U6. Contours

SJK002 Computer Vision

Master in Intelligent Sytems

UNIVERSITAT JAUME•I

Index

- Definitions
- Representation of curves
- Curve fitting
  - Error measures
  - Linear piecewise fitting
  - Other fittings
- Regression
  - Robust regression
- Hough transform

Definitions

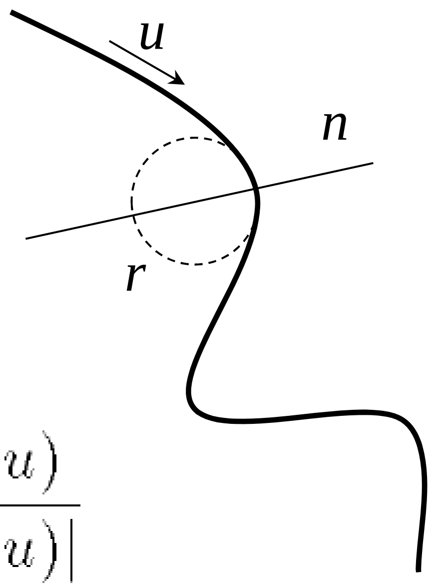
- Contour representation:
  - Simple and compact
  - Precise
  - Adequate to make operations easier

- Definitions:
  - Border list:
    - Set of ordered points
  - Contour:
    - Set of borders or curve
  - Frontier:
    - Closed contour that includes a region

Geometry of a curve

Representation types:

- Explicit $y = f(x)$
- Implicit $f(x, y) = 0$
- Parametric $p(u) = (x(u), y(u))$



Definitions:

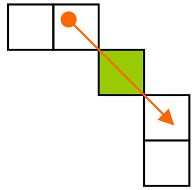
- Tangent unitary vector $t(u) = \dfrac{p'(u)}{|p'(u)|}$
- Normal vector $n(u) = p''(u)$
- Curvature $k = \dfrac{1}{r}$
- Curve length $\displaystyle\int_{u_1}^{u_2} \sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2}\, du$

Digital curves

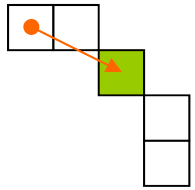
Basic representation: $p_i = (x_i, y_i)$ (border points)

Definitions:

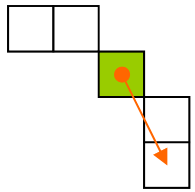
- K-slope (angle) $p_{i-k/2} \rightarrow p_{i+k/2}$



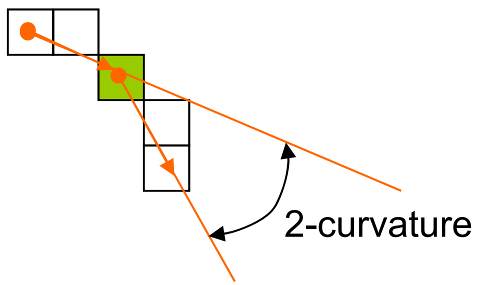
- Left K-slope $p_{i-k} \rightarrow p_i$



- Right K-slope $p_i \rightarrow p_{i+k}$



K-curvature: Difference between left K-slope and right K-slope

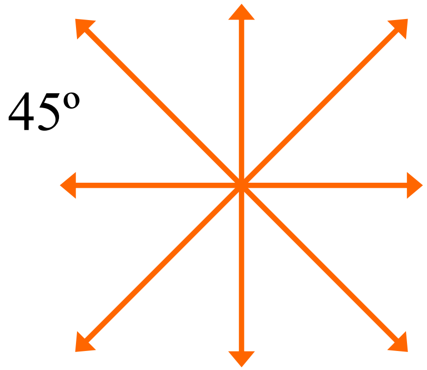


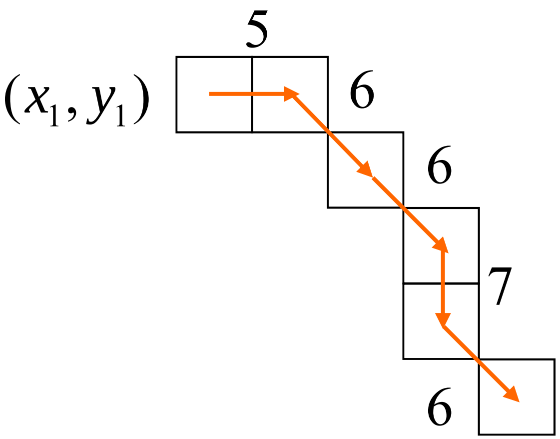
Contour length:

$$S = \sum_{i=2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

Representations: chain code

2	3	4
1	·	5
8	7	6





$$(x_1, y_1) \rightarrow 5, 6, 6, 7, 6$$

- Easy for rotations multiple of 45º
- Dervative is invariant to rotations

Curve fitting

- Types of fitting:
  - Interpolation: Curve goes through all points
  - Approximation: Curve pass near to points but not necessarily through them

- If all points are considered valid:

  Curve model:
  - Linear segments
  - Circular arcs, conic sections, cubic "*splines*", …

- If there are "outliers":

