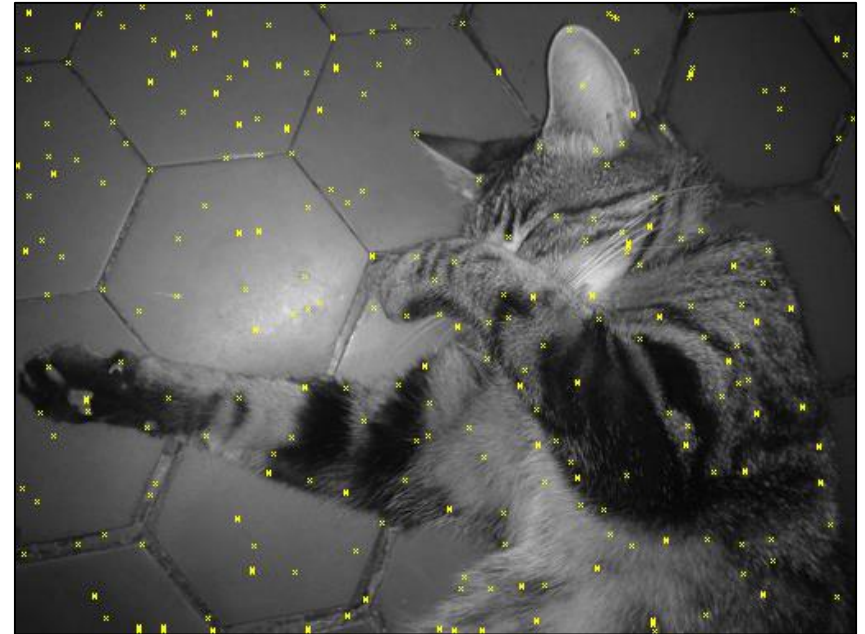
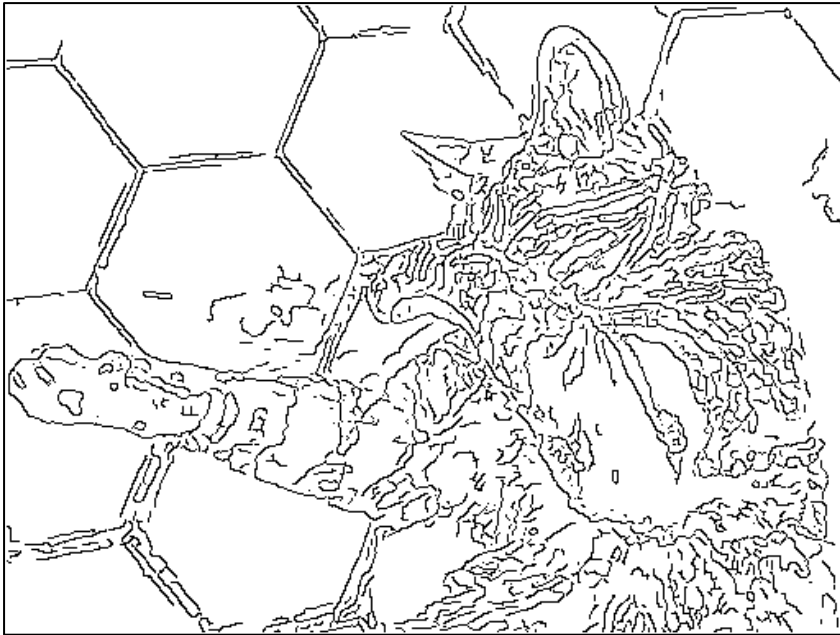


# Image Mining

## *MULTISCALE FEATURE EXTRACTION AND DESCRIPTION*



# MULTISCALE VISUAL FEATURES

Visual features aim at **representing** objects in order to **match** them within images (sequences, pairs, databases, models,...)

Feature extraction in images consists in:

- 1) **Reducing the support** of representation in images to a **significant** and **compact subset**.
- 2) Calculating a **function describing** this subset in a **discriminative, robust** and **efficient** manner.

**Local** characterisation is generally related to local (differential) **geometry**.

**Global** characterisation is generally related to **statistics**.

**Multiscale** estimation allows to:

- 1) Provide a well-founded **formalism** to differential calculus.
- 2) Establish a **continuum** between the local (geometry) and the global (statistics).

# IMAGE MINING: MULTISCALE VISUAL FEATURES

## Lecture outline:

- ❖ Introduction: what is a good visual feature?
- ❖ Basics differential geometry for images
- ❖ Beyond the local: multiscale derivatives
- ❖ Multiscale contour detection
- ❖ Feature points 1: Harris detector
- ❖ Feature points 2: SIFT point detector
- ❖ Local descriptors 1: Hilbert invariants
- ❖ Local descriptors 2: Orientation histograms
- ❖ From the local to the global: Visual Bag-of-Words
- ❖ A global descriptor: Fourier-Mellin invariants

## WHAT IS A GOOD VISUAL FEATURE?

**Goal:** Put in correspondence points / sets / images with other points / sets / images / classes / visual categories.

A good feature should be:

- **Robust:** it should faithfully represent the data without regard to its variation: geometric distortions, illumination changes, occlusions, intra-class variability...
- **Discriminative:** the represented data should be easily distinguished from other data, specially those from its close environment...
- **Efficient:** its computation should be fast, and its memory footprint low...

# BASICS OF DIFFERENTIAL GEOMETRY FOR IMAGES

Local geometry in an image is most naturally described in terms of differential géometry: direction, curvature,...

In the differential model, the image is assimilated to a continuous and differentiable function  $I: \mathbb{R}^2 \rightarrow \mathbb{R}$ .

Then the local behaviour in the image around every point can be predicted by its partial derivatives (Taylor Formula):

$$I(x_0 + \varepsilon, y_0 + \eta) = \sum_{k=0}^r \sum_{i=0}^k \frac{1}{(k-i)!i!} \varepsilon^{k-i} \eta^i \frac{\partial^k I}{\partial x^{k-i} \partial y^i} (x_0, y_0) + o\left((\varepsilon^2 + \eta^2)^{r/2}\right)$$

In discrete images, *derivability* is interpreted as a *local regularity* property.

Since such regularity can be explicitly imposed by filtering (convolution), the estimation of a derivative will be done through a convolution, and as such, will always be relative to a scale (scale spaces).

## ORDER 1: GRADIENT AND ISOPHOTE

At order 1, the basic measure is the gradient vector:

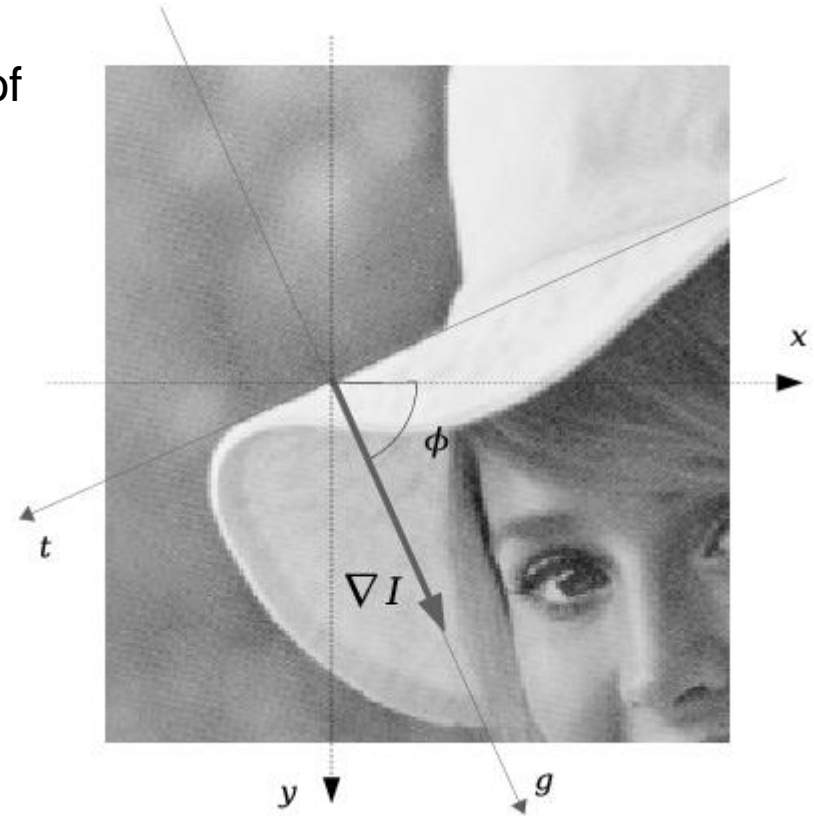
$$\nabla I = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)^T$$

- Its orientation,  $\arg \nabla I$ , corresponds to the direction of steepest ascent.
- Its magnitude,  $\|\nabla I\|$ , measures the local contrast.
- It allows to calculate the first derivative in any direction. Let  $v$  be a unitary vector:

$$\frac{\partial I}{\partial v} = \nabla I \cdot v^T$$

- So in the local frame  $(g, t)$  with  $g = \frac{\nabla I}{\|\nabla I\|}$  and  $t = g^\perp$ :

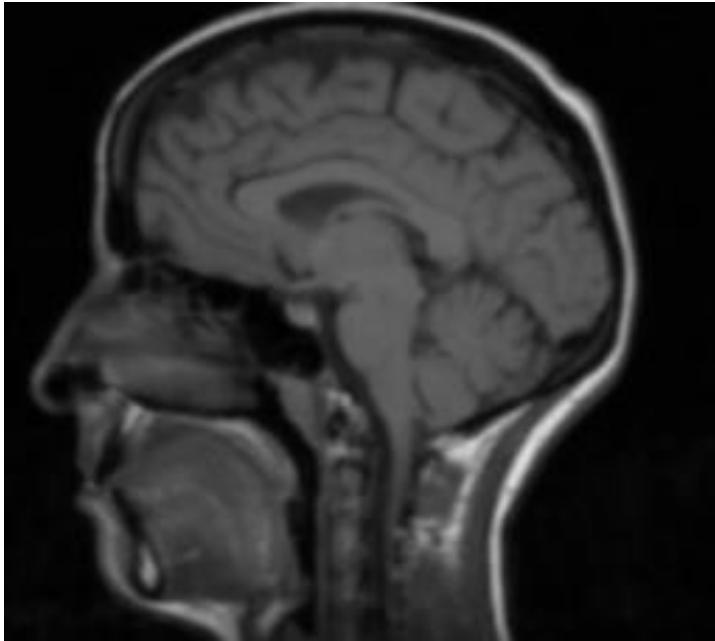
$$\frac{\nabla I}{\nabla g} = \|\nabla I\| \text{ (main direction)} ; \frac{\nabla I}{\nabla t} = 0 \text{ (isophote)}$$



