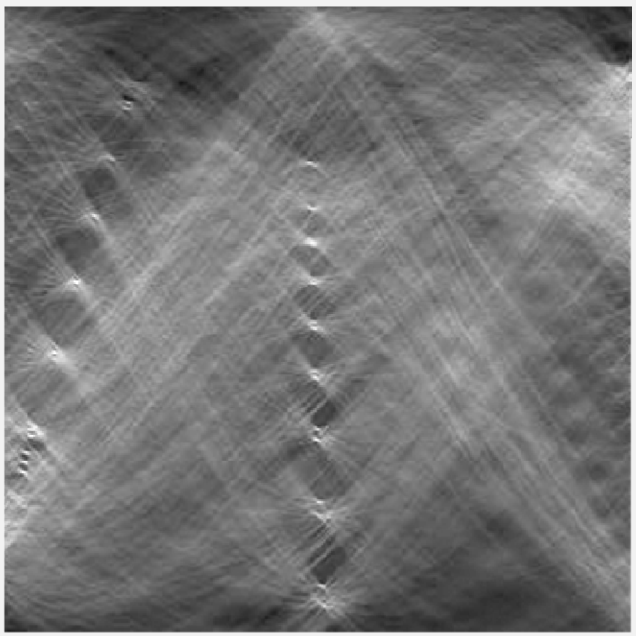
Department of Computer Science
University of Bristol

Image Processing and Computer Vision

www.ole.bris.ac.uk/bbcswebdav/courses/COMS30121_2018/content
www.ole.bris.ac.uk/bbcswebdav/courses/COMSM0020_2018/content



