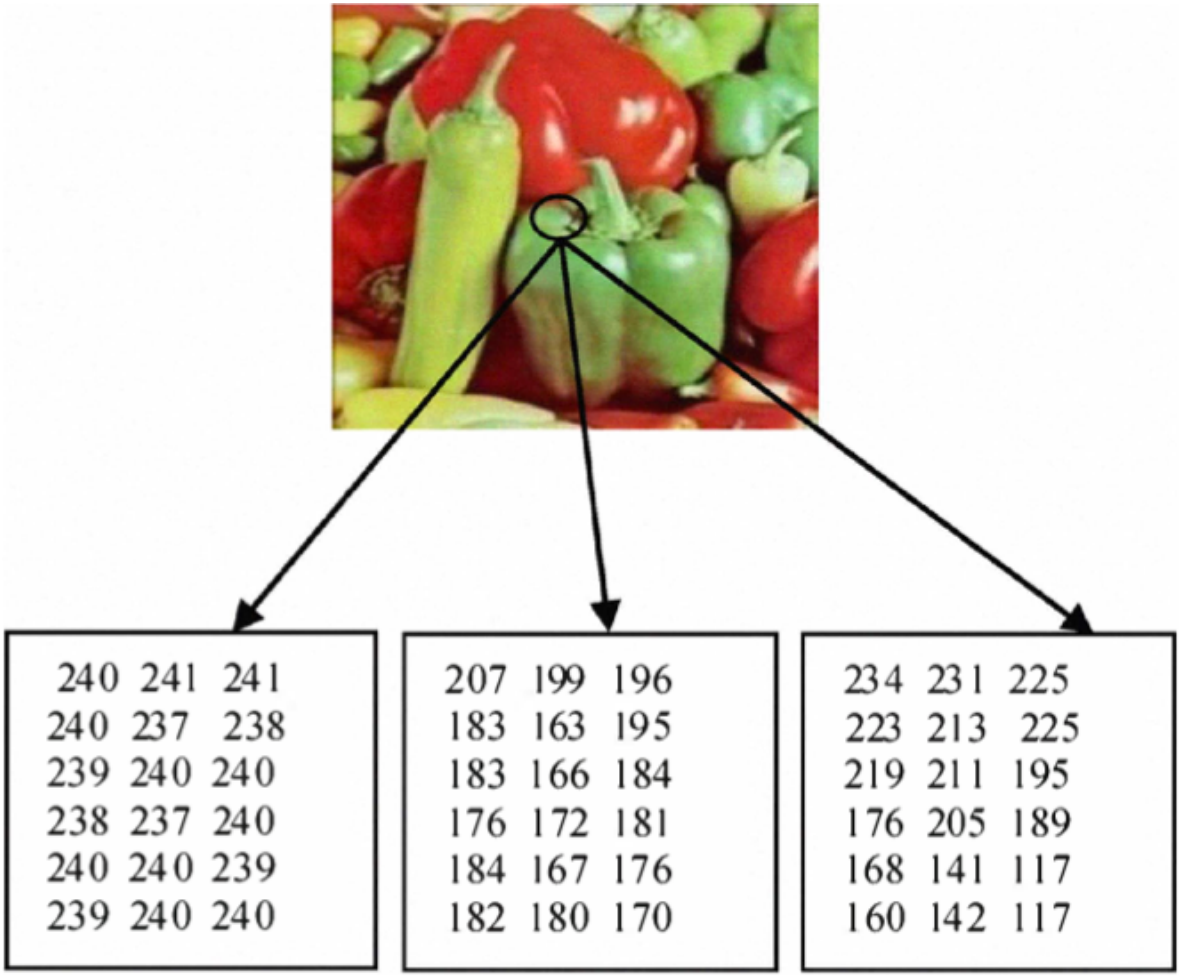
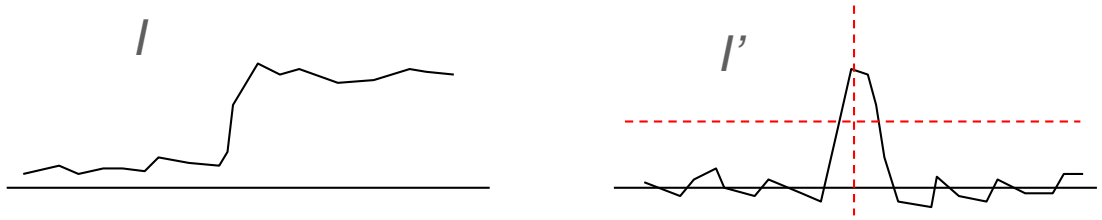
image features

jorge s. marques, josé santos-victor, 2017

What is an image ?

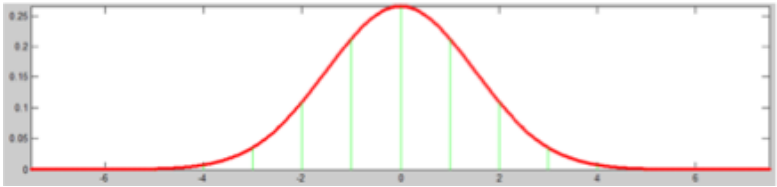


Edge detection 1D case



derivative amplifies the noise

Take a low pass filter *h(x)* (e.g. Gaussian) with bandwidth adapted to the scale of analysis.



noise reduction $v(x) = h(x) * u(x)$

derivative $w(x) = v'(x)$ ou $w(x) = h'(x) * u(x)$

detection $w(x_0) > T$ e w has a local maximum in x_0

Edge detection 2D case

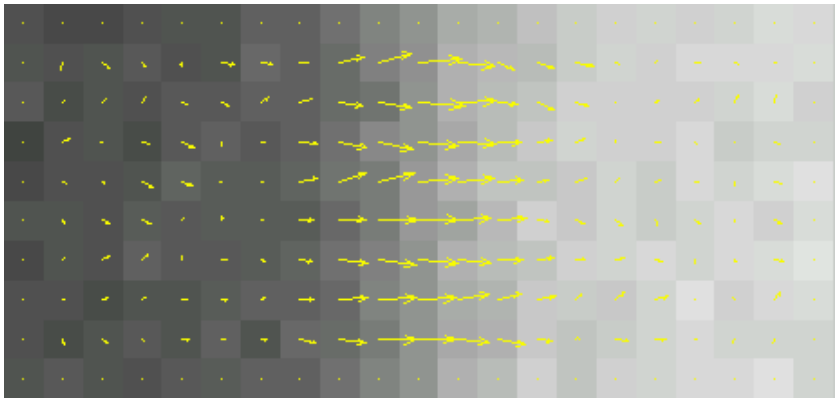
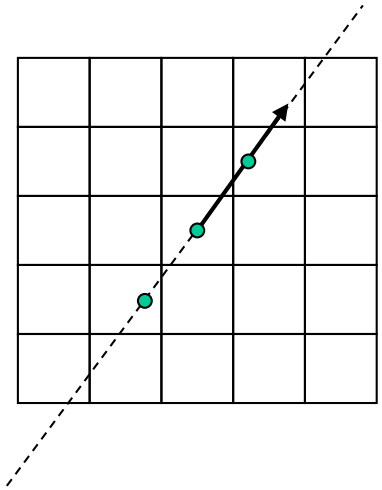


image gradient $$g(\boldsymbol{x}) = \frac{d}{d\boldsymbol{x}} I(\boldsymbol{x})$$

detection $$\|g(\boldsymbol{x})\| > T$$

The gradient direction is the direction of steepest ascent and it is orthogonal to the curves of constant intensity *I(x,y)=C* (level curves).

Non maximum suppression

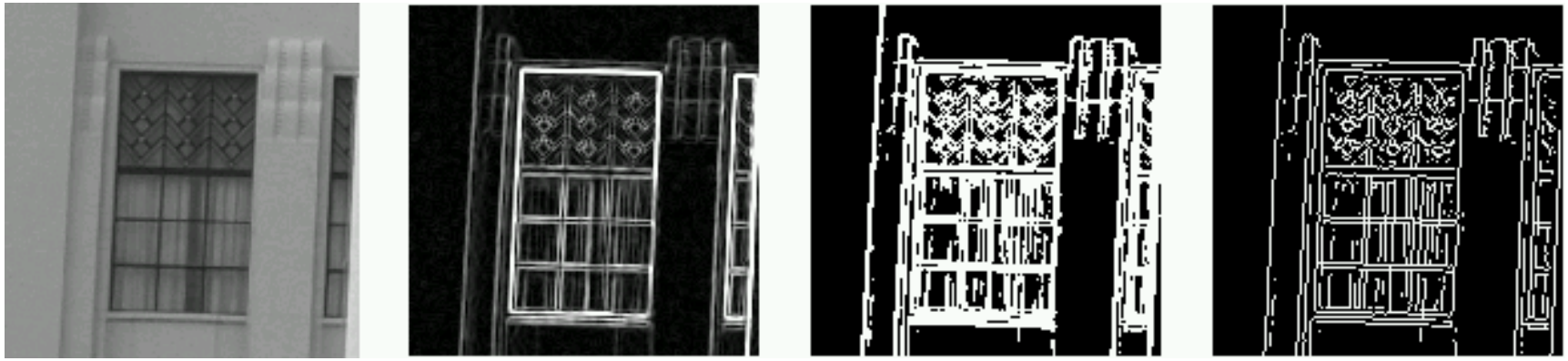


we accept only points in which ||g|| has a local maximum in the gradient direction.

$$|\, g(x) \,| > |\, g(x + \frac{g}{|g|}) \,| \quad \text{e} \quad |\, g(x) \,| > |\, g(x - \frac{g}{|g|}) \,|$$

$g(x + \frac{g}{|g|}), g(x - \frac{g}{|g|})$ computed using interpolation

example



image

gradient (Sobel)

binarization

non maximum suppression

example

image

gradient (Sobel)

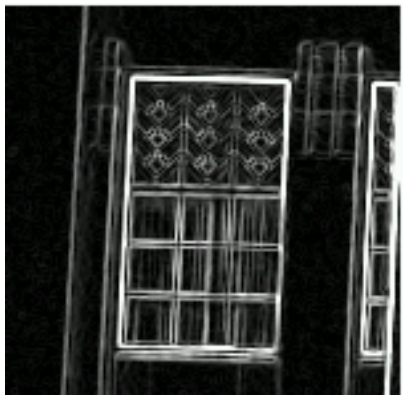
binarization

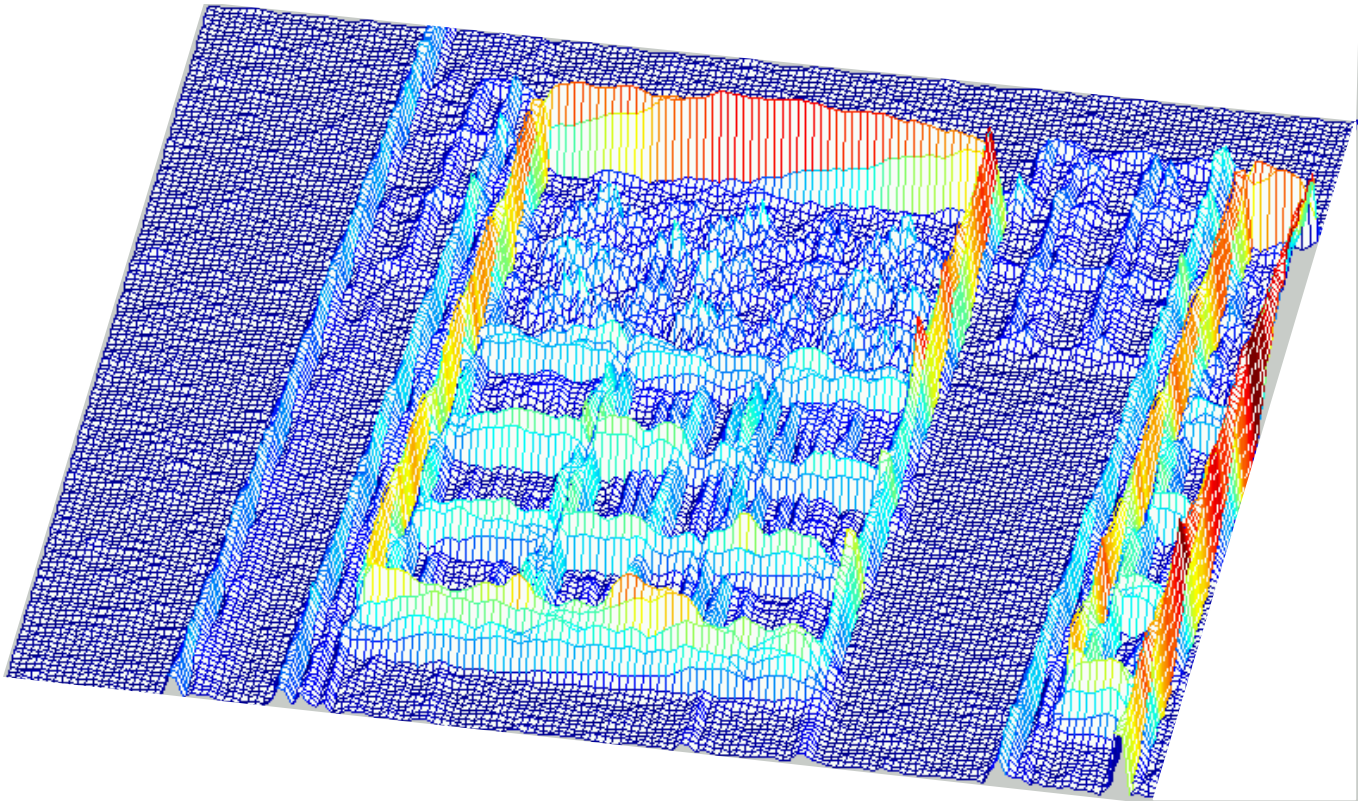
non maximum suppression

difficulties

- missing and false edges
- apparent edges (shadows)
- deviation

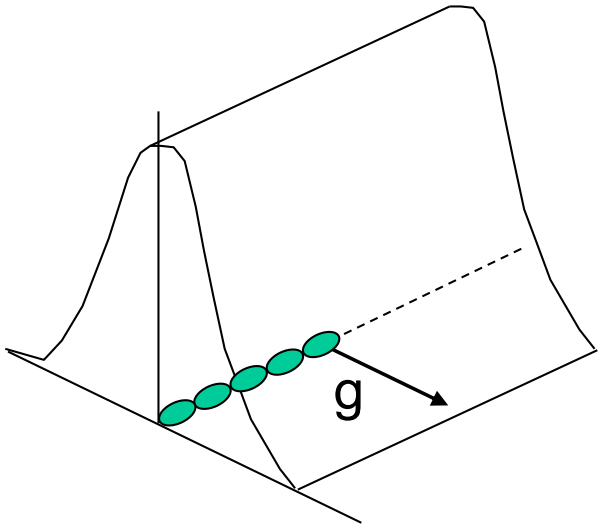
Canny method





goal: obtain edge segments

Canny method



select a point x such that $|g(x)|>T_a$

move x along the direction orthogonal to the gradient, i.e. the direction of the contour

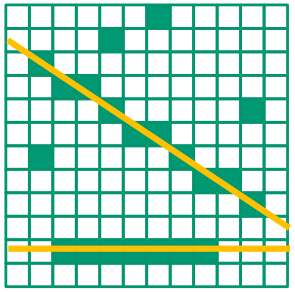
accept a new point x' if $|g(x')|>T_b$ $(T_b < T_a)$
and $|g(x')|$ is a local maximum in the direction of the gradient

repeat until all edge points have been tested.

hysteresis

line detection

how can we detect straight lines in a binary image?



difficulties:

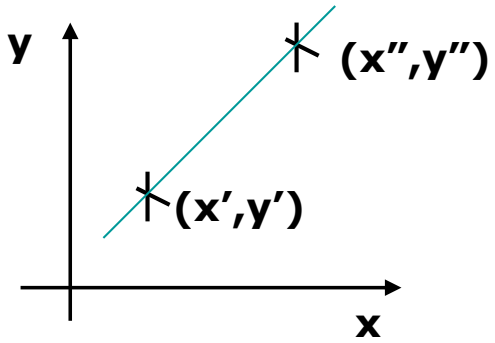
- multiple lines
- noise (outliers)

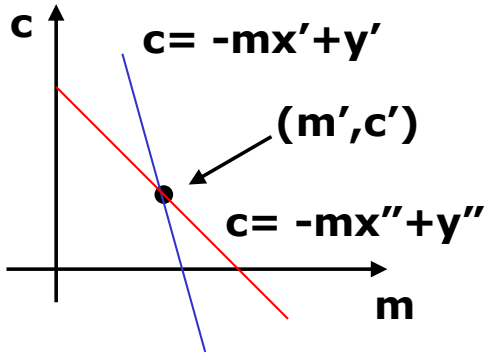
the Hough transform

jorge s. marques, josé santos-victor, 2017

Hough transform

The Hough transform can be used to find paramaterizable curves.
Main idea: parameterize the curve switch to the parameter space.





Example : (line detection)

1. For each contour point (x’,y’), consider all lines containing this point and increase counter A(m,c) corresponding to those lines.
2. Repeat for ther contour points.
3. When all points have been considered, choose those lines where A (m,c) is larger.

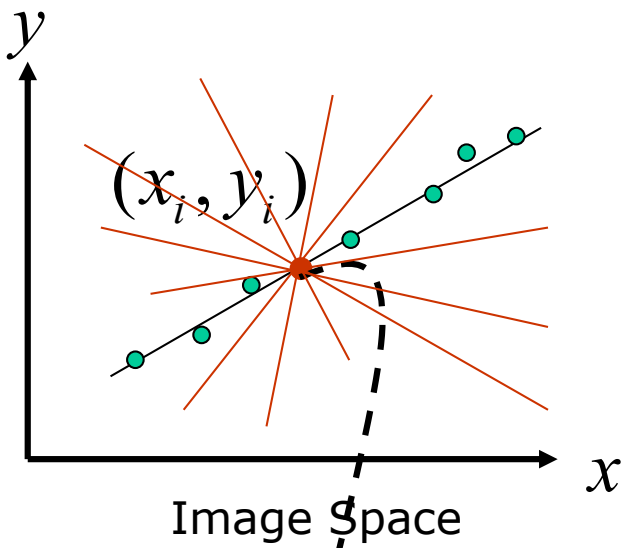
Hough transform: better parameterization

Line equation in polar coordinates:

$$x\cos\theta + y\sin\theta - \rho = 0$$

where $0 \le \theta \le 2\pi$

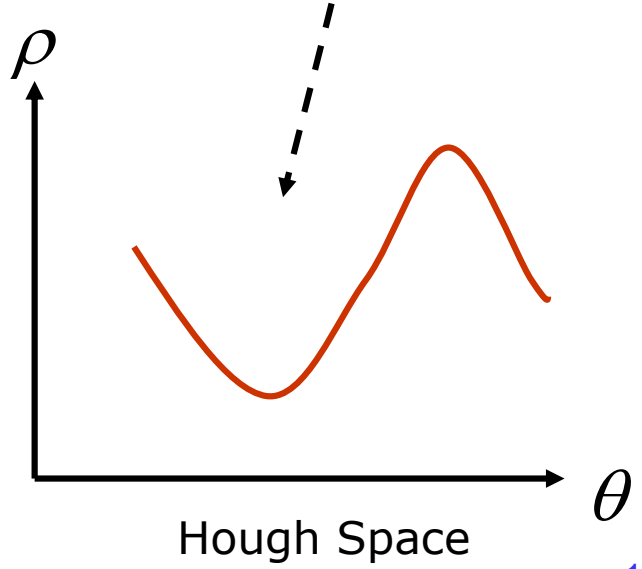
$$0 \le \rho \le \rho_{\max}$$



Given point (x_i, y_i) find (ρ, θ)

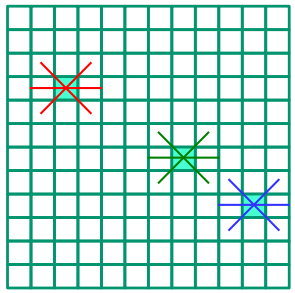
Hough Transform of one point (xi,yi) is sinusoid:

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

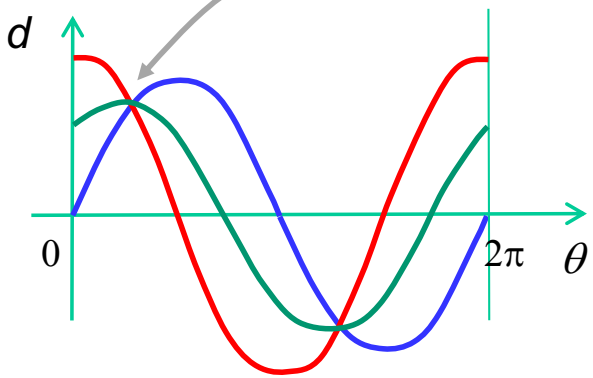


Hough transform

edge points



line parameters



$d \geq 0$

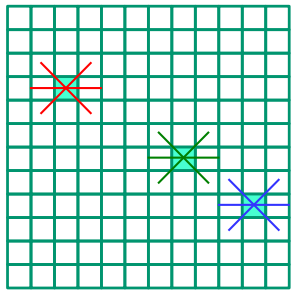
$$d = x\,cos\theta + y\,sin\theta$$

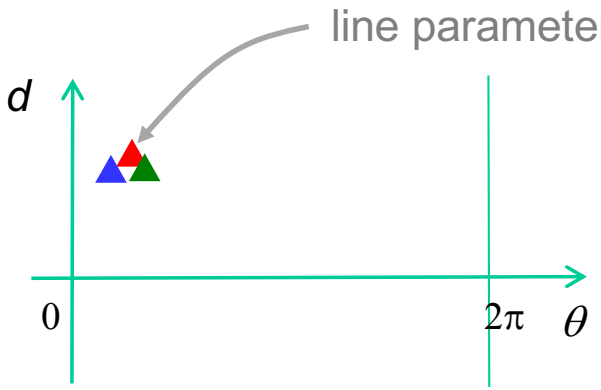
each image point is associated to a constraint (sinusoid) in the parameter space.