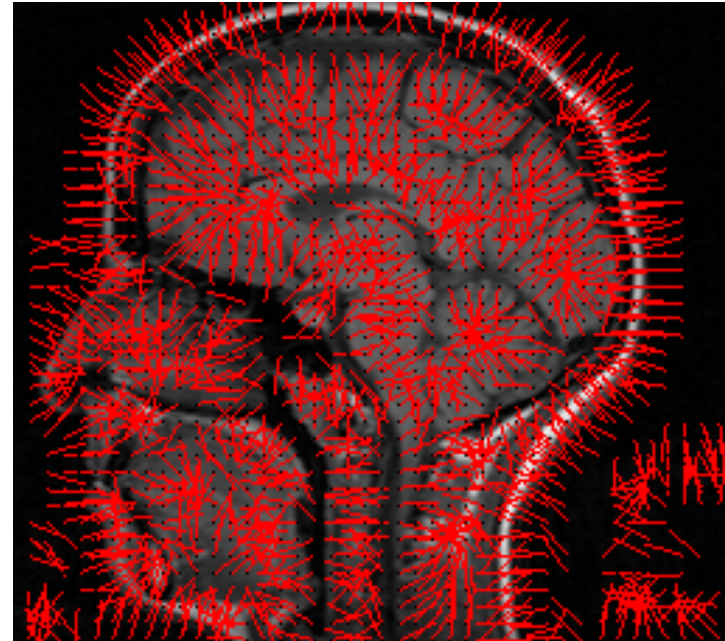


# DIFFERENTIAL QUANTITIES AT ORDER 1

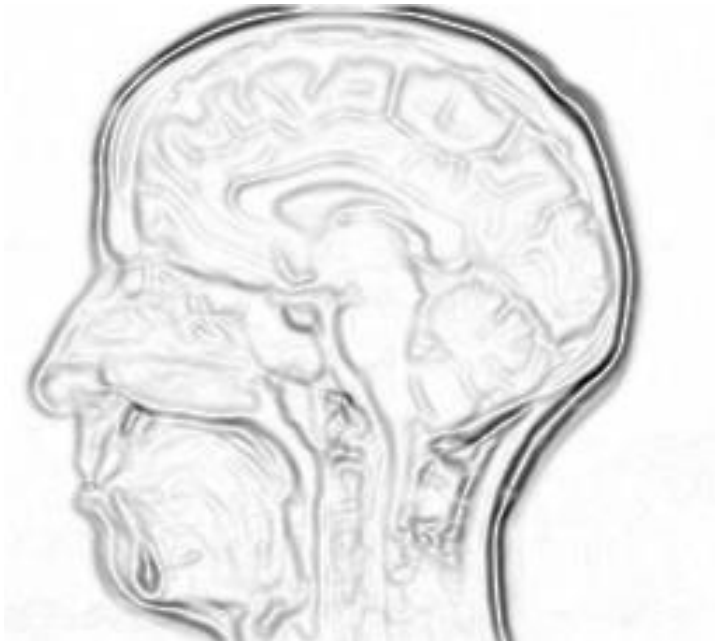
original  
 $I$



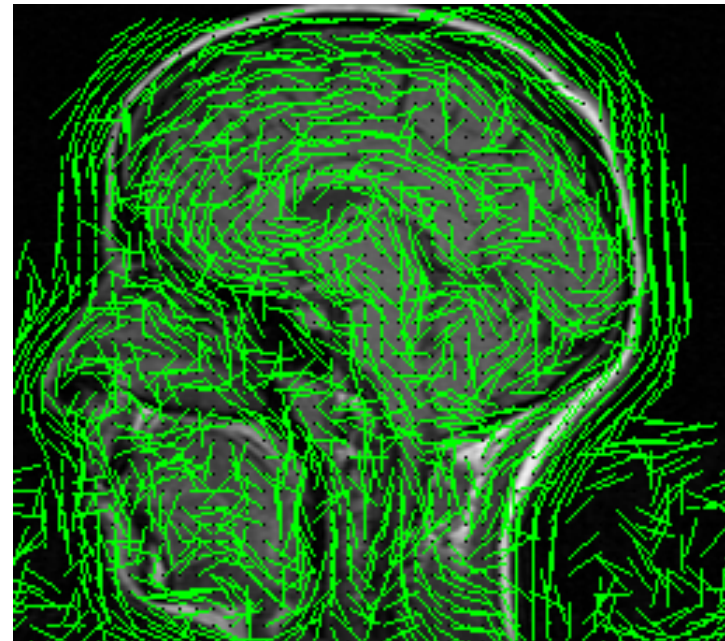
gradient  
direction  
 $\arg \nabla I$



gradient  
magnitude  
 $\|\nabla I\|$



isophote  
direction  
 $\arg \nabla I^\perp$

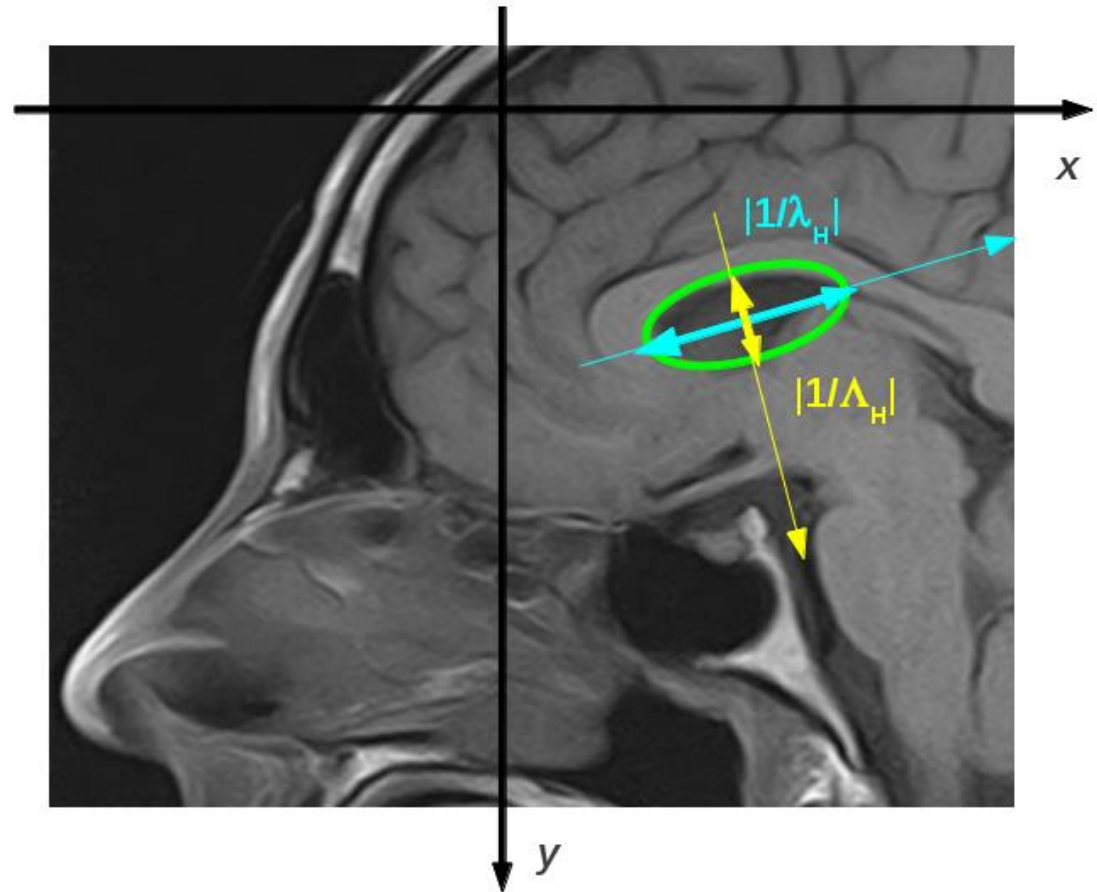


## ORDER 2: HESSIAN AND CURVATURE

At order 2, the basic measure is the Hessian matrix:

- Its eigen vectors (resp. eigen values  $\Lambda_H$  et  $\lambda_H$ ) correspond to principal curvature directions (resp. intensities).
- Its Frobenius norm,  $\|H_I\|_F$ , measures the intensity of global curvature.

$$H_I = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix}$$





## ORDER 2: HESSIAN AND CURVATURE

- Let  $u$  and  $v$  two unit vectors. The second derivative with respect to  $u$  and  $v$  is calculated as follows:

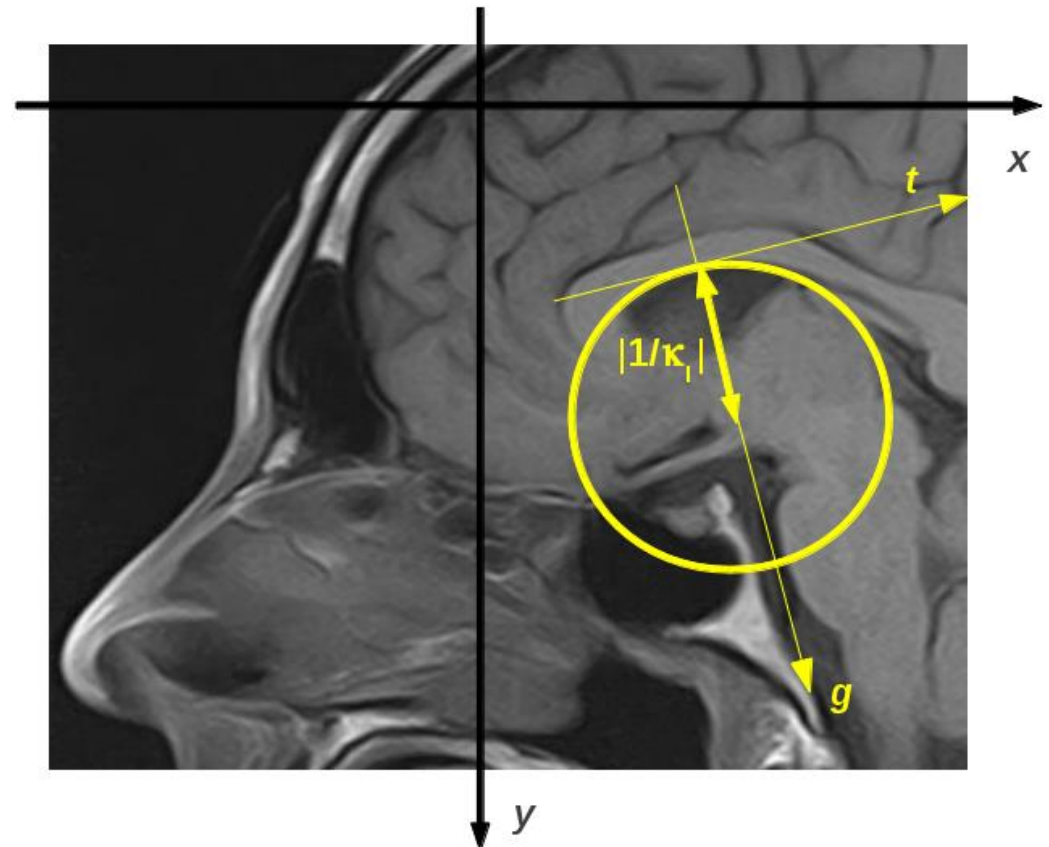
$$\frac{\partial^2 I}{\partial u \partial v} = u^T H_I v$$

- In particular the isophote curvature is related to the inverse radius of the osculating circle to the contour:

$$\kappa_I = -\frac{I_{tt}}{I_g} = -\frac{I_{xx}I_y^2 - 2I_xI_yI_{xy} + I_{yy}I_x^2}{\|\nabla I\|^3}$$

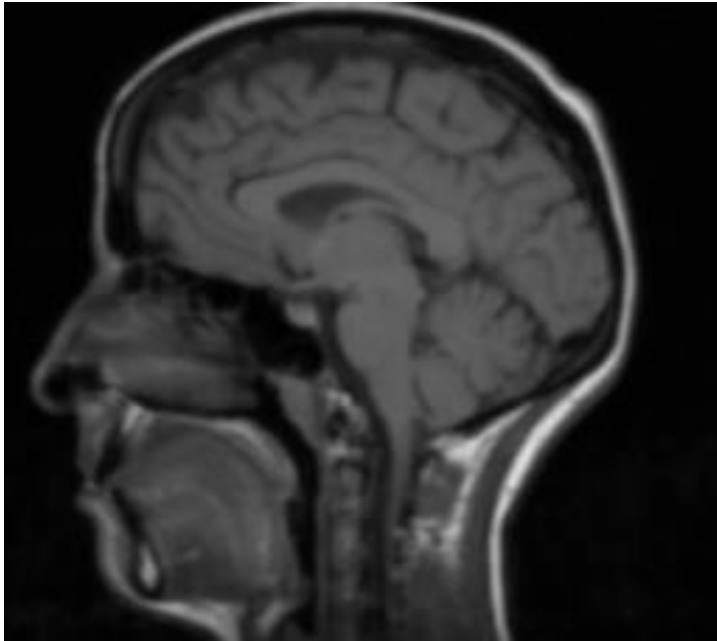
(Notations:

$$I_u = \frac{\partial I}{\partial u}; I_{uv} = \frac{\partial^2 I}{\partial u \partial v}, \text{ etc.})$$

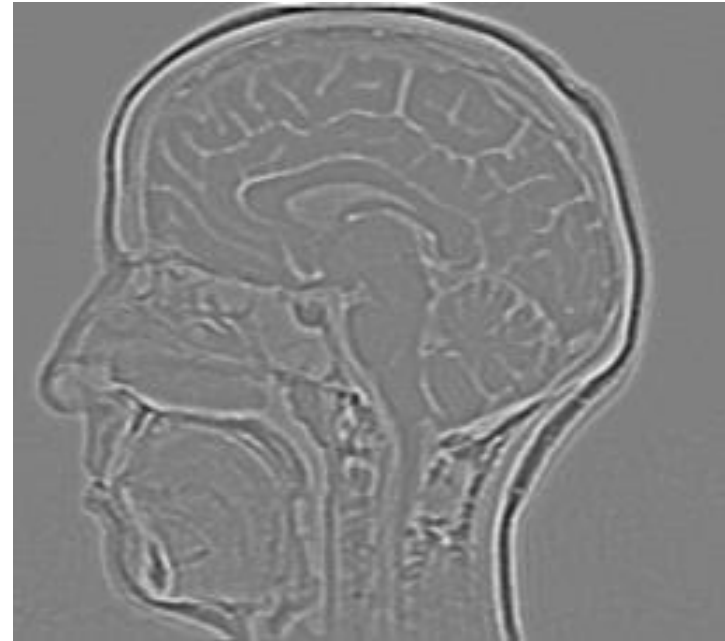


## DIFFERENTIAL QUANTITIES AT ORDER 2

*original*  
 $I$



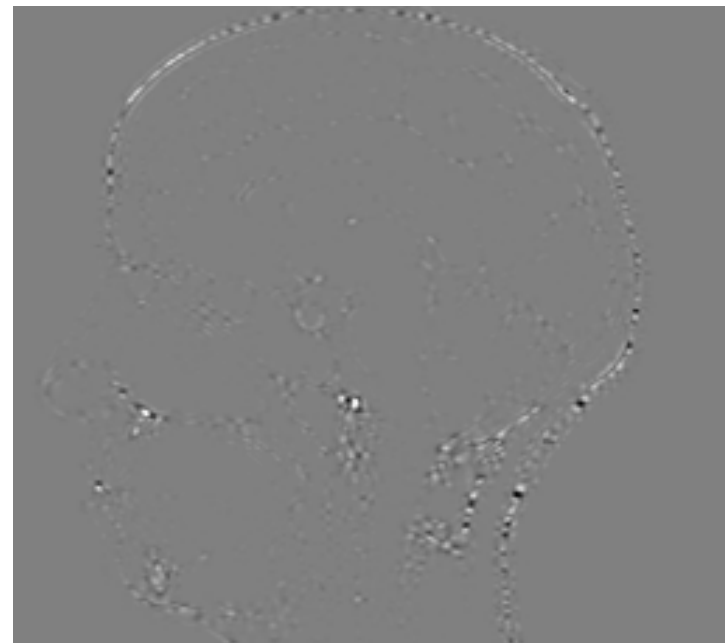
*Hessian trace,  
or total  
curvature  
= Laplacian*  
 $\Delta I$



*Hessian  
norm*  
 $\|H_I\|_F$



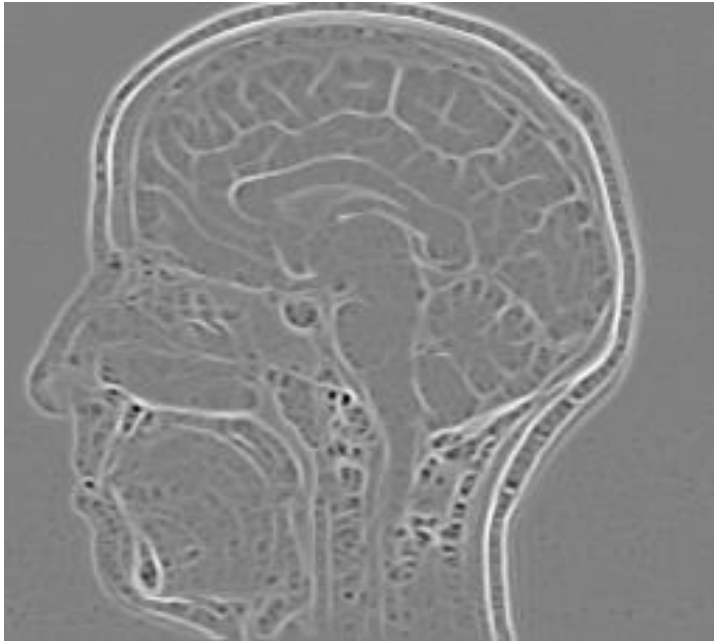
*Hessian  
determinant*  
 $\det\|H_I\|_F$



