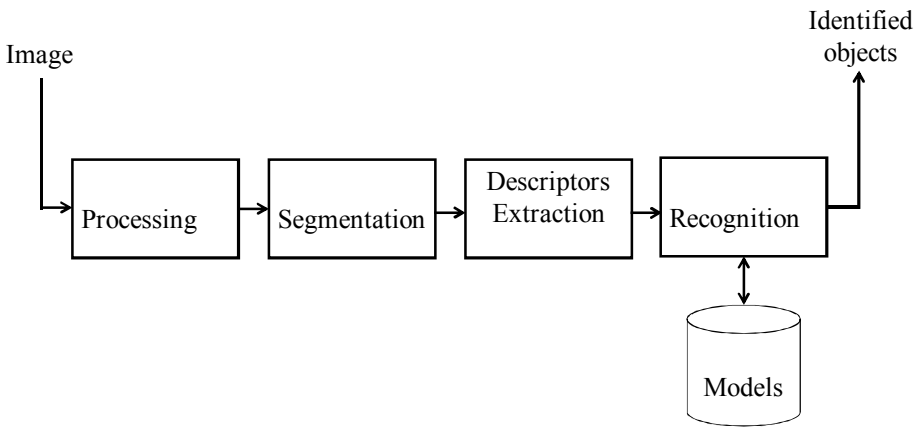
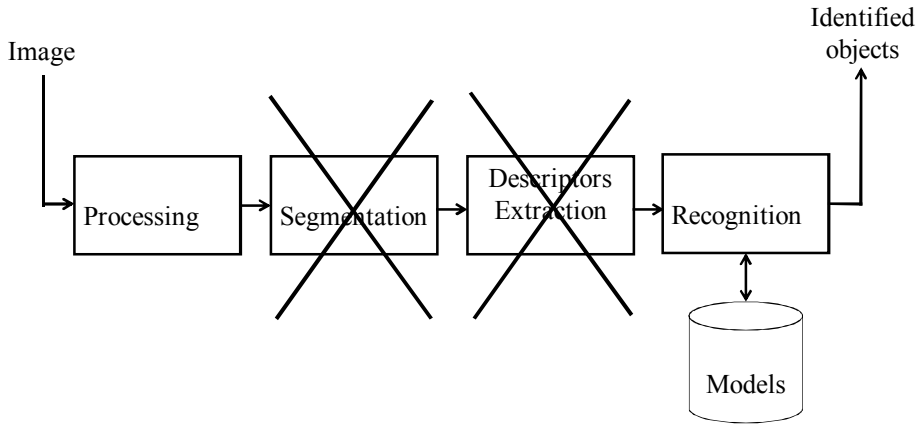
Local Features

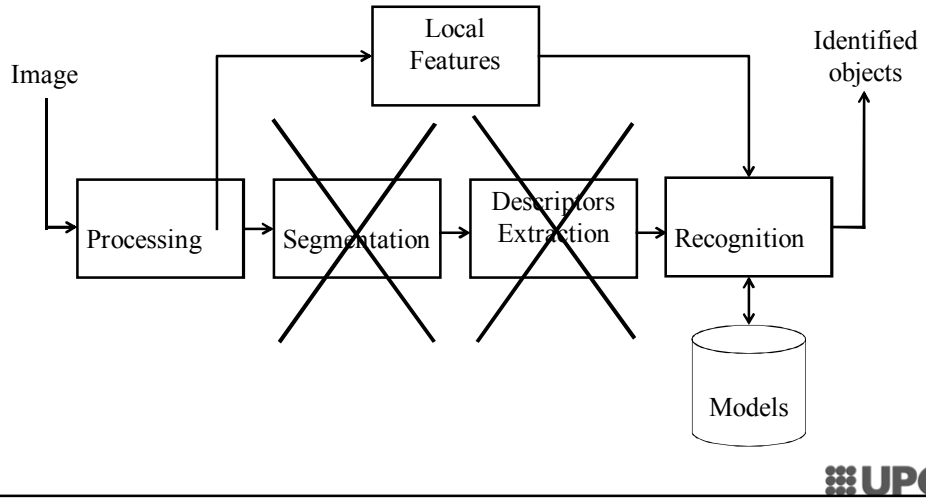
A computer vision system



A computer vision system



A computer vision system



Local Features

- Histograms
- Hough transform
- Corners
- Scale Invariant Feature transform (SIFT)
- Local Binary Patterns (LBP)
- Haar Features (face detection)

Local Features

- Global features are sensitive to noise and occlusion.
- An alternative is to use local features:
  - If some features are occluded, the object might still be recognized by identifying some visible features