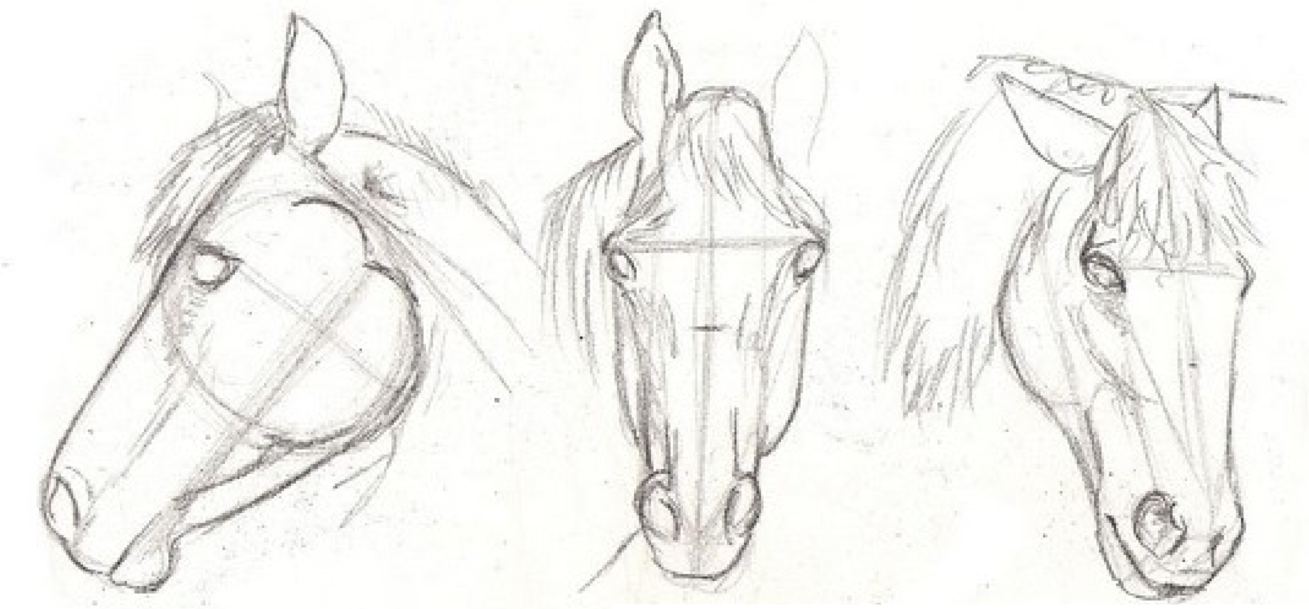
Lecture 04

Edges and Shapes

Andrew Calway | Tilo Burghardt | Sion Hanunna

21 Slides

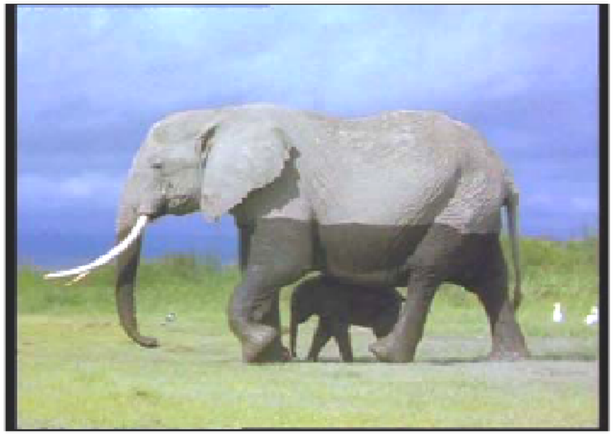
Edges in Artistic Drawings



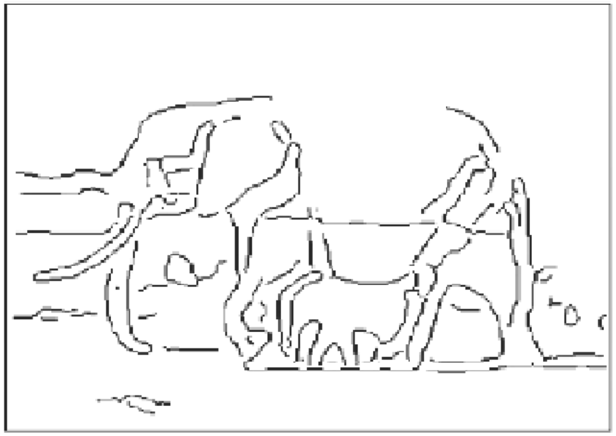
Source: http://www.ateliermagique.com

Motivation: Why detecting Edges?

- **Edges:** sharp changes of image brightness
- **Sources:** Object boundaries, Patterns, Shadows, …
- Meaningful edges ← → Nuisance edges
- For Segmentation: finding object boundaries
- For Recognition: extracting patterns
- For Motion Analysis: reliable tracking regions