The Hough transform splits the parameter space (θ,d) into cells and builds an histogram: each observation votes in all the cells intersected by the sinusoid.

Histogram peaks correspond to straight lines.

Hough transform (2)





$d \geq 0$

$$d = x\,cos\theta + y\,sin\theta$$

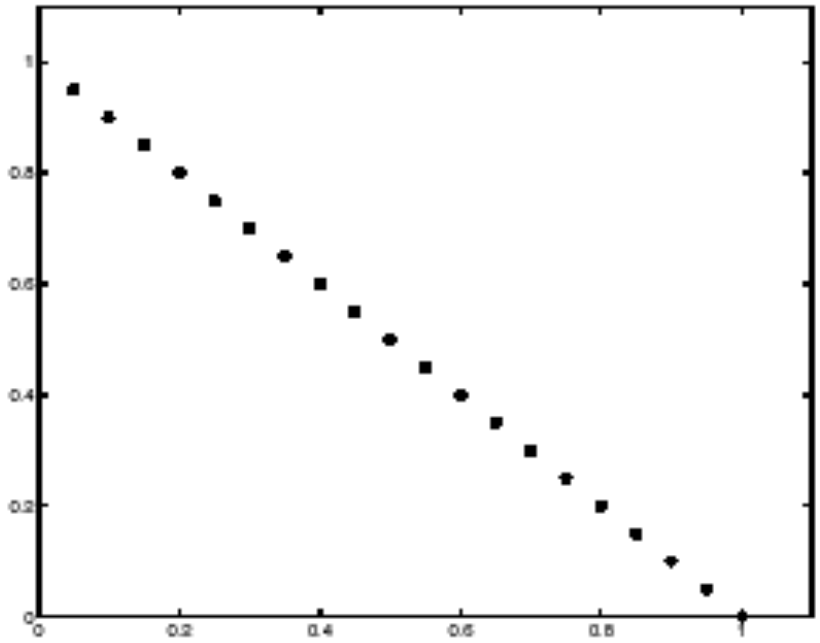
each image point is associated to a single point in the parameter space.

The image gradient should be orthogonal to the straight line and the direction of the gradient should be equal to θ (apart from a measurement error).

Therefore, we know the parameter θ associated to each edge point.

clustering in parameter space!

Straight line Hough Transform



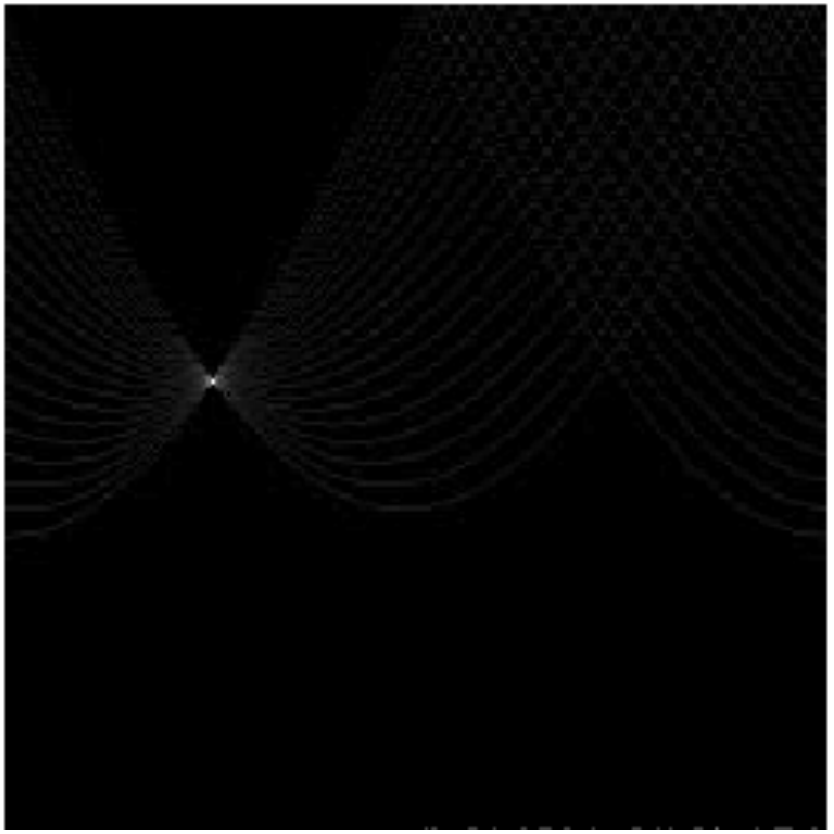
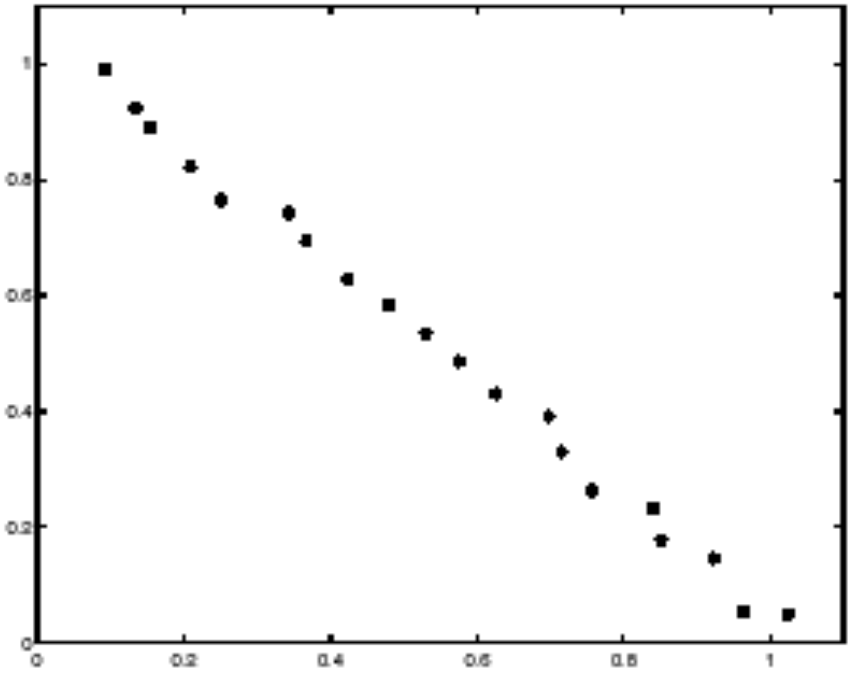


Image space

Votes/accumulator
Horizontal axis is θ,
Vertical is ρ.

Noisy line

Hough Transform



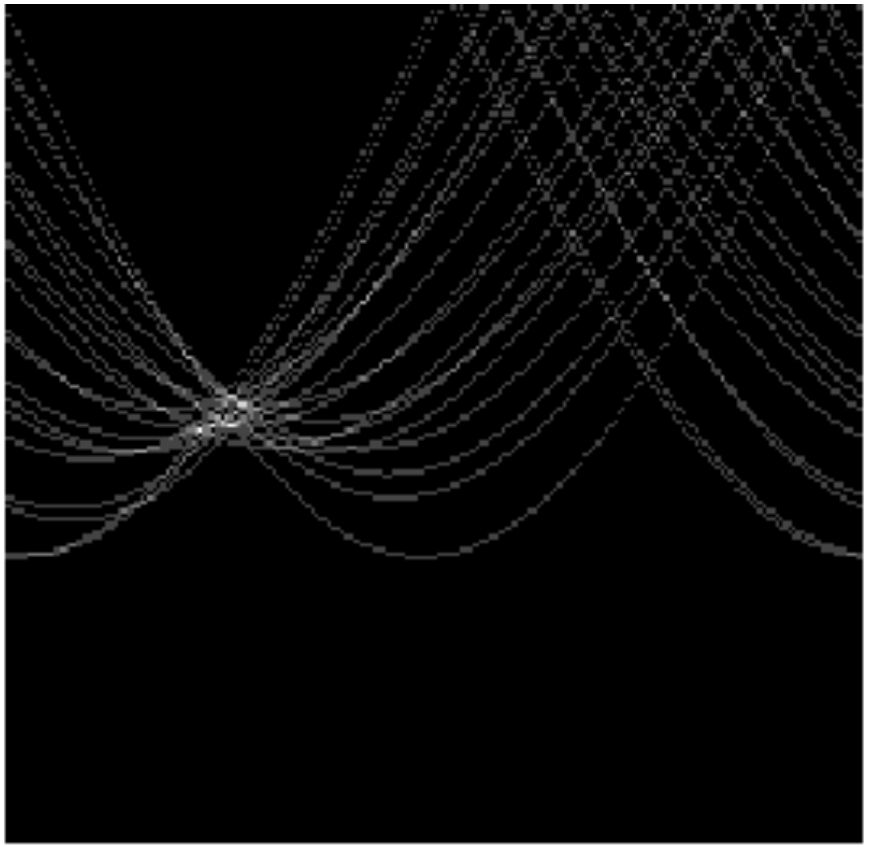
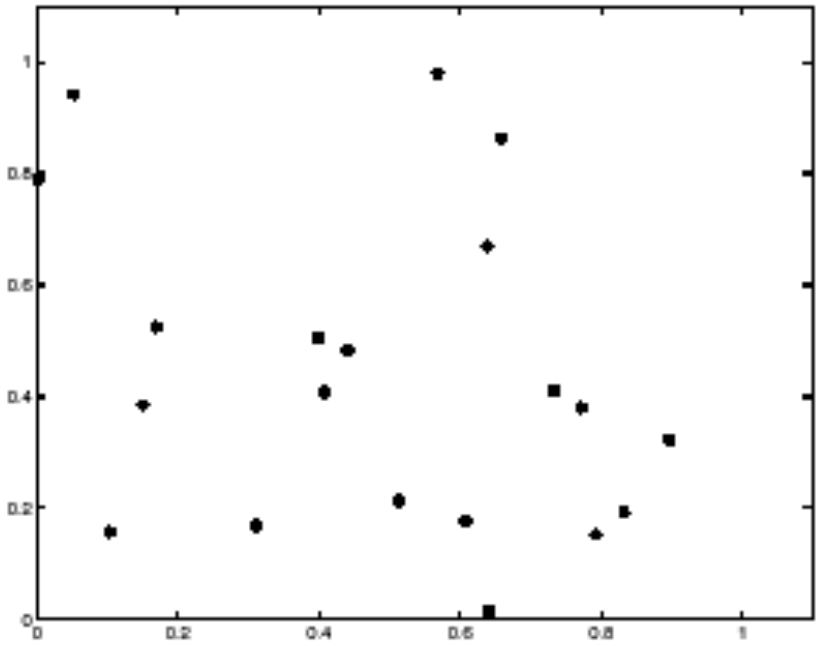


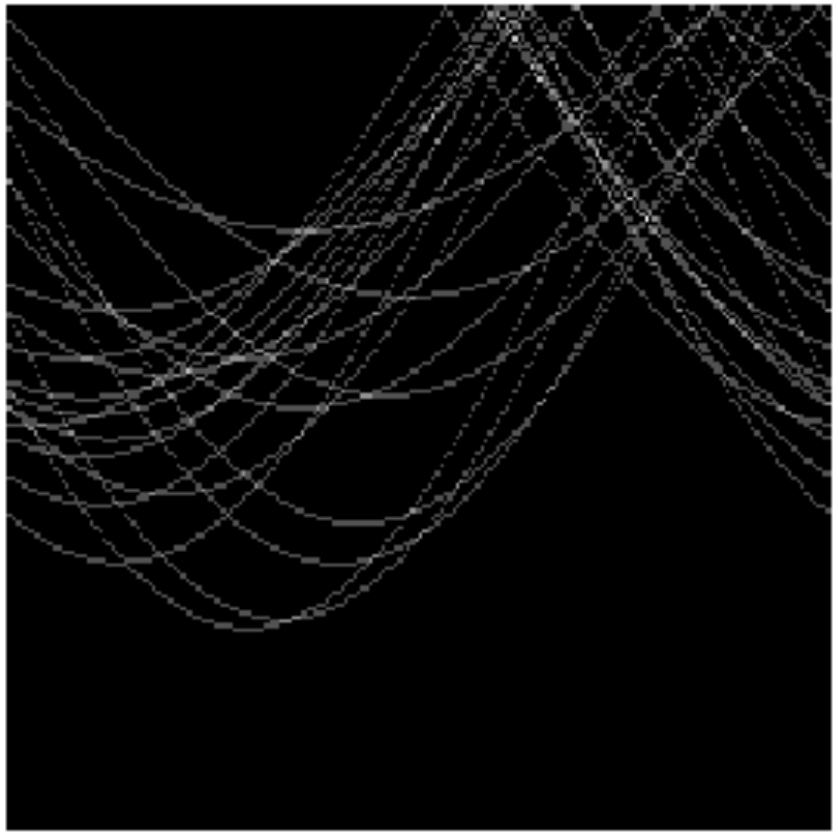
Image space

votes

Random points

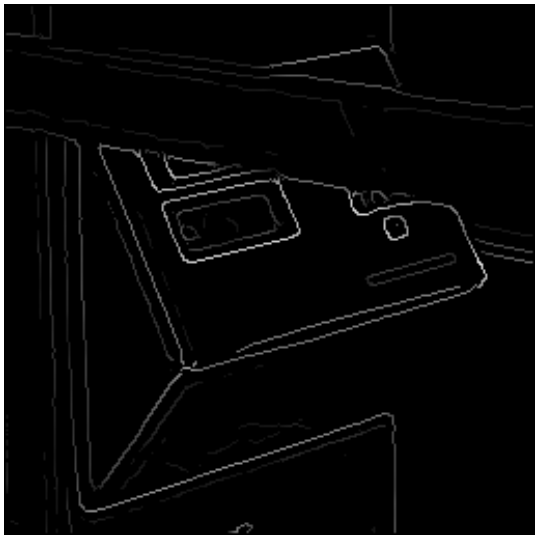
Hough Transform



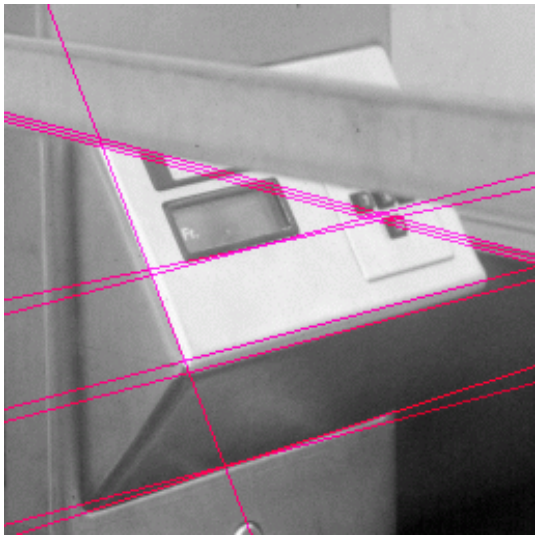


Real World Example

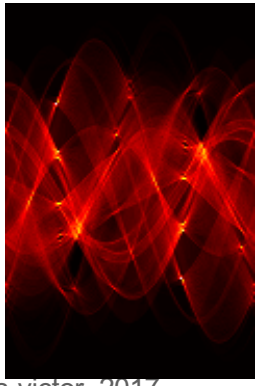
Original



Edge Detection



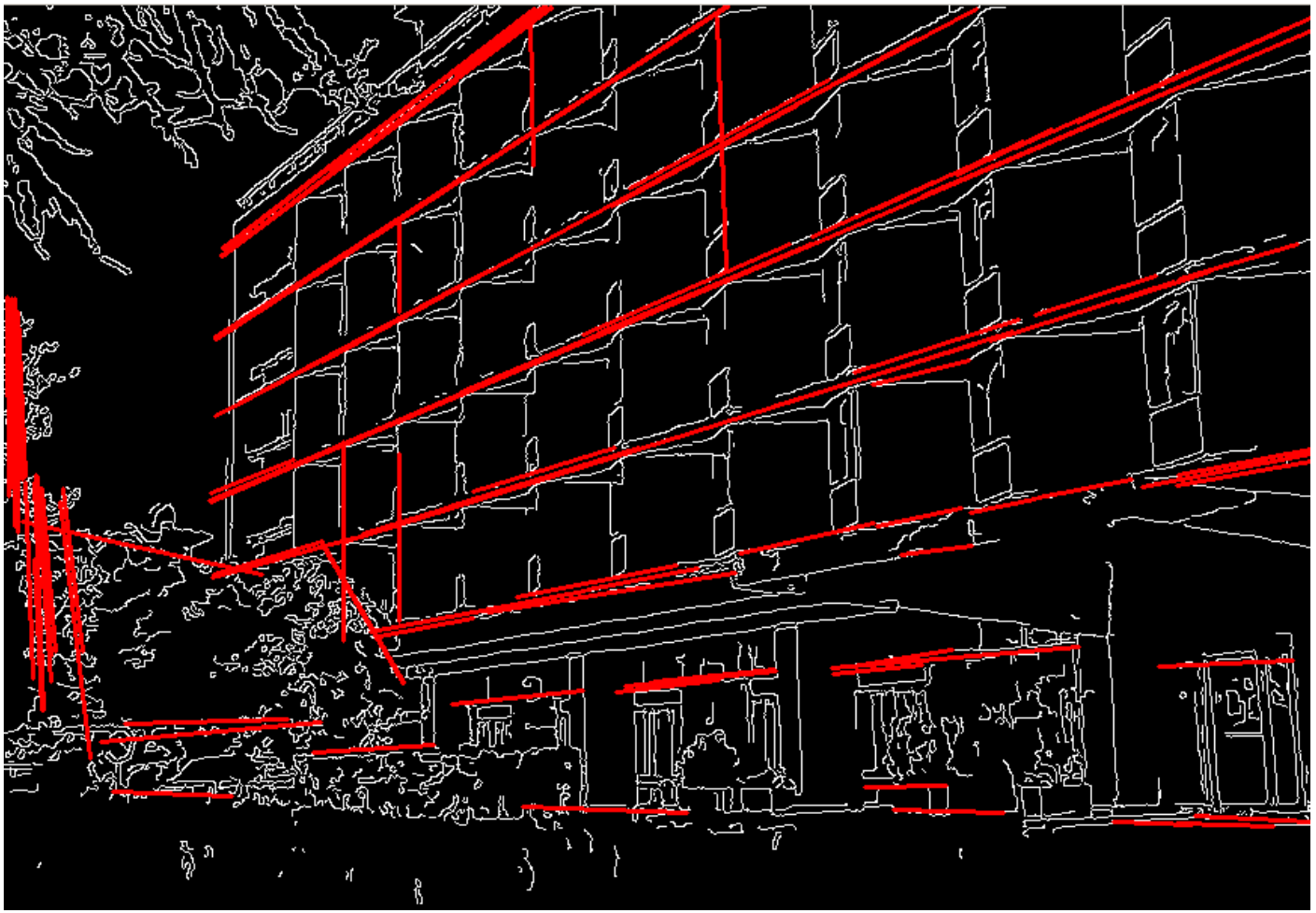
Found Lines



Parameter Space

Real World Example

Real World Example



Finding Circles by Hough Transform

Equation of Circle in the spatial domain:

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

- If radius is known (2D Hough Space):

- Hough Transform for a point (xi,yi) is a circle at center (xi,yi):