Corners are Simple Features

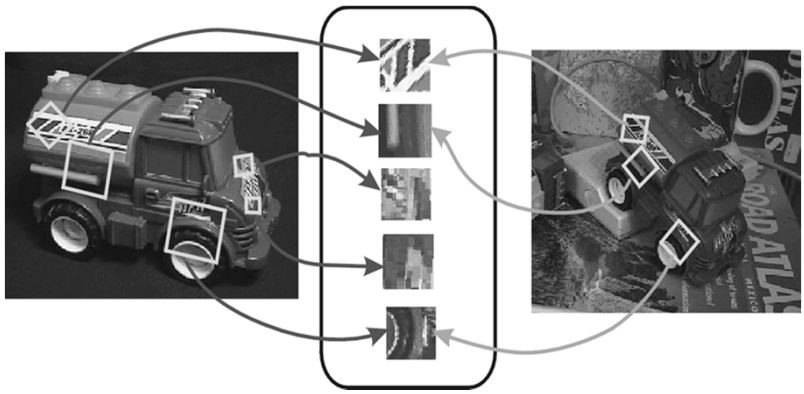
↘ Can you see common feature point between the two images?

UPC

Invariant local features

Find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, …



Feature Descriptors

Advantages of local features

Locality

- features are local, so robust to occlusion and clutter

Distinctiveness:

- can differentiate a large database of objects

Quantity

- hundreds or thousands in a single image

Efficiency

- real-time performance achievable

Generality

- exploit different types of features in different situations

Properties of 'good' features

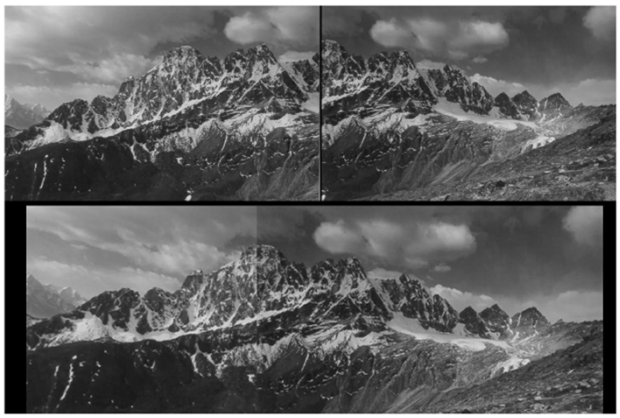
- Repeatability
  - Should be detectable at the same locations in different images despite changes in viewpoint and illumination
- Saliency (descriptiveness)
  - Same points in different images should have similar features
  - And vice-versa
- Compactness (affects speed of matching)
  - Fewer features
  - Smaller features

Applications of feature matching

- Image alignment and stitching
- Image retrieval
- Motion tracking
- Robot navigation
- Face detection
- Object recognition
- Gesture recognition
- Human action understanding
- Biometric identification

Applications of feature matching

Image Stitching (to generate panorama)



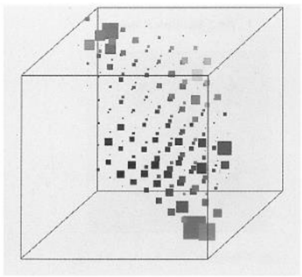
Applications of feature matching

Colour histograms

- Colour stays constant under geometric transformations
- Colour is a local feature
  - It is defined for each pixel
  - It is robust to partial occlusion
- Idea:
  - can use object colours directly for recognition, or
  - better – use statistics of object colours
- Colour histogram is a type of appearance features

Colour histograms

- Colour histograms are colour statistics
  - Here, RGB as an example
  - Given: tristimulus R, G, B for each pixel
  - Compute a 3D histogram
  - $h(R,G,B) = \#(\text{pixels with colour } (R,G,B))$



Colour histograms

Colour Normalization

↘ One component of the 3D colour space is intensity

- If a colour vector is multiplied by a scalar, the intensity changes but not the colour itself.
- This means colours can be normalized by the intensity.
- Note: intensity is given by $I = (R + G + B)/3$
- Chromatic representation:

$$r = \frac{R}{R+G+B} \qquad g = \frac{G}{R+G+B} \qquad b = \frac{B}{R+G+B}$$

Since $r + g + b = 1$, only 2 parameters are needed to represent colour (knowing r and g, we can deduce $b = 1 - r - g$).

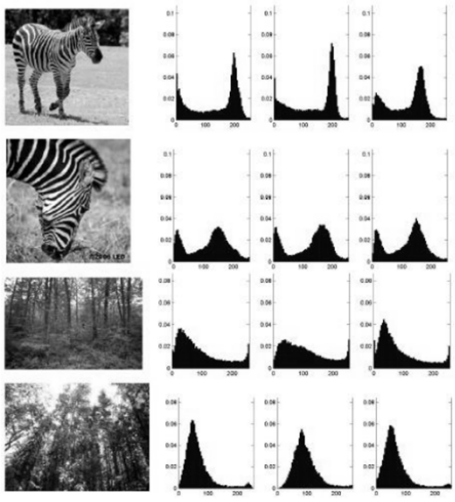
⇒ Can compute colour histogram using r, g, and b instead.