  Robust regression:
  - Least Median Squares
  - Ransac

- Model selection:
  - Depending on application

Error measures

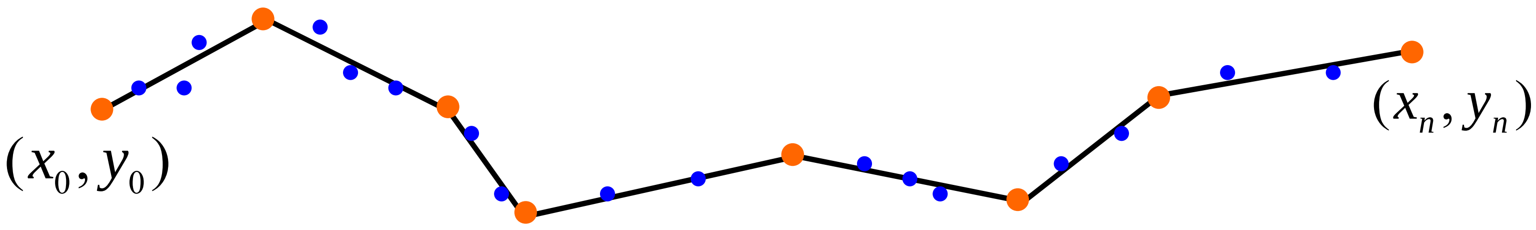
Distance of point i to the curve d_i

Maximum Absolut Error $\mathrm{MAE} = \max_i |d_i|$

Normalized Maximum Error $\varepsilon = \dfrac{\max_i |d_i|}{S}$

Mean Square Error $\mathrm{MSE} = \dfrac{1}{n} \sum_{i=1}^{n} d_i^2$

Linear segments



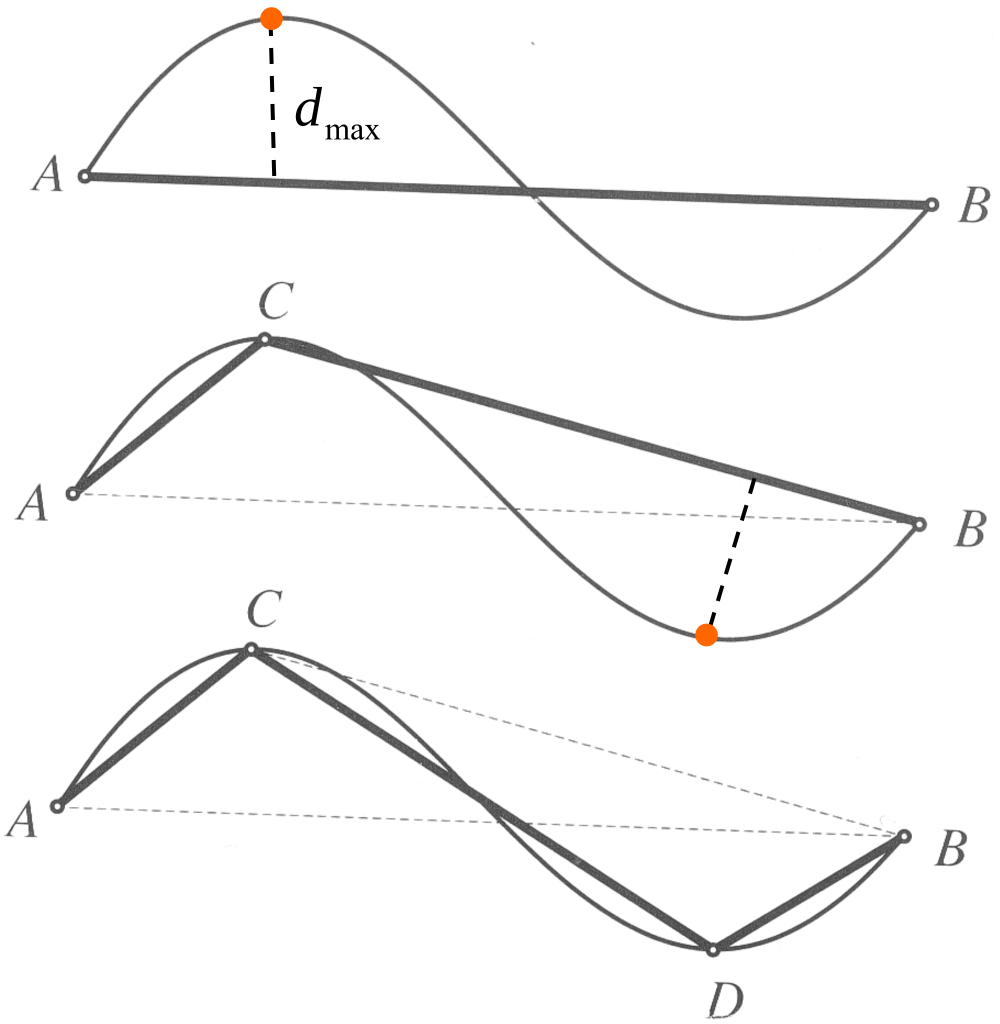
For two vertices:

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

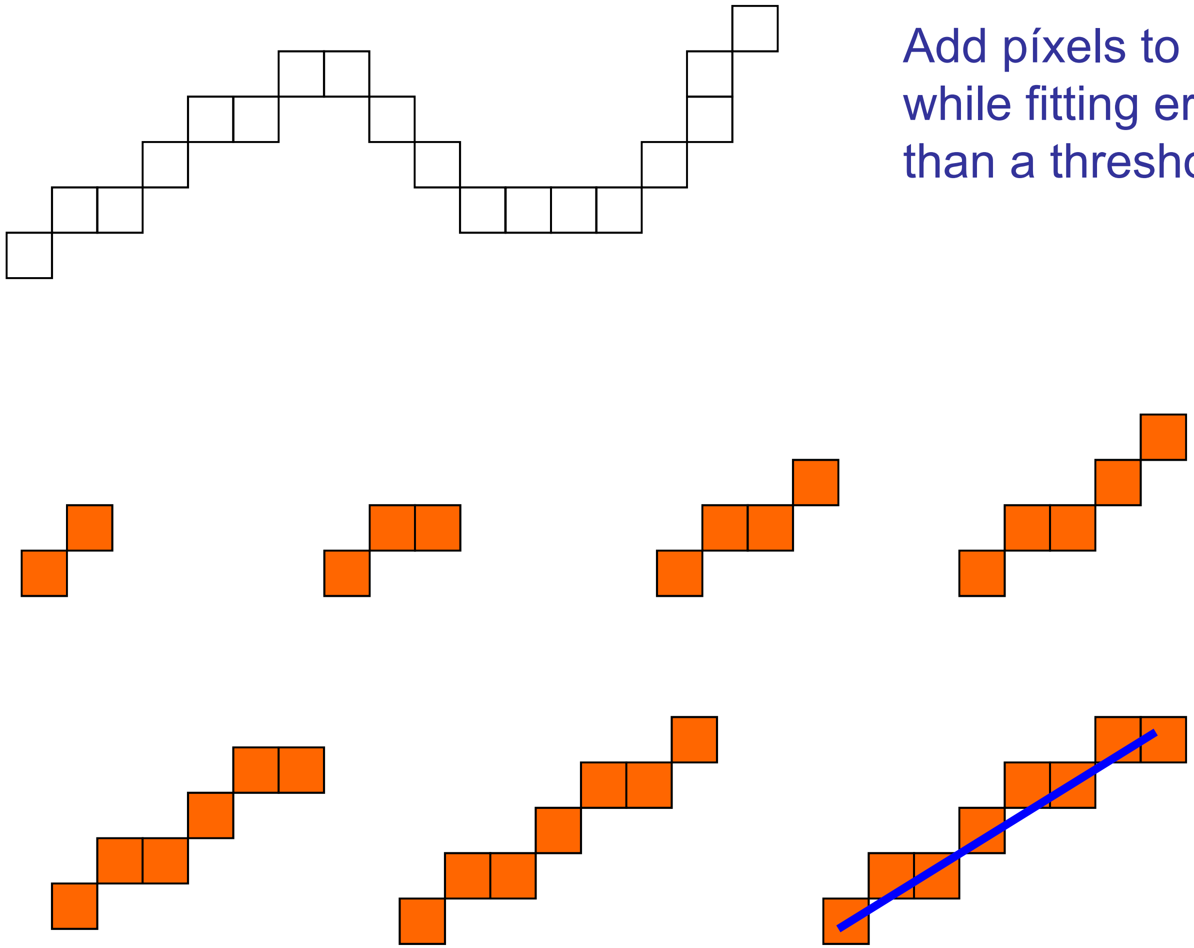
$$x\underbrace{(y_1 - y_2)}_{a} + y\underbrace{(x_2 - x_1)}_{b} + \underbrace{y_2 x_1 - y_1 x_2}_{c} = 0$$

$$a\,x + b\,y + c = 0$$

Vertices selection (Top-down)