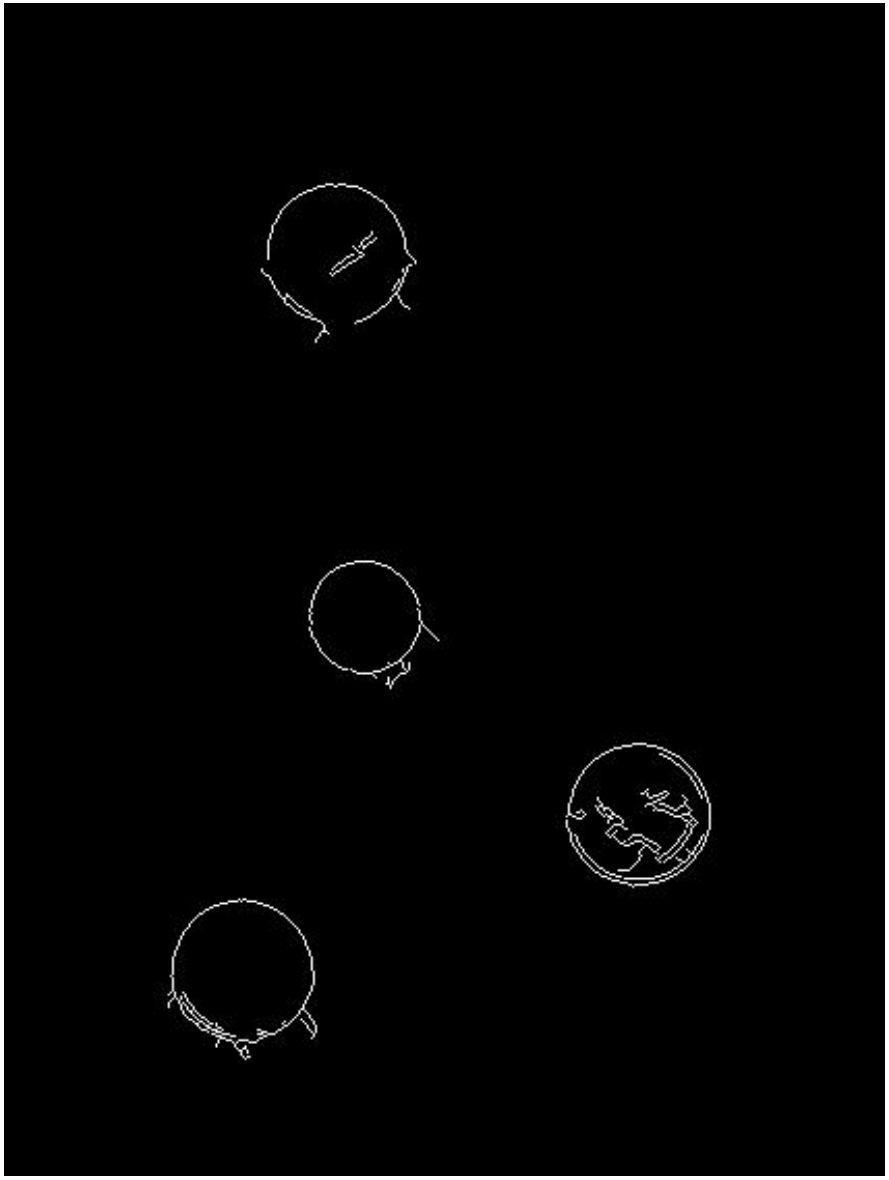
$$(a - x_i)^2 + (b - y_i)^2 = r^2$$

Accumulator Array $A(a,b)$

Finding Coins

Original

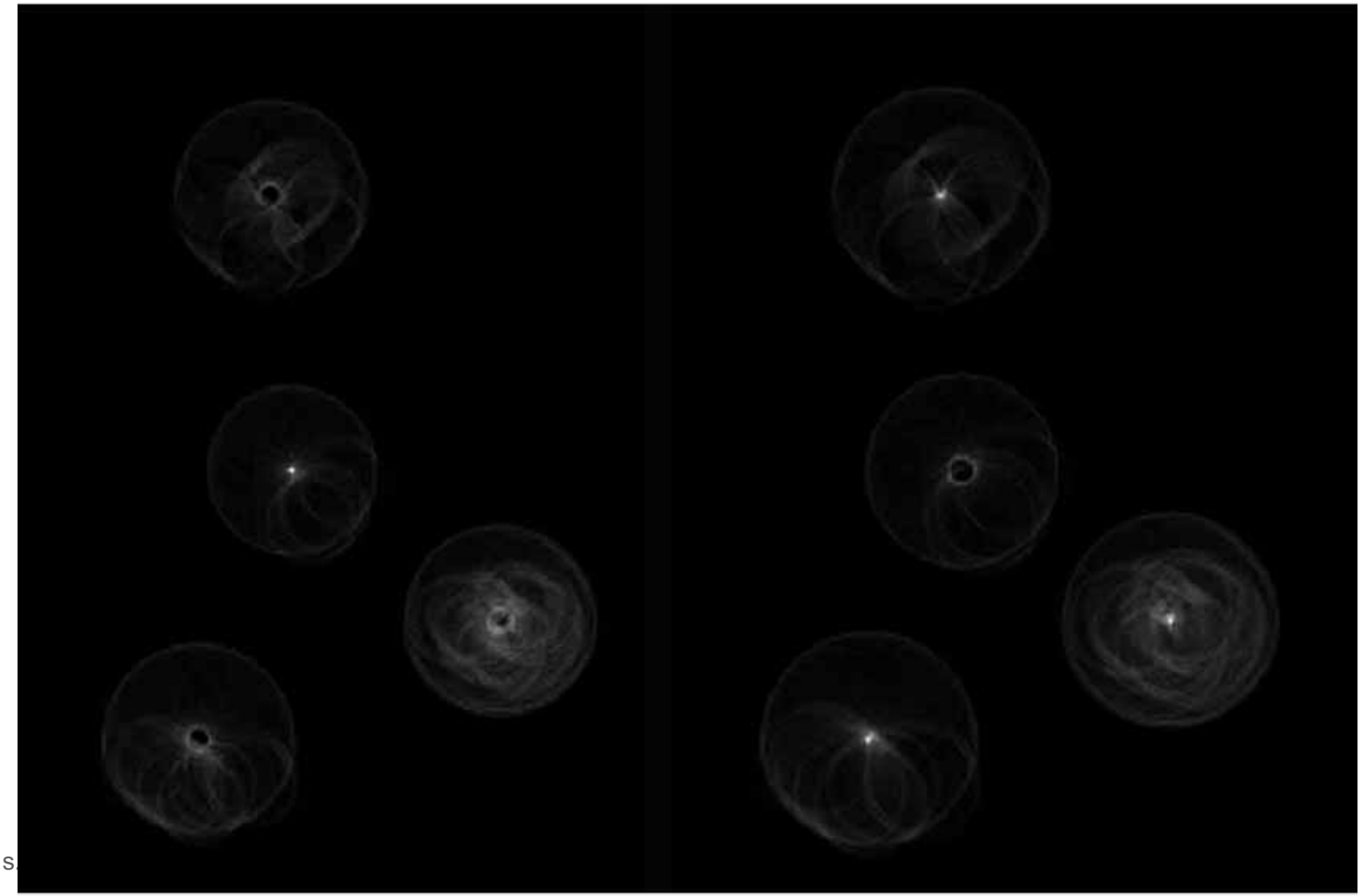
Edges (notice noise)



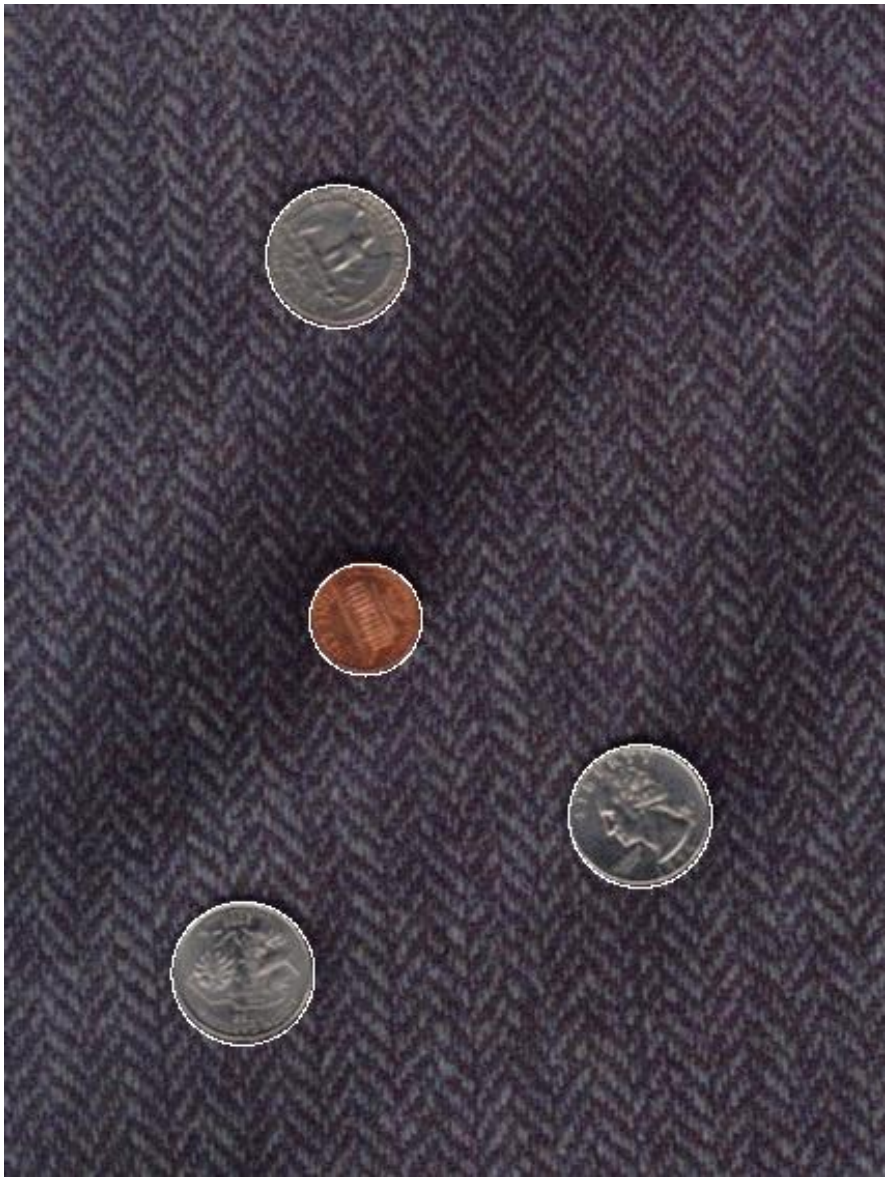
Finding Coins (continued)

Penny

Quarters



Finding Coins (Continued)

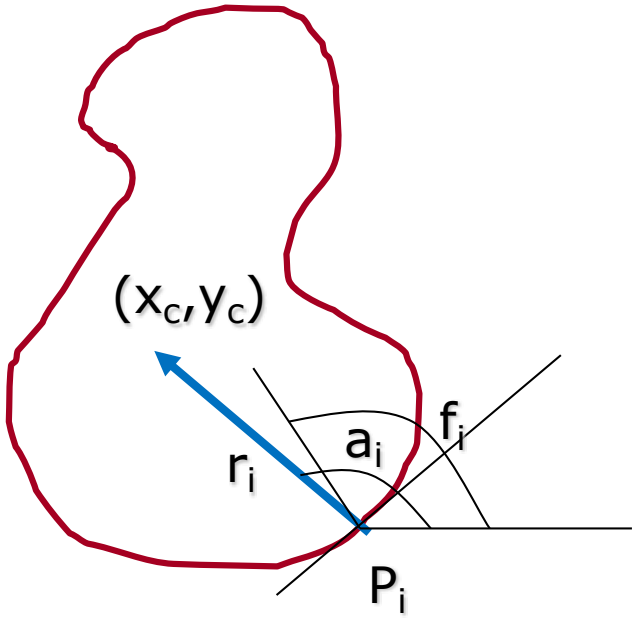


Note that because the quarters and penny are different sizes, a different Hough transform (with separate accumulators) was used for each circle size.

Coin finding sample images from: Vivik Kwatra

Generalized Hough Transform

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



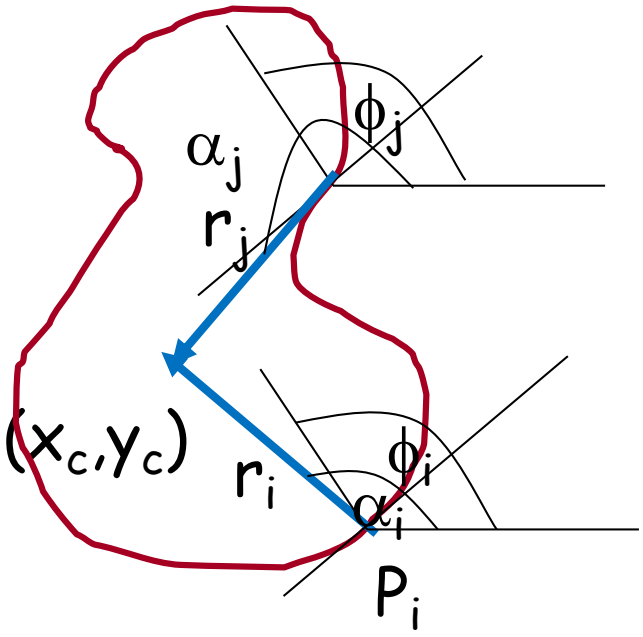
1. Pick a reference point (x_c,y_c)
2. For i = 1,…,n :
   1. Draw segment to P_i on the boundary.
   2. Measure its length r_i, and its orientation a_i.
   3. Write the coordinates of (x_c,y_c) as a function of r_i and a_i
   4. Record the gradient orientation f_i at P_i.
3. Build a table with the data,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

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

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

Generalized Hough Transform

Suppose, there were m ***different*** gradient orientations:
(m <= n)



ϕ_1	$(r^1_1,\alpha^1_1),(r^1_2,\alpha^1_2),\ldots,(r^1_{n1},\alpha^1_{n1})$
ϕ_2	$(r^2_1,\alpha^2_1),(r^2_2,\alpha^1_2),\ldots,(r^2_{n2},\alpha^1_{n2})$
.	.
.	.
.	.
ϕ_m	$(r^m_1,\alpha^m_1),(r^m_2,\alpha^m_2),\ldots,(r^m_{nm},\alpha^m_{nm})$

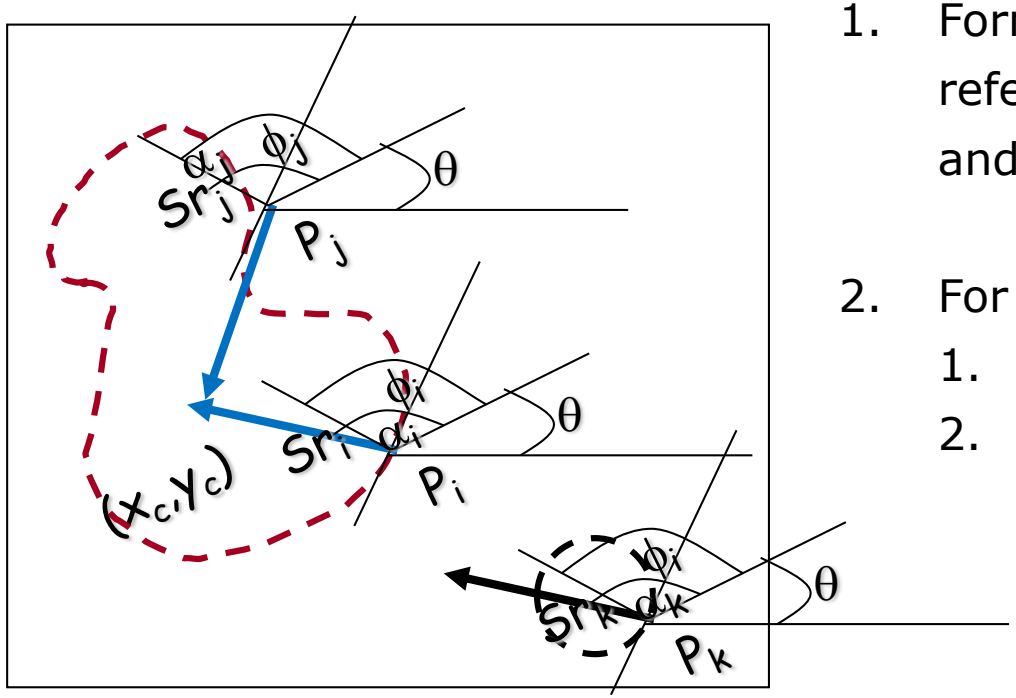
$x_c = x_i + r_i\cos(\alpha_i)$

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

H.T. table

Generalized Hough Transform

Finds a rotated, scaled, and translated version of the curve:



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

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

1. Form an A accumulator array of possible reference points (x_c,y_c), scaling factor Σ and Rotation angle θ.

2. For each edge (x,y) in the image:
   1. Compute f(x,y)
   2. For each (r,α) corresponding to $\phi(x,y)$ do:
      1. For each Σ and θ:
         1. $x_c = x_i + r(f)\ \Sigma \cos[\alpha(\phi) + \theta]$
         2. $y_c = y_i + r(f)\ \Sigma \sin[\alpha(\phi) + \theta]$
         3. $A(x_c,y_c,\Sigma,\theta)$ ++

3. Find maxima of A.

Hough Transform Summary

- H.T. is a "voting" scheme
  - points vote for a set of parameters describing a line or curve.
- The more votes for a particular set
  - the more evidence that the corresponding curve is present in the image.
- Can detect MULTIPLE curves in one shot.
- Computational cost increases with the number of parameters describing the curve.

RANSAC

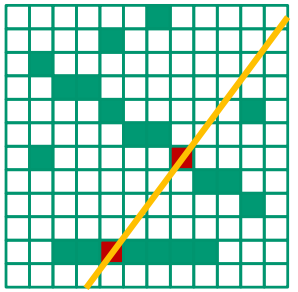
Fischler & Bolles, 1981

repeat

- randonly select 2 edge points
- compute the line coefficients θ,d
- count the number of edge points close to the line (inliers)

select the model with larger number of inliers

Perform a least squares fit using the inliers



keypoints and patches

Keypoints

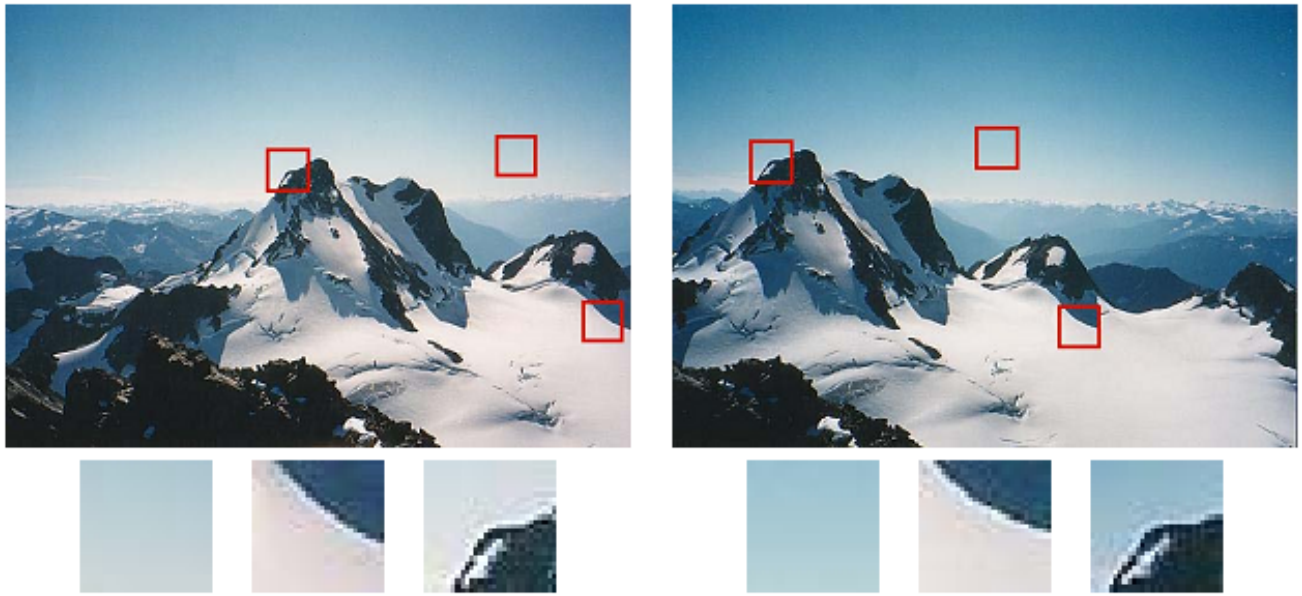
Keypoints are points which allow us to extract useful information from the image (e.g., for alignment or motion estimation).

They should be detected in a robust way and contain useful information.

They correspond to textured regions; their neighborhood should contain intensity variations in more than one direction.

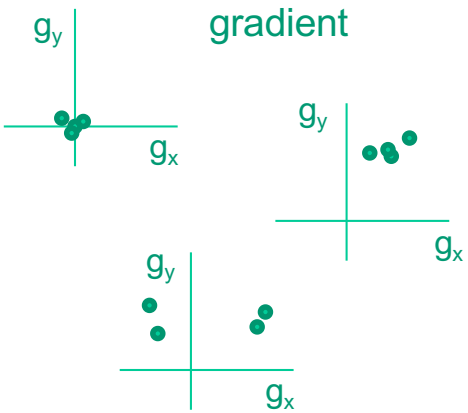
keypoints are sometimes called interest points or corners.