Colour histograms

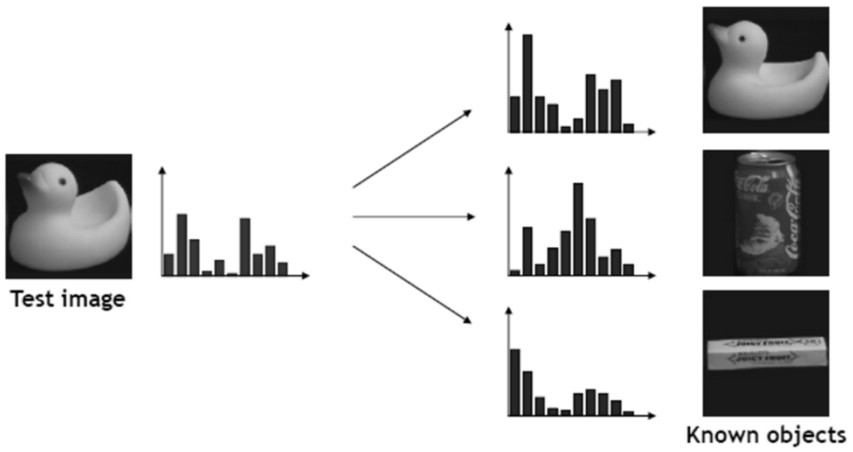
Object Recognition based on Colour Histograms

Colour histograms

- are discrete approximation of the colour distribution of an image.
- contain no spatial information ⇒ invariant to translation, scale, and rotation



Colour histograms



Object recognition using histograms

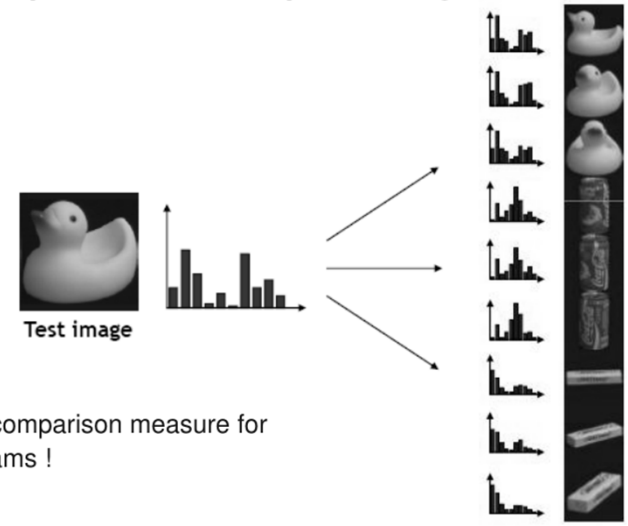
- Simple algorithm

1. Build a set of histograms $H = \{\boldsymbol{h}_i\}$ for each known object.
   - More exactly, for each view of each object.
2. Build a histogram $\boldsymbol{h}_t$ for the test image.
3. Compare $\boldsymbol{h}_t$ with each $\boldsymbol{h}_i \in H$ using a suitable histogram comparison measure.
4. Select the object with the best matching score;
   or reject the test image if no object is similar enough.

This is known as the "**nearest-neighbour**" strategy.

Colour histograms

Histogram Comparison with Multiple Training Views

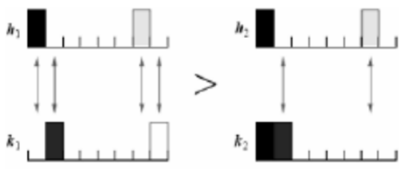


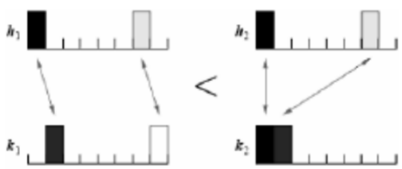
⇒ Need a good comparison measure for colour histograms !

Histogram comparison

What is a Good Comparison Measure?

↘ How to define matching cost?



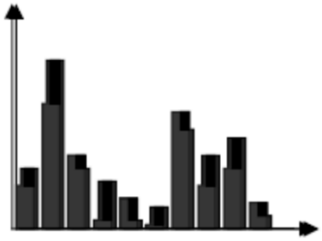


Histogram comparison

Euclidean distance (L_2 norm)

$$d(\boldsymbol{q},\boldsymbol{v}) = \sum_i (q_i - v_i)^2$$

- Motivation of the Euclidean distance:
  - Focuses on the differences between the histograms.
  - Interpretation: distance in the feature space.
  - Range: $[0,\infty)$.
  - All cells are weighted equally.
  - Not very robust to outliers !