Vertices selection (Bottom-up)



Add píxels to the segment while fitting error is less than a threshold

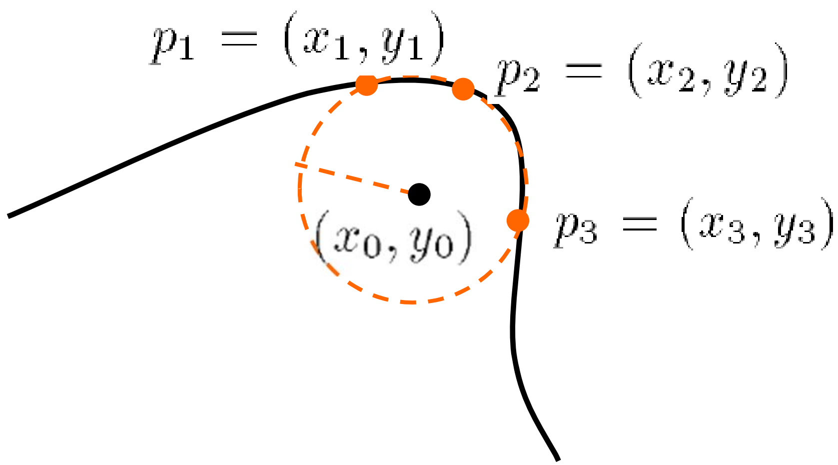
Split & merge

- Algorithm:
  - Top-down division
  - Union of adjacent segments using the “normalized error” measure
  - Repeat until no changes

- Observations:
  - After a union step, a segment could be divided at a different point
  - The use of normalized error in the union step allows merging into a single segment

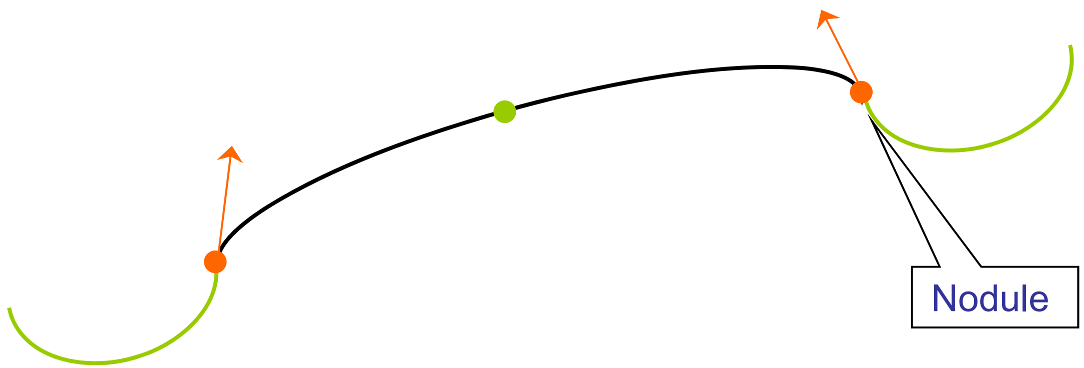
Circular arcs

Circumference: $$(x - x_0)^2 + (y - y_0)^2 = r^2$$



Conic sections

$$f(x, y) = ax^2 + 2hxy + by^2 + 2ex + 2gy + c = 0$$



- Each conic section is defined by:
  - 2 points (nodules)
  - 2 tangents (at the nodules)
  - 1 additional point of the curve

Cubic *splines*

- *Spline*:
  - Piecewise curve of any type of function
- Cubic *spline*: order 3 polynomial
  - Very much used
  - It enforces continuity of the tangent at the nodules

$$p(u) = (x(u), y(u)) = a_0 + a_1 u + a_2 u^2 + a_3 u^3$$

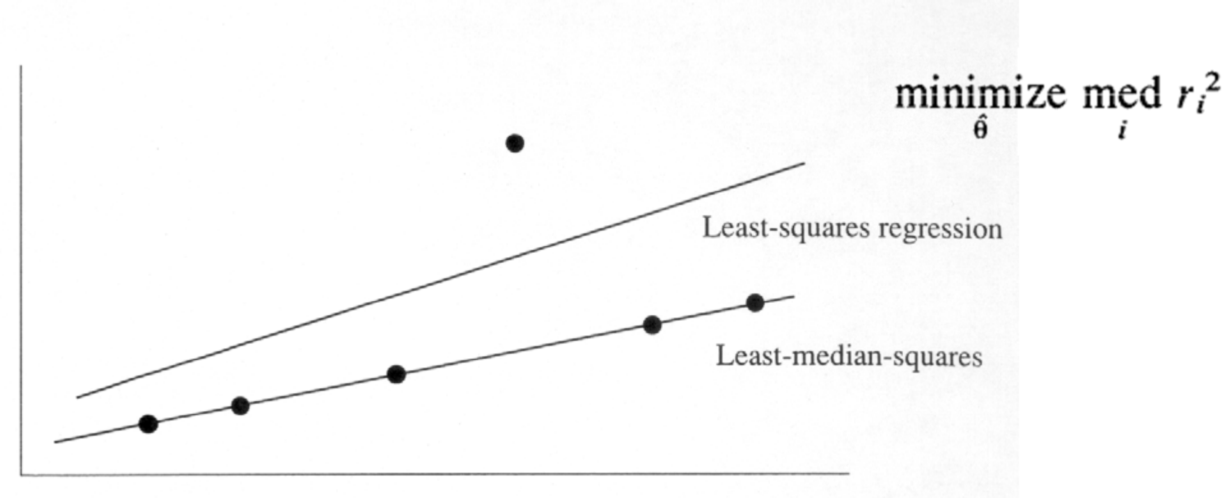
$$\begin{cases} u \in [0,1] \\ a_0, a_1, a_2, a_3 \text{ are vectors} \qquad (a_{ix}, a_{iy}) \end{cases}$$

Regression

- Requires:
  - Definition of a model
  - Definition of an error mesure between samples and model
  - Solution ➡ Error minimization
- Example: Linear regression minimizing Least Square Error (LSE).
- Problem: One or several “outliers” can affect the fitting in a significant way

Robust regression

- Try several subsets of points puntos and choose the one with better result (minimum error).
- Least Median Squares Regression:
  - Choose a random subset of points
  - Fit the model to the point subset
  - Calculate the median of the square errors
  - Repeat until the error is less than a threshold or until the maximum number of iterations is reached



