frac{\sigma_1}{\sigma_2} = 1.6$)

$$LoG(x, y, \frac{\sigma_1}{\sigma_2}) \approx DoG(x, y, \sigma_1, \sigma_2) = g(x, y, \sigma_1) - g(x, y, \sigma_2) \tag{7}$$





- Used for fast computation of pyramidal multi-resolution schemes

Second order approaches

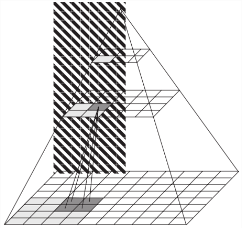
Pyramidal multi-resolution: construction (image of size $2^K \times 2^K$)

- We start with the original image (highest resolution, level 1) I_0, application of a Gaussian kernel g_σ: $I_0' = I_0 \star g_\sigma$
- Next resolution (level 2) is obtained by sub-sampling (factor 2) I_0': $I_1 = (I_0 \star g_\sigma) \downarrow 2$
- This process is iterated: I_{k+1} is sub-sampled (factor 2) from I_k': $I_{k+1} = (I_k \star g_\sigma) \downarrow 2$
- Up to an resolution of 2×2 pixels (level K, lowest resolution)

Pyramidal multi-resolution and edge detection

- The Gaussian filtering detects edges of given thickness (see Ramp edge model):
    - small value of σ: detects thin edges (Ramp function with high slope: quick intensity variation), good localization
    - high value of σ: detects thick edges (Ramp function with a low slope: slow intensity variation), bad localization
- Pyramid of resolutions: σ varies, each level of the pyramid detects edges at a given size ("scale")
- Multiscale fusion: edge detection at low resolution (bad localization), refine the localization using detection at higher resolutions

Pyramidal multi-resolution and edge detection



Pyramidal multi-resolution and edge detection: level 0

Pyramidal multi-resolution and edge detection: level 1

Pyramidal multi-resolution and edge detection: level 2

First and second order approaches

Pyramidal multi-resolution and edge detection: level 3

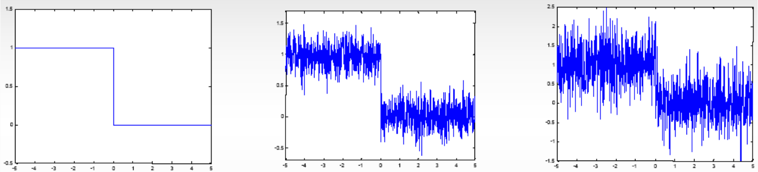
Outline

Continuous approaches

Optimal filtering: Canny (1986)

- 1-D edge model: a step function
- With noise: $C(x) = A\Theta(x) + n(x)$, n Gaussian noise
- Looking for a LTI and derivative filter f such that:

$$f \star C = \Theta \tag{8}$$



Canny approach: 3 optimality criteria

1. Good detection: Σ criterion
2. Good localization: Λ criterion
3. Uniqueness of the response

$\Rightarrow$ Looking for a derivative filter f maximizing Σ and Λ under the constraint of uniqueness response

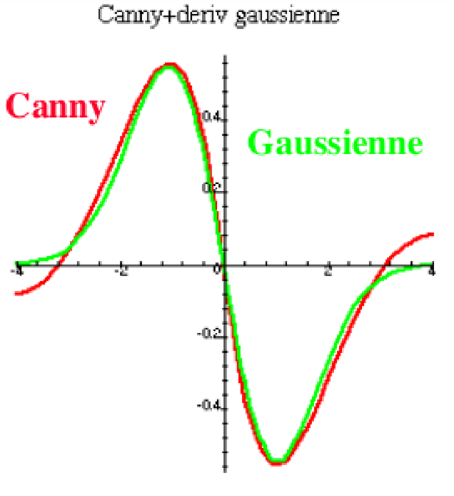
Canny approach: solution with a FIR

- Canny: solution with a finite impulse response (FIR) filter

  $$f(x) = a_1 e^{\frac{x}{\sigma}} \sin(wx) + a_2 e^{\frac{x}{\sigma}} \cos(wx) + a_3 e^{\frac{-x}{\sigma}} \sin(wx) + a_4 e^{\frac{-x}{\sigma}} \cos(wx)$$