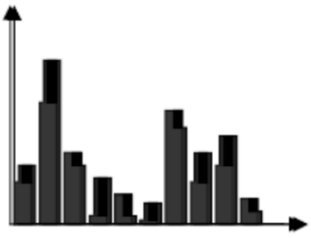
UPC

Histogram comparison

Chi-Square distance:

$$d(\boldsymbol{q},\boldsymbol{v}) = \sum_i \frac{(q_i - v_i)^2}{q_i + v_i}$$

- Motivation of the χ^2 distance:
  - Statistical background
  - Test if two distributions are different.
  - Possible to compute a significance score.
  - Range: $[0,\infty)$.
  - Cells are not weighted equally !
  - More robust to outliers than the Euclidean distance, if the histograms contain enough observations…



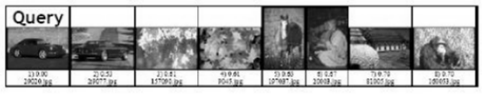
UPC

Histogram comparison

- Which measure is the best?
  - It depends on the application
  - Euclidean distance is often not robust enough.
  - Generally, χ^2 distance gives good performance for histograms
  - KL/Jeffreys divergence works well sometimes, but is expensive
  - EMD is the most powerful, but also very expensive.

Colour histograms for image retrieval

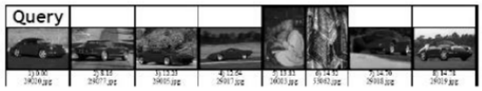
- The image retrieval problem concerns the retrieval of those images in a database that best match a query image.



L2 distance

Jeffrey divergence

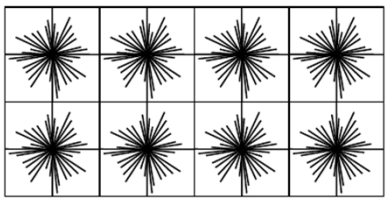
χ^2 distance



Earth Movers Distance

Histograms of oriented gradients (HOGs)

- Objective: object recognition
- Basic idea
  - Local shape information often well described by the distribution of intensity gradients or edge directions even without precise information about the location of the edges themselves.

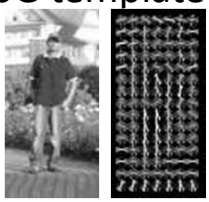


HOG algorithm

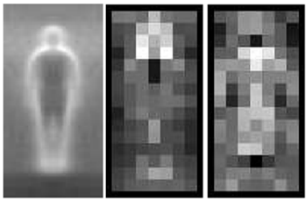
- Divide image into small sub-images: "cells"
- Accumulate a histogram of edge orientations within that cell
- The combined histogram entries are used as the feature vector describing the object
- To provide better illumination invariance (lighting, shadows,etc.) normalize the cells across larger regions incorporating multiple cells: "blocks"