## DIFFERENTIAL QUANTITIES AT ORDER 2

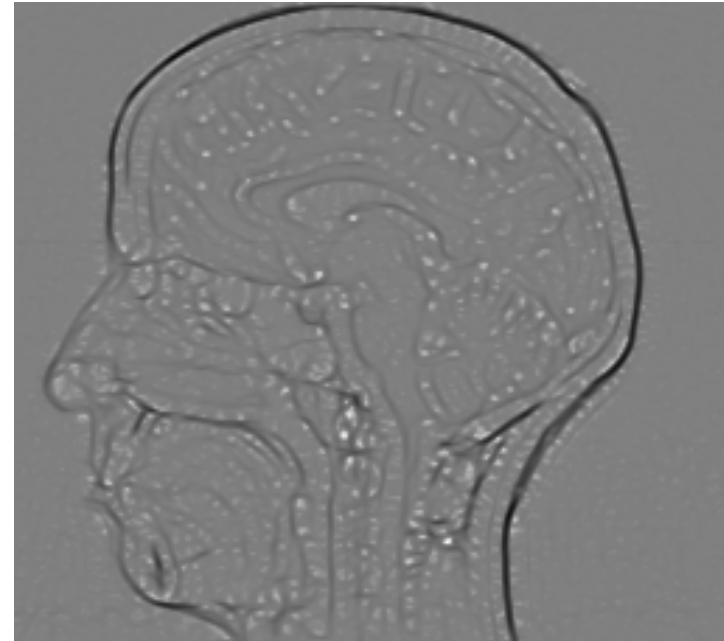
*largest  
eigen  
value*

$$\Lambda_I$$

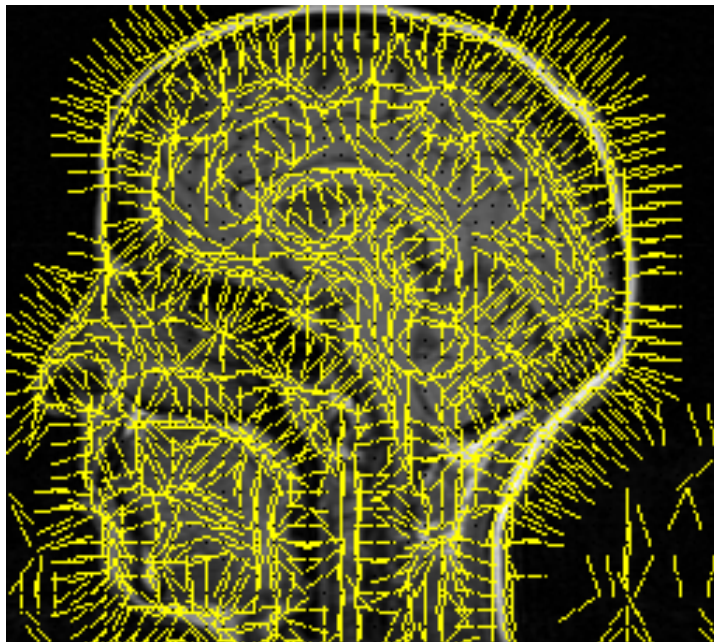


*smallest  
eigen  
value*

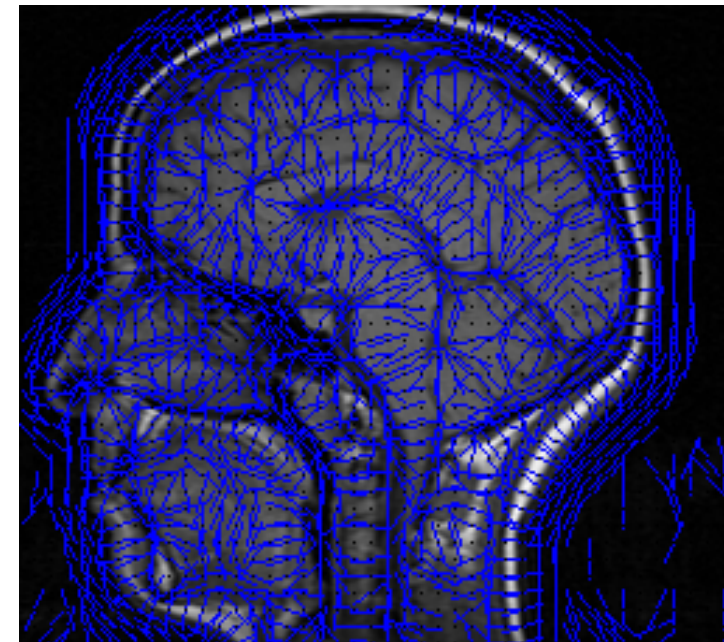
$$\lambda_I$$



*direction  
of "large"  
eigen  
vector*



*direction  
of "small"  
eigen  
vector*



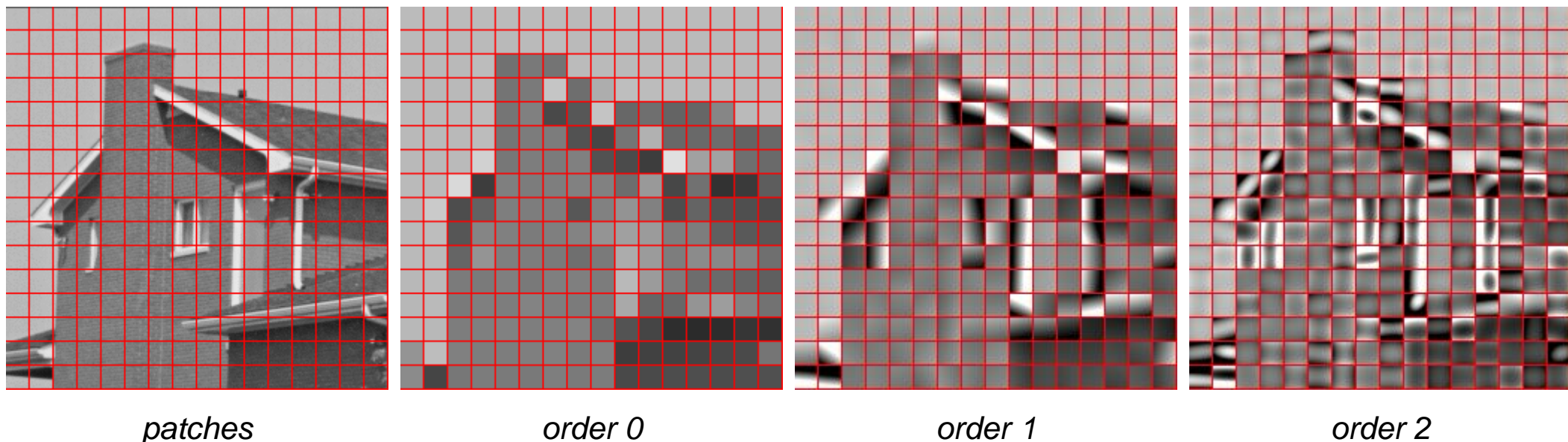


# REPRESENTATION BY LOCAL DERIVATIVES

Expressing Taylor's formula at order 2, using the gradient vector and Hessian matrix:

$$I(x_0 + \varepsilon, y_0 + \eta) = I(x_0, y_0) + (\varepsilon, \eta)^T \cdot \nabla I + \frac{1}{2}(\varepsilon, \eta)^T \cdot H_I \cdot (\varepsilon, \eta) + o(\varepsilon^2 + \eta^2)$$

Reconstructing image patches from partial derivatives estimated at the centre of the patch, at orders 0, 1 and 2 :



*patches*


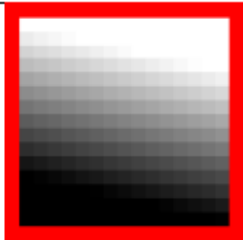
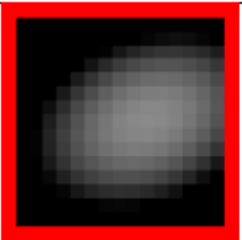
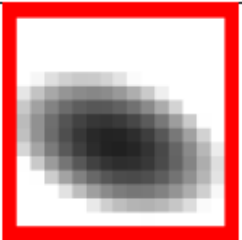
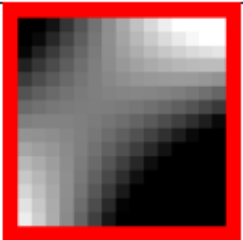
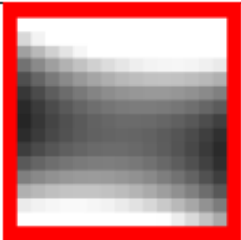
*order 0*

*order 1*

*order 2*

# CATEGORISING IMAGE PATCHES BY THEIR DERIVATIVES

The values of derivatives up to order 2 allow dividing, depending on the dominating order, the local geometry of pixels into 4 categories (6 if considering the polarity):

0	1	2			
$  \nabla_I   \simeq 0$ $  H_I  _F \simeq 0$ Plateau	$  \nabla_I   \gg 0$ $  H_I  _F \simeq 0$ Contour	$  H_I  _F \gg 0$			
		$\Lambda_I \lambda_I > 0$ Courbure elliptique		$\Lambda_I \lambda_I < 0$ Courbure tubulaire	
					
		$\Lambda_I < 0$ $\lambda_I < 0$	$\Lambda_I > 0$ $\lambda_I > 0$	$\Lambda_I < 0$ $\lambda_I > 0$	$\Lambda_I > 0$ $\lambda_I < 0$

## ESTIMATION OF DERIVATIVES AND SCALE SPACES

The key notion of scale spaces for image processing is that any physical (as opposed to mathematical) quantity is relative to an estimation scale.

In particular a derivative only makes sense as estimated to a given scale, corresponding to a regularity hypothesis that is explicitly realised by image smoothing. This estimation is based on the commutativity property that links derivation and convolution:

$$\partial^n(I \star g) = I \star (\partial^n g)$$

In the Gaussian scale space framework, the convolution kernel  $g$  is identified to the 2d Gaussian kernel with standard deviation  $\sigma$ :

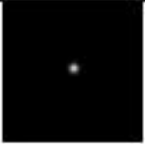




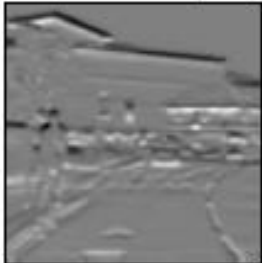

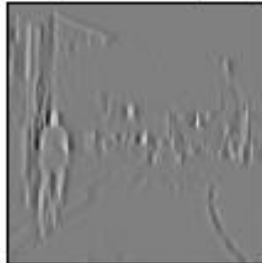

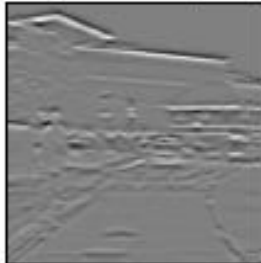

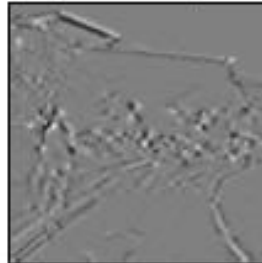
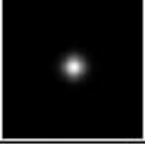

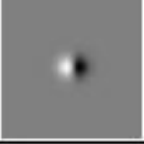
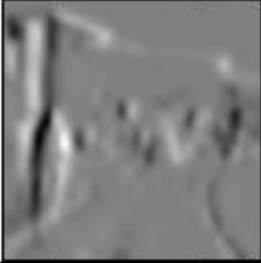
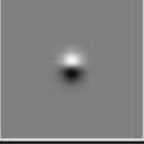
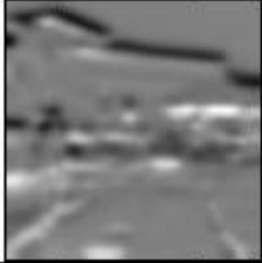

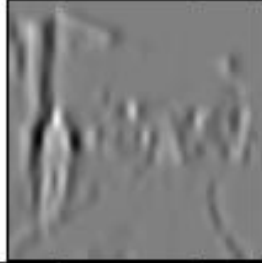

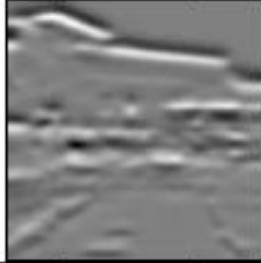
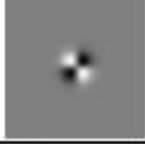
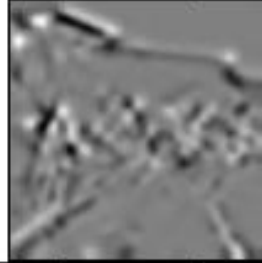


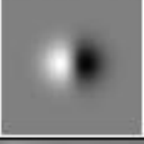


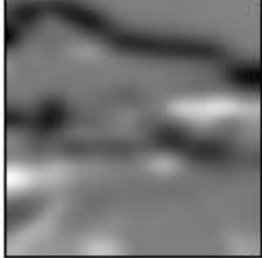
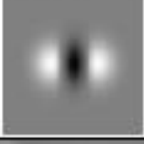

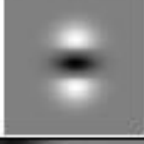
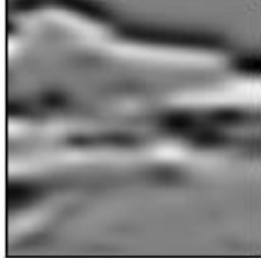
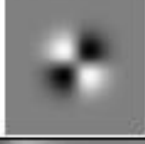
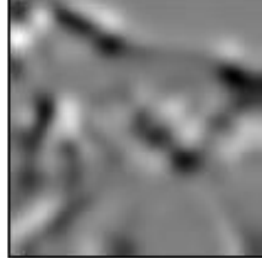
$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The derivatives of image  $I$  estimated at scale  $\sigma$  are thus defined by the convolutions with the corresponding Gaussian derivatives:

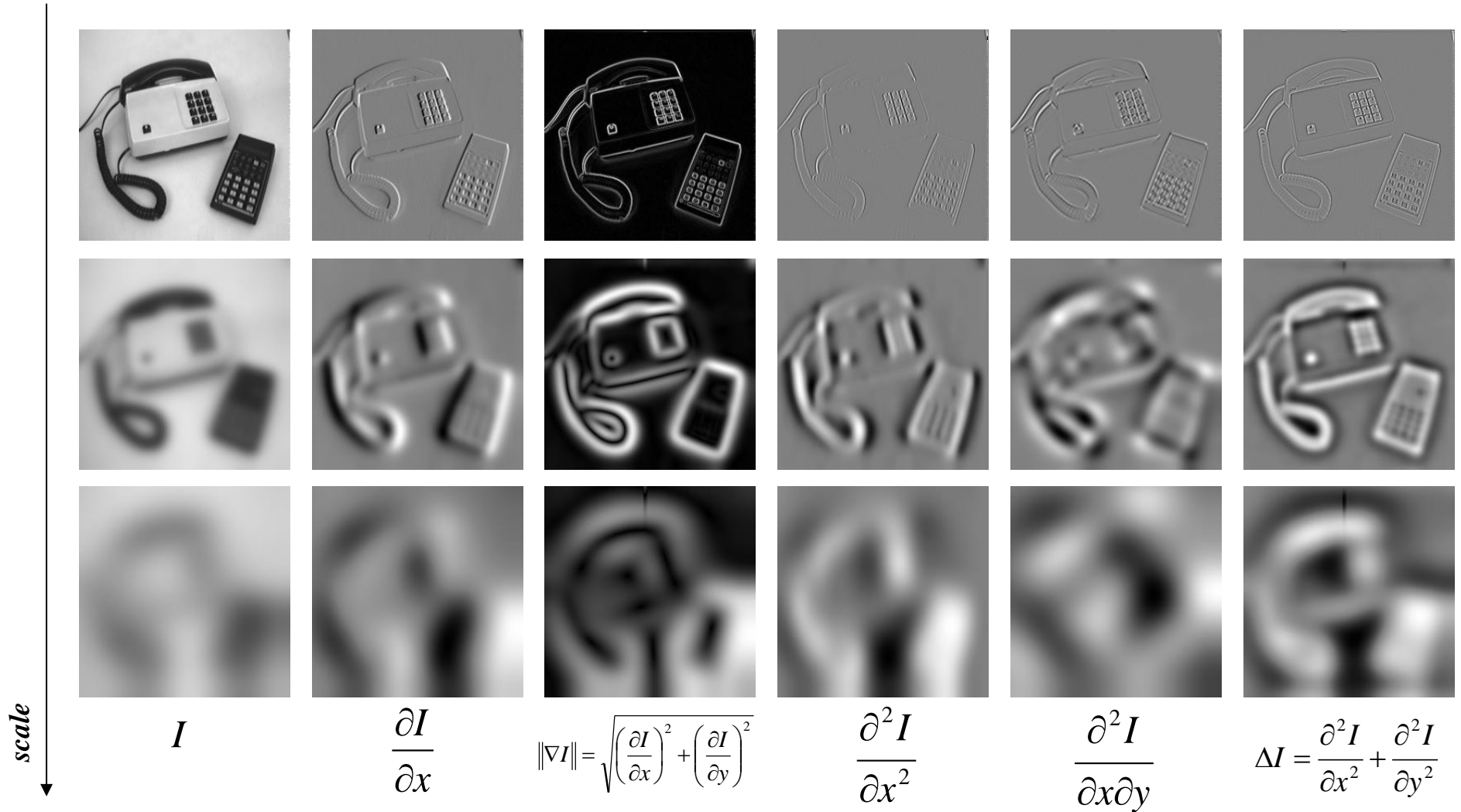
$$\left( \frac{\partial^{i+j} I}{\partial x^i \partial y^j} \right)_\sigma \stackrel{\text{def.}}{=} I \star \left( \frac{\partial^{i+j} G_\sigma}{\partial x^i \partial y^j} \right)$$