What are the good points (patches) to track?



Patches:

- no texture: blank wall problem
- one edge: aperture problem
- changes in multiple directions: good!!

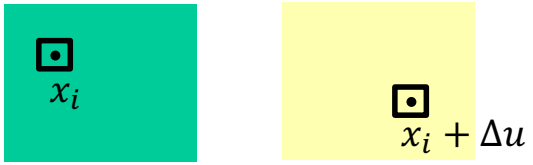


What are the good points (patches) to track

Alignment criterion

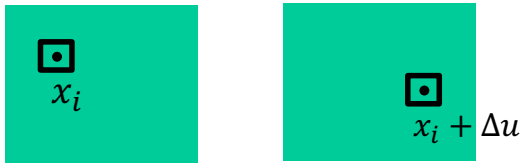
Sum of square differences

$$E(\Delta u) = \sum_i w(x_i)\,[I_1(x_i + \Delta u) - I_0(x_i)]^2$$



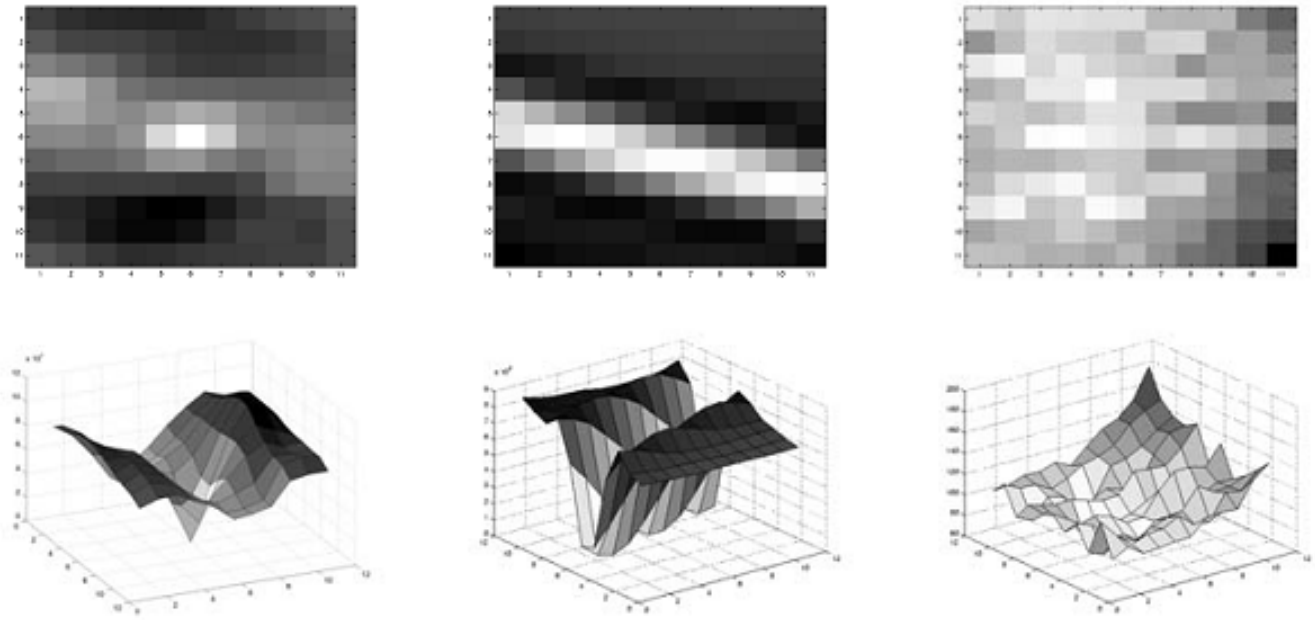
Autocorrelation function

$$E(\Delta u) = \sum_i w(x_i)\,[I_0(x_i + \Delta u) - I_0(x_i)]^2$$



autocorrelation energy

(a)



Harris corner detector

$$E(\Delta u) = \sum_i w(x_i)\,[I_0(x_i + \Delta u) - I_0(x_i)]^2$$

expanding the image in Taylor series $I_0(x_i + \Delta u) \cong I_0(x_i) + \nabla I_0(x_i)\Delta u$

we obtain

$$E(\Delta u) = \Delta u^T M\ \Delta u \qquad M = \sum_x w(x_i)\nabla I_0(x_i)\nabla I_0(x_i)^T$$

The window w may move along the image.

$$M(x,y) = w(x,y) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \qquad I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

Mathematics

$$E(u,v) \cong [u,v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

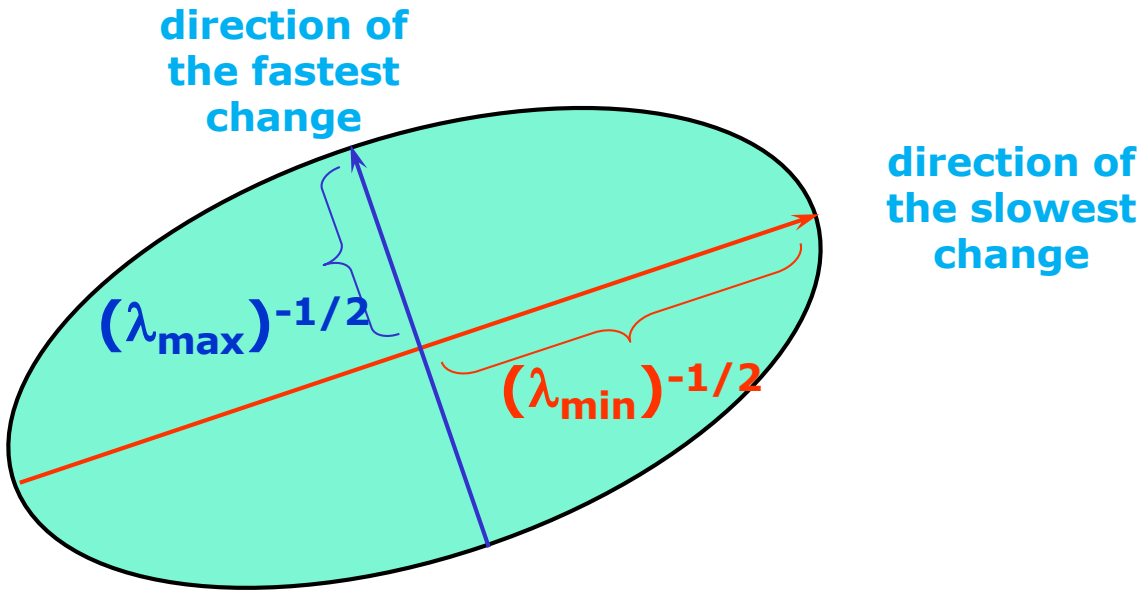
$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

λ_1, λ_2 *eigenvalues of M*

We can visualize *M* as an ellipse with axis lengths determined by the eigenvalues and orientation determined by *R*

Ellipse E(u,v) = const.

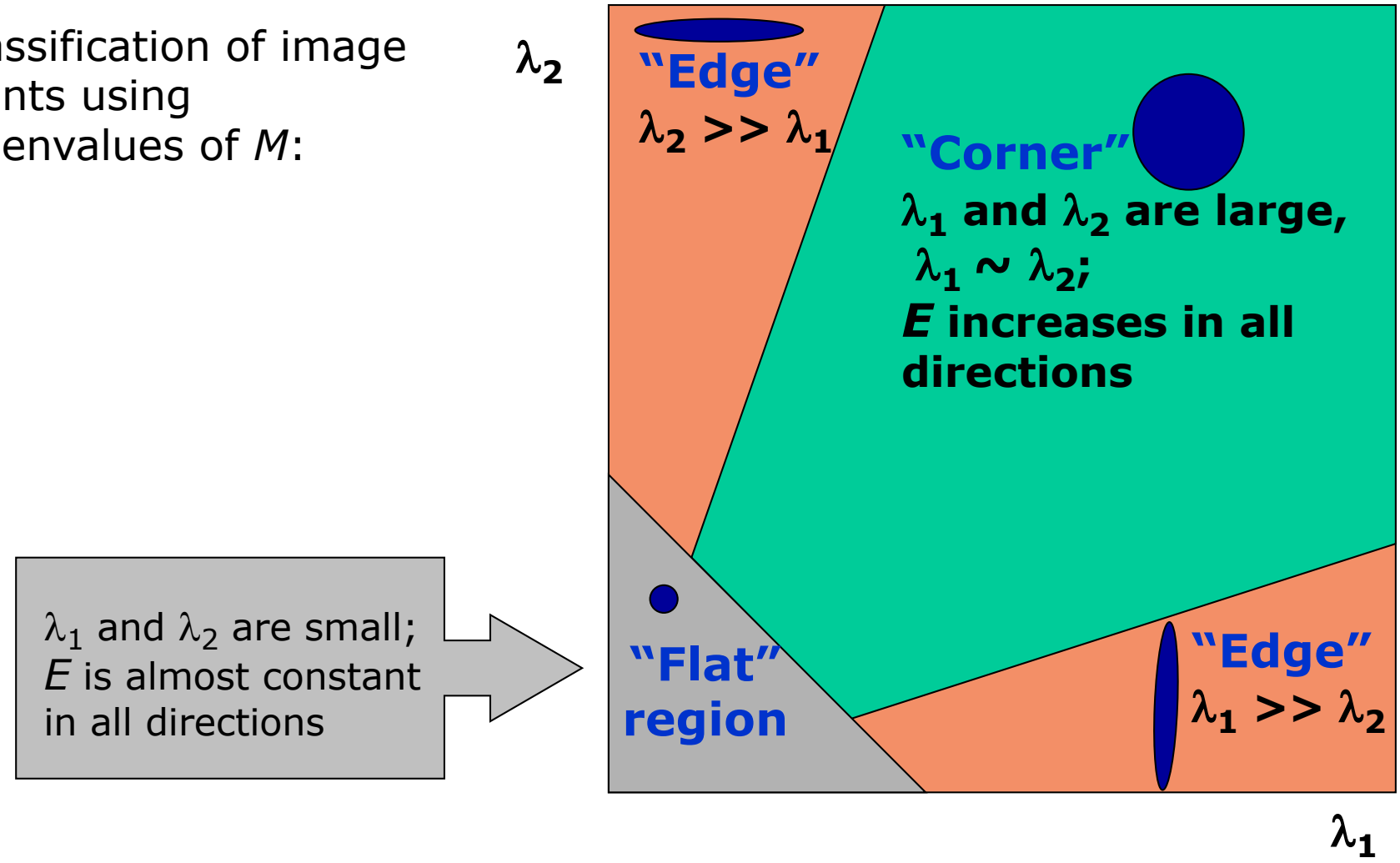
Iso-intensity contour of E(u,v)



Slide adapted from Darya Frolova, Denis Simakov.

Interpreting the eigenvalues

Classification of image points using eigenvalues of *M*:



Harris corner detector

$$M(x,y) = w(x,y) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \qquad I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

Matrix M should have large eigenvalues

Harris criterion

$$R = det(M) - k\, trace^2(M) = \lambda_0 \lambda_1 - k(\lambda_0 + \lambda_1)^2; \quad k \in [0.04, 0.15]$$

A point *x* is detected if:

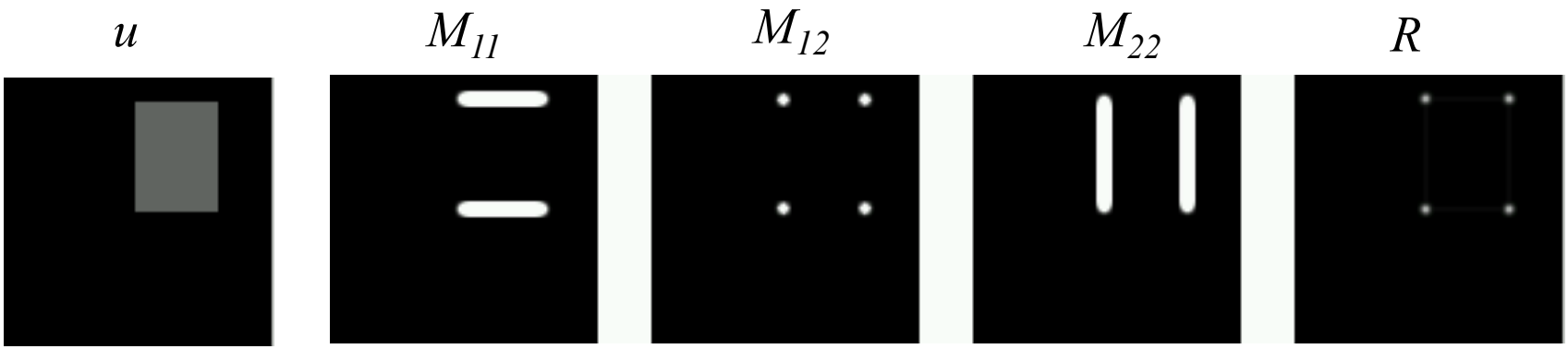
i) R has a local maximum at x and — non-maximum supression

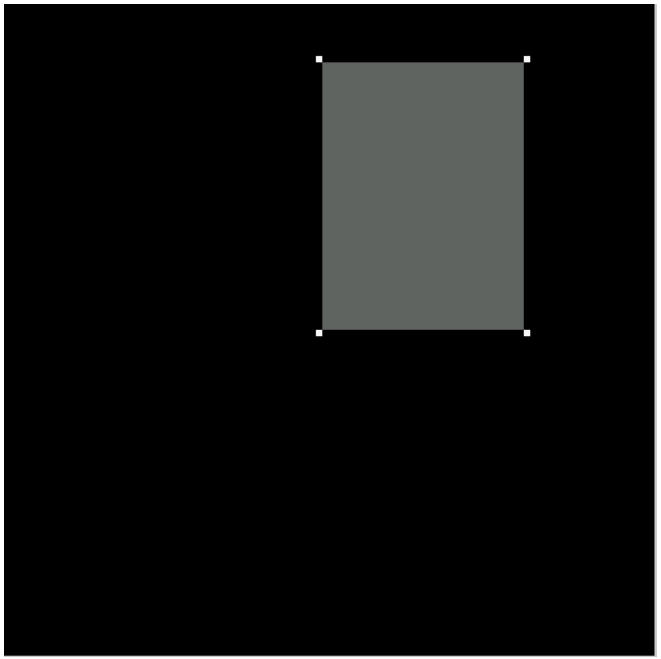
ii) R>T. — thresholding

Harris corner detector

- compute the gradient of the image (e.g., Sobel)
- compute 3 images with the products of the partial derivatives and filter them with a lowpass filter.
- compute R at each point
- determine local maxima of R and compare them with a threshold T

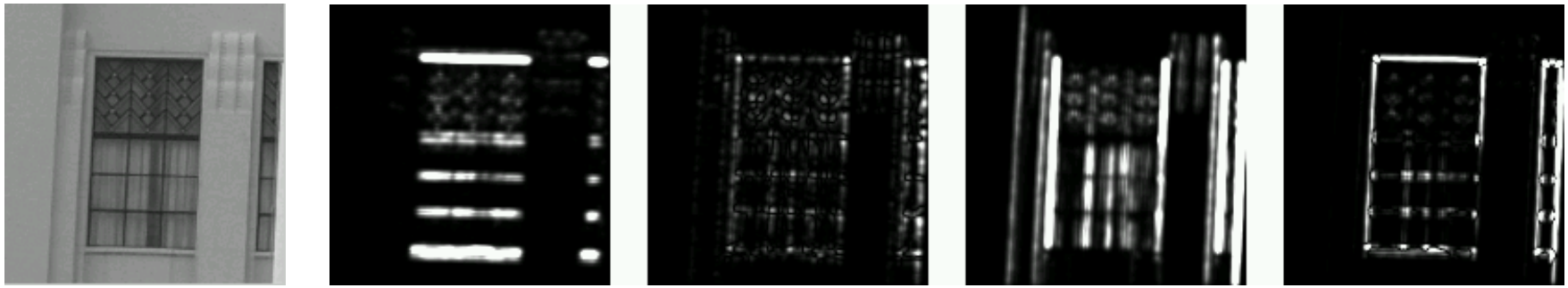
example

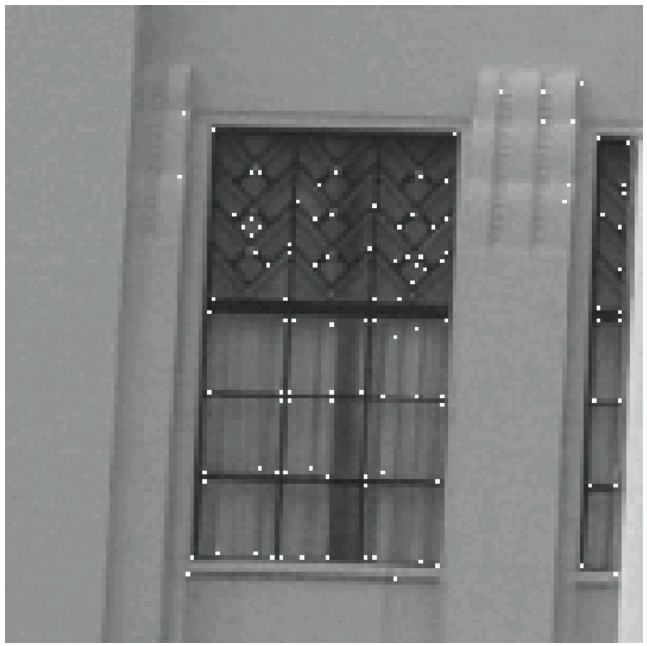




example

u M_{11} M_{12} M_{22} R



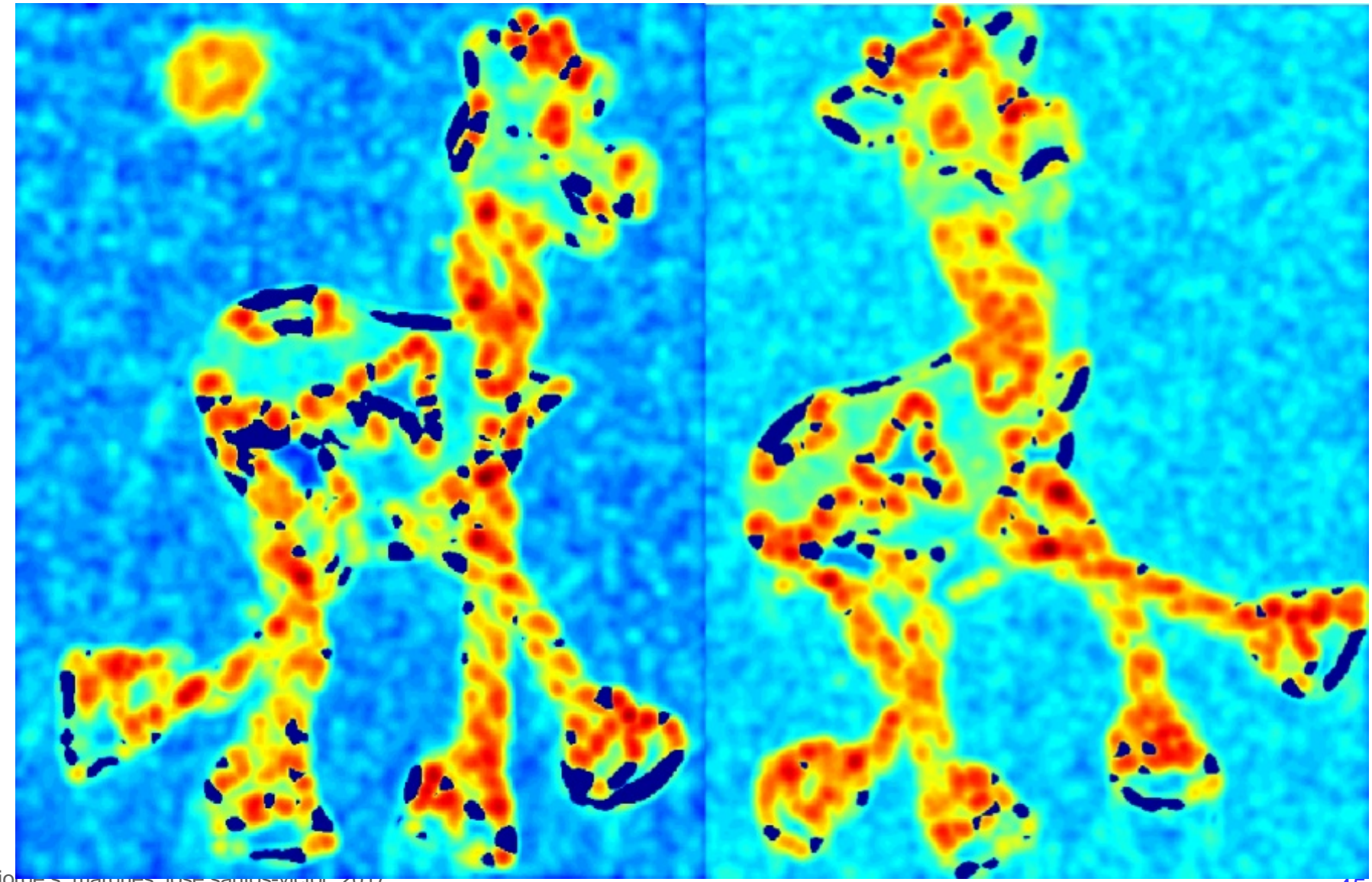


Harris Detector: Workflow

Slide adapted form Darya Frolova, Denis Simakov, Weizmann Institute.