Person detection, Dalal-Triggs 2005

1. Represent each example with a single, fixed HoG template



2. Learn a single [linear] SVM as a detector



Code available: http://pascal.inrialpes.fr/soft/olt/

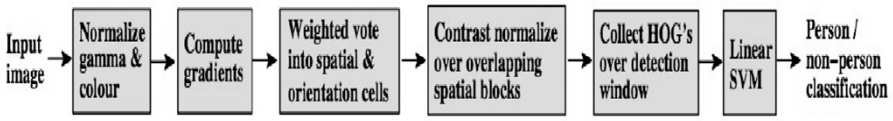
Positive and negative examples

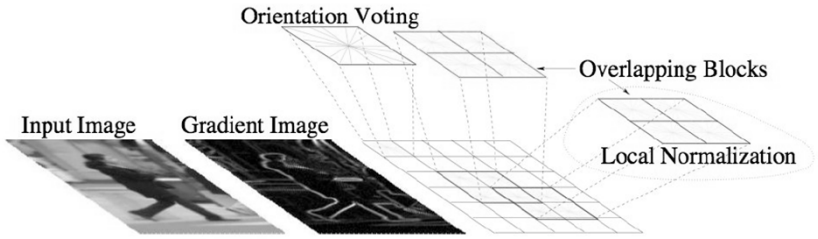
+ thousands more…

+ millions more…

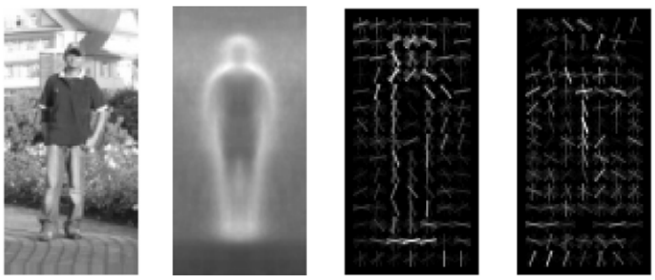
HOGs. Aplicacions

A worked example: Dalal and Triggs CVPR'05





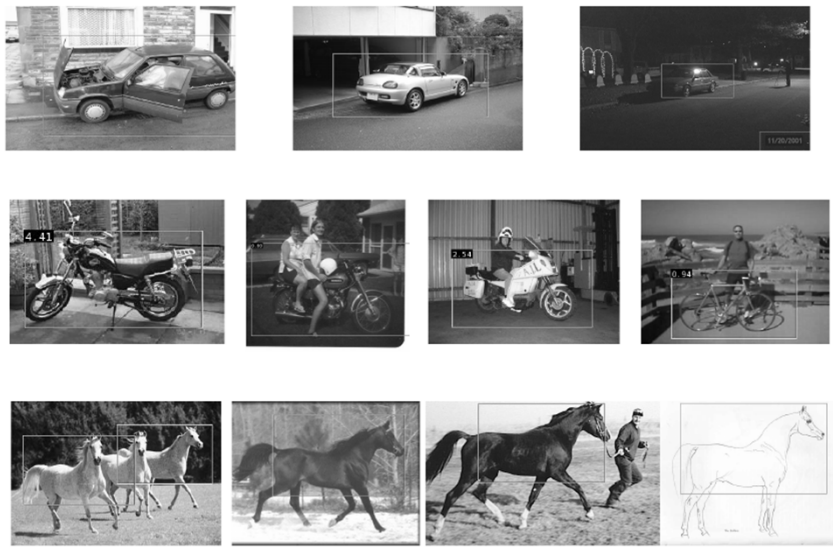
HOGs. Descriptor cues



Most important cues are head, shoulder, leg silhouettes

Vertical gradients inside a person are counted as negative

Applications to Other Classes

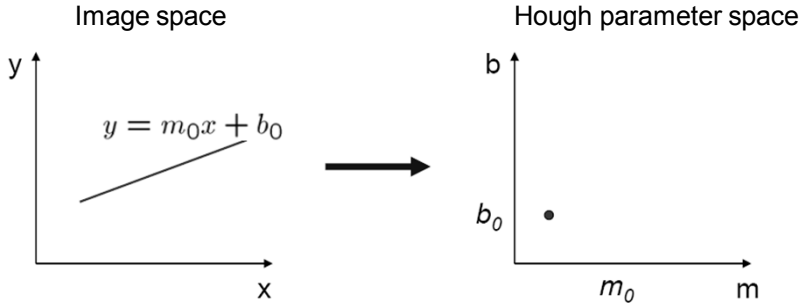


M. Everingham et al. *The 2005 PASCAL Visual Object Classes Challenge.* Proceedings of the PASCAL Challenge Workshop, 2006. 25

Transformada de Hough

Transformada de Hough

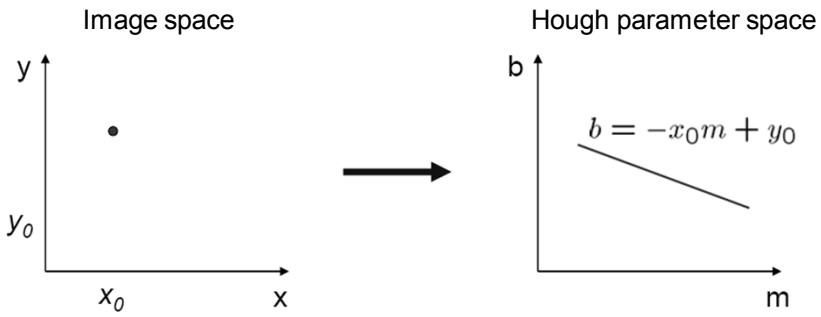
- Dissenyada originalment per a la detecció de rectes
- Molt robusta a soroll i imperfeccions
- Cada punt de la recta original es transforma en una recta en el pla transformat