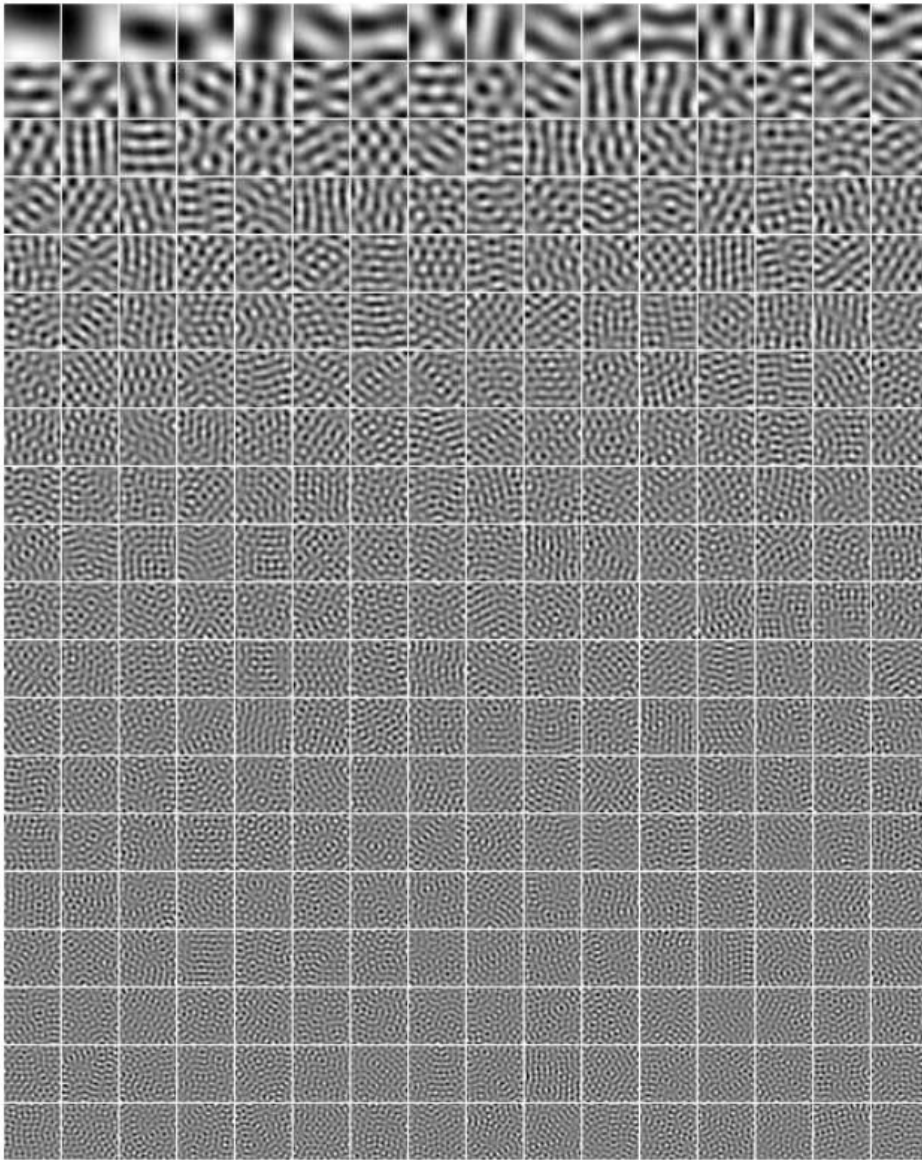
# MULTISCALE DERIVATIVES AND ASSOCIATED DERIVATION KERNELS

$\sigma$	$I$	$I_x$	$I_y$	$I_{xx}$	$I_{yy}$	$I_{xy}$
2	 	 	 	 	 	 
5	 	 	 	 	 	 
10	 	 	 	 	 	 

# MULTISCALE DIFFERENTIAL QUANTITIES

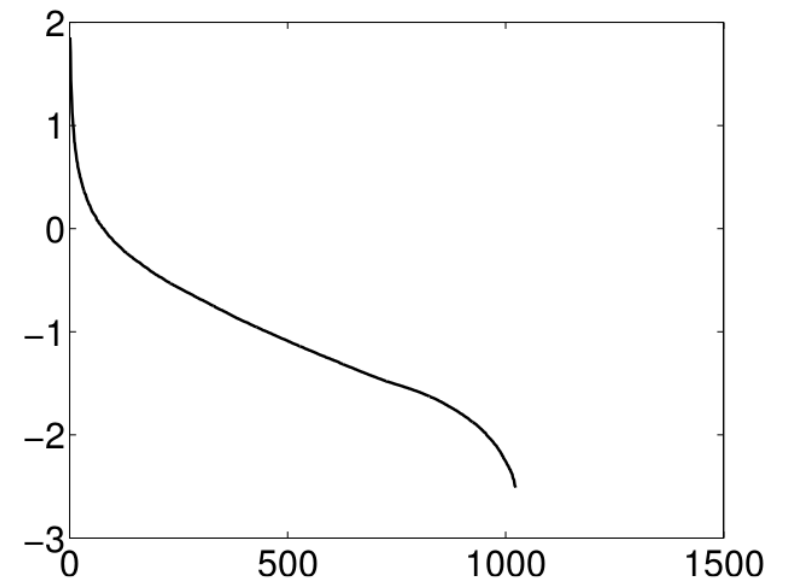


# PCA AND NATURAL IMAGE STATISTICS



On the left, the 320 first eigen vectors calculated by a Principal Component Analysis (PCA) applied to a set of 32x32 patches randomly sampled from a natural image dataset.

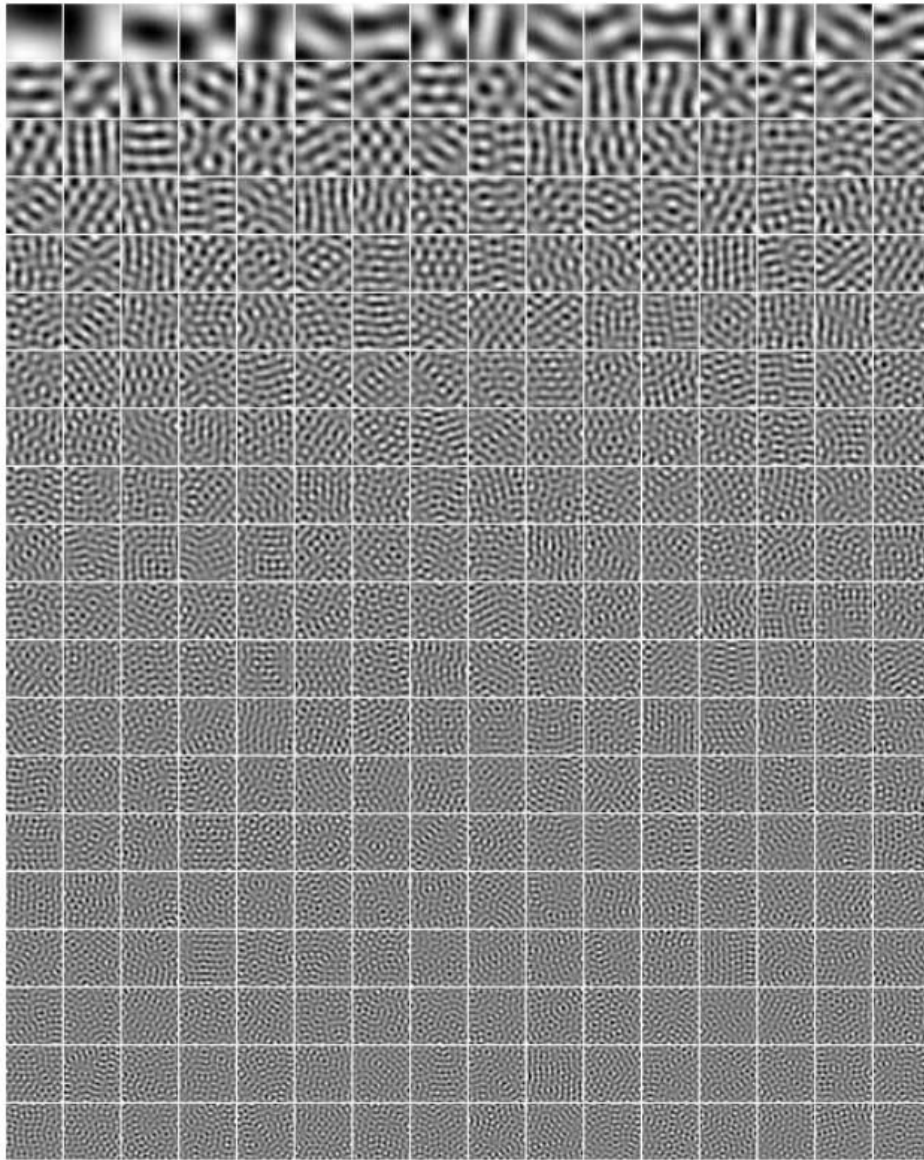
Hereunder, the log-variance associated to each eigen vector (principal component), as a function of its rank, for the whole set of patches.



*[Hyvriinen 09]*



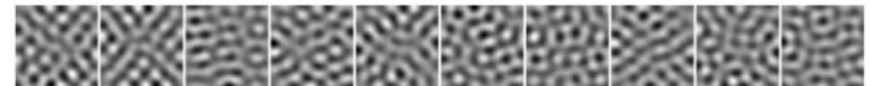
# PCA AND NATURAL IMAGE STATISTICS



Number 1 Principal Component obtained for 10 distinct random sets:



Number 100 Principal Component obtained for 10 distinct random sets:



**Note the similarity between the first principal components and the first derivatives of Gaussian.**

*[Hyvriinen 09]*

# VISUAL PRIMITIVES FOR RECOGNITION AND TRACKING

The representation level, from strictly local to fully global, is a fundamental property of visual features.

**Local: more geometry (direction, curvature,...)**



**Global: more statistics (histogram, frequency spectrum,...)**

The scale spaces act as continuum from the local to the global.

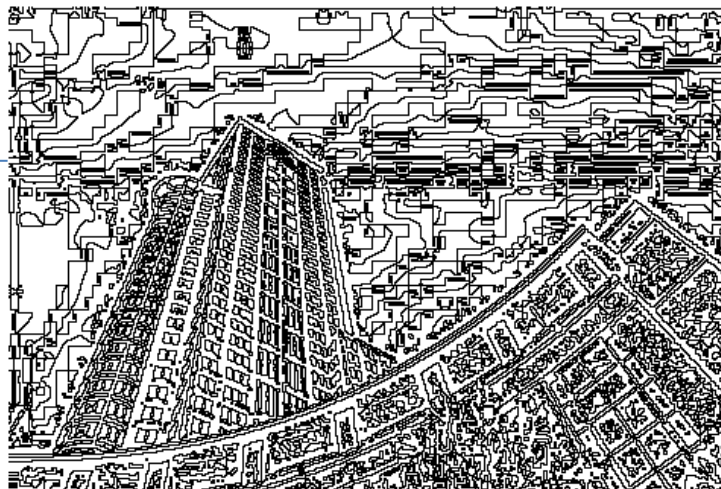
In the next slides:

- Contours detection (Zero crossing of the Laplacian)
- Corner points detection (Harris)
- Blobs detection (SIFT)
- Local descriptors (differential invariants).

# CONTOURS: ZERO-CROSSINGS OF THE LAPLACIAN

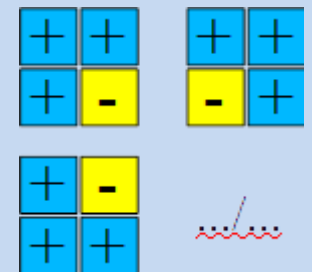


Laplacian



Zero-crossings

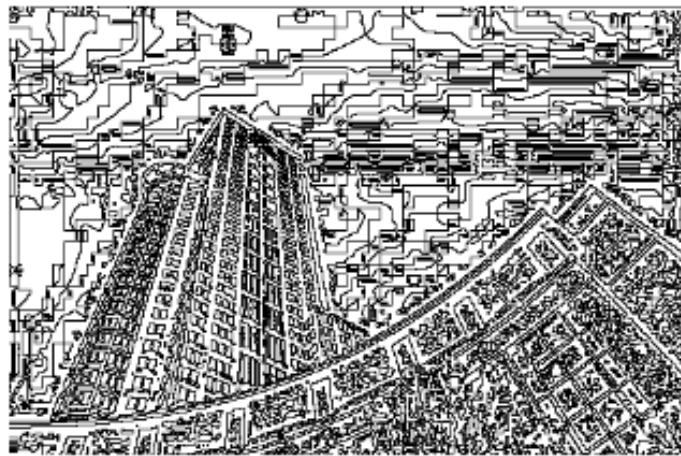
- Select zero-crossings w.r.t. contrast
- Select structures w.r.t. scales



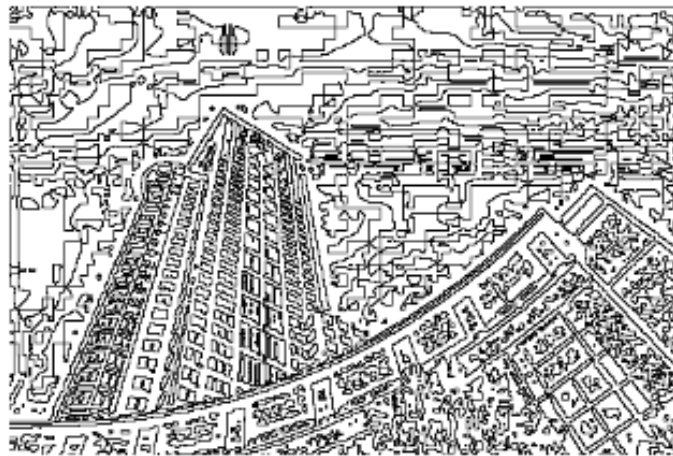
Sign change  
detection in 2x2  
neighbourhoods