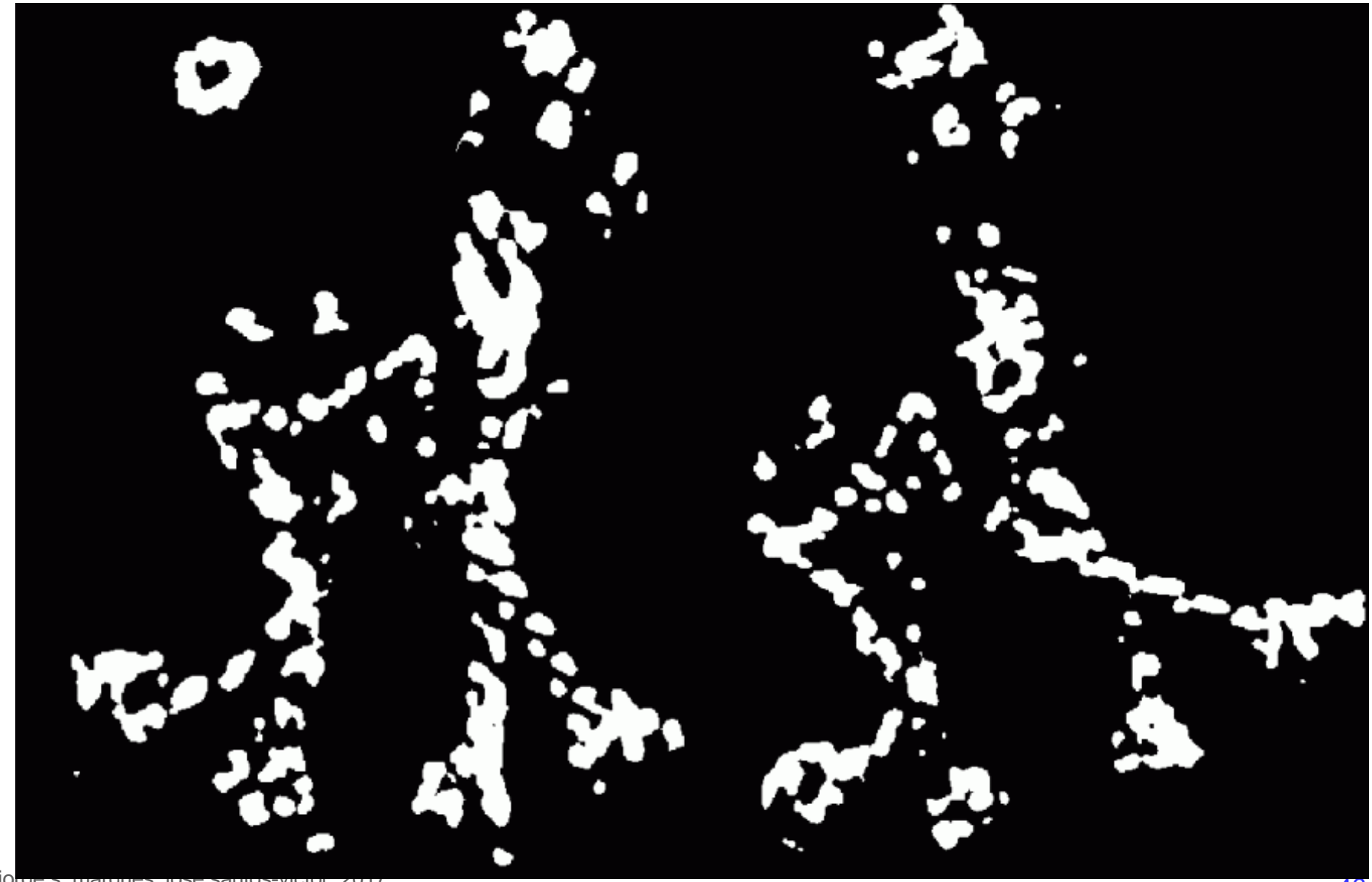
Harris Detector: Workflow

Compute corner response *R*



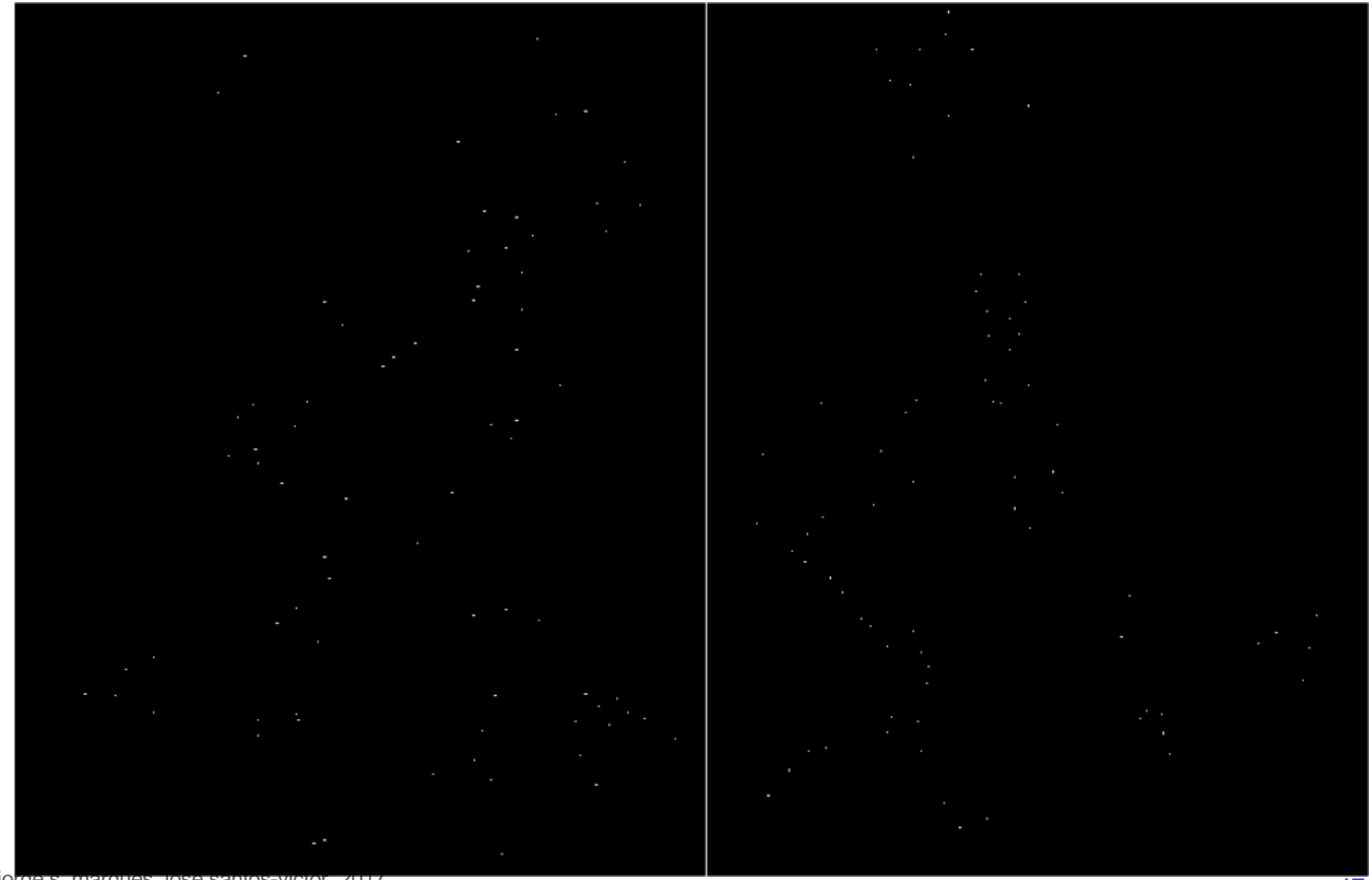
Harris Detector: Workflow

Find points with large corner response: *R*>threshold<<



Harris Detector: Workflow

Take only the points of local maxima of *R*

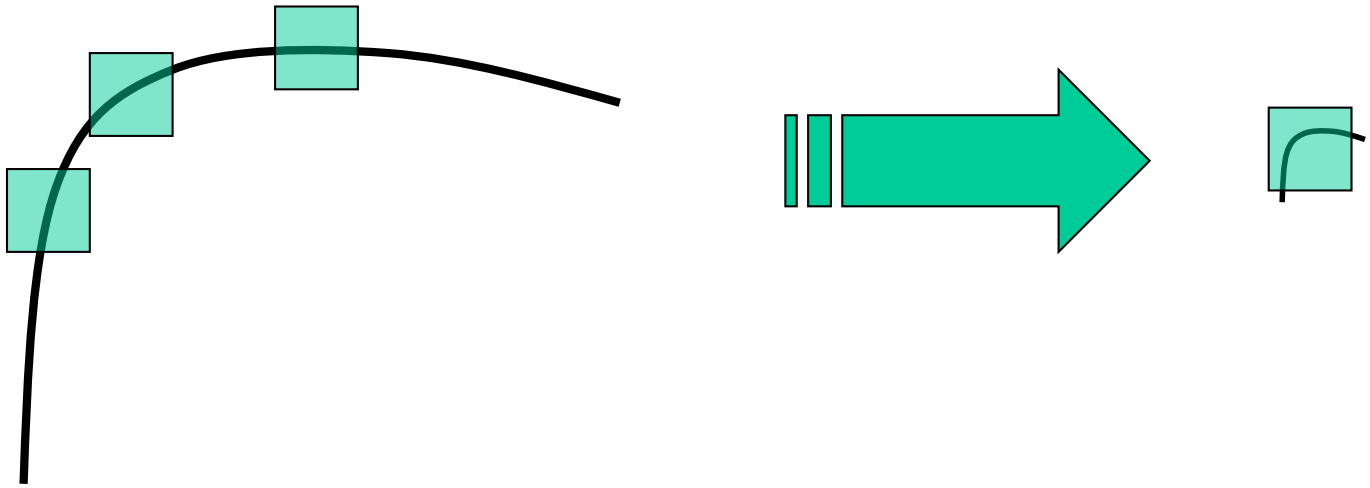


Harris Detector: Workflow

coping with scale

Harris Detector: Properties

Not invariant to image scale

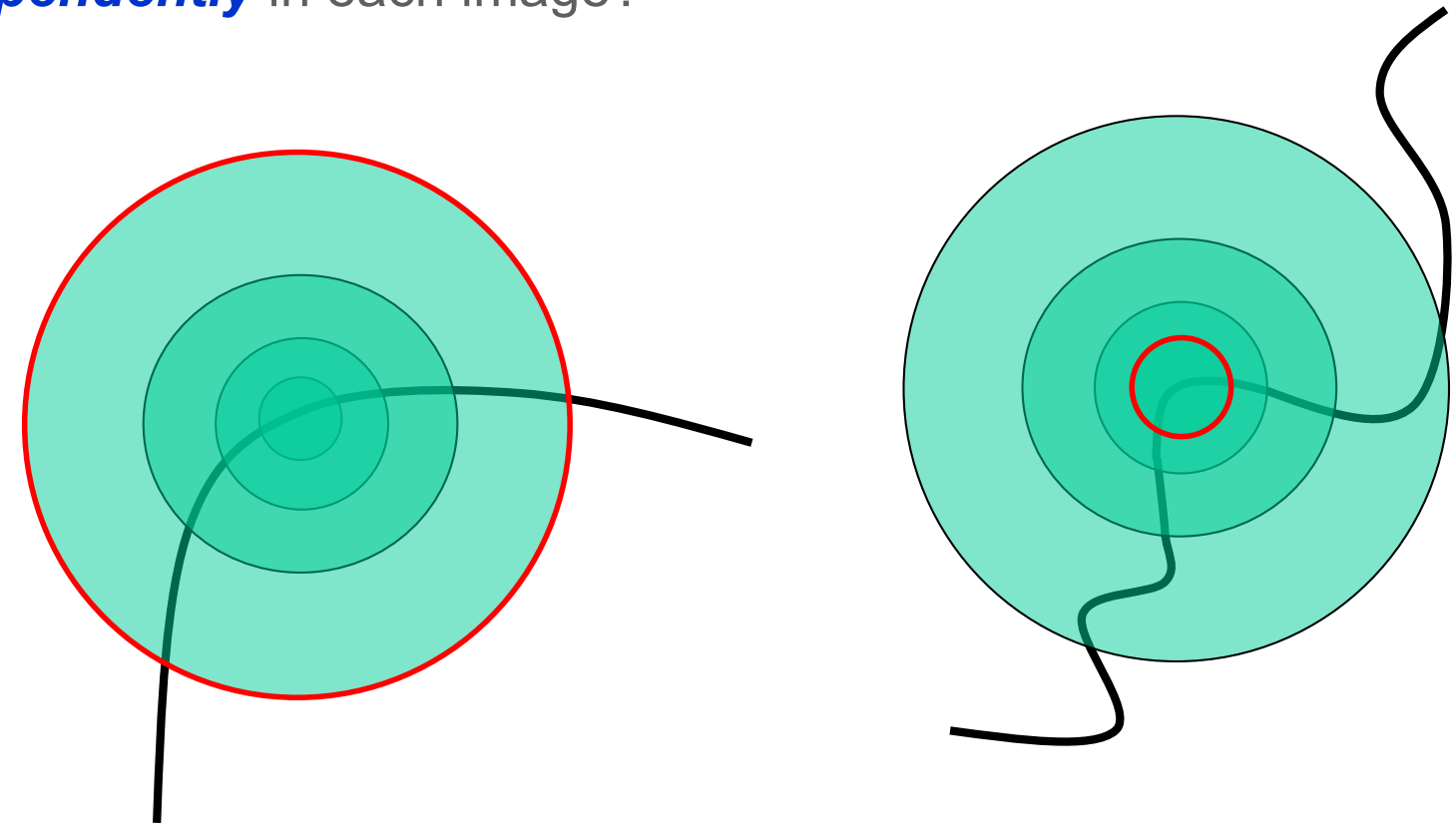


All points will be classified as edges

Corner !

dealing with scale

The problem: how do we choose corresponding circles ***independently*** in each image?

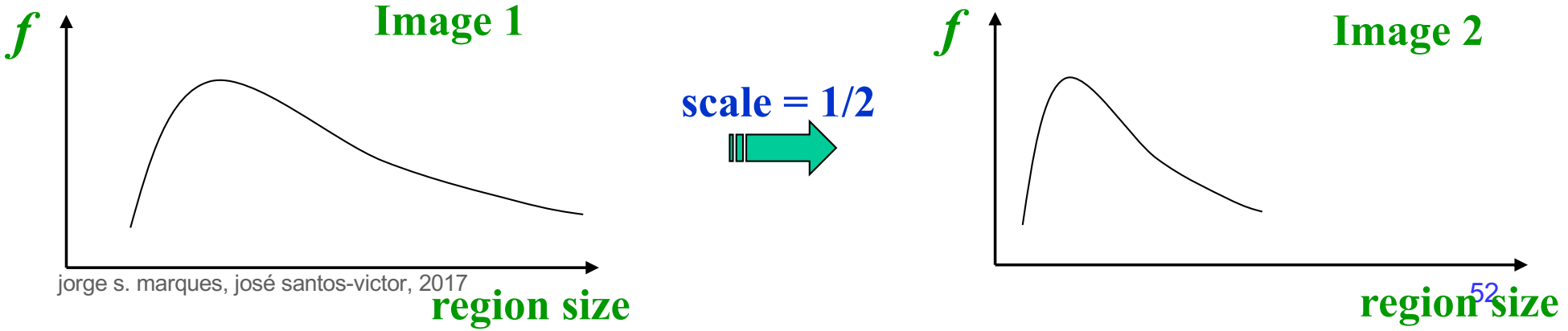


Scale Invariant Detection

- Solution:
  - Design a function on the region (circle), which is "scale invariant" (the same for corresponding regions, even if they are at different scales)

Example:
average intensity. For corresponding regions (even of different sizes) it will be the same.

  - For a point in one image, we can consider it as a function of region size (circle radius)



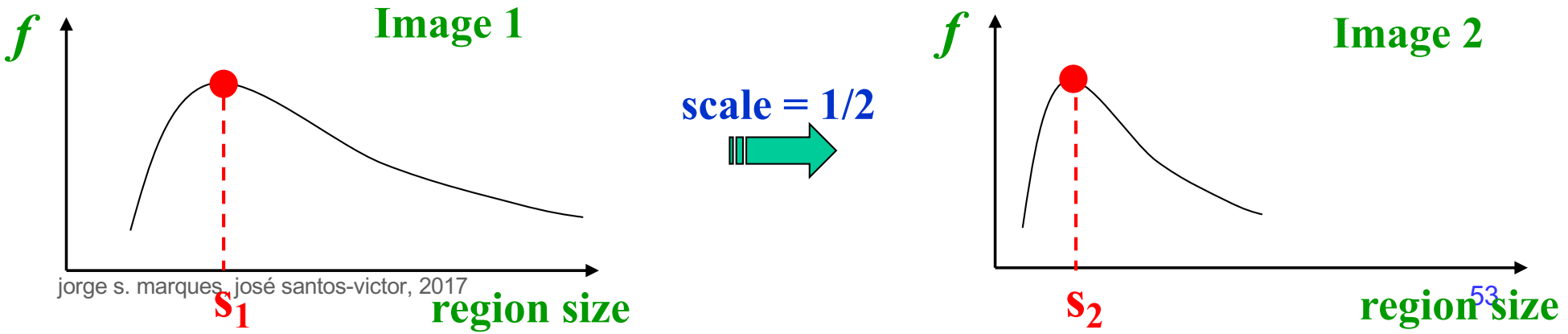
Scale Invariant Detection

- Common approach:

  Take a local maximum of this function

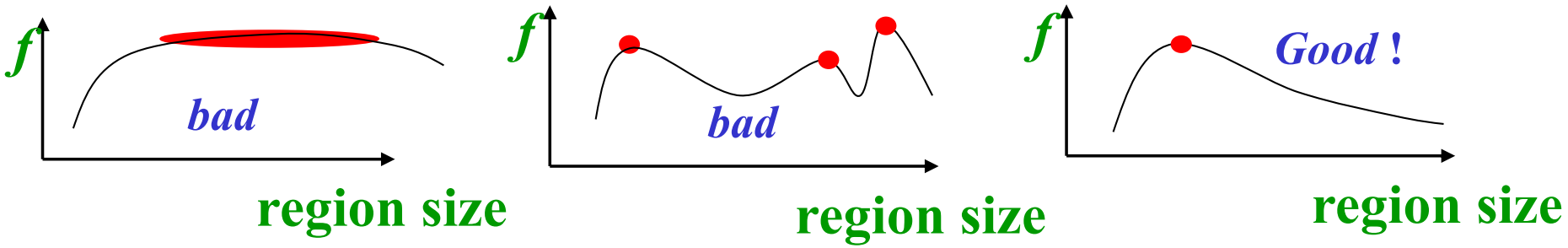
  Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.

Important: this scale invariant region size is found in each image independently!