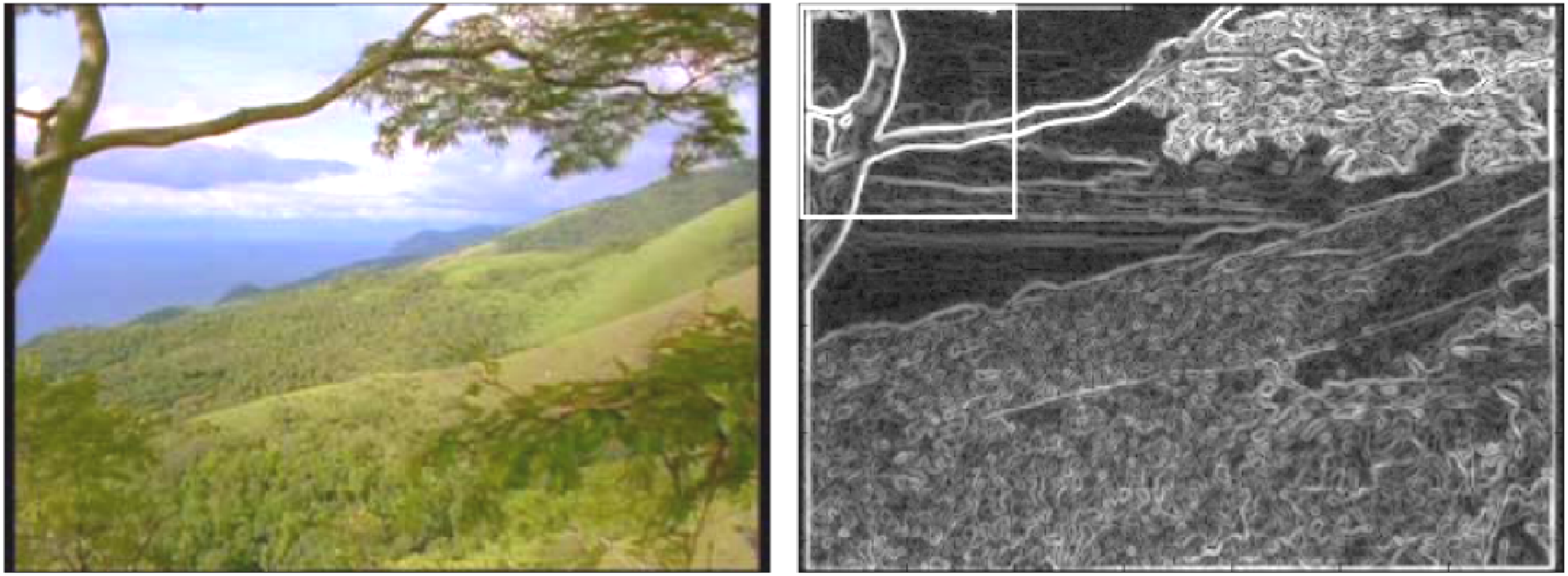
→



Edge Detection Strategy

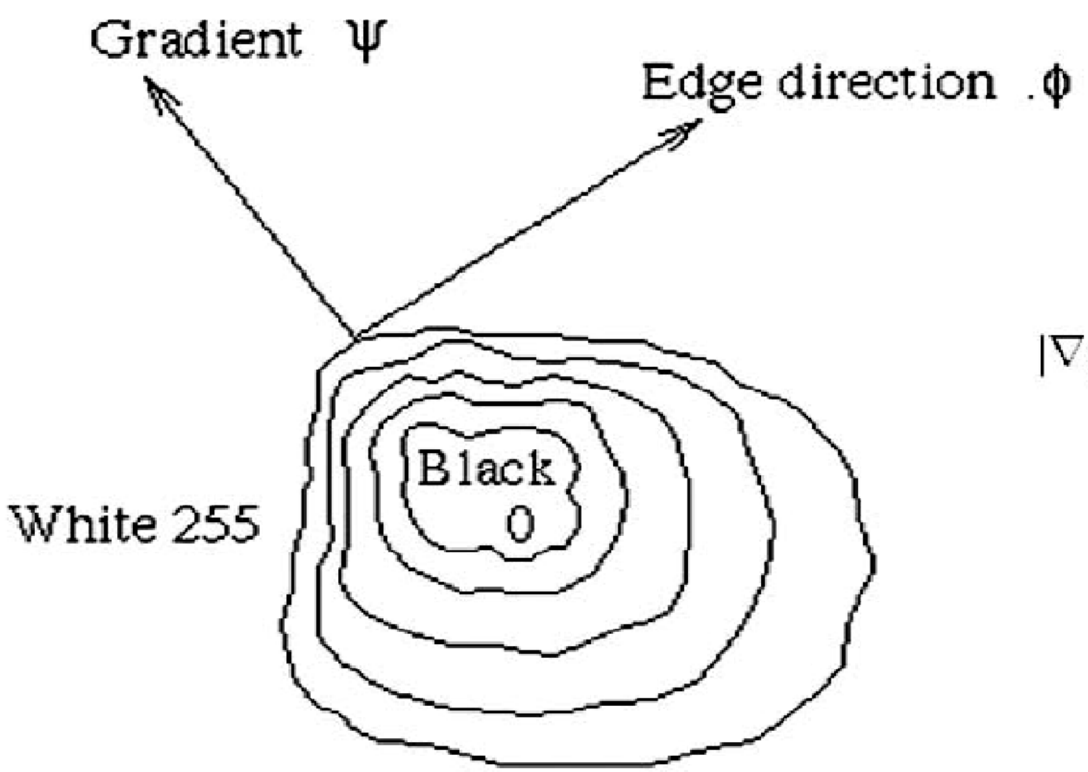
- **Recognition Strategy:**
  Determine a 'measure of change' in the pixel's neighbourhood

- First derivation in 2D space → Image Gradient



The Image Gradient

- A *vector* variable
  - Direction ψ of the maximum growth of the function
  - Magnitude $|\nabla f(x,y)|$ of the growth
  - Perpendicular to the edge direction



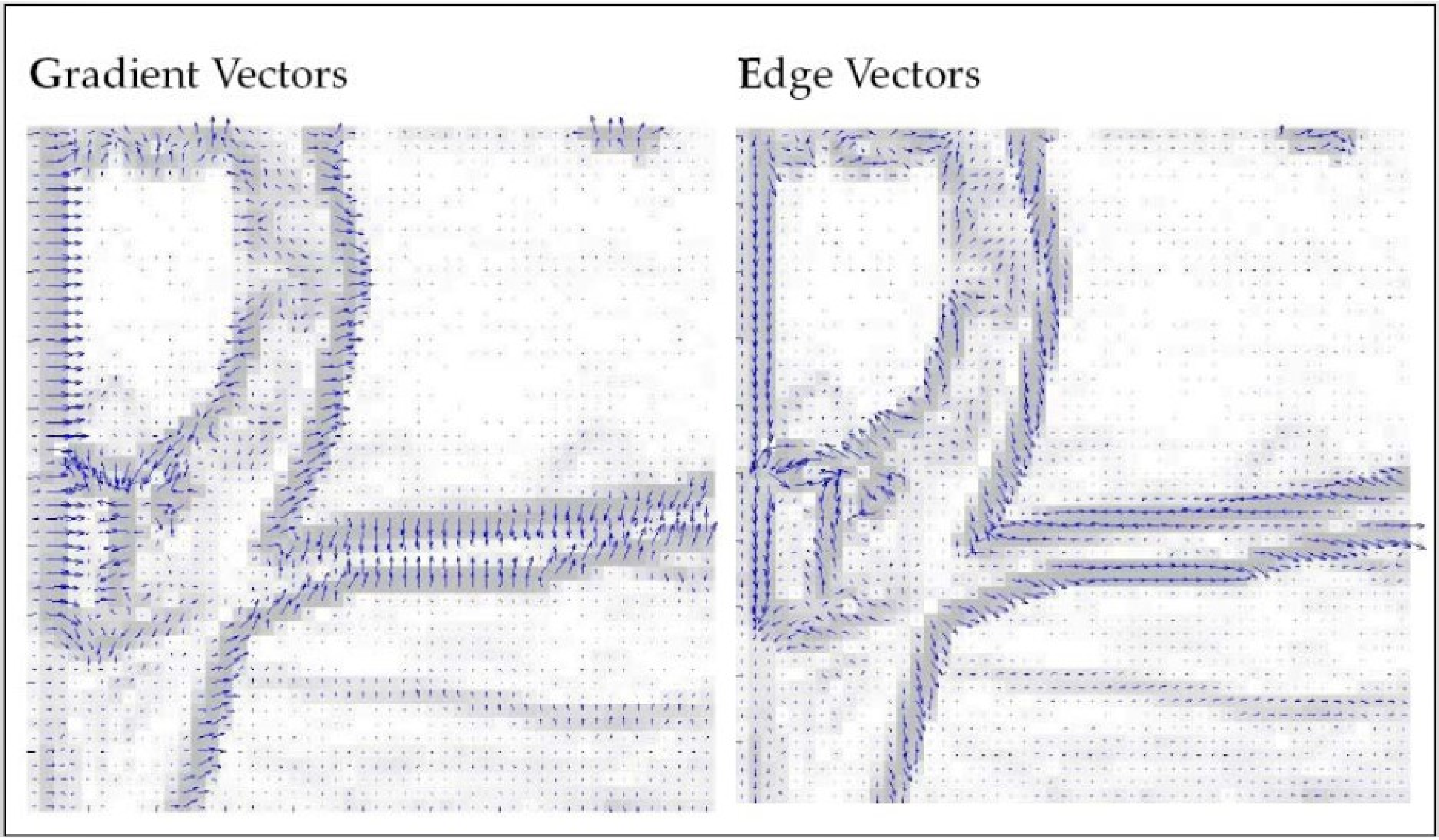
$$\nabla f(x,y) = \frac{\partial f}{\partial x}\hat{\mathrm{x}} + \frac{\partial f}{\partial y}\hat{\mathrm{y}}$$