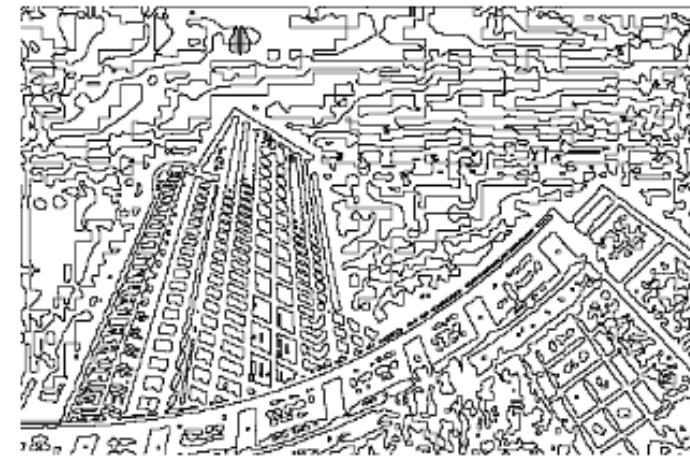
# MULTISCALE CONTOURS



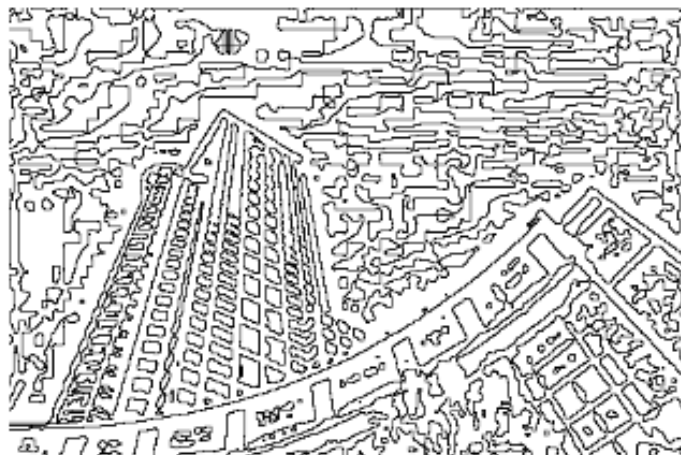
$\sigma = 1.0$



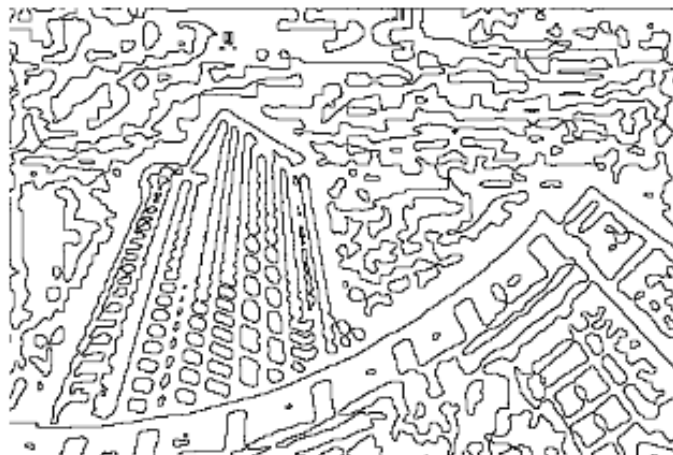
$\sigma = 1.5$



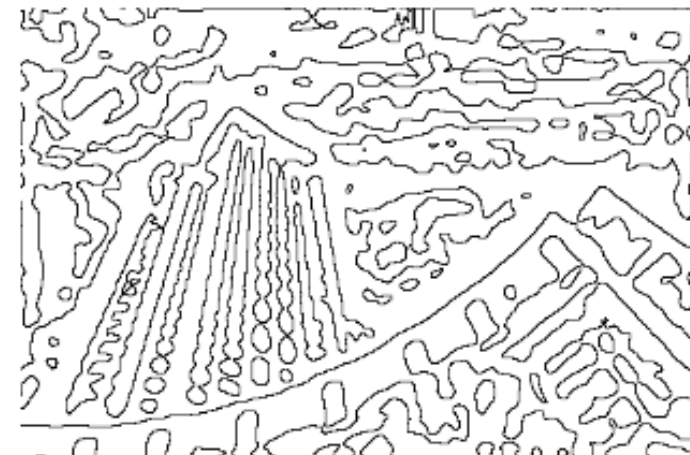
$\sigma = 2.0$



$\sigma = 2.5$



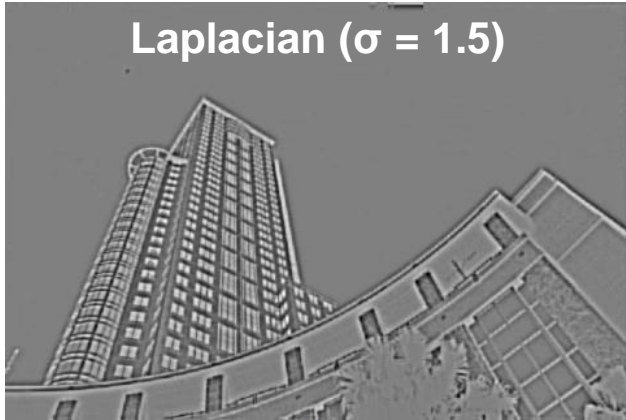
$\sigma = 3.5$



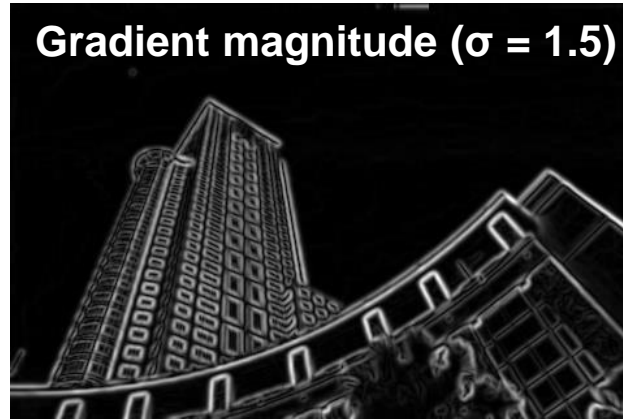
$\sigma = 5.0$

# CONTOURS AND CONTRAST

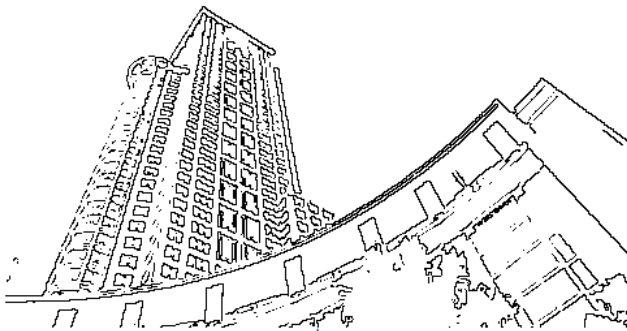
Laplacian ( $\sigma = 1.5$ )



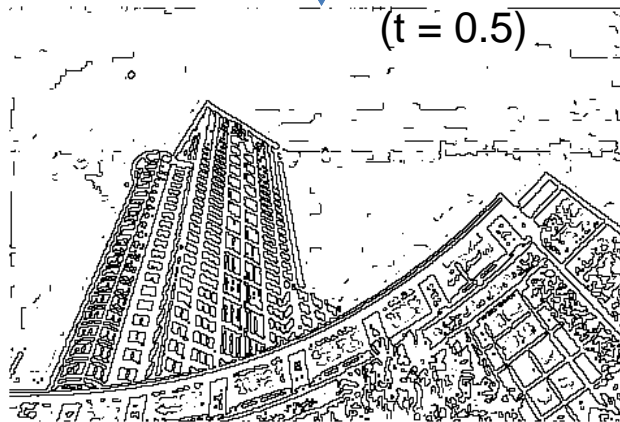
Gradient magnitude ( $\sigma = 1.5$ )



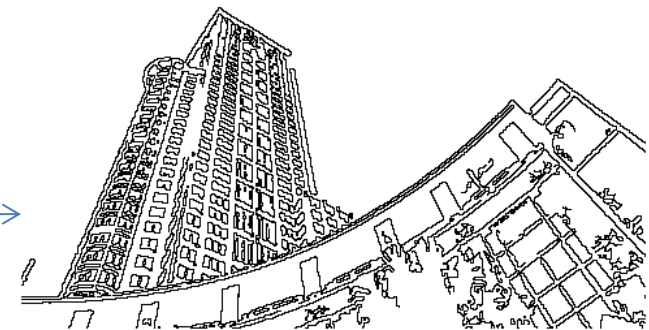
high  
threshold  
( $t = 8.0$ )



low  
threshold  
( $t = 0.5$ )

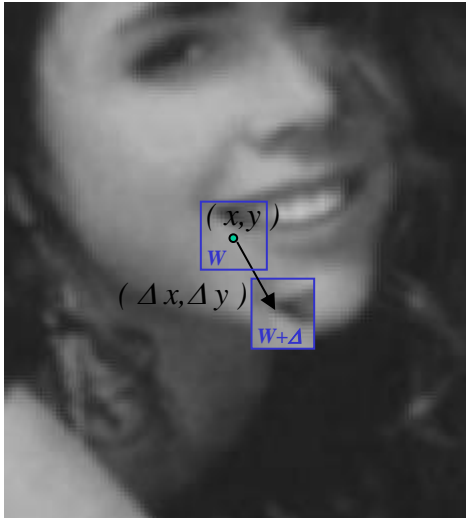


hysteresis  
threshold



# CORNER POINTS AND AUTOCORRELATION MATRIX

Corner (or Interest) points are points that carry much information relatively to the image. At the neighbourhood of these points, the image is expected to *vary significantly in more than one directions*.



One measure of the local variations of image  $I$  at point  $(x, y)$  associated to a displacement  $(\Delta x, \Delta y)$  is the *autocorrelation* function:

$$\chi(x, y) = \sum_{(x_k, y_k) \in W} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2$$

Where  $W$  is a window centred at point  $(x, y)$ .

Now by using a first order approximation:

$$I(x_k + \Delta x, y_k + \Delta y) \approx I(x_k, y_k) + \left( \frac{\partial I}{\partial x}(x_k, y_k) \quad \frac{\partial I}{\partial y}(x_k, y_k) \right) \cdot \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

And then:

$$\chi(x, y) = \sum_{(x_k, y_k) \in W} \left( \left( \frac{\partial I}{\partial x}(x_k, y_k) \quad \frac{\partial I}{\partial y}(x_k, y_k) \right) \cdot \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right)^2 = \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \underbrace{\begin{pmatrix} \sum_{(x_k, y_k) \in W} \left( \frac{\partial I}{\partial x}(x_k, y_k) \right)^2 & \sum_{(x_k, y_k) \in W} \frac{\partial I}{\partial x}(x_k, y_k) \cdot \frac{\partial I}{\partial y}(x_k, y_k) \\ \sum_{(x_k, y_k) \in W} \frac{\partial I}{\partial x}(x_k, y_k) \cdot \frac{\partial I}{\partial y}(x_k, y_k) & \sum_{(x_k, y_k) \in W} \left( \frac{\partial I}{\partial y}(x_k, y_k) \right)^2 \end{pmatrix}}_{\Xi(x, y)} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

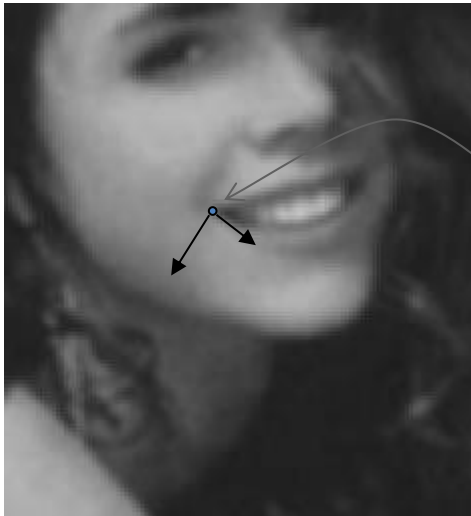
*Autocorrelation matrix* of image  $I$  at  $(x, y)$



# AUTOCORRELATION MATRIX AND THE HARRIS DETECTOR

$$\Xi(x, y) = \begin{pmatrix} \sum_{(x_k, y_k) \in W} \left( \frac{\partial I}{\partial x}(x_k, y_k) \right)^2 & \sum_{(x_k, y_k) \in W} \frac{\partial I}{\partial x}(x_k, y_k) \cdot \frac{\partial I}{\partial y}(x_k, y_k) \\ \sum_{(x_k, y_k) \in W} \frac{\partial I}{\partial x}(x_k, y_k) \cdot \frac{\partial I}{\partial y}(x_k, y_k) & \sum_{(x_k, y_k) \in W} \left( \frac{\partial I}{\partial y}(x_k, y_k) \right)^2 \end{pmatrix}$$

The autocorrelation matrix  $\Xi$  represents the local variation of  $I$  at  $(x, y)$ .  $(x, y)$  will be a corner point of  $I$  if for any displacement  $(\Delta x, \Delta y)$ , the quantity  $(\Delta x, \Delta y) \cdot \Xi(x, y) \cdot (\Delta x, \Delta y)^t$  is large.



Corner points are those points  $(x, y)$  for which the autocorrelation matrix  $\Xi(x, y)$  has *two large eigen values*.

This corresponds to points for which there locally exists a basis of eigen vectors of  $\Xi$  that describe major local variations for the image.

The *Harris detector* actually calculates an *interest map*  $\Theta(x, y)$ :

$$\Theta(x, y) = \det \Xi - \alpha \text{trace} \Xi$$

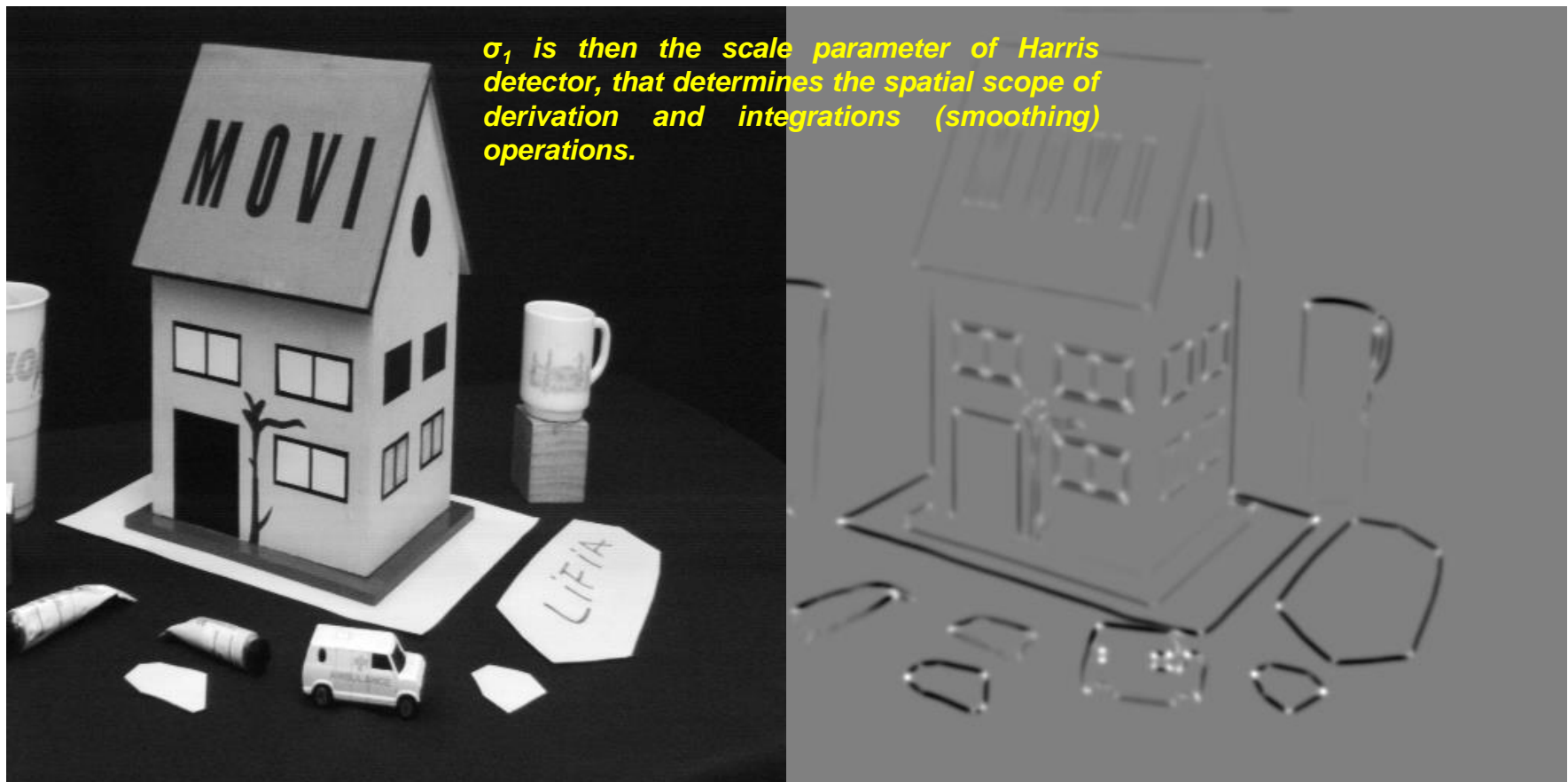
The first term corresponds to the product of eigen values, the second term penalises contour points with one single large eigen value.

Corner points correspond to local maxima of function  $\Theta$  that are beyond a certain threshold (typically, 1% of  $\Theta_{\max}$ ).

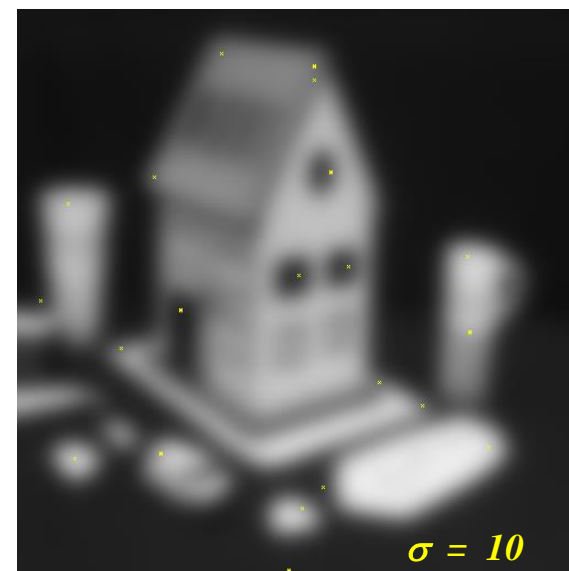
*[Harris 88]*

# COMPUTING HARRIS INTEREST MAP $\Theta$

1. Compute the first derivatives using Gaussian derivatives (standard deviation  $\sigma_1$ )
2. Compute the components of the autocorrelation matrix  $\Xi$  by using a Gaussian smoothing instead of summing on window  $W$  (standard deviation  $\sigma_2$ , typically  $\sigma_2 = 2 \sigma_1$ )
3. Compute the interest map:  $\Theta = \det(\Xi) - \alpha \text{trace}(\Xi)$  (typically  $\alpha = 0,06$ ).
4. Compute the local maxima of  $\Theta$  larger than a certain threshold (typically 1% of  $\Theta_{\max}$ ).