Scale Invariant Detection

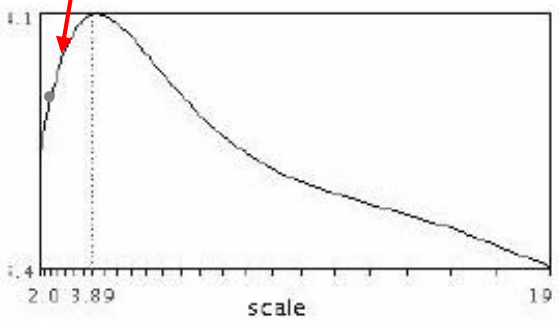
- A "good" function for scale detection:
  has one stable sharp peak



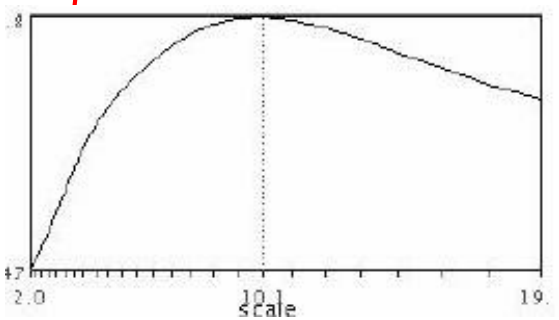
- **For usual images: a good function would be a one which responds to contrast (sharp local intensity change)**

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



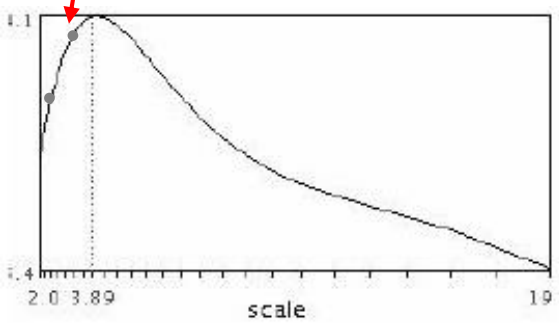
$$f(I_{i_1 \ldots i_m}(x,\sigma))$$



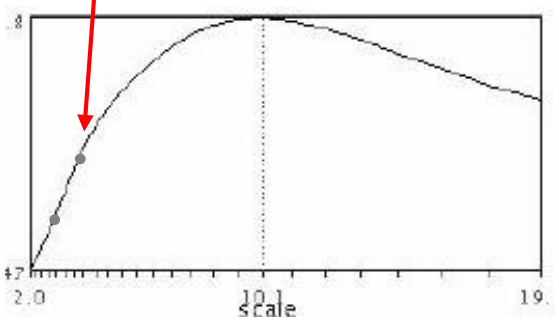
$$f(I_{i_1 \ldots i_m}(x',\sigma))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



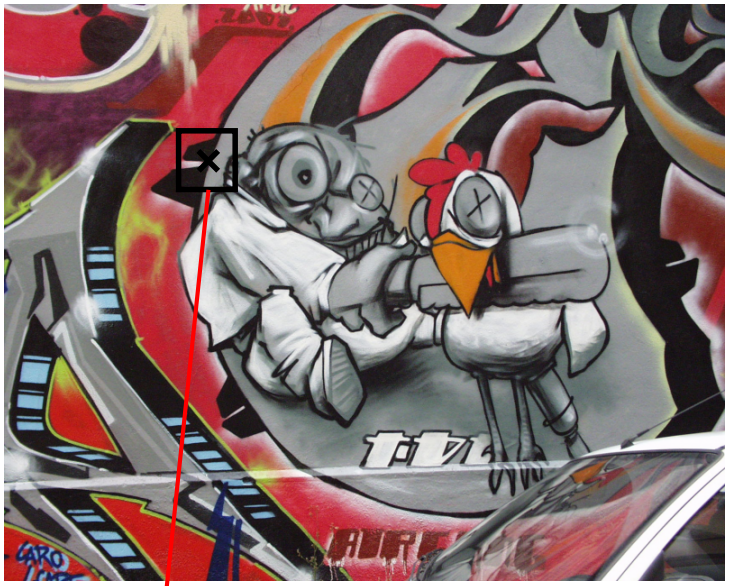
$f(I_{i_1 \ldots i_m}(x,\sigma))$

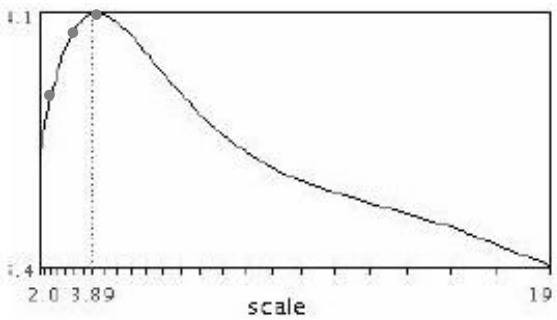


$f(I_{i_1 \ldots i_m}(x',\sigma))$

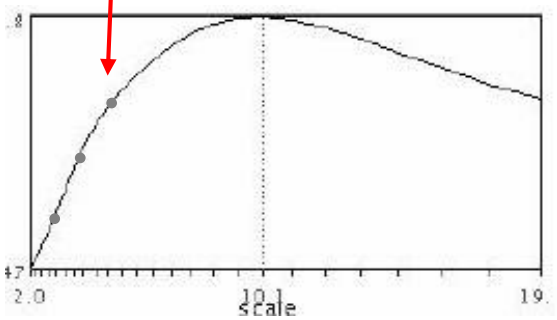
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





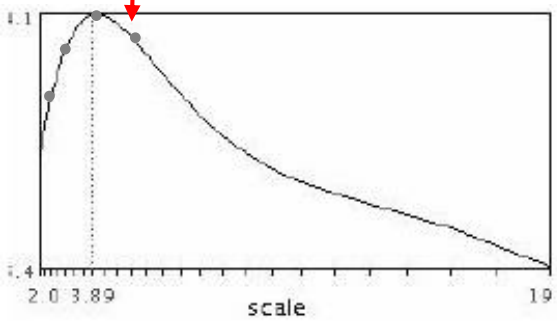
$$f(I_{i_1 \ldots i_m}(x,\sigma))$$



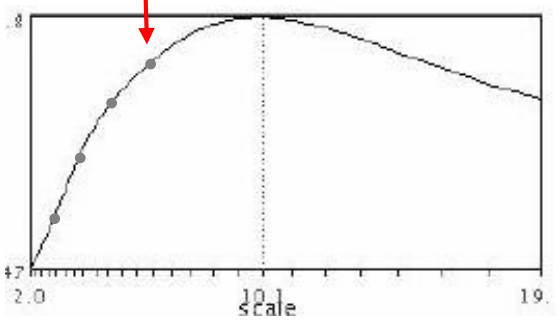
$$f(I_{i_1 \ldots i_m}(x',\sigma))$$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



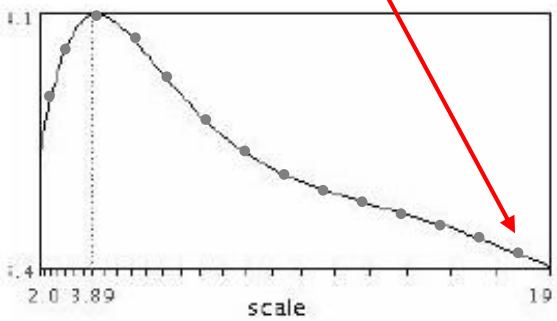
$f(I_{i_1 \ldots i_m}(x,\sigma))$



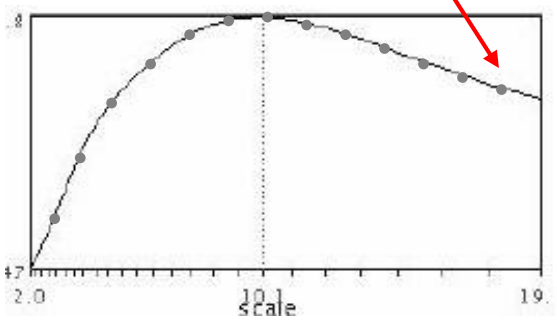
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



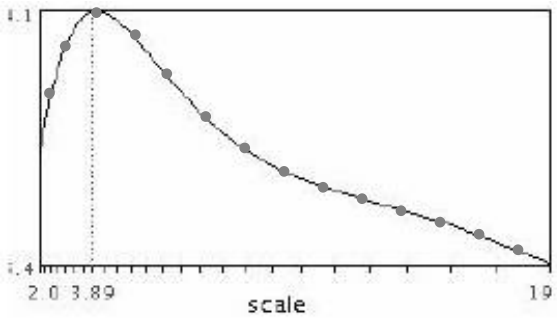
$f(I_{i_1 \ldots i_m}(x,\sigma))$



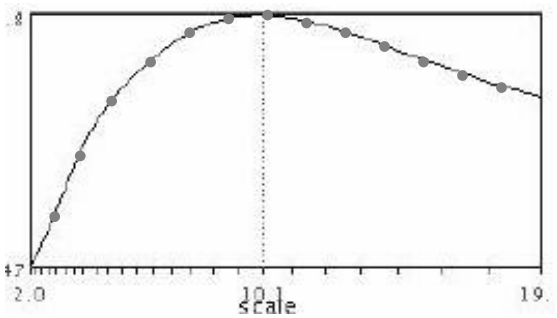
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$

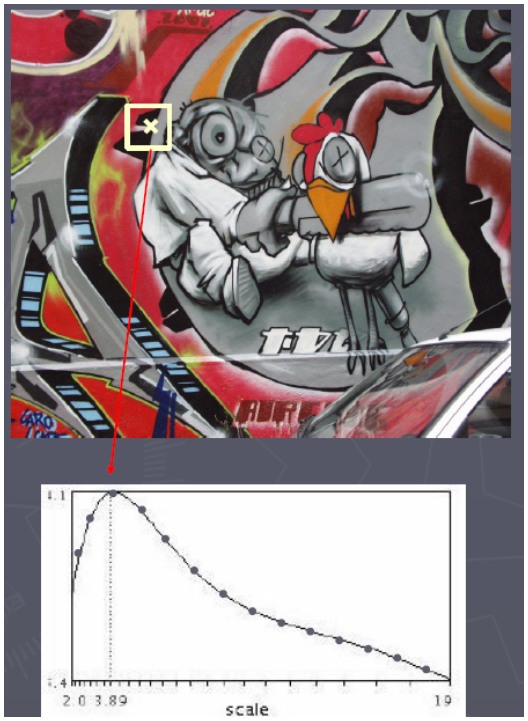


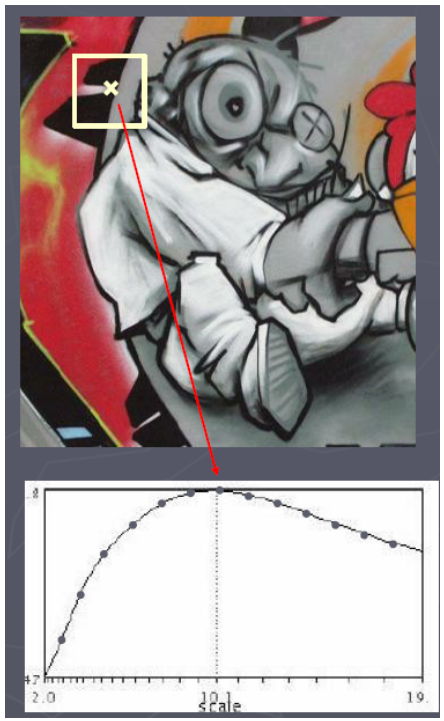
$f(I_{i_1 \ldots i_m}(x',\sigma'))$

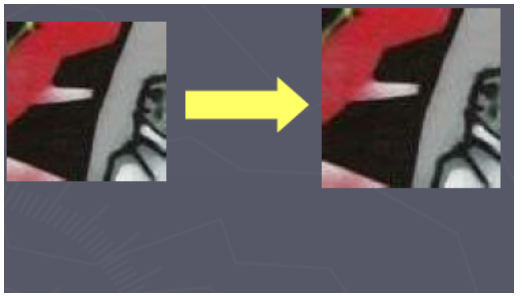
K. Grauman, B. Leibe

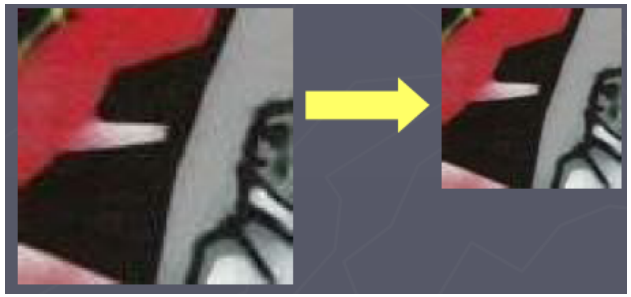
Scale selection

- Use the scale determined by detector to compute descriptor in a normalized frame





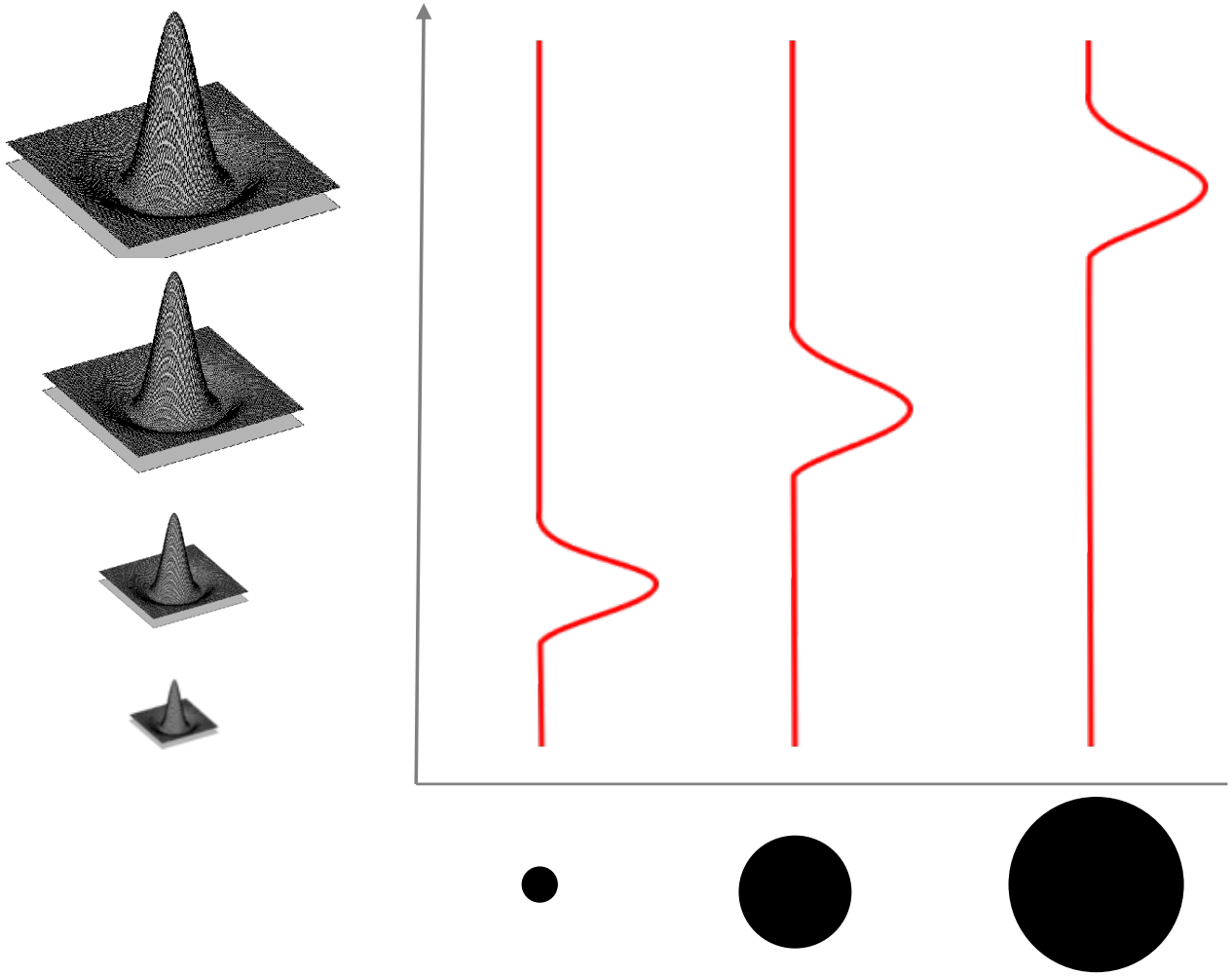




[Images from T. Tuytelaars]

what is a useful signature function?

- Laplacian-of-Gaussian = "blob" detector



K. Grauman, B. Leibe

dealing with scale

If the scale of the patches is unknown, the detector should be stable with respect to scale.

Two approaches:

Laplacian of Gaussian: $L(x, y, \sigma) = \nabla^2 G(x, y, \sigma) * \mathrm{I}(x, y)$ Lindberg, 1993

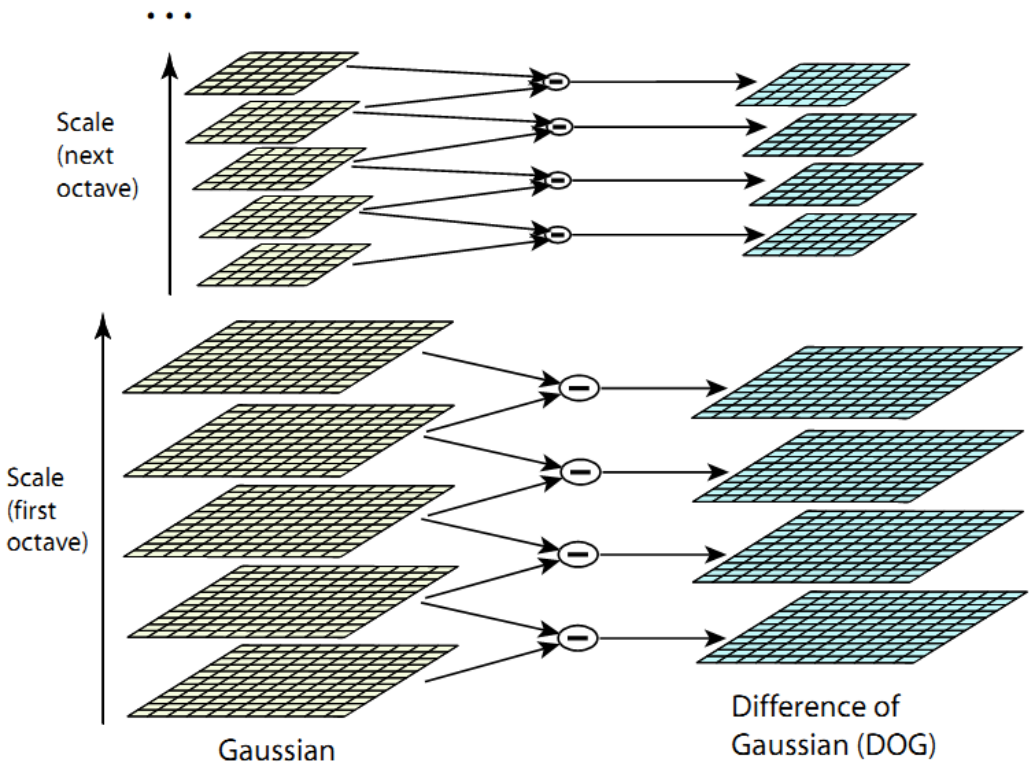
Gaussian kernel

Difference of Gaussians: (SIFT) $D(x, y, \sigma) = G(x, y, \sigma) * \mathrm{I}(x, y) - G(x, y, k\sigma) * \mathrm{I}(x, y)$ Lowe, 2004

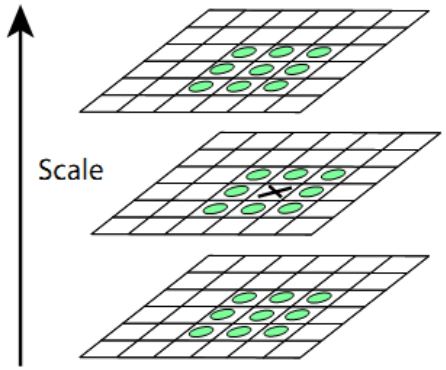
approximate solution

Keypoints located at located at local minima and maxima. They provide (x,y) coordinates and scale.