UPC

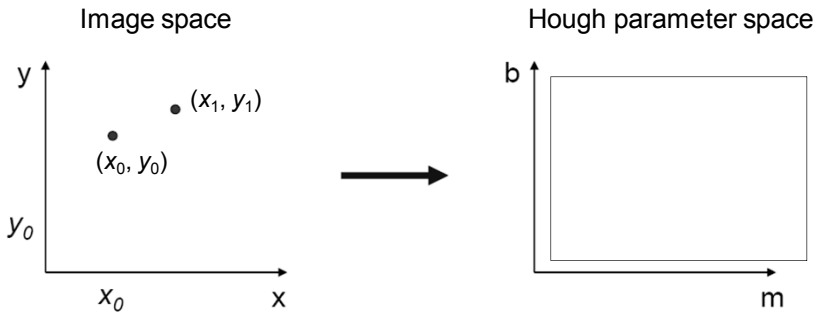
Parameter space representation

- What does a point (x_0, y_0) in the image space map to in the Hough space?
  - Answer: the solutions of $b = –x_0m + y_0$
  - This is a line in Hough space



Parameter space representation

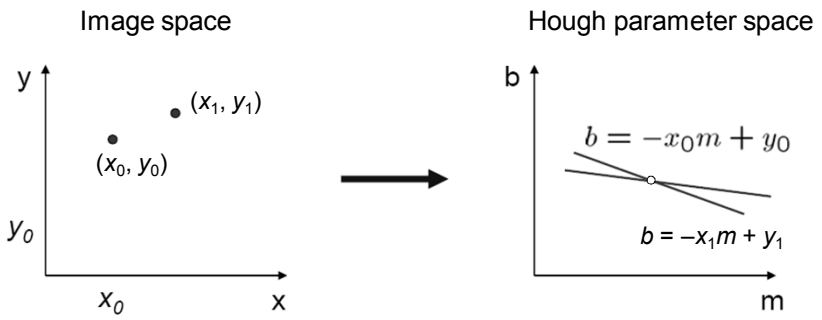
- Where is the line that contains both (x_0, y_0) and (x_1, y_1)?



Source: S. Seitz

Parameter space representation

- Where is the line that contains both (x_0, y_0) and (x_1, y_1)?
  - It is the intersection of the lines $b = –x_0m + y_0$ and $b = –x_1m + y_1$



Source: S. Seitz

Transformada de Hough

Algorisme:

-1. Crear la matriu de l'espai m-b i dimensionar-la.

-2. Cada posició de la matriu és un acumulador. Posar-los tots a 0.

-3. Per a cada píxel (x,y) de la imatge, incrementar les posicions de la matriu m-bque satisfan l'equació.

-4. Els màxims de la matriu m-b es corresponen a la presència de rectes en la imatge.

- Si hi ha diverses rectes a la imatge, tindrem diversos màxims a la matriu.

-Es difícil determinar si un màxim és determinant o no.

Transformada de Hough

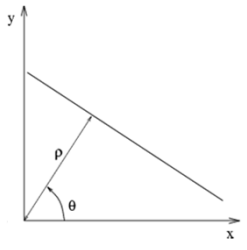
-Problema: afitar m i b.

-Solució: S'usa la forma polar de la recta:

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

- ara els punts (x,y) queden mapejats com funcions sinusoïdals en l'espai (r,α).

- r representa la llargada i α l'orientació del vector normal desde l'origen de la imatge a la línia.