  $+$ boundary conditions
- Can be approximated by a Gaussian derivative: $f(x) \approx -xe^{\frac{-x^2}{2\sigma^2}}$
- 2-D: smooth derivative in a direction is combined with a Gaussian derivative in other direction:
  - direction x: $f_x(x, y) = -xe^{\frac{-x^2}{2\sigma^2}} e^{\frac{-y^2}{2\sigma^2}}$
  - direction y: $f_y(x, y) = -ye^{\frac{-x^2}{2\sigma^2}} e^{\frac{-y^2}{2\sigma^2}}$

Canny: approximation with a Gaussian derivative



Canny: example of edge detection

filtre horizontal (contours verticaux)

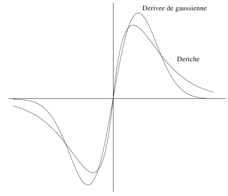
filtre vertical

norme du gradient

Contours (norme du gradient, seuillé et aminci)

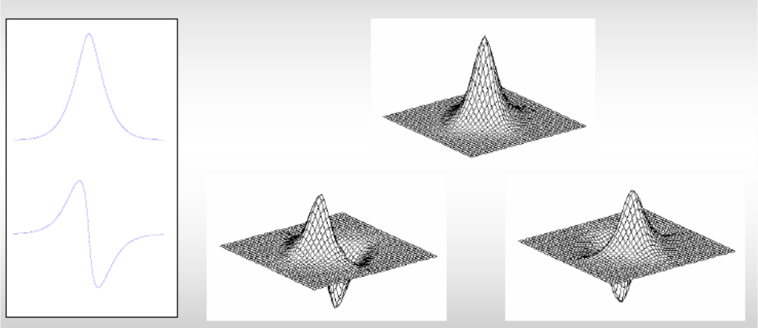
Recursive implementation of Canny filter: Canny-Deriche filter

- Solution of the Canny criteria for an infinite impulse response (IIR) filter:
$$f(x) = -ce^{-\alpha|x|}\sin(wx) \tag{9}$$
with $c = (1 - e^{-\alpha})^2/e^{-\alpha}$
- Can be implemented in a recursive way: let x_n be the signal to process and y_n the processed signal. There exists α such that (Deriche, 1987): $y_n = y_{n-1} + \alpha x_{n-1}$



Canny-Deriche: graph of 2-D filters

- Let k be a primitive of f, then the 2-D Canny-Deriche filters write:
  $f_x(x,y) = f(x)k(y)$ and $f_y(x,y) = f(y)k(x)$



Canny-Deriche: results

α=0.2 α=0.7 α=1.5

- Influence of $\alpha \iff$ scale factor, $\alpha \approx \frac{1}{\sigma}$ for a Gaussian filter
- High α value: bad robustness to noise, good localization
- Low α value: good robustness to noise, bad localization
- α depends on the image signal to noise ratio

Other optimal filters

- General schema to design an edge detector: a Cartesian product of:
    1. a low-pass filter, symmetrical, driven by a parameter $\alpha \iff$ noise and edges thickness (the scale of contours)
    2. a high-pass filter, anti-symmetrical
- Shen-Castan: $f(x) = c\,\mathrm{sign}(x)e^{-\alpha|x|}$
- Numerous filters in the scientific literature: Spacek, Petrou, *etc.*