# MULTISCALE HARRIS CORNER POINTS



*Harris corner points obtained by calculating the first derivatives by convolution with a derivative of Gaussian of standard deviation  $\sigma$ .*



# SIFT DETECTOR: EXTREMA IN SCALE SPACE

The SIFT (Scale Invariant Feature Transform) detector uses a different approach of interest point that better fits large scales compared to corners:

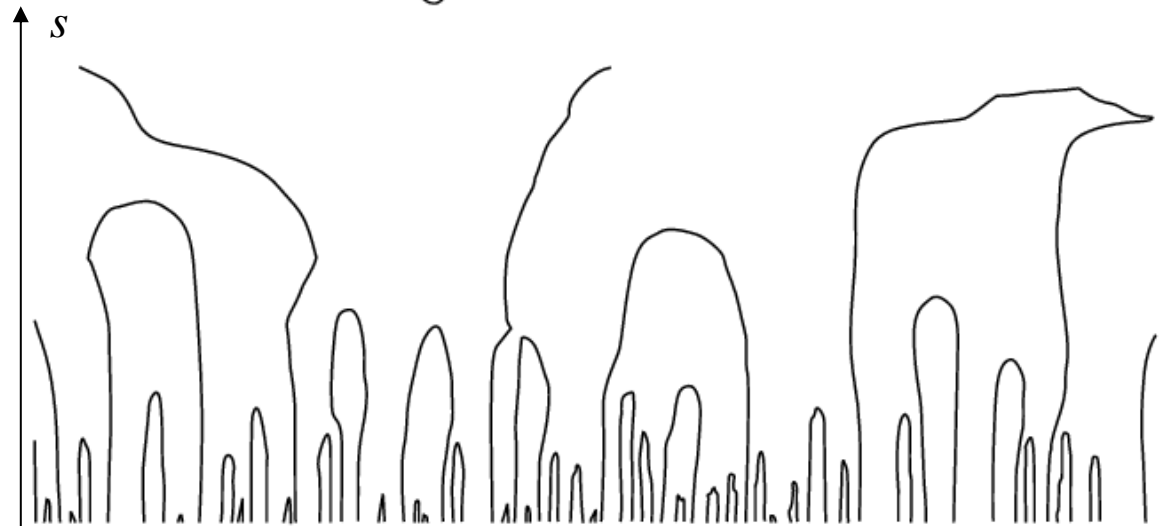
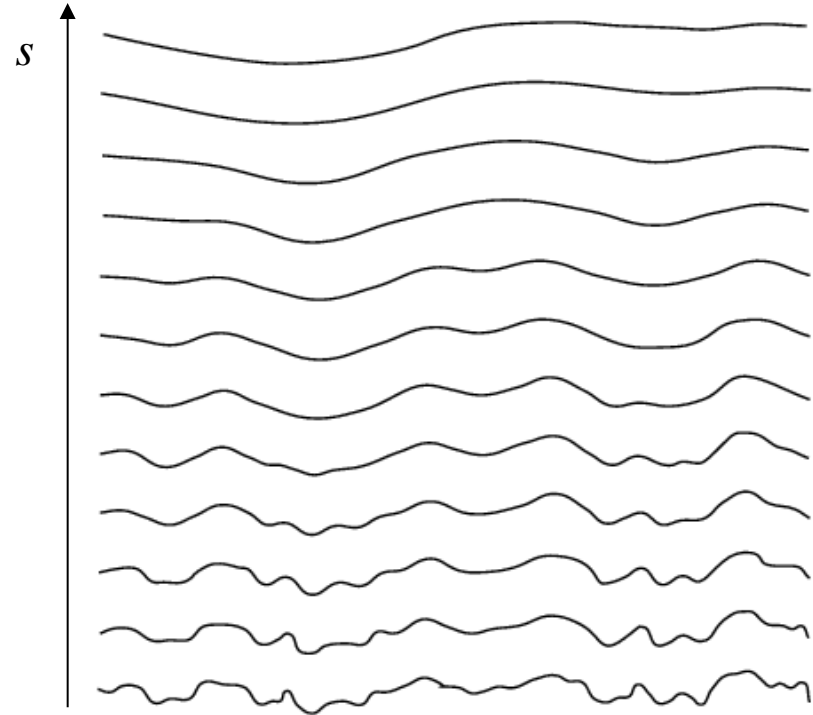
The *blob* (elliptical structure)

Such structure can be uniformly characterised at all scales and corresponds to a point of the mixed scale-space  $(x,y,s)$  where a local extremum disappears.

This relates to the causality principle of scale spaces.

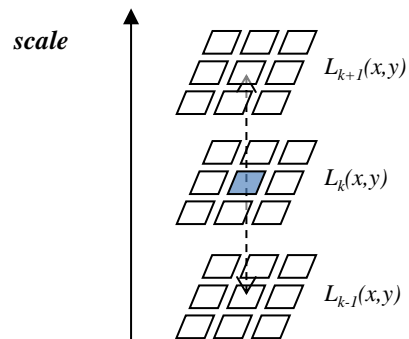
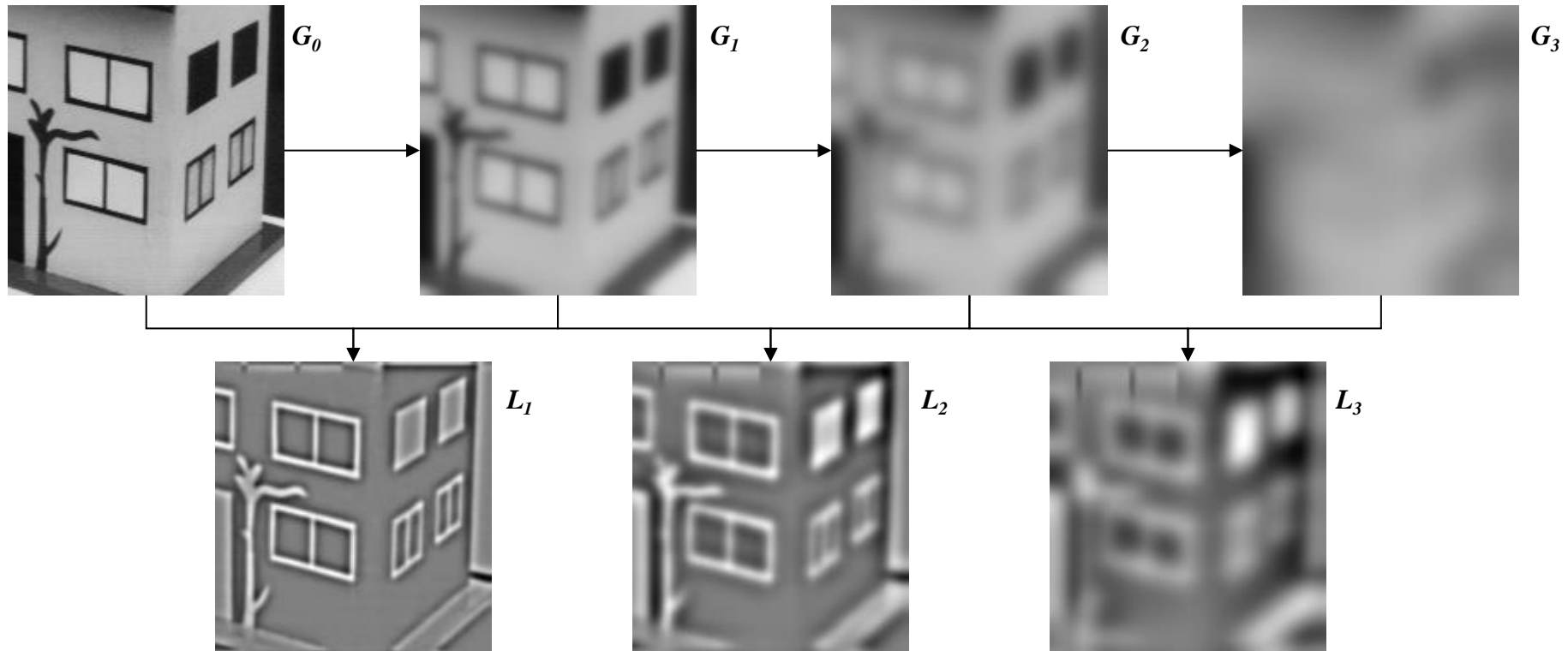
In 1d (on the right): point of maximal scale  $s$  on each curve of the scale space fingerprint.

[Witkin 83]



*Gaussian scale space of a 1d signal (top) and scale-space fingerprint, showing the position of extrema in  $(x,s)$  (bottom).*

# SIFT DETECTOR: EXTREMA IN SCALE SPACE



The function  $G_k(x,y) = G(x,y,k\sigma)$  is the image convolved by a Gaussian of standard deviation  $k\sigma$ . The functions  $L_k(x,y)$  correspond to the difference (normalised here) between two successive Gaussians.

The function  $L_k(x,y)$  is a Laplacian representation of the image, that corresponds to a spatially localised frequency decomposition: *i.e.* contribution of structures of scale (or size)  $k\sigma$  at point  $(x,y)$ .

The points selected by SIFT are the local maxima and minima locaux of function  $L_k(x,y)$ , both in the current scale and in the adjacent scales (see on the left).

[Lowe 04]

# SIFT INTEREST POINTS

For each scale-space extremum of the Laplacian representation (SIFT interest point), the associated orientation is calculated as follows:

$$\theta(x, y) = \arctan \left( \frac{G_y^\sigma(x, y)}{G_x^\sigma(x, y)} \right)$$

with  $G_x^\sigma(x, y) = \frac{\partial}{\partial x} G(x, y, \sigma) = I(x, y) * \frac{\partial}{\partial x} g_\sigma(x, y)$

(where  $\sigma$  is the selected scale)

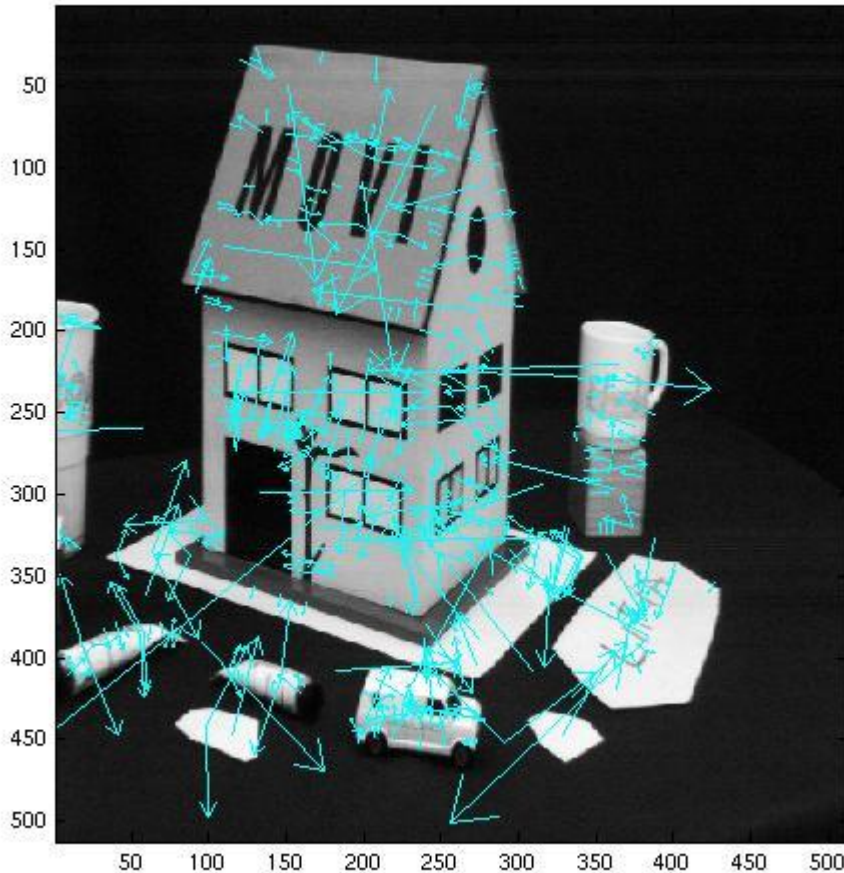


Image 1: 589 detected points.

On the left, SIFT interest points: the direction of the arrow represents the orientation  $\theta$  and its length the associated scale.

*[Lowe 04]*

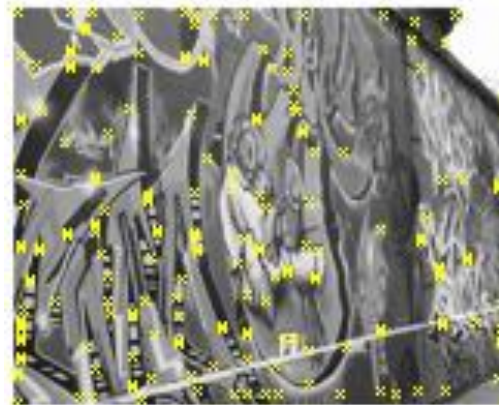
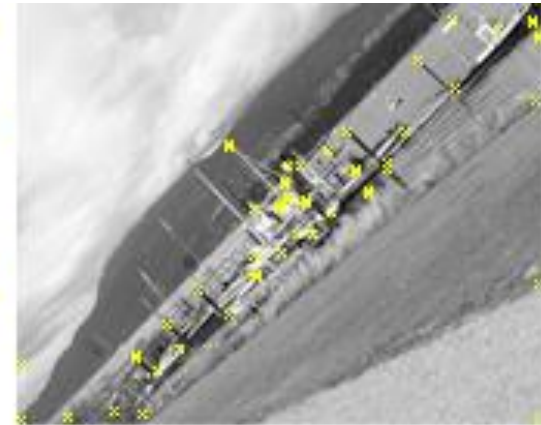
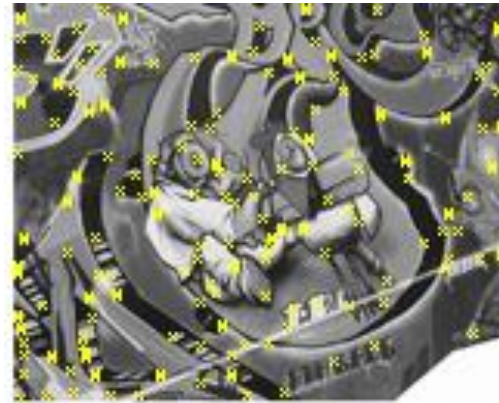
# EVALUATION OF INTEREST POINT DETECTORS

Most interest point detectors are designed independently of the descriptor they will be used with. It then makes sense to evaluate them alone.