UNIVERSITAT JAUME•I

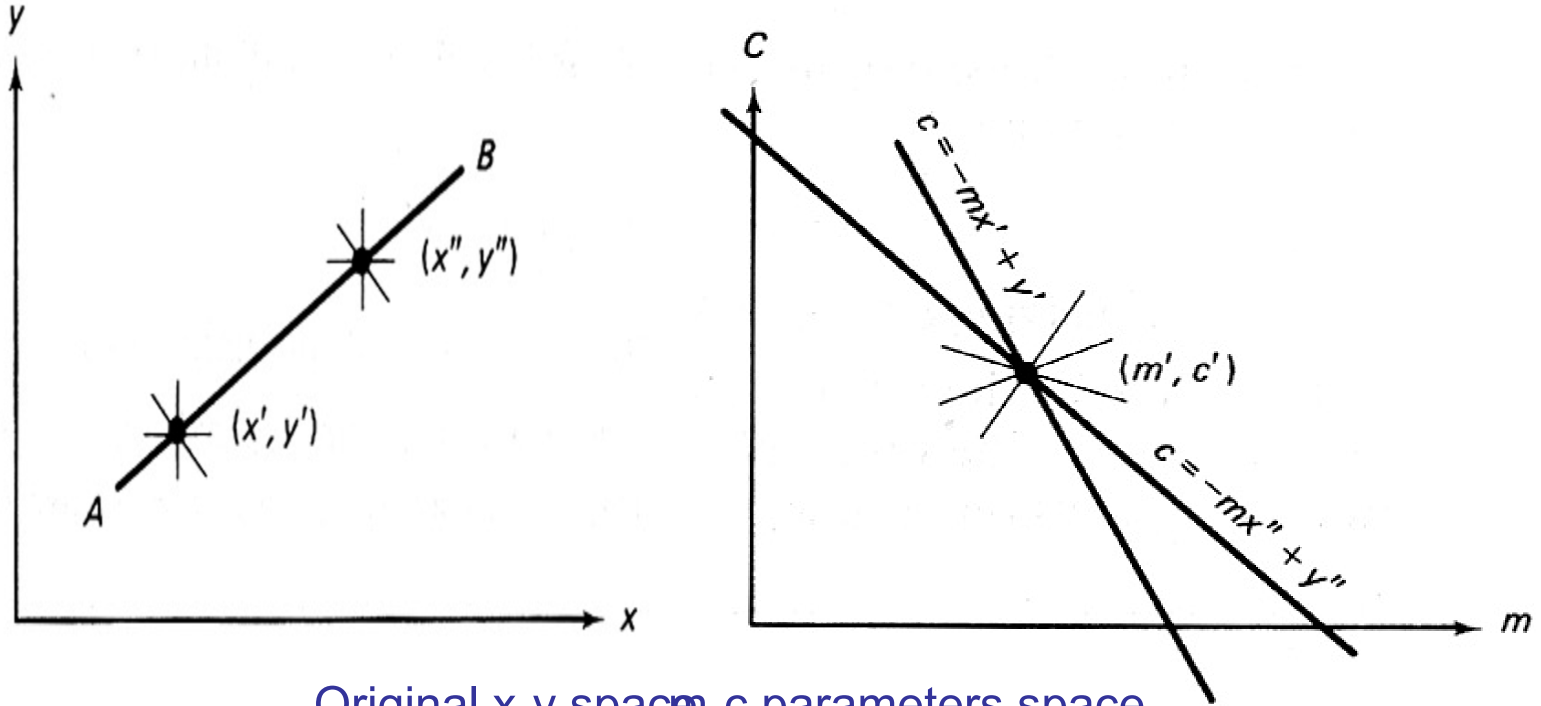
Ransac

- Ransac (Random sample consensus) algorithm:
  - Choose a random subset of points
  - Fit the subset of points to the chosen model
  - Calculate the number *K* of samples such that they fit to the model according to an error threshold
  - Estimate the fitting error using the *K* inliers
  - Repeat until *K* is big enough (success) or until the maximum number of iterations is reached (fail)