Transformada de Hough

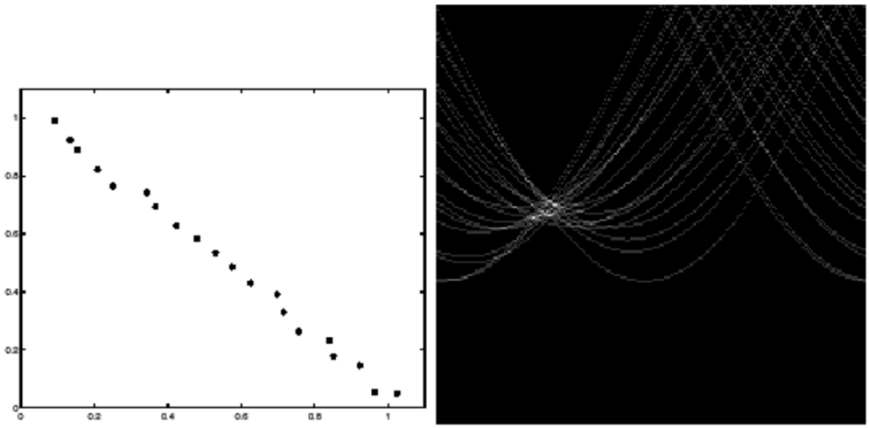
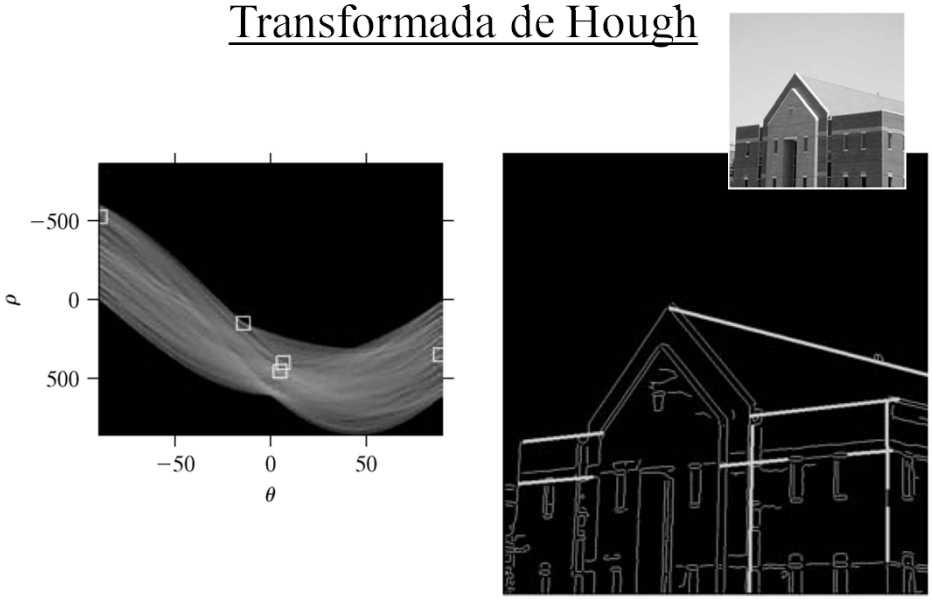
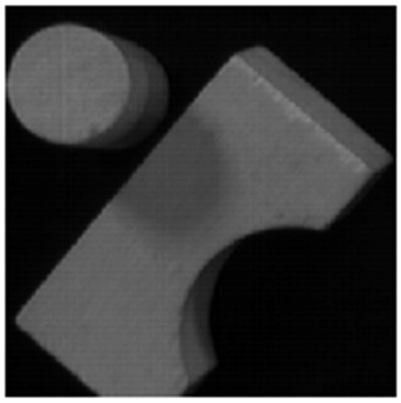


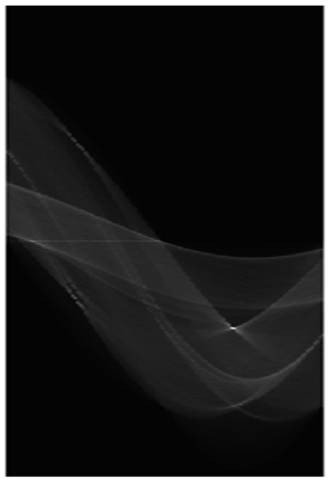
Image space **Votes**

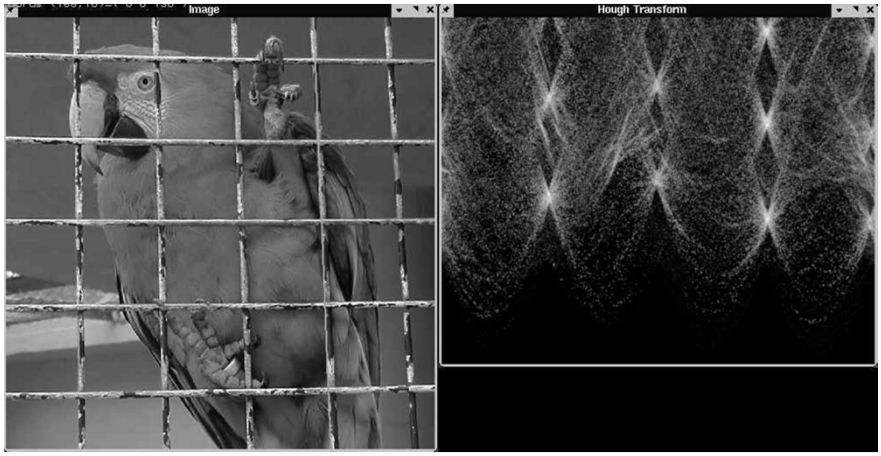
(Horizontal axis is θ, vertical is rho)