A good detector should be:

- **Repeatable:** a point should appear at the very same place whatever the deformation.
- **Representative:** the points should be as numerous as possible.
- **Efficient:** it should be fast to compute (see SURF, FAST)

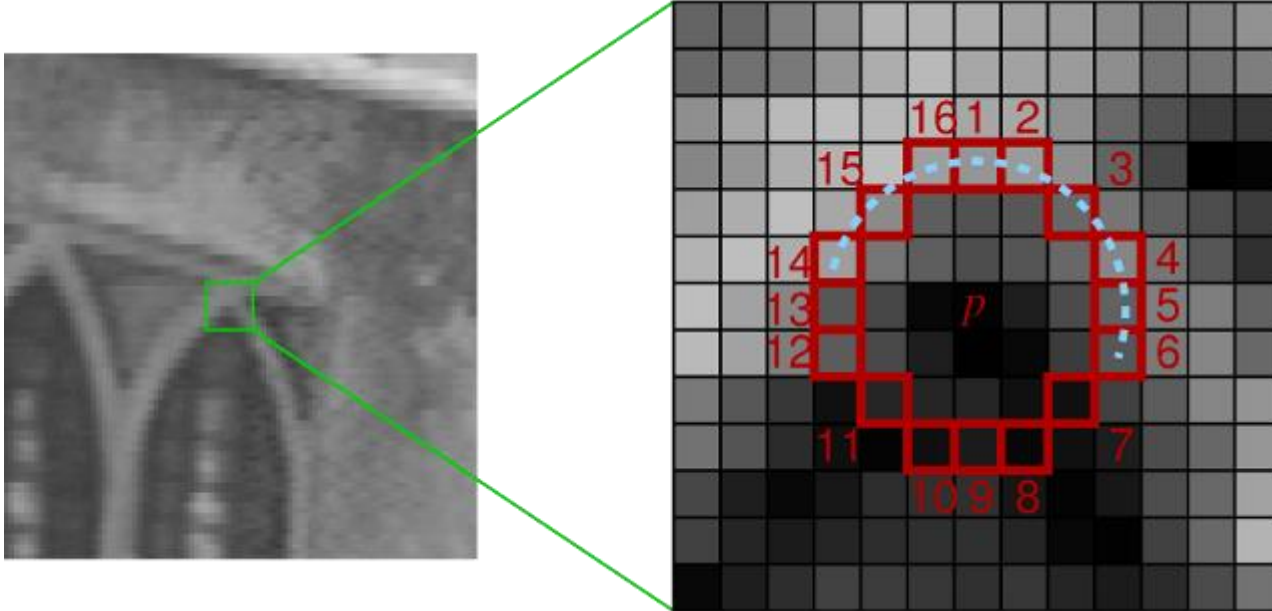
(NB: repeatability and representativity are not independent!)



*[Schmid 2000]*



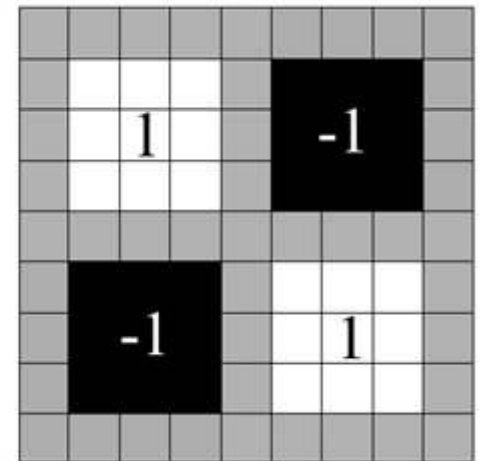
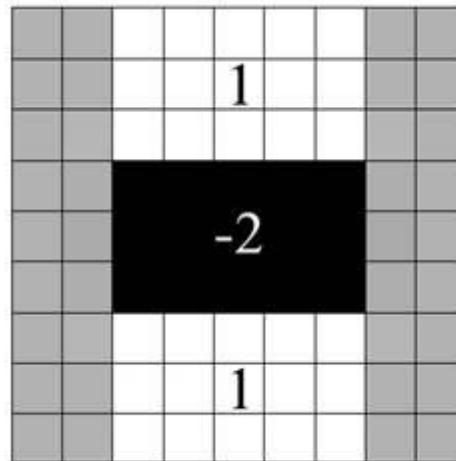
## WHAT ABOUT COMPUTATIONAL EFFICIENCY?



The FAST detector selects points  $p$  whose circular neighbourhood shows long contiguous runs with values significantly brighter (resp. darker) than  $p$ .

*[Rosten 05]*

The SURF detector approximates the second derivatives using rectangular convolution kernels computed with integral images, then selects the local maxima of the determinant of the Hessian.



*[Bay 06]*

# DESCRIPTORS: DIFFERENTIAL INVARIANTS

**Goal:** represent interest points by *indexes* that are *rotation* and *scale* invariant.

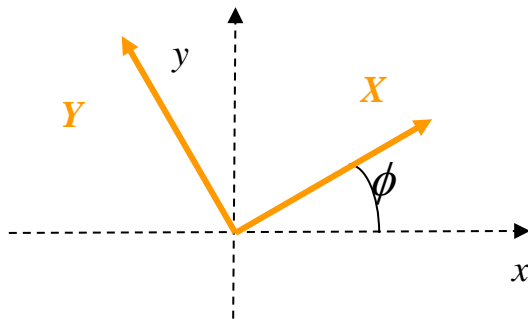
The principle used here is based on multiscale spatial derivatives:

The *local jet* of  $I$ :  $I_{ij}^\sigma = I * G_{ij}^\sigma$  with:  $G_{ij}^\sigma = \frac{\partial^{i+j}}{\partial x^i \partial y^j} G^\sigma$  and:  $G^\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$

Notation:  $\{I_{ij}^\sigma; 0 \leq i + j \leq 3\} = \{I, I_x, I_y, I_{xx}, I_{xy}, I_{yy}, I_{xxx}, I_{xxy}, I_{xyy}, I_{yyy}\}$

The idea is to *combine these derivatives* to obtain *rotation invariant* quantities:

As an example, the Laplacian  $I_{xx} + I_{yy}$  is rotation invariant:



$$\begin{cases} x = X \cos \phi + Y \sin \phi \\ y = X \sin \phi - Y \cos \phi \end{cases}$$

$$\begin{cases} X = x \cos \phi + y \sin \phi \\ Y = -x \sin \phi + y \cos \phi \end{cases}$$

$$\begin{cases} I_X = I_x \cos \phi + I_y \sin \phi \\ I_Y = I_x \sin \phi - I_y \cos \phi \end{cases}$$

$$\begin{cases} I_{XX} = I_{xx} \cos^2 \phi + 2I_{xy} \cos \phi \sin \phi + I_{yy} \sin^2 \phi \\ I_{YY} = I_{xx} \sin^2 \phi - 2I_{xy} \cos \phi \sin \phi + I_{yy} \cos^2 \phi \end{cases}$$

And then:  $I_{XX} + I_{YY} = I_{xx} + I_{yy}$ .



## DESCRIPTORS: DIFFERENTIAL INVARIANTS

More generally, a whole family of independent rotation invariant differential quantities can be built: the Hilbert differential invariants. For example, at order 2, the following descriptor is obtained:

$$\Psi_2 = \begin{pmatrix} I \\ I_x^2 + I_y^2 \\ I_{xx}I_x^2 + 2I_xI_yI_{xy} + I_{yy}I_y^2 \\ I_{xx} + I_{yy} \\ I_{xx}^2 + 2I_{xy}^2 + I_{yy}^2 \end{pmatrix}$$

NB: the rotation invariance also relies on the isotropy of the Gaussian kernels!

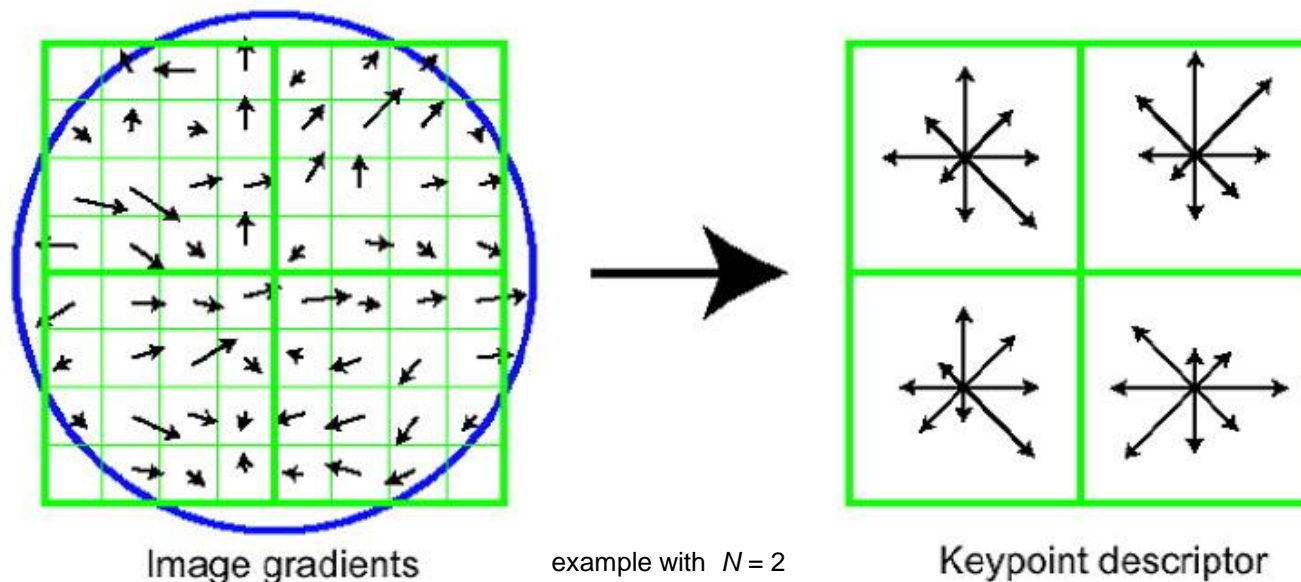
The vectors  $\Psi$  are then calculated for all interest points at different scales, and then matched using a certain metrics (e.g. Euclidean distance).

*[Schmid et Mohr 97]*

# SIFT DESCRIPTOR: GRADIENT ORIENTATION HISTOGRAMS

The descriptors associated to SIFT points are orientation histograms computed around the interest point:

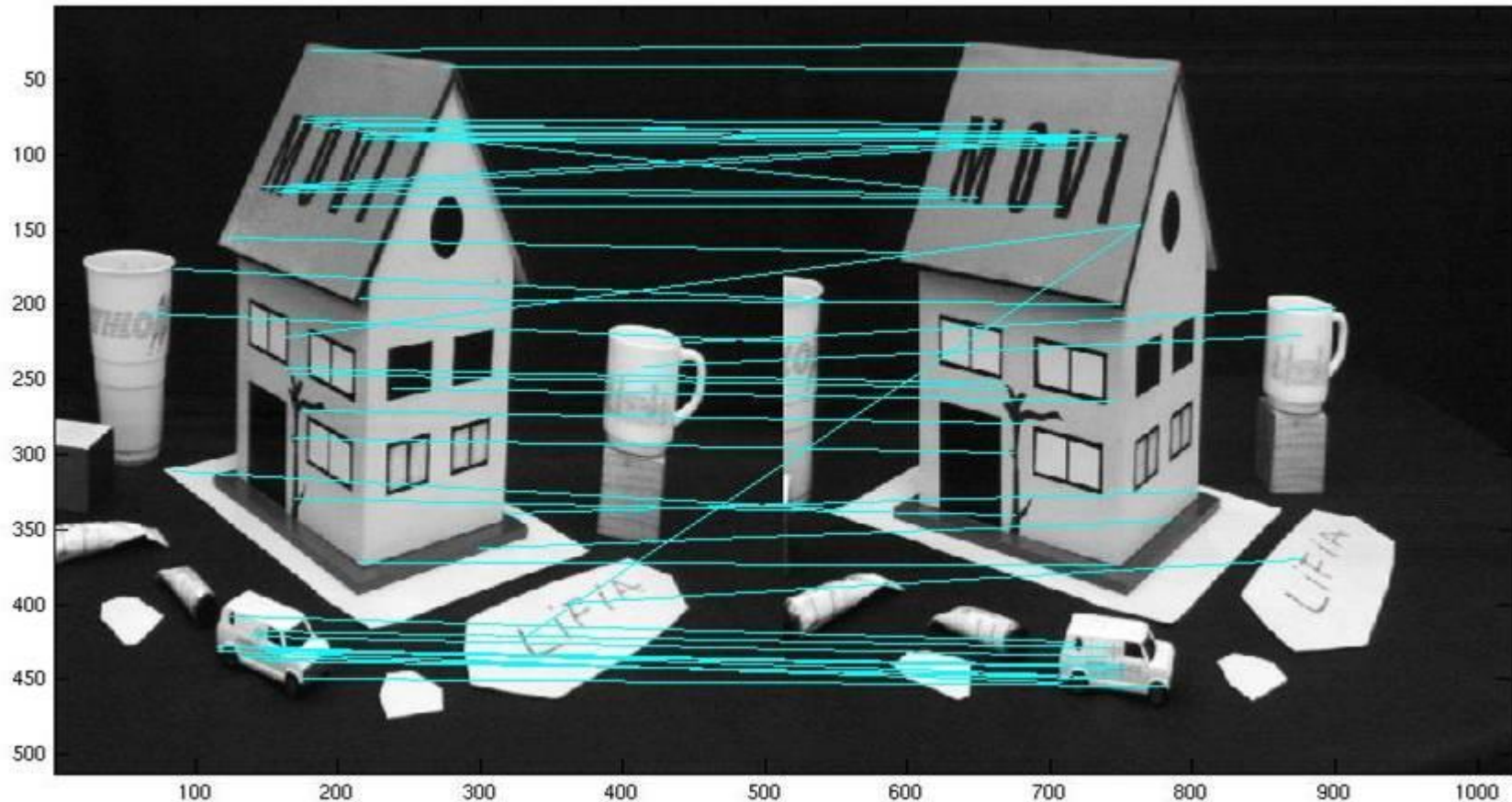
- The space is divided around each point  $(x,y)$  into  $N^2$   $4 \times 4$  squares.
- The gradient  $(G_x(a,b,\sigma), G_y(a,b,\sigma))$  is calculated for the  $4 \times 4 \times N^2$  points  $(a,b)$ .
- For each  $4 \times 4$  square, a histogram of orientations quantised to 8 directions is computed, by weighting the occurrences using: (1) the gradient magnitude (2) the inverse distance to the interest point  $(x,y)$ .
- For rotation invariance purposes: the local orientation of the interest point  $\theta(x,y)$  is used as the reference (zero) orientation of histograms.



The resulting descriptors are then  $8 \times N^2$  vectors, that will be compared using a distance (e.g. Euclidean distance)

*[Lowe 04]*

# SIFT POINT MATCHING EXAMPLE



*SIFT points matching result between image (2) on the left (510 detected points), and image (1) on the right (589 detected points). 51 matches were selected as acceptable here.*

*Exercise: Which criteria can be used for such selection?*

*[Lowe 04]*

## MATCHING FEATURES: METRICS

Matching features then relies on pairwise comparison of descriptors. Ideally, this should be measured with a simple metrics:

The Euclidean distance:

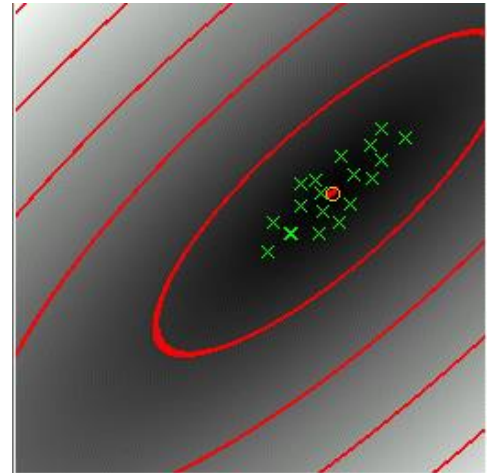
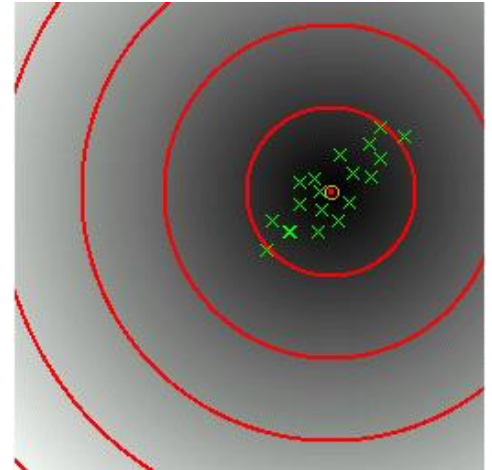
$$\delta_e(x, x')^2 = (x - x')^T (x - x')$$

However this distance does not take into account differences in range, nor correlations that can exist between the different components of the descriptor.