Hough transform

$$y = mx + c$$

$$c = -mx + y$$



Original x-y space

m-c parameters space

Line ➡ Point

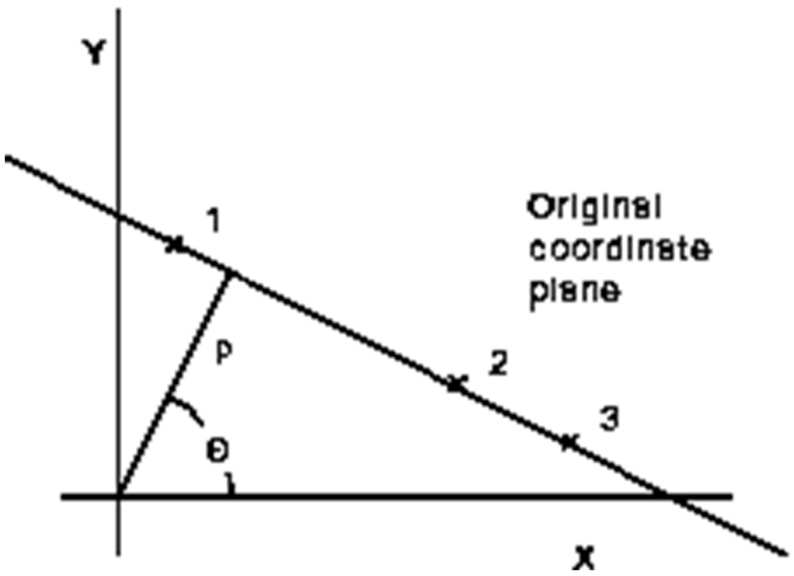
Point ➡ Line

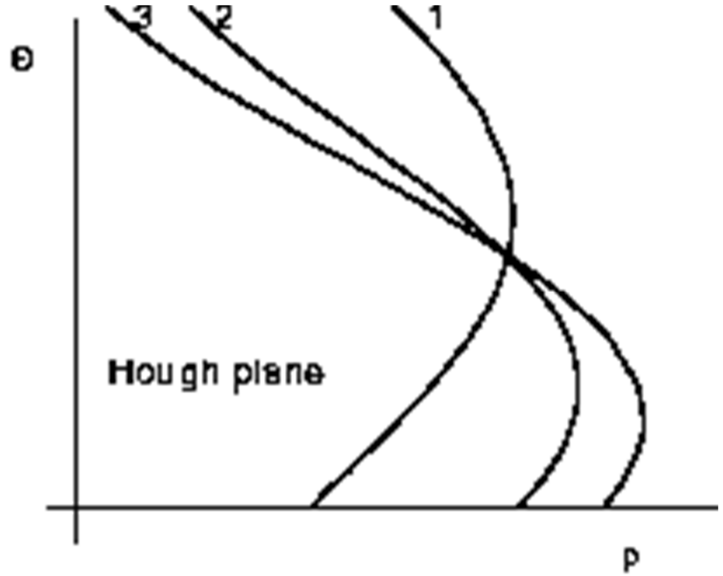
Hough transform

Problem: $m \in [-\infty, \infty]$

Solution: use normal form of a line equation

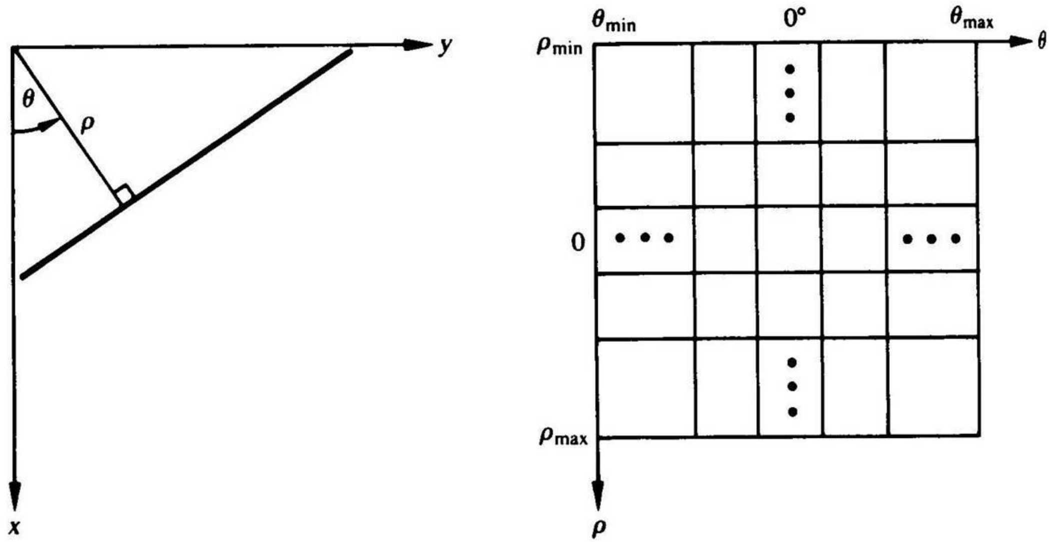
$$x\cos\theta + y\sin\theta = \rho$$



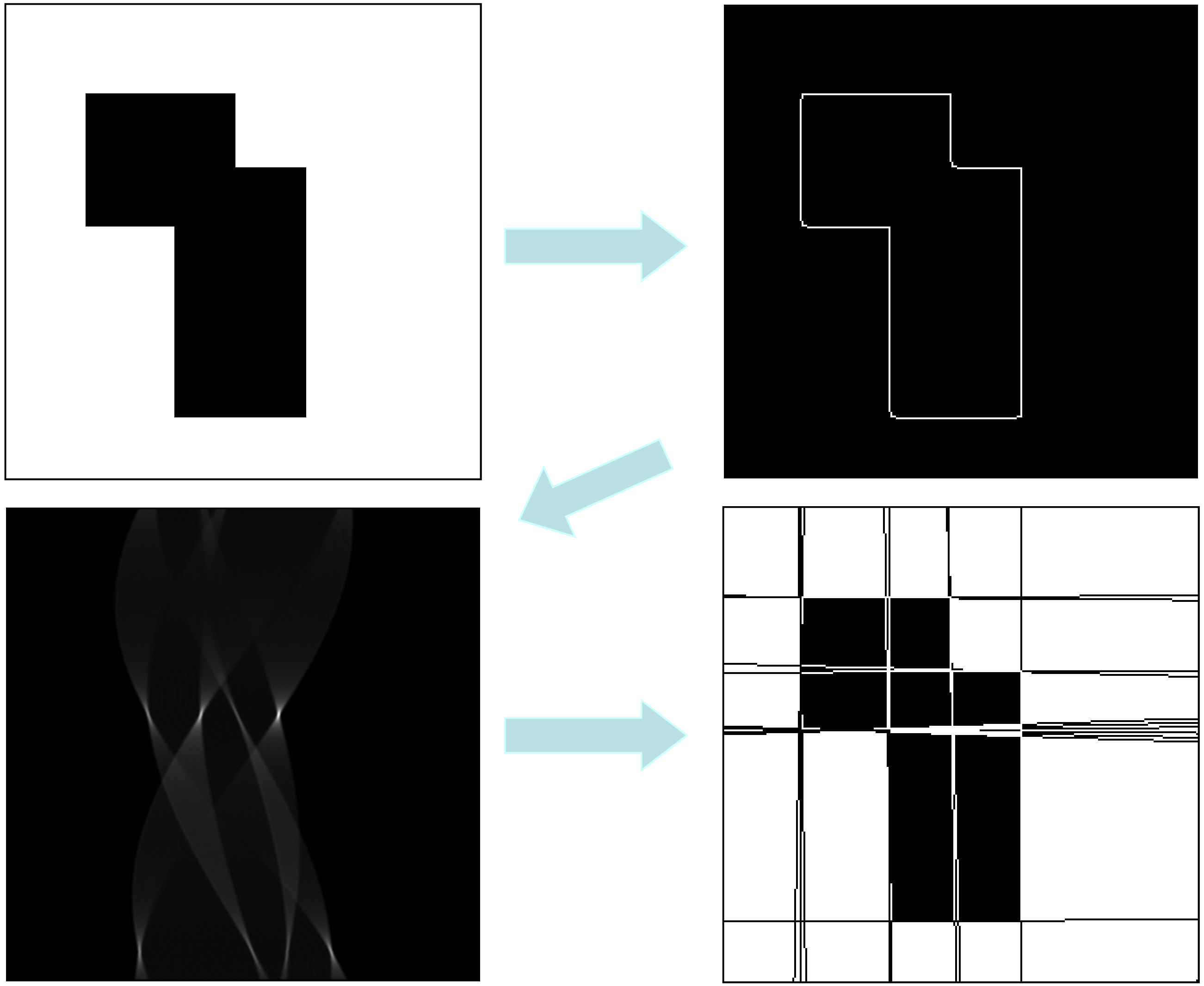


Hough transform: algorithm

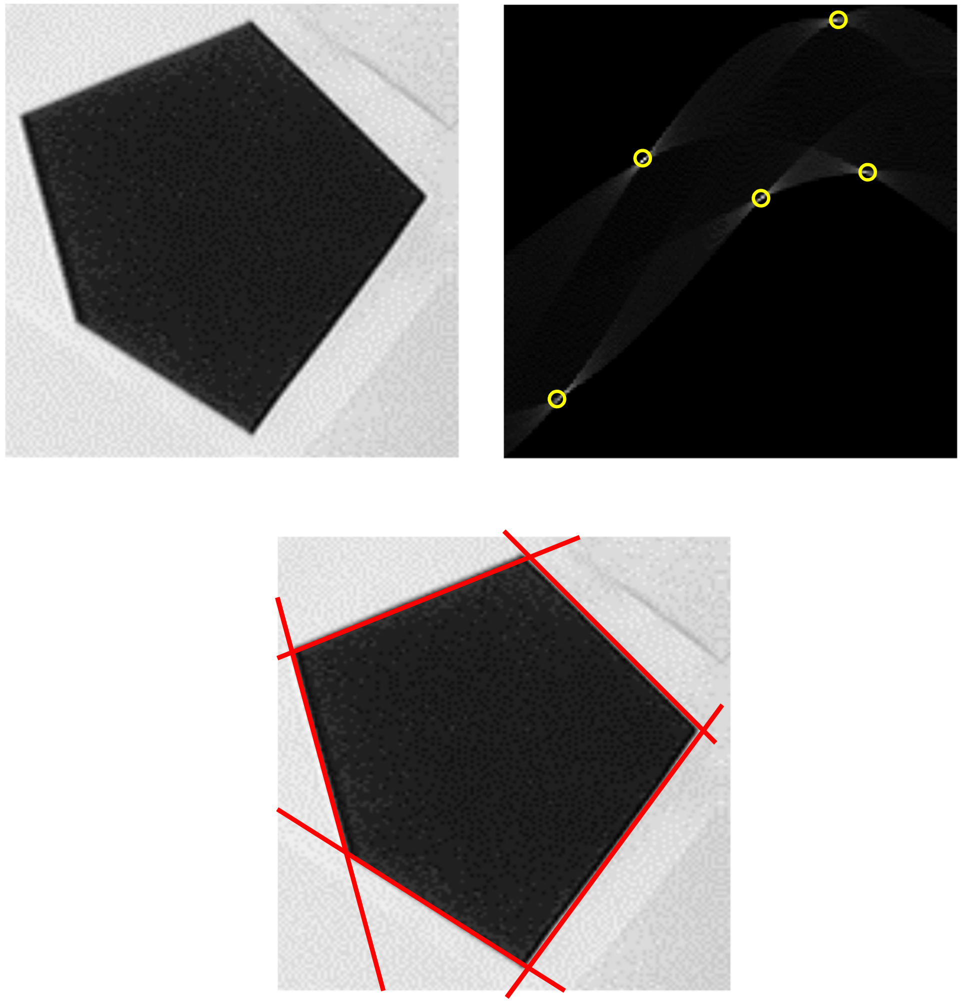
- Discretize the parameter space (rho-theta)
- Consider each cell as an accumulator initialized to zero
- For each *edgel* (x,y) increment the cell accumulators that hold the previous equation
- The cell accumulators with máximum values define the parameters of the model