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

$$\psi = \arctan\left(\frac{\partial f/\partial y}{\partial f/\partial x}\right)$$

$$\phi = \psi - \frac{\pi}{2}$$

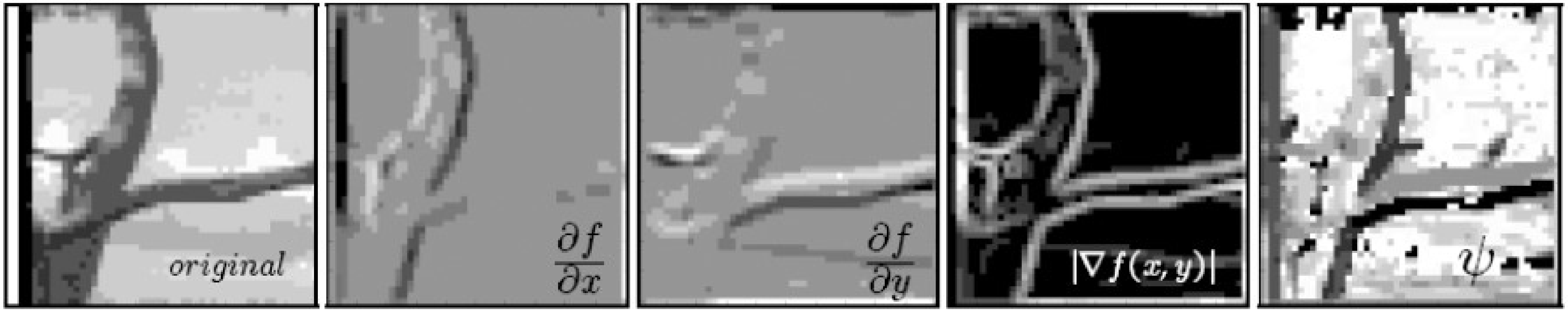
Example: Gradient & Edge Vectors



Gradient Extraction via Filtering

$$\frac{\partial f}{\partial x} \approx \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \frac{\partial f}{\partial y} \approx \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$grad(f) = |\nabla f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \qquad \psi = \arctan\left(\frac{\partial f/\partial y}{\partial f/\partial x}\right)$$



Roberts Operator

- Oldest gradient approximation kernel

- $h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

- $\nabla f = \begin{array}{l} |f(i,j) - f(i+1,j+1)| + \\ |f(i,j+1) - f(i+1,j)| \end{array}$