Scale invariante keypoint detection

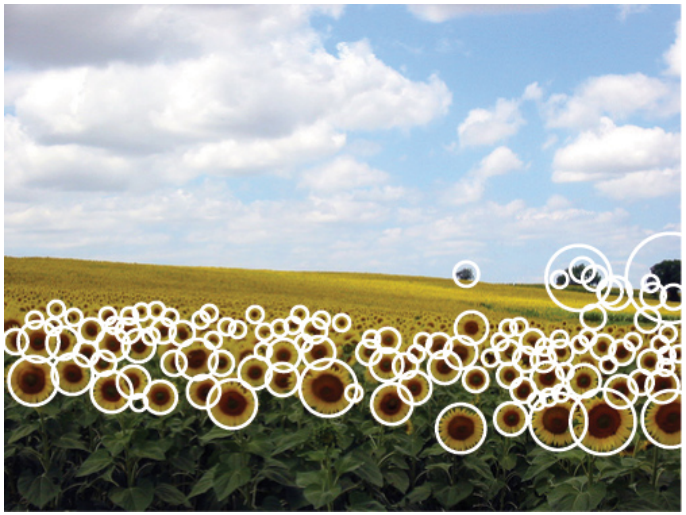


difference of Gaussians (DoG) representation



detection of local maxima and minima

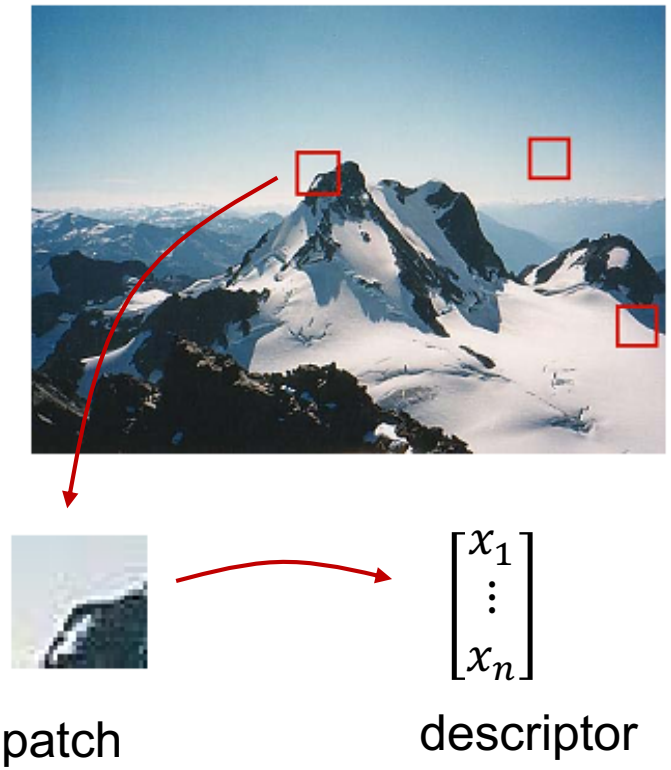
Patch representation



Desired properties

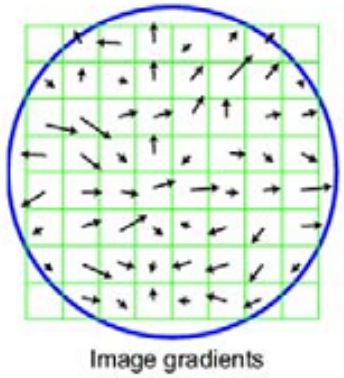
Keypoint should be represented by a small set of features (descriptor), robust with respect to the following changes:

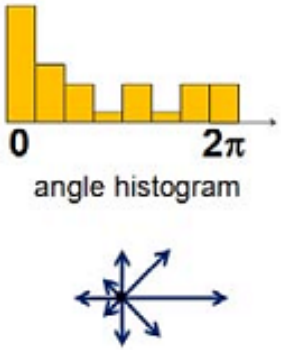
- translation ✓
- rotation
- scaling ✓
- affine transforms
- intensity changes



Feature representation (SIFT)

invariance to rotation

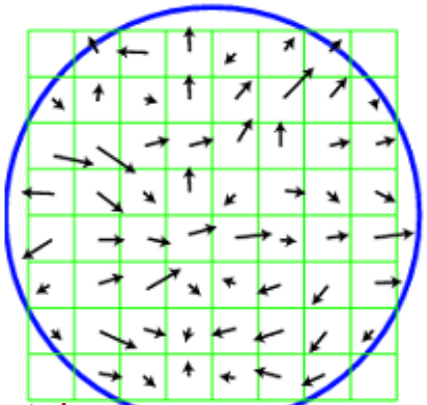




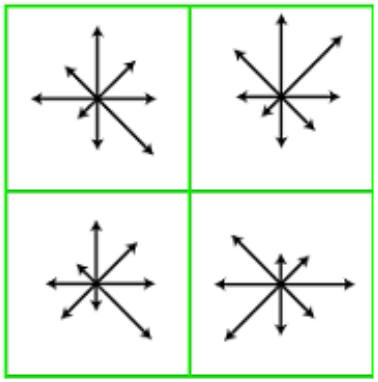
32 bins

local histograms

This figure is simplified. The SIFT features use 4x4 local histograms.



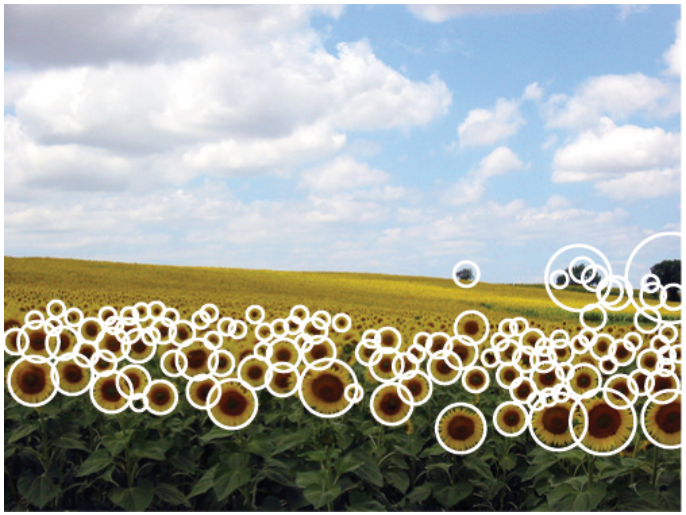
rotated patch



4x4 histograms of 8 bins (not 2x2)

The descriptor contains 4x4x8=128 features

Matching



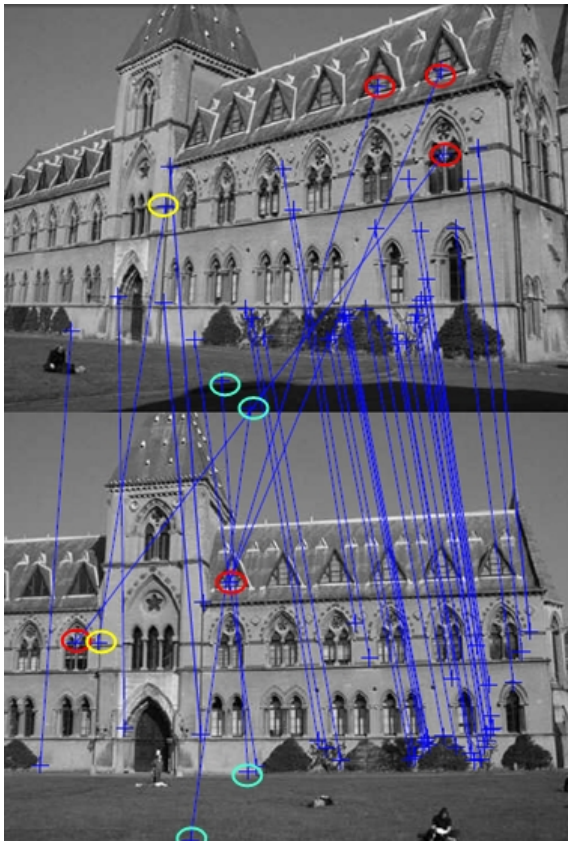
Keypoint matching

Given a set of keypoints and descritors of 2 images

image I_0: $\{x_{0i}, d_{0i}\}$

image I_1: $\{x_{1j}, d_{1j}\}$

how do we match the keypoints in both images?

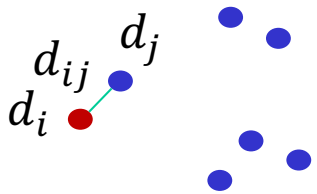


Imaging.utk.edu

Keypoint matching (SIFT)

Given two feature vectors $d_{0i,}$ d_{1j} extracted from different images we can measure their similarity by computing the Euclidean norm

$$d_{ij}=\|d_i - d_j\|$$

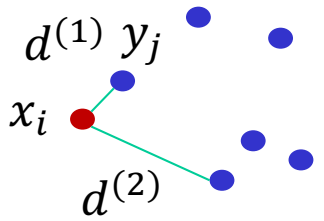


For each $d_{0i,}$ we select the nearest d_{1j} and accept the match if the distace is small enough

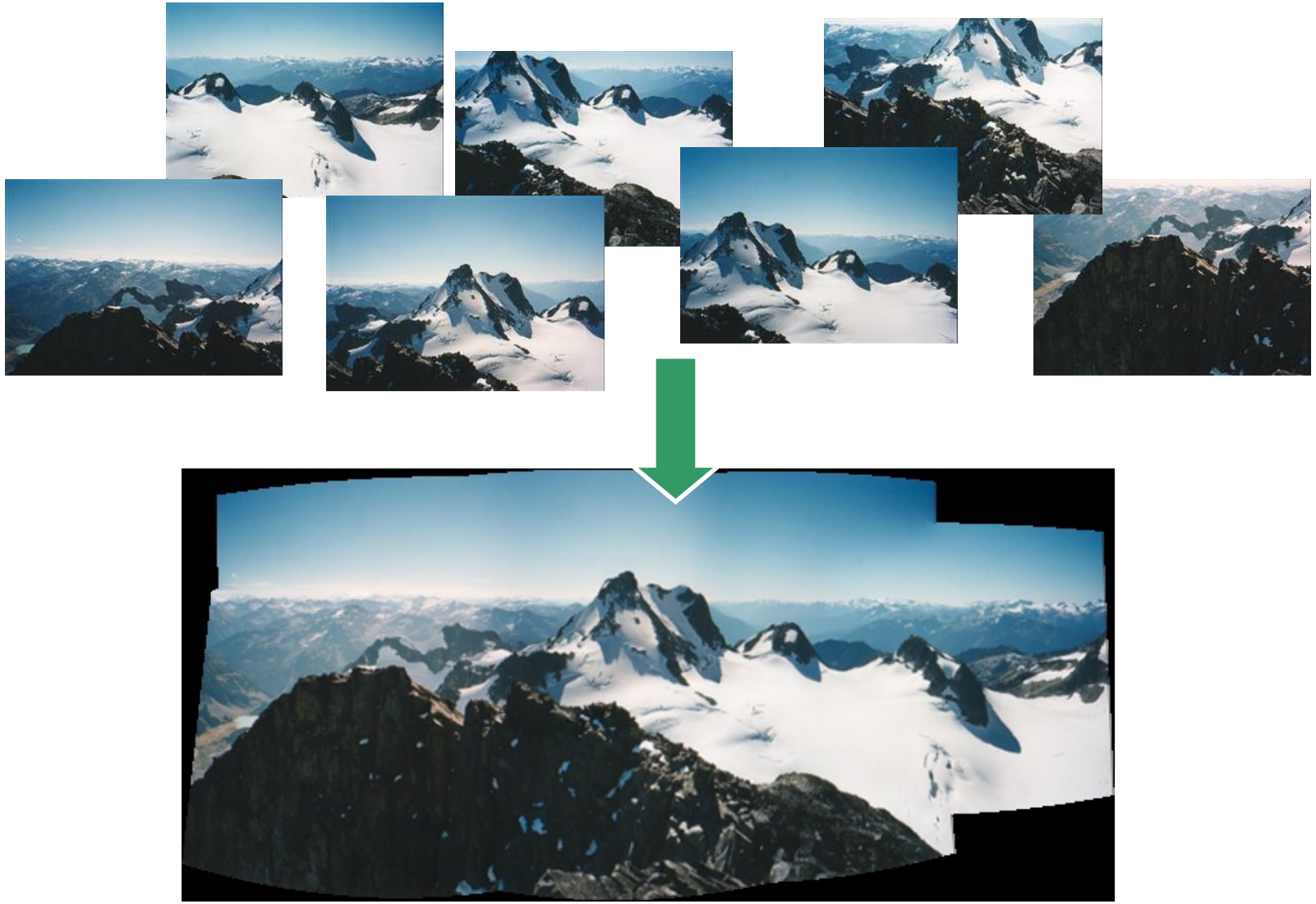
$$\min_j d_{ij} <\lambda$$

This criteria does not work well in practice. Better results are obtained if we use the two closest neighbors

$$\frac{d^{(1)}}{d^{(2)}} < \lambda$$



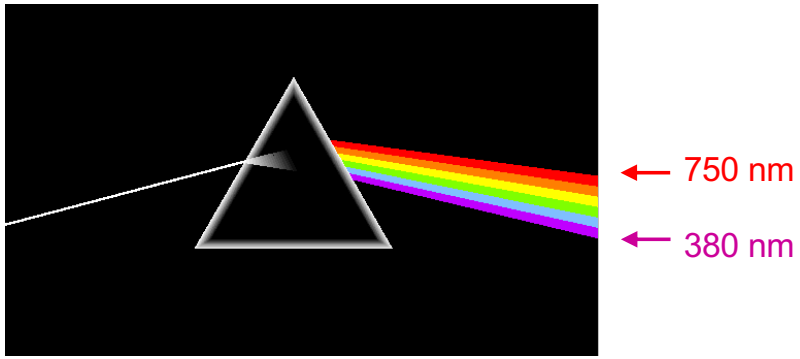
Application: Image Stitching



[Microsoft Digital Image Pro version 10]

color

Visible spectrum



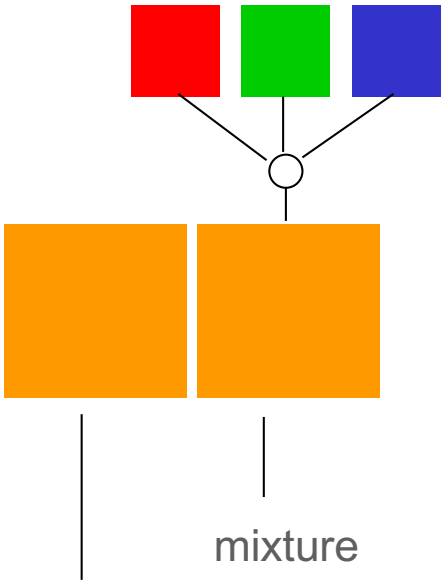
380 nm 750 nm

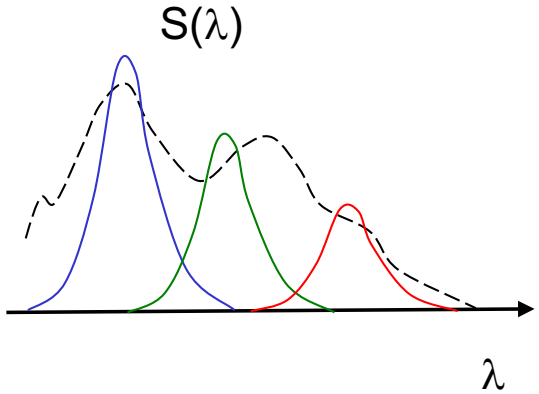
Trichromatic theory

Proposed by Tomas Young (XVIII Cent.) and Helmoltz (XIX Cent.)

Helmoltz showed that any color (?) can be synthesized by a mixture of three primary colors.

experience





$$\hat{S}(\lambda) = c_1 P_1(\lambda) + c_2 P_2(\lambda) + c_3 P_3(\lambda)$$

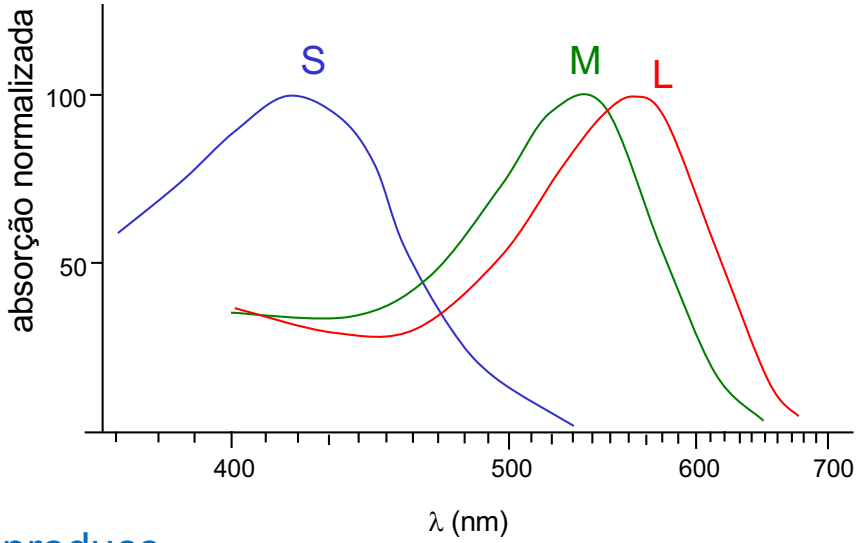
counterintuitive, with 3 primary colors we cannot approximate a given spectrum S(λ) well!

Young believed that there were 3 color sensitive structures in the human retinae. True?...

Retina

As Young guessed, there are three types of photosensitive cells in the retina, called cones.

The three types of cones have different spectral sensitivity (absorption).



Two incident spectra may be different and still produce the same color perception. → meromorfism

The response of the i^{th} cone to a light source with spectrum $S(\lambda)$ is

$$R_i(S) = \int S(\lambda) S_i(\lambda) d\lambda$$

Color perception caused by two spectra is the same if the output of the three cones is the same.

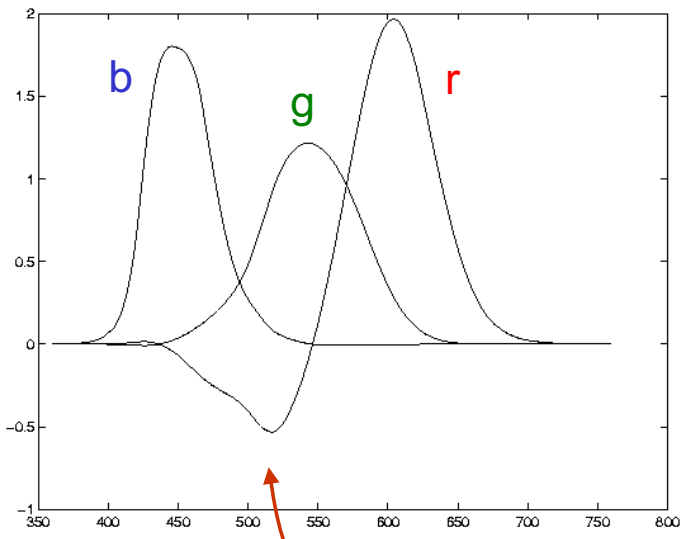
RGB color space

CIE, 1931 – primary spectral system

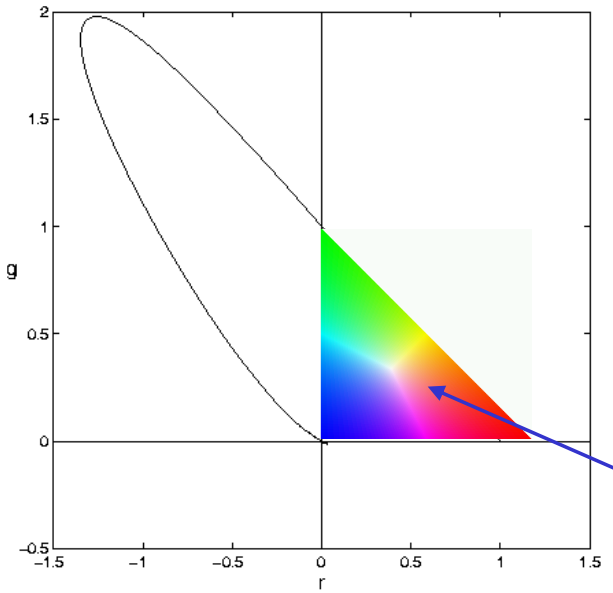
$$P_j(\lambda) = \delta(\lambda - \lambda_j)$$

$\lambda_1 = 700nm$ (R) $\lambda_2 = 546.1nm$ (G) $\lambda_3 = 435.8nm$ (B)

color matching functions



chromaticity diagram



$$r = \frac{R}{R+G+B}$$

$$g = \frac{G}{R+G+B}$$

feasible colors

mixture coefficients may be negative?

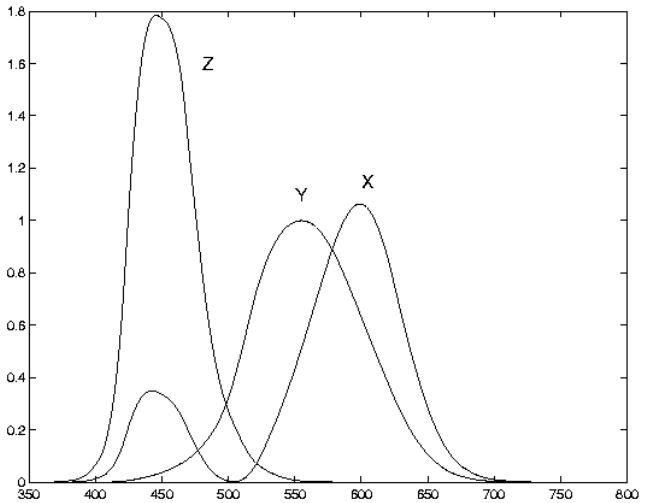
XYZ color space

CIE

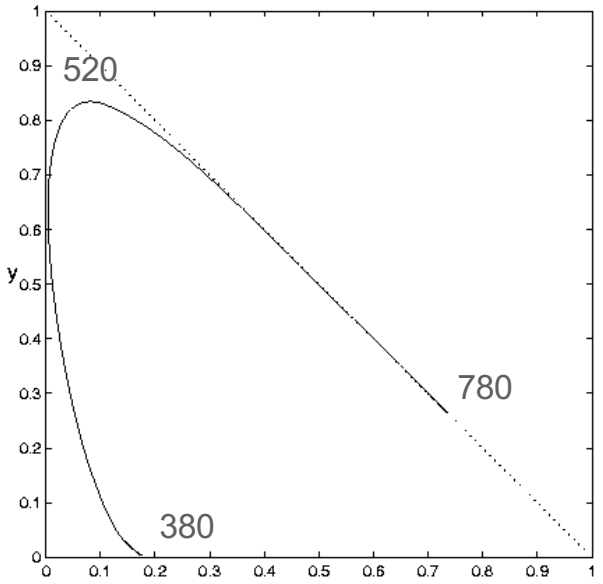
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Y - luminance

color matching functions



chromaticity diagram

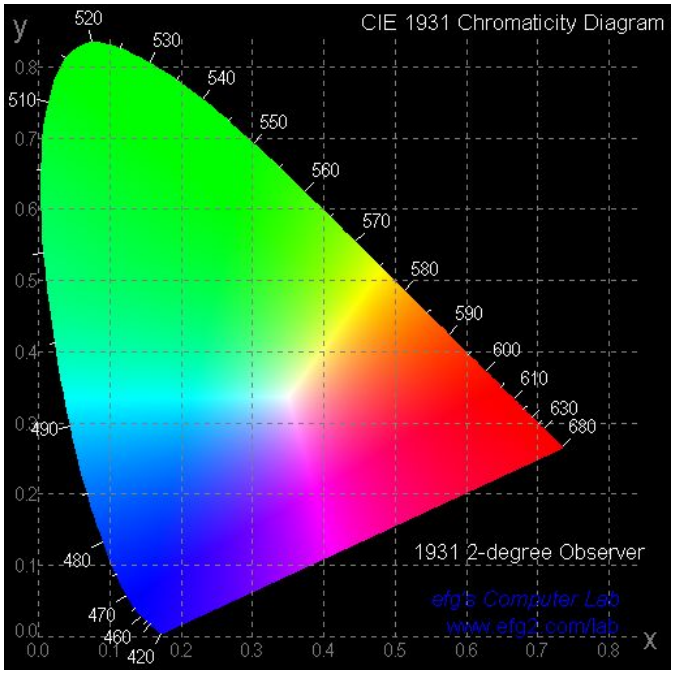


$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

The coefficients are always positive but primary colors are unfeasible!