Hough transform: detection process



Hough transform: example

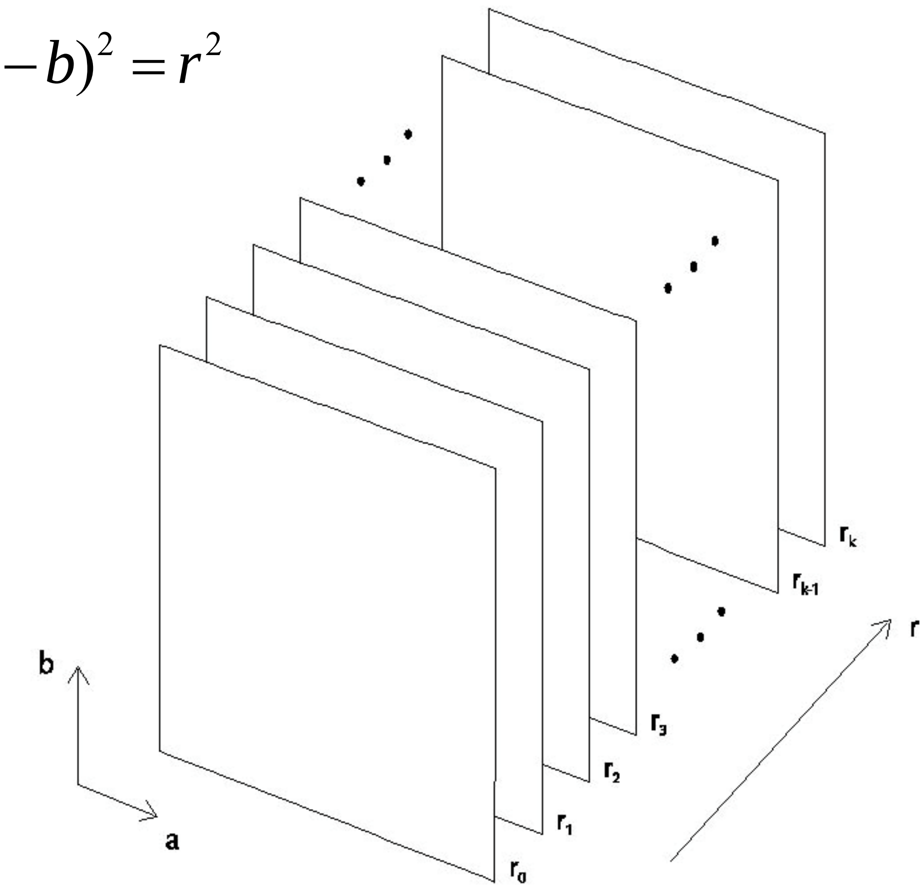


Hough transform: other geometries

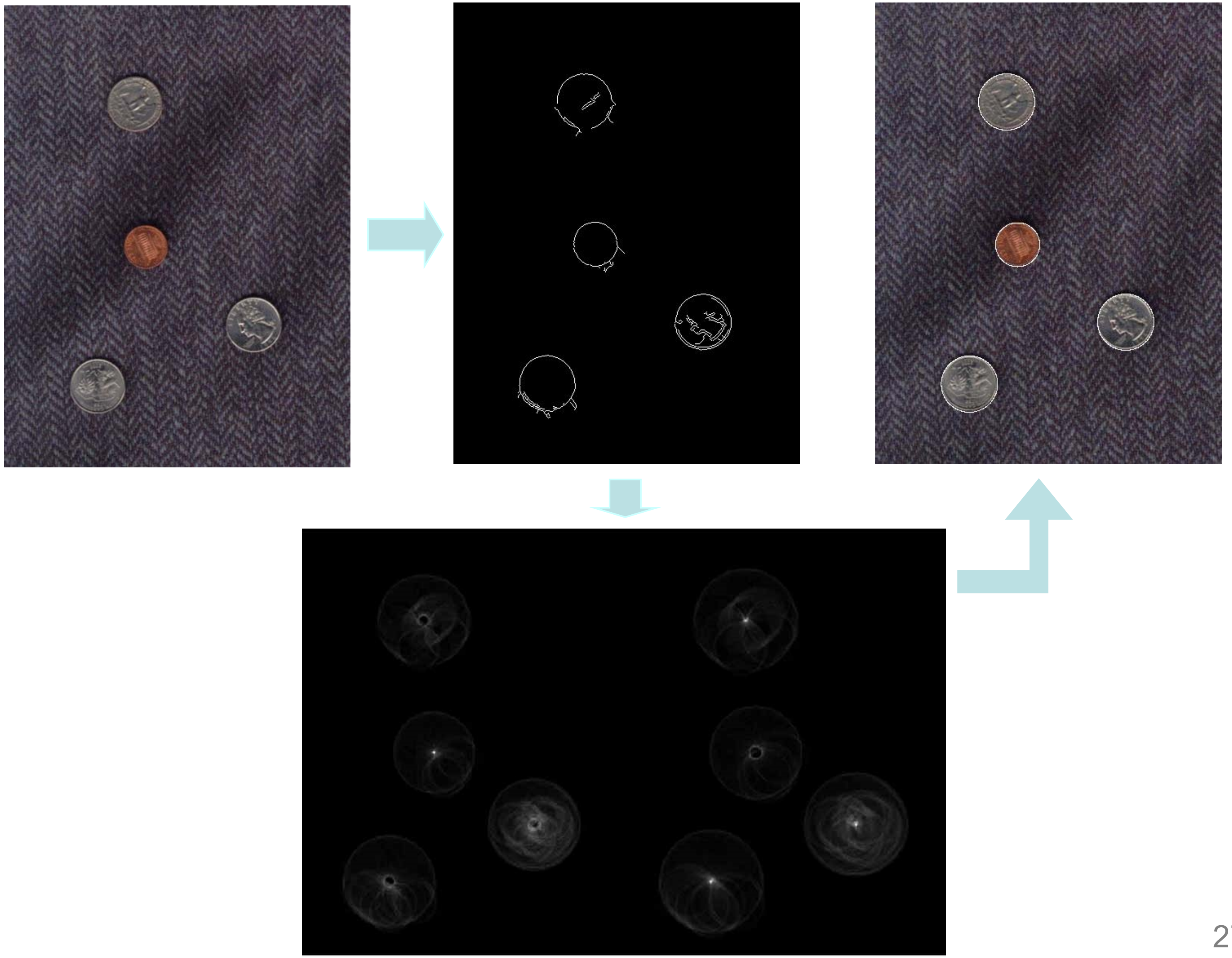
- Circumference $(x-a)^2+(y-b)^2=r^2$
- Ellipsis $\dfrac{(x-x_0)^2}{a^2}+\dfrac{(y-y_0)^2}{b^2}=1$
- …
- Any type of curve that can be analytically expressed

Círcumferences: *naïve* method

$$(x-a)^2+(y-b)^2=r^2$$



Circumferences: example



Ellipsis detection

Original

Borders

Detected ellipsis (in white)

Hough transform

- Advantages:
  - Robust to noise and occlusions
  - Robust to presence of other forms
  - Detection of multiple instances at the same time

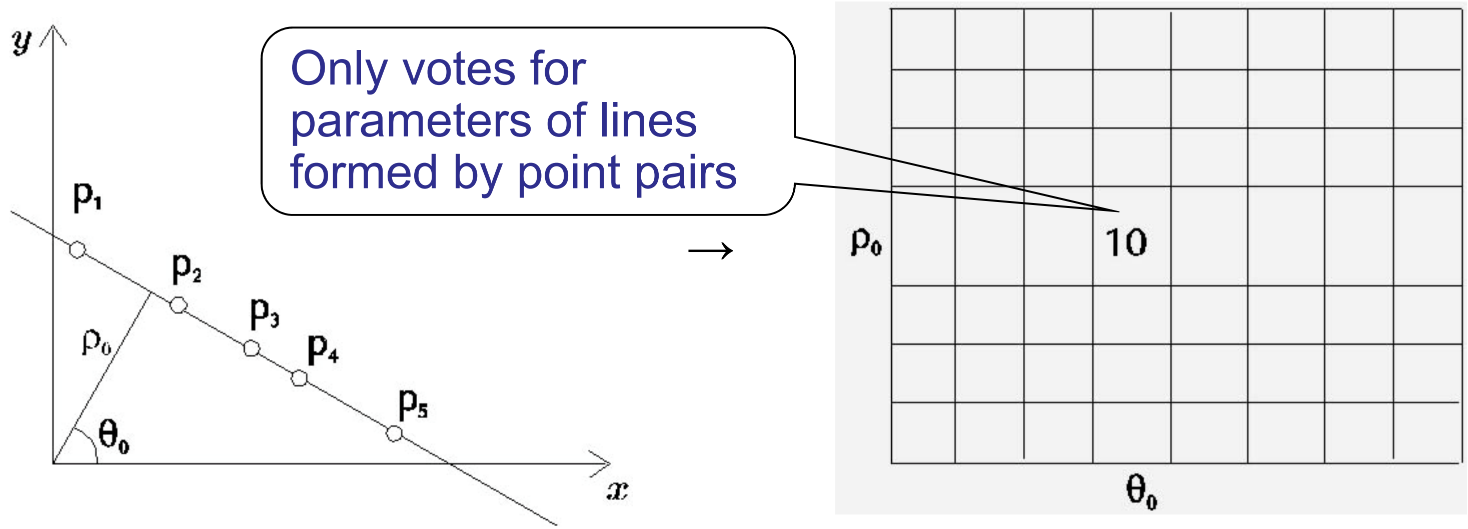
- Disadvantages:
  - Detection of false positives
  - Computational cost
  - Resolution of accumulator space
  - Peak localization

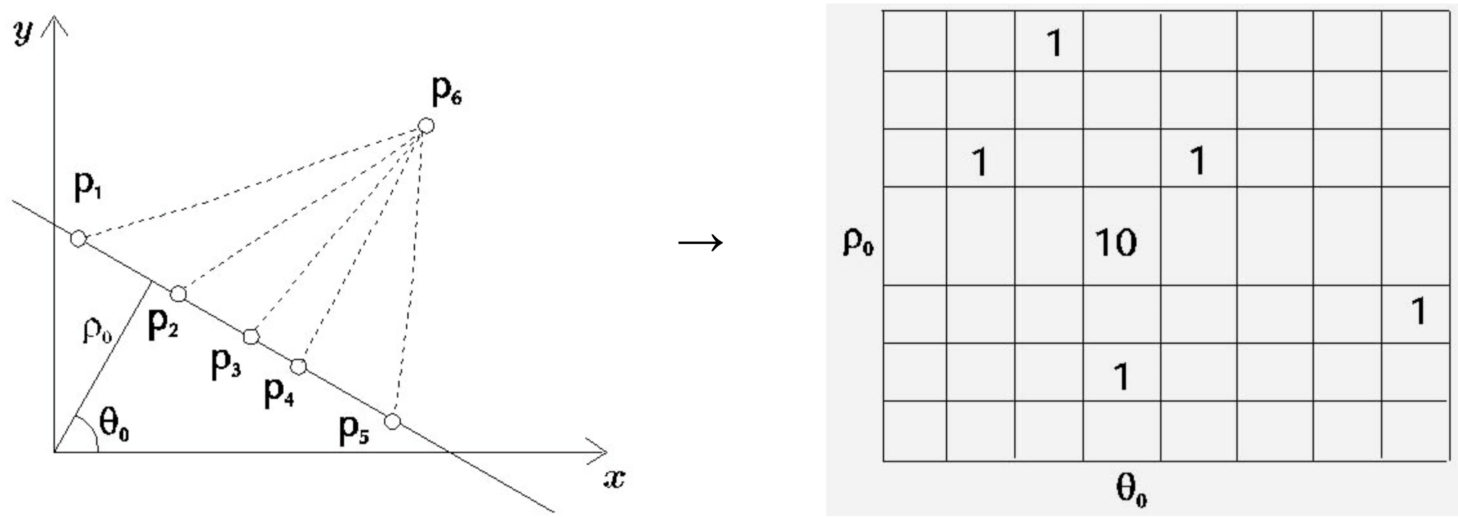
Hough transform: computational cost

- Computational cost depends on:
  - Image size
  - Accumulator space size
    - Nº parameters and resolution
  - Number of *edgels* and amount of noise