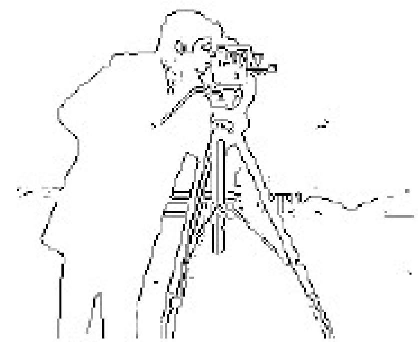
- Very efficient - only 2x2 mask size

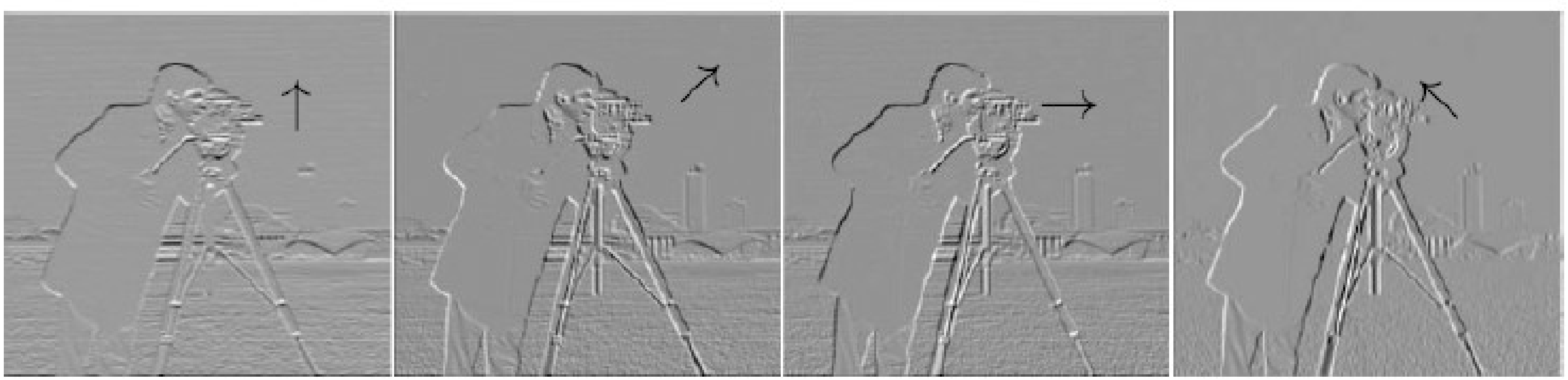
- High sensitivity to noise

Prewitt Operator

- Central difference $\frac{\delta f}{\delta x} \approx \frac{f(x+1) - f(x-1)}{2}$
- Mask [-1 0 1] is very sensitive to noise
- Smoothing in the perpendicular direction
- For 3x3 mask, ∇f estimated in 8 directions

$$h_{hor} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, h_{dia} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \ldots$$



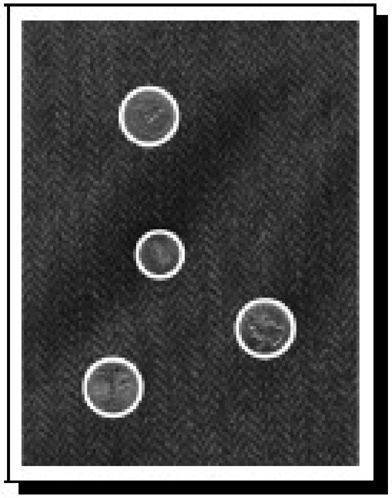


Sobel Operator

- As Prewitt, relies on central differences
- Greater weight to the central pixels
- $\frac{\partial}{\partial x} \approx \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \frac{\partial}{\partial y} \approx \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
- Can be approx as derivative of a Gaussian
- First Gaussian smoothing, then derivation
- $\frac{\partial}{\partial x}(I * G) = I * \frac{\partial G}{\partial x}$

Shape Recognition via Hough Transform

- How to detect, locate and describe simple geometrical shapes?
- Basic recognition task
- Choice of feature set and processing domain
- Detecton/Recognition algorithm

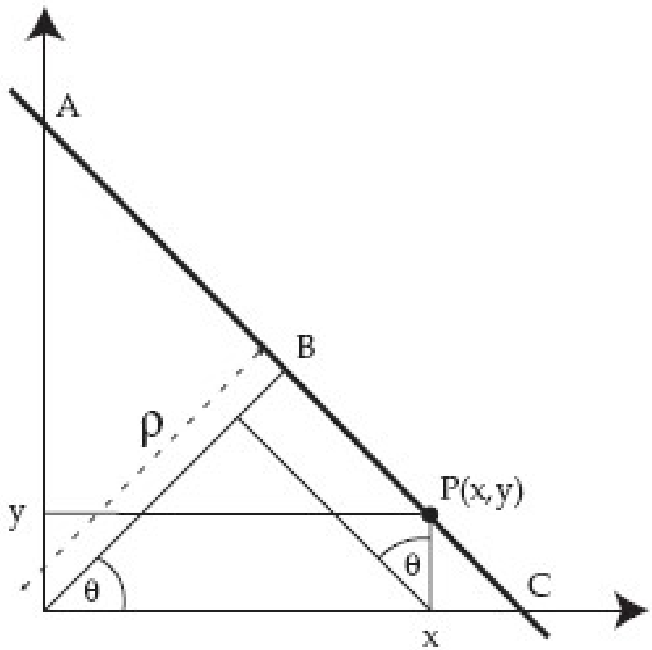



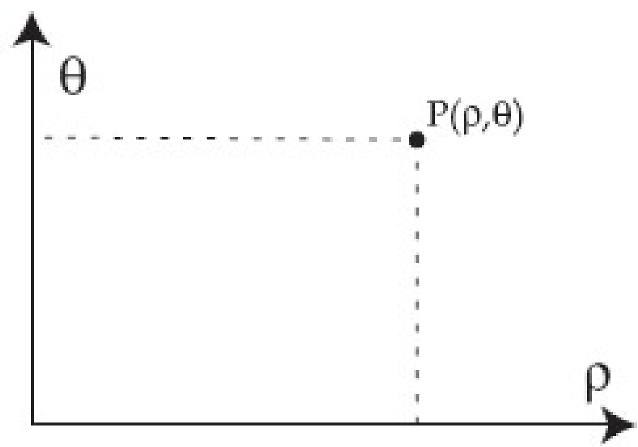

Line Representation

A straight line in 2D space described by this parametric equation:

$$f(x, y, \rho_0, \theta_0) = x\ cos\ \theta_0 + y\ sin\ \theta_0 - \rho_0 = 0$$

can be represented in the 2D parameter space by a point (ρ_0, θ_0), where ρ_0 is the distance between the straight line and the origin, and θ_0 is the angle between the distance vector and the positive x-direction.