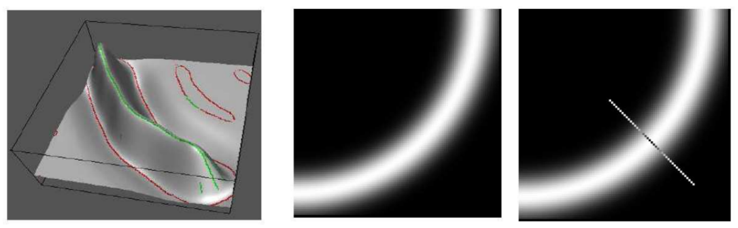
Outline

From image derivative to map of contours

- Recall: edge $\neq$ contour:
  - edges: places of significant intensity variation, possibly belonging to a contour
  - contours: the frontier (closed path) of a region/object
- Image derivatives: probability/likelihood to belong to a **contour**
  1. first order: high values indicate contour
  2. second order: values close to zero indicate a contour
- To extract contours of an image:
  1. edge detection (first order)
  2. suppression of non maxima in the gradient direction
  3. thresholding
  4. edge linking

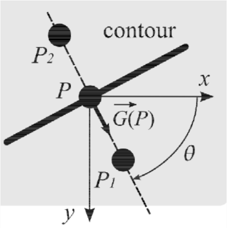
Non maxima suppression

- Recall: the gradient magnitude is maximal along the gradient direction
    1. Given a pixel P, determine the gradient magnitude along the straight line passing through P and oriented in the direction of gradient of P
    2. Check whether the gradient magnitude is maximal at P along this straight line



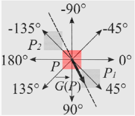
Non maxima suppression

- Recall: the gradient magnitude is maximal along the gradient direction
- P is a local maxima in the gradient direction $\iff$ $\|\overrightarrow{G(P)}\| > \|\overrightarrow{G(P_1)}\|$ and $\|\overrightarrow{G(P)}\| > \|\overrightarrow{G(P_2)}\|$

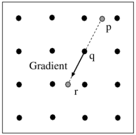


Non maxima suppression: practical case

1. Approximation of the direction:
   - four possible directions (8 neighbors): 0, $\frac{\pi}{4}$, $\frac{\pi}{2}$, $\frac{3\pi}{4}$
   - for one direction: two neighboring pixels
2. **Or** interpolation:
   - The value of gradient magnitude of subpixels is derived by interpolation of two neighboring pixel values



(1)



(2)

3. Suppression of P if $\|\overrightarrow{G(P)}\|$ non maximal

Non maxima suppression: practical case

- Insure to have thin edges (thickness of 1 pixel)

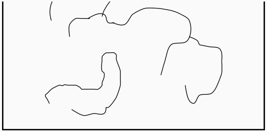
original image

$\|\overrightarrow{G}\|$ NMS Thin edges

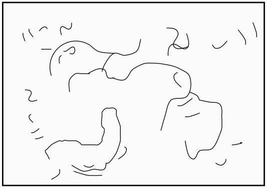
Thresholding

1. Basic thresholding: one keeps pixels such as $\|\overrightarrow{G(x,y)}\| > S$
    - Difficulty to have a unique and relevant threshold value S for the whole image
    - $\hookrightarrow$ contrast variation, illumination variation, *etc.*
2. Hysteresis thresholding:
    - Given two values of threshold: $S_{\text{low}} < S_{\text{high}}$
    - Basic thresholding with S_{high}: select edges such that $\|\overrightarrow{G(x,y)}\| > S_{\text{high}}$
    - Recursive thresholding with S_{low}: select low thresholded edges, $\|\overrightarrow{G(x,y)}\| > S_{\text{low}}$, **which are connected to high thresholded** edges

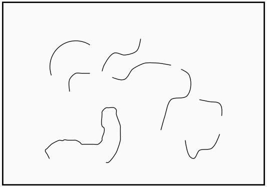
reconstruction



low threshold



high threshold



Edge linking and boundary detection

- The final step: extract the border of a region, an object. Required for many applications.
- Local approaches: consider the pixels as a graph and try to find closed or shorter path (dynamical programming).
- Regional and global approaches: find parametric curves matching the edges in the region or the whole image (regression, Hough transform).
- Out of the scope of this lecture...

THE END

QUESTIONS ?