- Some strategies to reduce cost:
  - Pre-calculate sinus and co-sinus values
  - Multi-resolution: start with little resolution
  - Divide image into sub-images
  - Combinatorial Hough transform
  - Parallelize the implementation

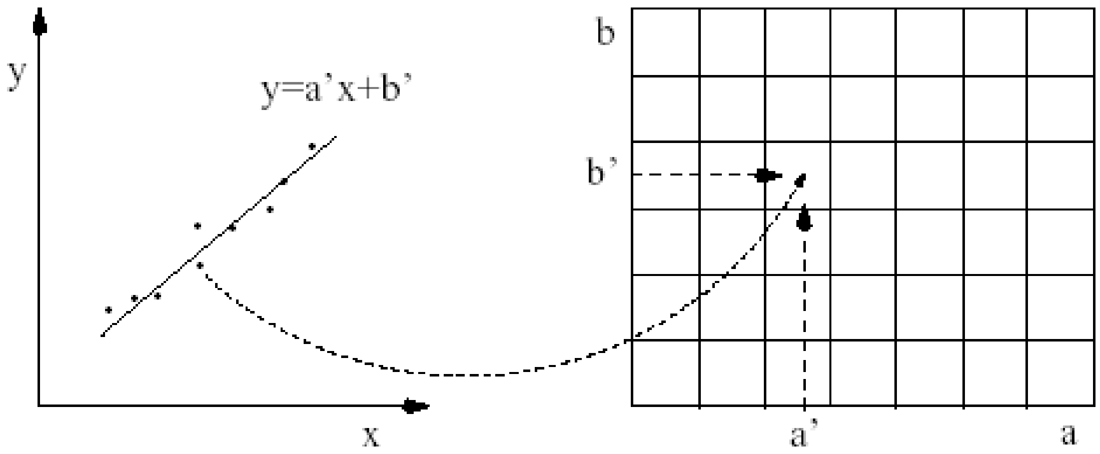
Combinatorial Hough transform



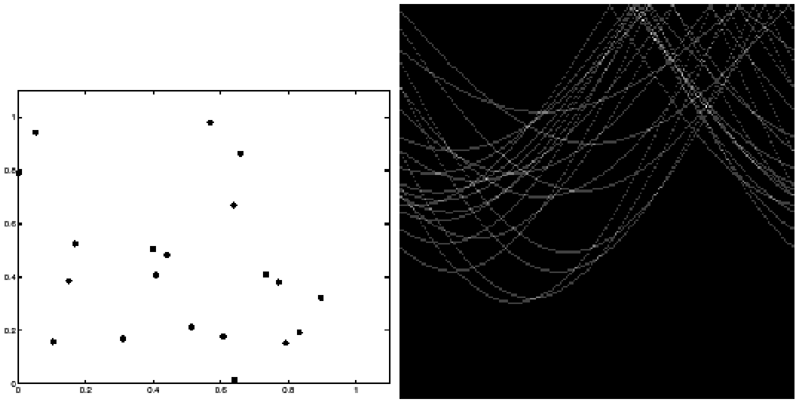


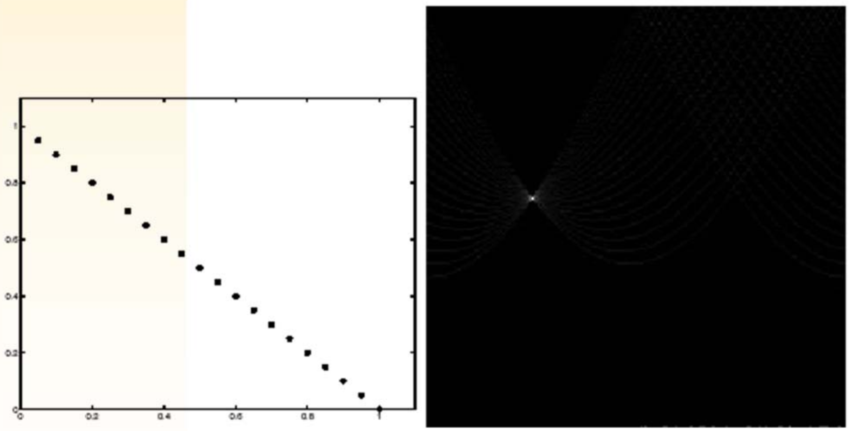
Accumulator space size

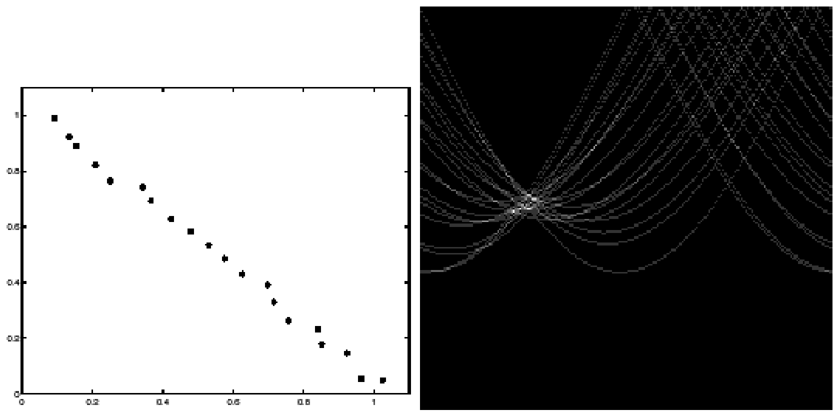
- What is the appropriate size of the accumulator space?
  - Too small:
    - Low precision
    - More tolerance to noise
  - Too big
    - A lot of computational resources



Peak sparsity







Peak detection

- Smooth accumulator space before peak search
- Use clustering techniques
- “Eliminate” detected peaks after each iteration
- How many peaks? Which ones are “true” peaks?
  - Set a threshold for cell votes
  - Prior knowledge
  - Problem constrains

Detection of line *segments*

- There are two problems after the line detection:
  - Each peak represents a line, *not a segment*
  - Estimated/found parameters are according to *accummulator resolution*

- Strategies:
  - For each Hough space cell, keep *edgels*, not only votes
  - "Explore" image near the line

HT: complete example