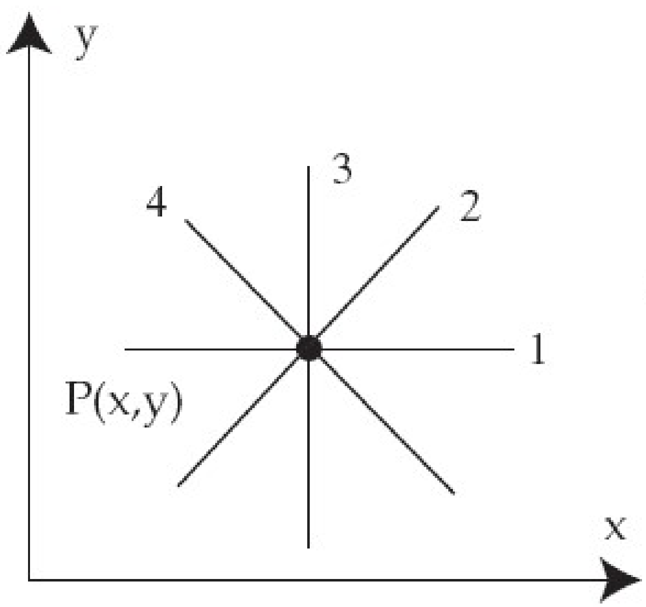


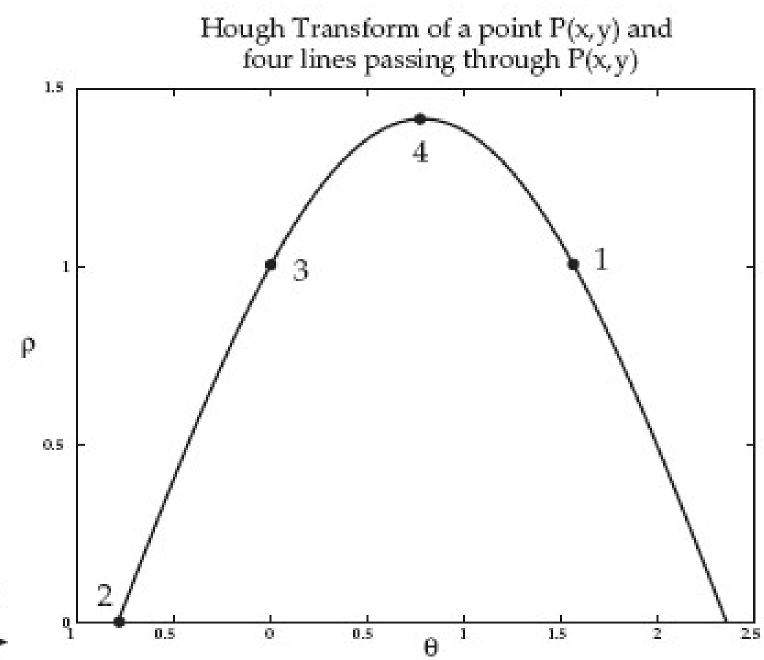


The Hough Space

A *point* (x_0, y_0) in the image space is transformed into a *sinusoidal curve* in the parameter space. A *point* (θ, ρ) on this sinusoidal curve represents a *straight line* passing through the point (x_0, y_0) in the image space.

	point 1	point 2	point 3	point 4
θ	$\pi/2 = 1.571$	$-\pi/4 = -0.785$	0	$\pi/4 = 0.785$
ρ	1	0	1	1.4142





Line Detection Algorithm

1. Make available an $n = 2$ dimensional array $H(\rho, \theta)$ for the parameter space;
2. Find the gradient image: $G(x, y) = |G(x, y)| \angle G(x, y)$;
3. For any pixel satisfying $|G(x, y)| > T_s$, increment all elements on the curve $\rho = x\ cos\ \theta + y\ sin\ \theta$ in the parameter space represented by the H array:

$$\forall\theta \quad | \quad \rho = x\ cos\ \theta + y\ sin\ \theta$$

$$H(\rho, \theta) = H(\rho, \theta) + 1;$$

4. In the parameter space, any element $H(\rho, \theta) > T_h$ represents a straight line detected in the image.

Line Detection using Gradient Information

This algorithm can be improved by making use of the gradient direction $\angle G$, which, in this particular case, is the same as the angle θ. Now for any point $|G(x,y)| > T_s$, we only need to increment the elements on a small segment of the sinusoidal curve. The third step in the above algorithm can be modified as:

1. Make $n = 2$ dimensional array $H(\rho, \theta)$
2. Find the gradient image: $G(x,y) = |G(x,y)|\angle G(x,y)$;
3. For any pixel satisfying $|G(x,y)| > T_s$,

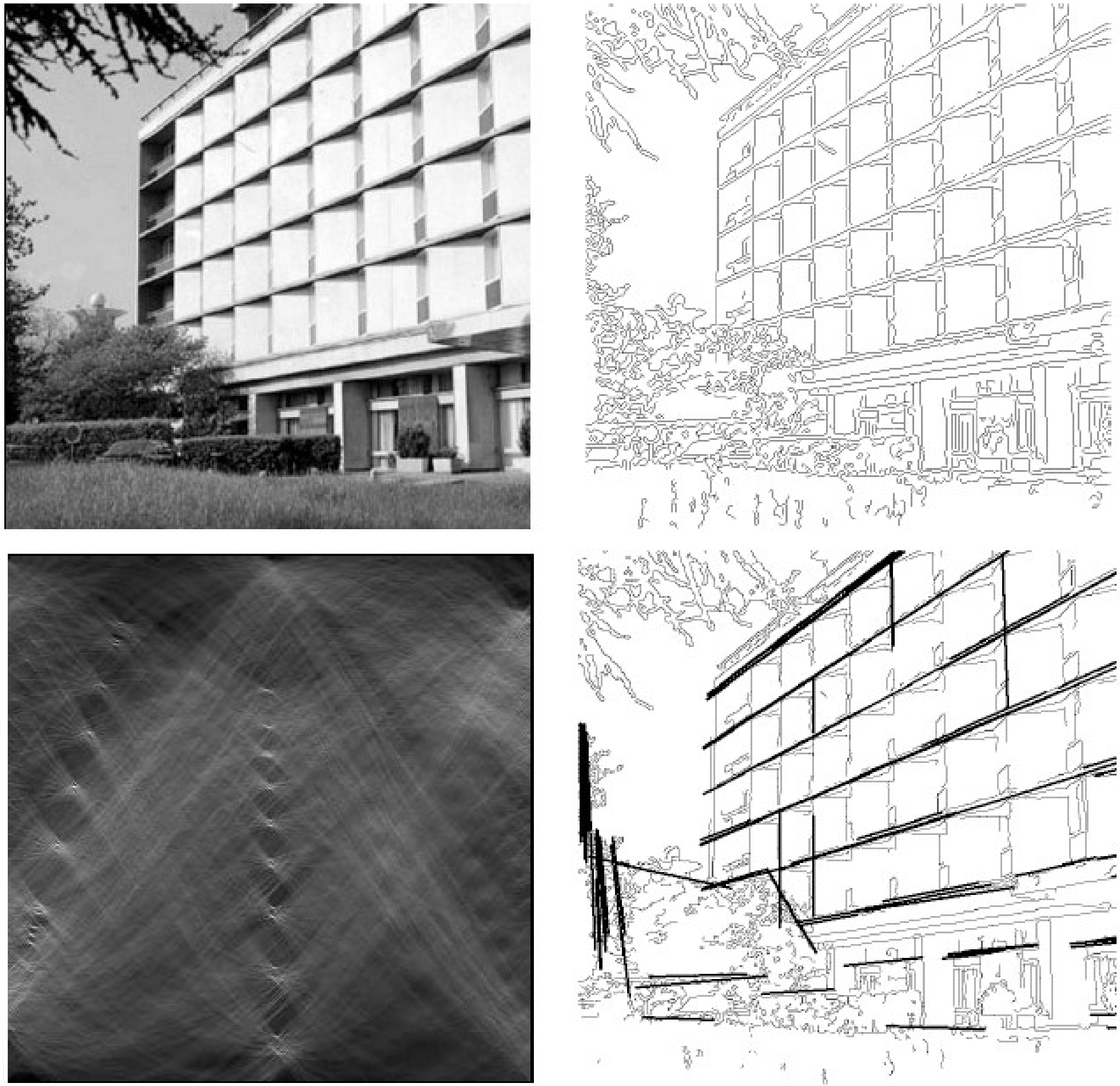
   $$\forall\theta \quad | \quad \angle G(x,y) - \Delta\theta \le \theta \le \angle G(x,y) + \Delta\theta$$

   $$\rho = x \; cos \; \theta + y \; sin \; \theta$$

   $$H(\rho,\theta) = H(\rho,\theta) + 1;$$

   where $\Delta\theta$ defines a small range in θ to allow some room for error in $\angle G$.
4. Any element $H(\rho,\theta) > T_h$ represents a straight line