Transformada de Hough









http://ostatic.com/files/images/ss_hough.jpg

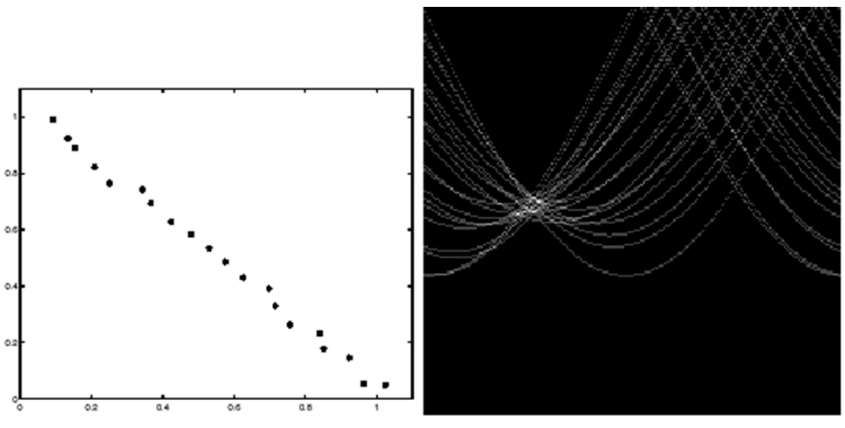
Transformada de Hough

Mechanics of the Hough transform

- Difficulties
  - how big should the cells be? (too big, and we merge quite different lines; too small, and noise causes lines to be missed)
  - small cell sizes => more computation!
  - large cell sizes => less accuracy!
- How many lines?
  - Count the peaks in the Hough array
  - Treat adjacent peaks as a single peak, i.e. must perform non-maximum suppression

UPC

Effect of noise



features votes

Peak gets fuzzy and hard to locate

Dealing with noise

- Choose a good grid / discretization
  - Too coarse: large votes obtained when too many different lines correspond to a single bucket
  - Too fine: miss lines because some points that are not exactly collinear cast votes for different buckets
- Increment neighboring bins (smoothing in accumulator array)
- Try to get rid of irrelevant features
  - Take only edge points with significant gradient magnitude

Hough Transform Algorithm

Input is an edge image (E(i,j)=1 for edgels)

1. Discretize θ and ρ in increments of $d\theta$ and $d\rho$. Let A(R,T) be an array of integer accumulators, initialized to 0.
2. For each pixel E(i,j)=1 and h=1,2,…T do
   1. $\rho = i \cos(h * d\theta) + j \sin(h * d\theta)$
   2. Find closest integer k corresponding to ρ
   3. Increment counter A(h,k) by one
3. Find local maxima in A(R,T)

PENNSTATE

Hough Transform Speed Up

- If we know the orientation of the edge – usually available from the edge detection step
  - We fix theta in the parameter space and increment **only one** counter!
  - We can allow for orientation uncertainty by incrementing a *few* counters around the "nominal" counter.

PENNSTATE

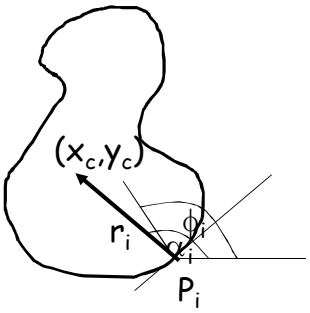
Hough Transform for Curves

- The H.T. can be generalized to detect any curve that can be expressed in parametric form:
  - Y = f(x, a1,a2,…ap)
  - a1, a2, … ap are the parameters
  - The parameter space is p-dimensional
  - The accumulating array is LARGE!

PENNSTATE

Generalizing the H.T.

The H.T. can be used even if the curve has not a simple analytic form!



1. Pick a reference point (x_c,y_c)
2. For i = 1,…,n :
   1. Draw segment to P_i on the boundary.
   2. Measure its length r_i, and its orientation α_i.
   3. Write the coordinates of (x_c,y_c) as a function of r_i and α_i
   4. Record the gradient orientation ϕ_i at P_i.
3. Build a table with the data, indexed by ϕ_i .

$x_c = x_i + r_i\cos(\alpha_i)$

$y_c = y_i + r_i\sin(\alpha_i)$

PENNSTATE

Transformada de Hough

Finding Circles by Hough Transform

- Equation of Circle:

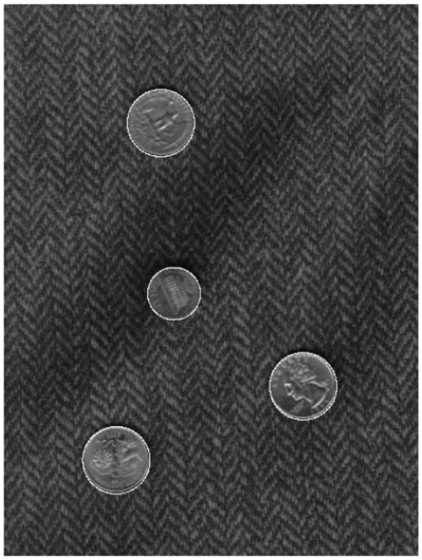
$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- If radius is known: (2D Hough Space)
  - Accumulator Array $A(a,b)$
- If radius is not known: 3D Hough Space!
  - Use Accumulator array $A(a,b,r)$

UPC

Transformada de Hough

Finding Coins



Note that because the quarters and penny are different sizes, a different Hough transform (with separate accumulators) was used for each circle size.