Chromaticity diagram xy, CIE 1931



Color components

R

G

B

Lab colour space

CIE 1976 – non linear coordinates

$$L^* = 116\left(\frac{Y}{Y_0}\right)^{1/3} \text{-16}$$

$$a^* = 500\left[\left(\frac{X}{X_0}\right)^{1/3} - \left(\frac{Y}{Y_0}\right)^{1/3}\right]$$

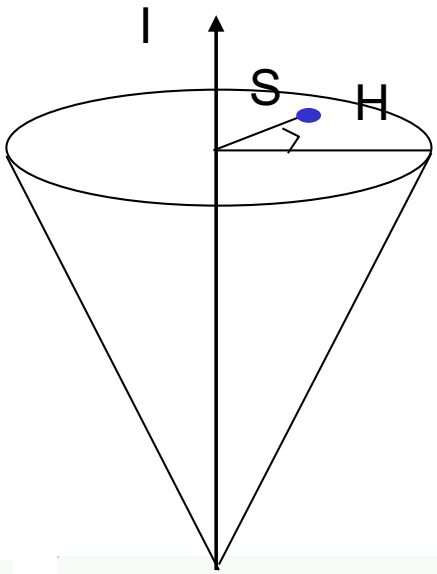
$$b^* = 200\left[\left(\frac{Y}{Y_0}\right)^{1/3} - \left(\frac{Z}{Z_0}\right)^{1/3}\right]$$

L* - brightness
a* - amount of red green
b* - amount of blue-yellow
X_0, $Y_{0,}$, Z_0 – reference white coordinates

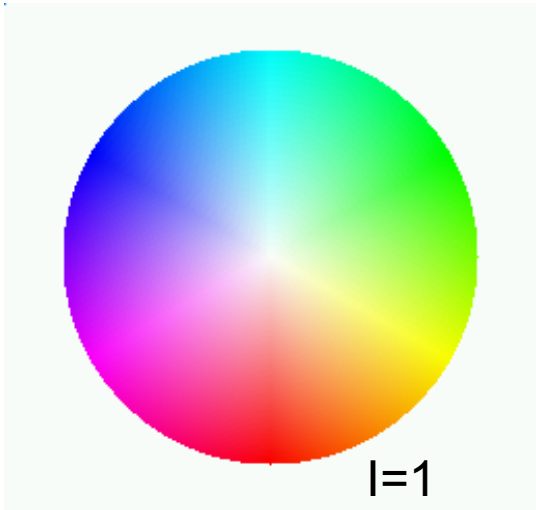
Apropriate for colorimetry: measurement of color differences

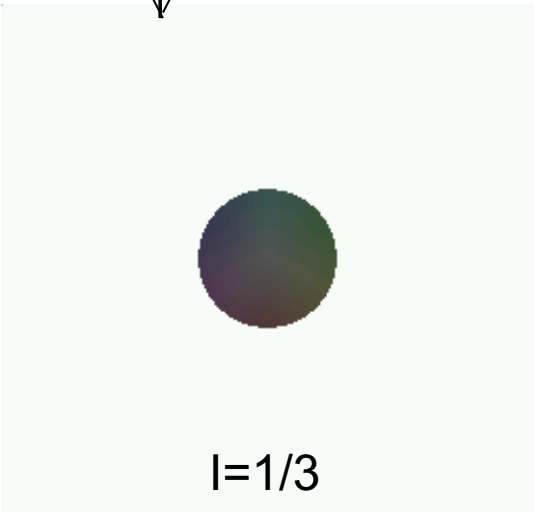
HSI colour space

H – hue
S – saturation
I – intensity



Representation close to the human colour perception





Conversion formulas

$$I = \frac{1}{3}(R + G + B)$$

Gonzalez & Woods

$$S = 1 - \frac{3\min(R,G,B)}{R+G+B}$$

$$H = \begin{cases} \frac{1}{2\pi}\cos^{-1}\left(\frac{0.5\,(R-G+R-B)}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right) & \text{B>G} \\ 1-\frac{1}{2\pi}\cos^{-1}\left(\frac{0.5\,(R-G+R-B)}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right) & \text{B}\leq\text{G} \end{cases}$$

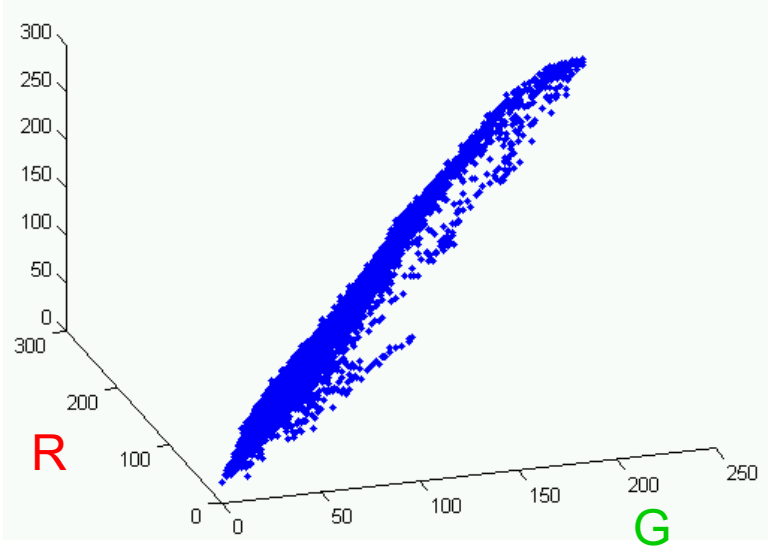
$$B = \frac{1}{3}(1\text{-}S)\,,\quad R = \frac{1}{3}\left[1+\frac{S\cos\theta}{\cos(\theta-\frac{\pi}{3})}\right],\quad G = 1\text{ - }(B+R)\,,\qquad \theta = \cos(2\pi H),\qquad 0 \le H < \frac{1}{3}$$

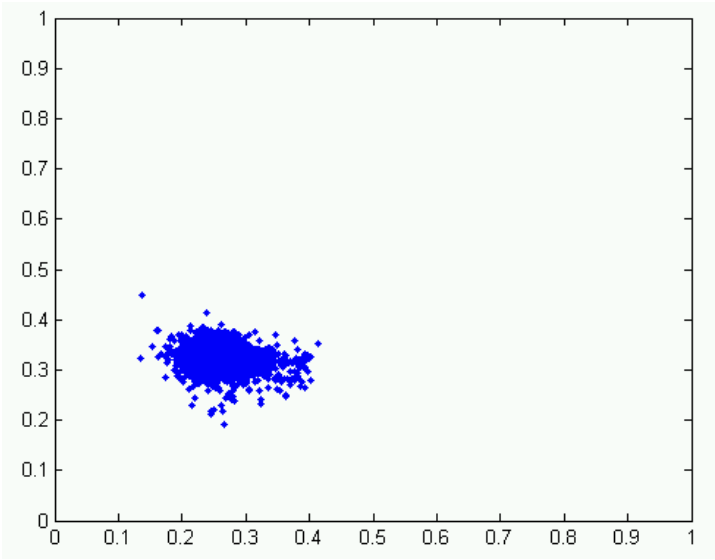
$$R = \frac{1}{3}(1\text{-}S)\,,\quad G = \frac{1}{3}\left[1+\frac{S\cos\theta}{\cos(\theta-\frac{\pi}{3})}\right],\quad B = 1\text{ - }(R+G),\qquad \theta=\cos(2\pi(H-\tfrac{1}{3})),\qquad \frac{1}{3}\le H<\frac{2}{3}$$

$$G = \frac{1}{3}(1\text{-}S)\,,\quad B = \frac{1}{3}\left[1+\frac{S\cos\theta}{\cos(\theta-\frac{\pi}{3})}\right],\quad R = 1\text{ - }(G+B)\,,\quad \theta = \cos(2\pi(H-\tfrac{2}{3})),\qquad \frac{2}{3} \le H < 1$$

How can we describe the content of a color image ?

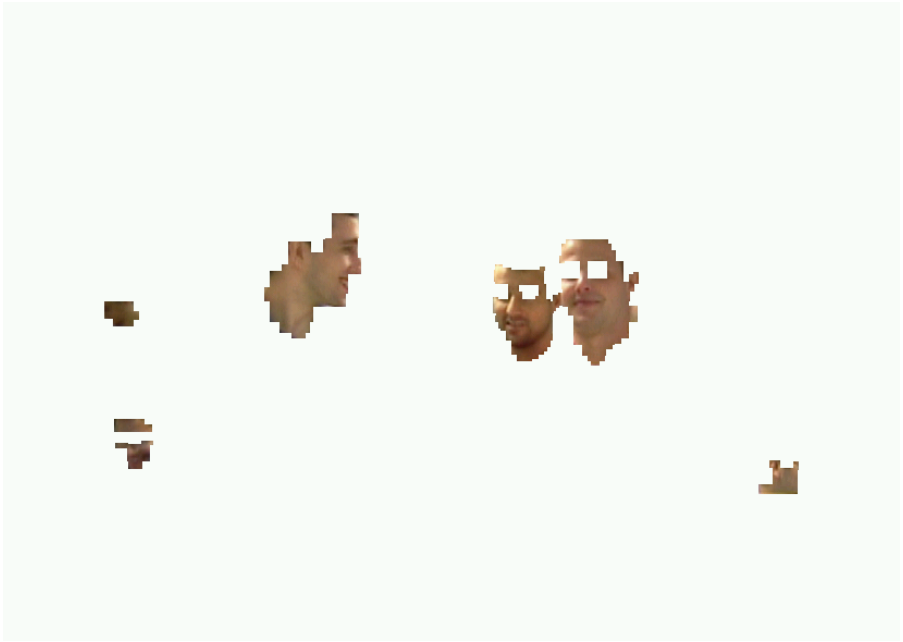
example: shirt

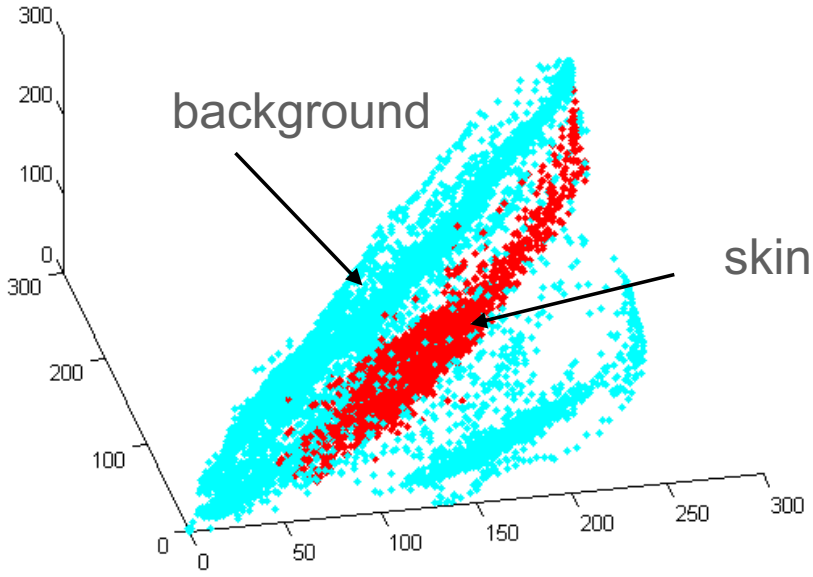


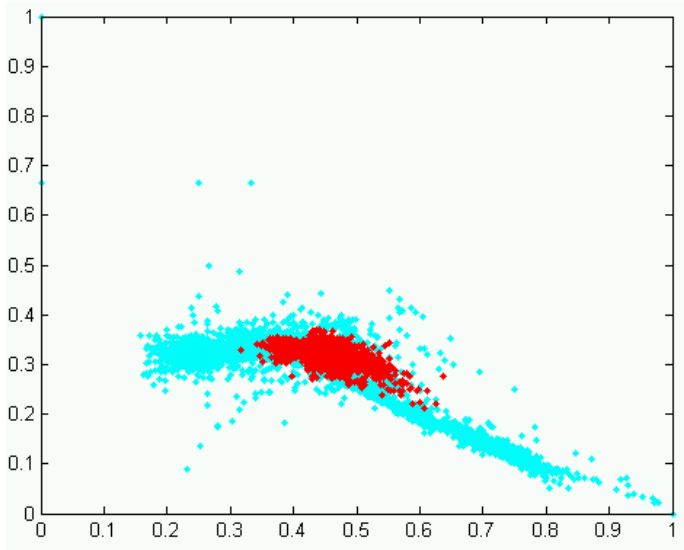


color variations are smaller in rg space

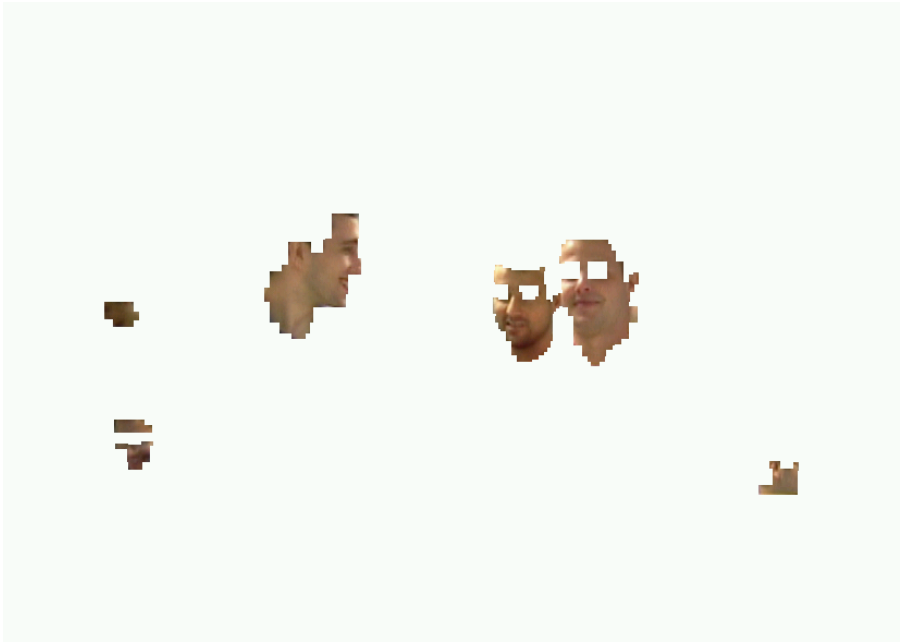
example: skin

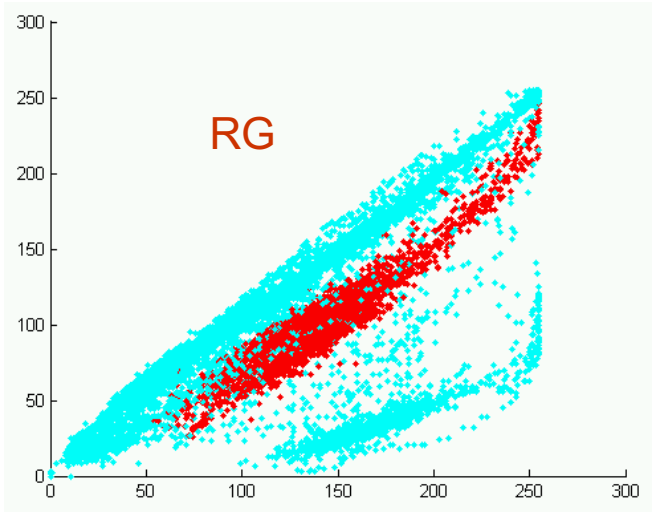


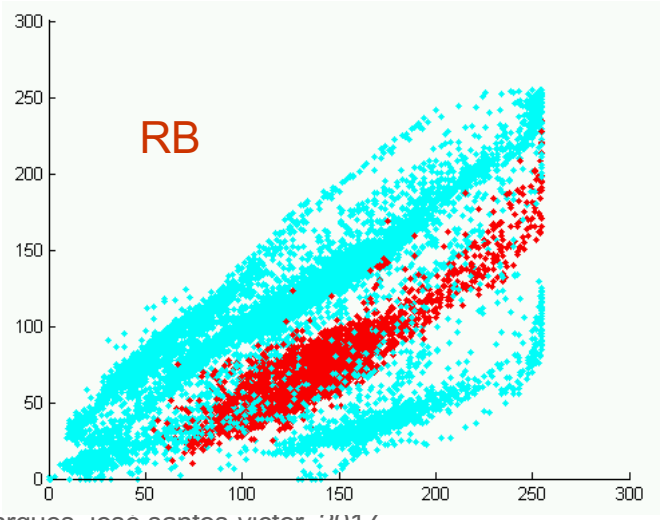


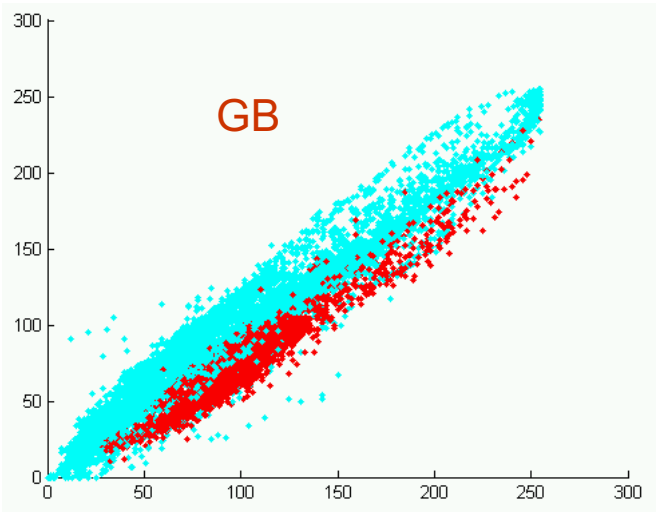


example - skin

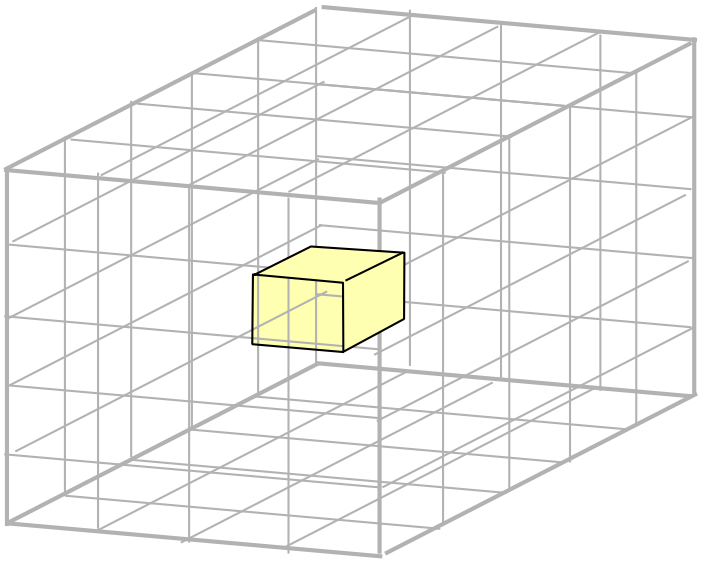








Color histogram



Count the number of pixels falling (voting) in each cell.

h_k - number of pixels in the k-th cell

Spatial information is destroyed in the histogram computation.

The RGB color space is very popular but other color spaces are used as well.

Probabilistic interpretation

We may assume that image pixels are realizations of a discrete random variable with probability distribution

$$P(k) = C\, h_k$$

C – normalization constant

The comparison between a pair of images using their histograms becomes a comparison between probability distributions.

Histogram measures

Euclidean norm $\sum_k [P_1(k) - P_2(k)]^2$

Minimum distance $\sum_k min\{P_1(k), P_2(k)\}$

Kullback divergence $\sum_k P_1(k), \log \frac{P_2(k)}{P_1(k)}$