Line Detection Example



Circle Detection Algorithm

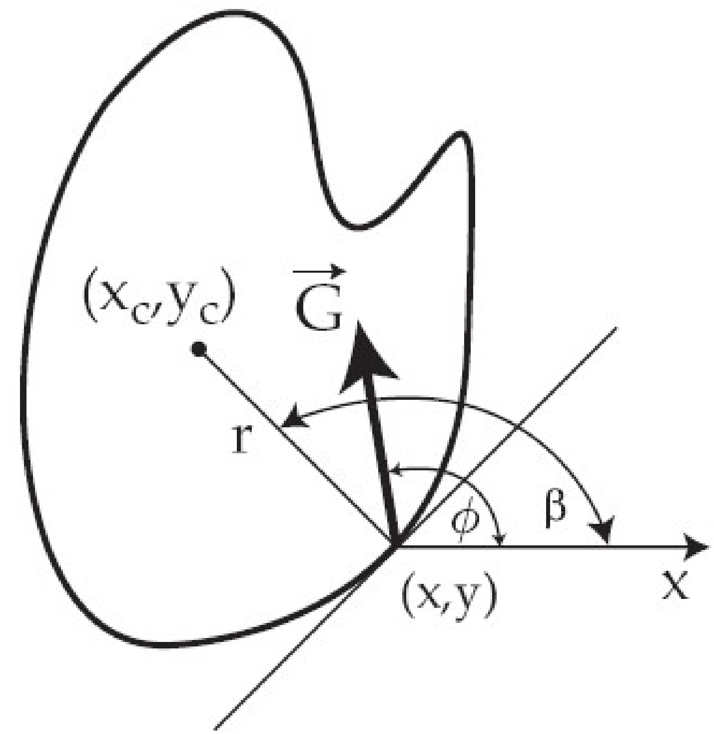
1. For any pixel satisfying $|G(x,y)| > T_s$, increment all elements satisfying the two simultaneous equations

$$\forall r, \quad \begin{cases} x_0 = x \pm r\ cos\ \angle G \\ y_0 = y \pm r\ sin\ \angle G \end{cases}$$

$$H(x_0, y_0, r) = H(x_0, y_0, r) + 1;$$

2. In the parameter space, any element $H(x_0, y_0, r) > T_h$ represents a circle with radius r located at (x_0, y_0) in the image.

Encoding Shapes Generally



$\phi_1 = 0$	$(r,\beta)_{1_1}$	$(r,\beta)_{1_2}$	$\cdots$	$(r,\beta)_{1_{n_1}}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
ϕ_j	$(r,\beta)_{j_1}$	$(r,\beta)_{j_2}$	$\cdots$	$(r,\beta)_{j_{n_1}}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$\phi_k = \pi$	$(r,\beta)_{k_1}$	$(r,\beta)_{k_2}$	$\cdots$	$(r,\beta)_{k_{n_1}}$

General Hough Parameters

No analytical form of the targeted shape $\Rightarrow$ Generate an approximation by calculating θ & ϕ in k points as follows:

- Prepare a table with k entries each indexed by an angle $\phi_i, \ (i = 1, \cdots, k), \Delta\phi = 180/k$

- Define a reference point (x_c, y_c) (e.g., center of gravity) $\forall P(x, y)$ on the boundary of the shape, find

$$\begin{cases} r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \\ \beta = tan^{-1} \ (y - y_c)/(x - x_c) \end{cases}$$

  and the gradient direction $\angle G$. Add the pair (r, β) to the table entry with its ϕ closest to $\angle G$.

- Prepare a 2D Hough array $H(x_c, y_c)$ initialized to 0.

Generalised Hough Transform

- For each image point (x, y) with $|G(x, y)| > T_s$, find the table entry with its corresponding angle ϕ_j closest to $\angle G(x, y)$

- For each of the n_j pairs $(r, \beta)_i$ $(i = 1, \cdots, n_j)$ in this table entry, find

$$\begin{cases} x_c = x + r\ cos\ \beta \\ y_c = y + r\ sin\ \beta \end{cases}$$

- Increment the corresponding element in the H array by 1:

$$H(x_c, y_c) = H(x_c, y_c) + 1$$

All elements in the H table satisfying $H(x_c, y_c) > T_h$ represent the locations of the shape in the image.

Invariant Generalised Hough Transform

It is desirable to detect a certain 2D shape independent of its ***orientation and scale***, as well as its location. Two additional parameters, a scaling factor S and a rotational angle θ, are needed to describe the shape. Now the Hough space becomes 4-dimensional $H(x_c, y_c, S, \theta)$.
$\forall P(x, y)$ with $|G(x, y)| > T$, find the proper table entry with $\phi_j = \angle G(x, y)$. Then for each of the n_j pairs $(r, \beta)_i$ $(i = 1, \cdots, n_j)$ in this table entry, do the following for all S and θ: find

$$\begin{cases} x_c = x + r\ S\ cos(\beta + \theta) \\ y_c = y + r\ S\ sin(\beta + \theta) \end{cases}$$

and increment the corresponding element in the 4D H array by 1:

$$H(x_d, y_c, S, \theta) = H(x_c, y_c, S, \theta) + 1$$

All elements in the H table satisfying $H(x_c, y_c, S, \theta) > T_h$ represent the scaling factor S, rotation angle θ of the shape, as well as its reference point location (x_c, y_c) in the image.