The Mahalanobis distance:

$$\delta_m(x, x')^2 = (x - x')^T C^{-1} (x - x')$$

with  $C = (\text{cov}(x_i, x_j))_{i,j}$  the covariance matrix calculated on the descriptors dataset, take those properties into account by deforming the Euclidean distance in the principal covariance directions.





## MATCHING FEATURES: METRICS

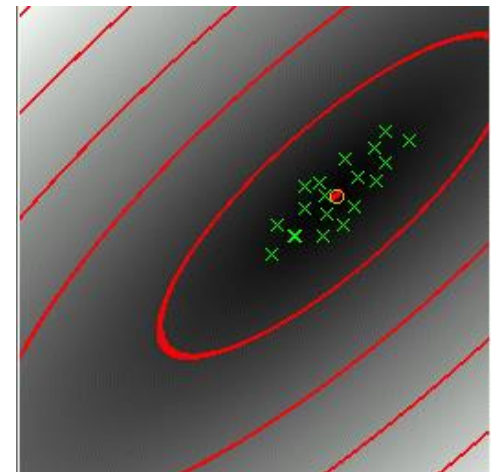
In the case of large descriptor dataset (image mining), the covariance matrix is calculated and updated off-line. By diagonalising  $C^{-1}$ , the computation is simplified to a Euclidean distance on normalised components:

$$C^{-1} = P^T D P$$

$$\delta_m(x, x') = \sqrt{(x - x')^T C^{-1} (x - x')} = \underbrace{\|\sqrt{D} P x - \sqrt{D} P x'\|}_{\text{ellipsoidal distance}}$$

Then for each descriptor dataset update, one should:

- Update the covariance matrix  $C$
- Calculate and diagonalise  $C^{-1}$
- Normalise all vectors to  $x \rightarrow \sqrt{D} P x$



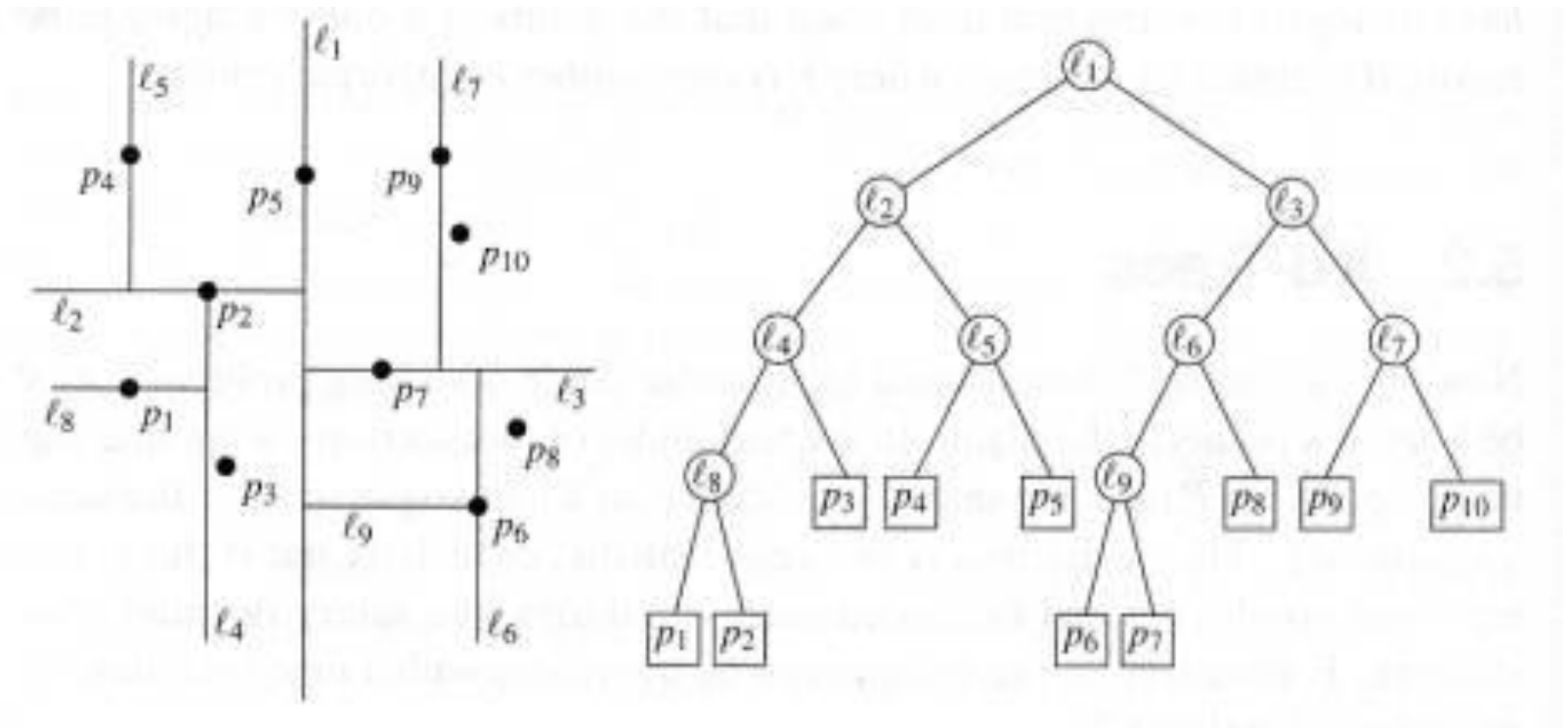
# DESCRIPTORS DATASET AND DATA MINING

In the case of a large descriptor database, it is desirable to limit the search to a limited neighbourhood of the unknown descriptor. This problem is strongly related to the way the descriptor vectors are *stored* within the database.

Cutting the descriptor base into hypercubes



Representing the base by a Kd-tree



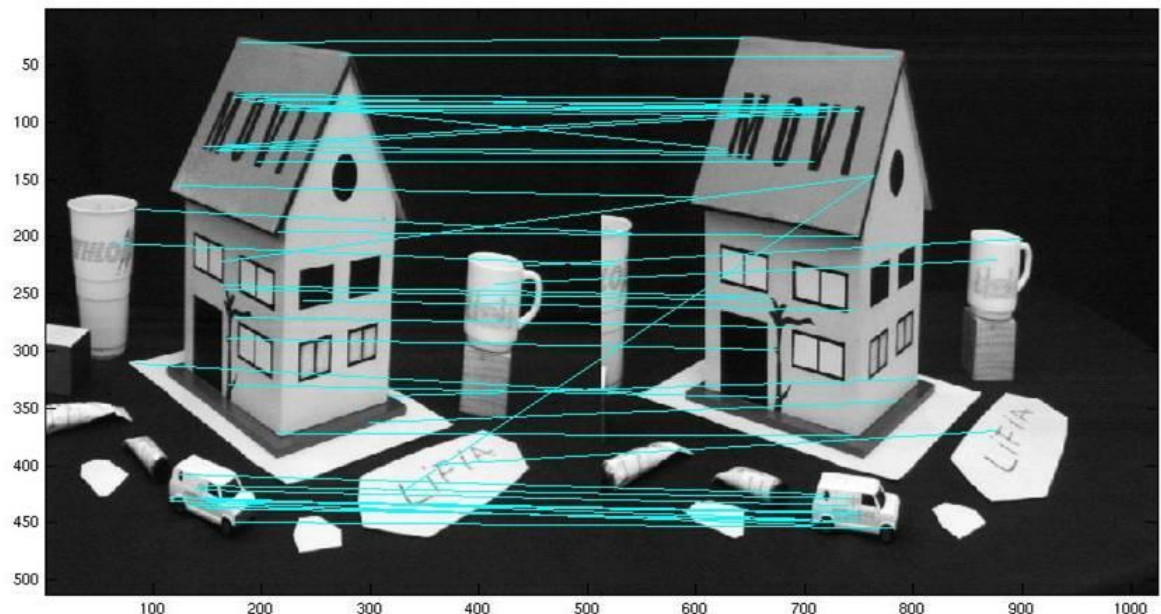
# FROM LOCAL TO GLOBAL: CONSENSUS OF LOCAL DESCRIPTORS

The visual features are often used to make a global decision: class label (recognition, categorisation), displacement parameters (visual odometry).

How to make such collective decision from the set of descriptors?

**Voting consensus:** every local descriptor is classified and the global class is attributed based on a majority voting (e.g.: room recognition, image categorisation...)

**Selection by consistence:** a subset of the local matches is (iteratively) selected so that a consistent decision is made (e.g.: visual odometry...)



## FROM LOCAL TO GLOBAL: VISUAL BAG-OF-WORDS

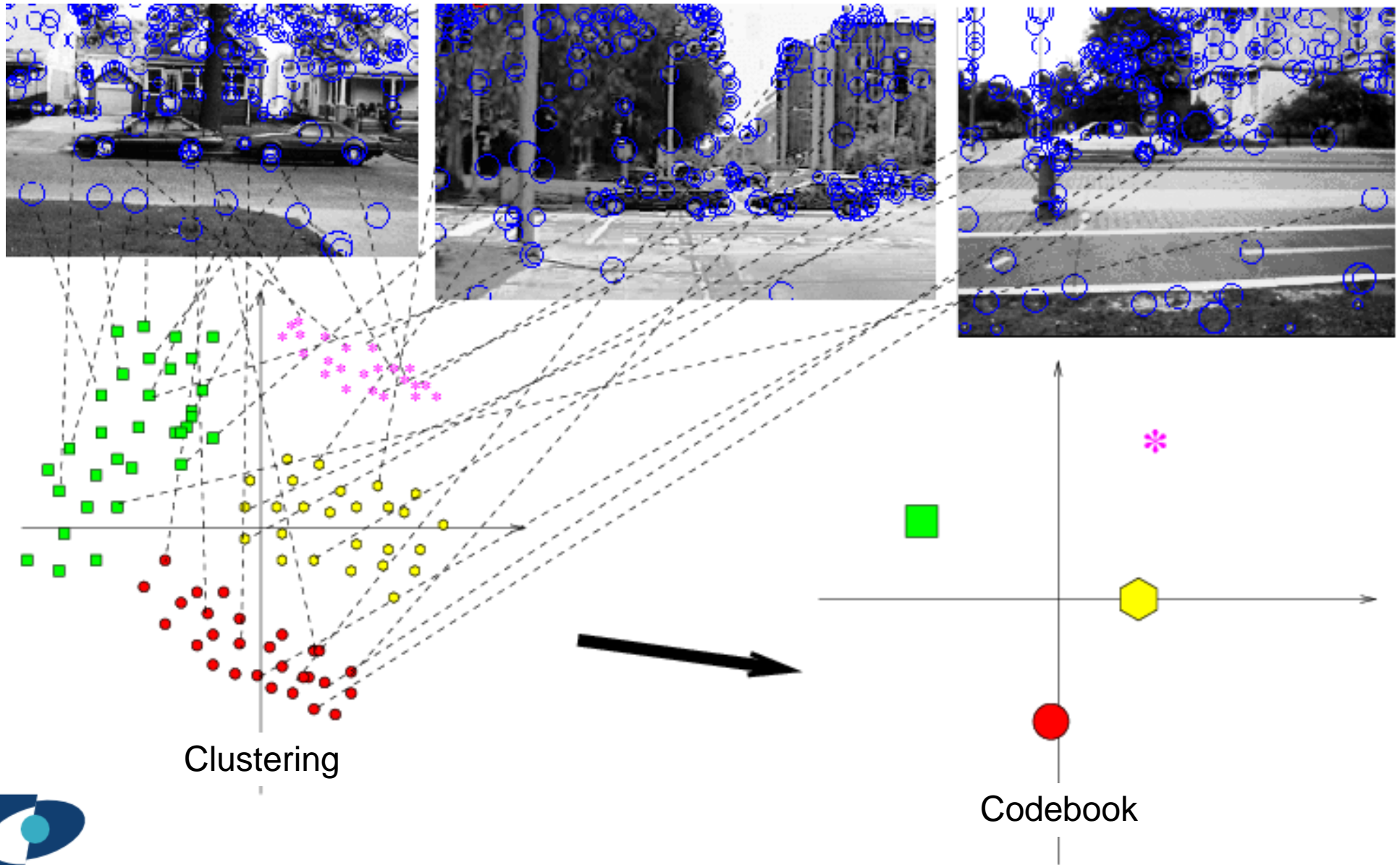
Another popular method consists in building a global descriptor from statistics of local descriptors:

- The descriptor space is reduced to a limited number of labels (words) by using a vector quantisation (or clustering) algorithm to form a *codebook* of local descriptors → **Unsupervised learning phase**.
- Histograms of visual words are used as global descriptors of example objects, then used to train a classifier → **Supervised learning phase**.
- For a unknown image, the codebook is used to encode the local descriptors (using for example Nearest Neighbour approach...) → **Local classification**.
- The histogram of visual words is then fed to the classifier to predict the image class → **Global classification**.

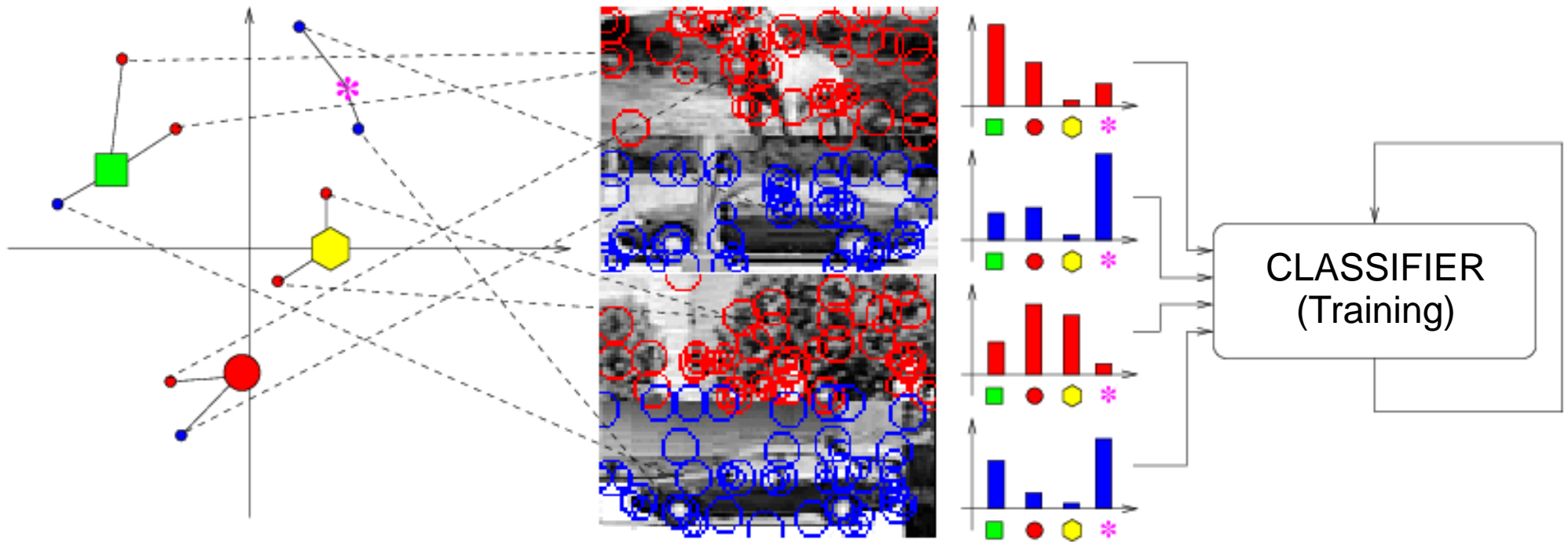
*[Csurka 2004]*



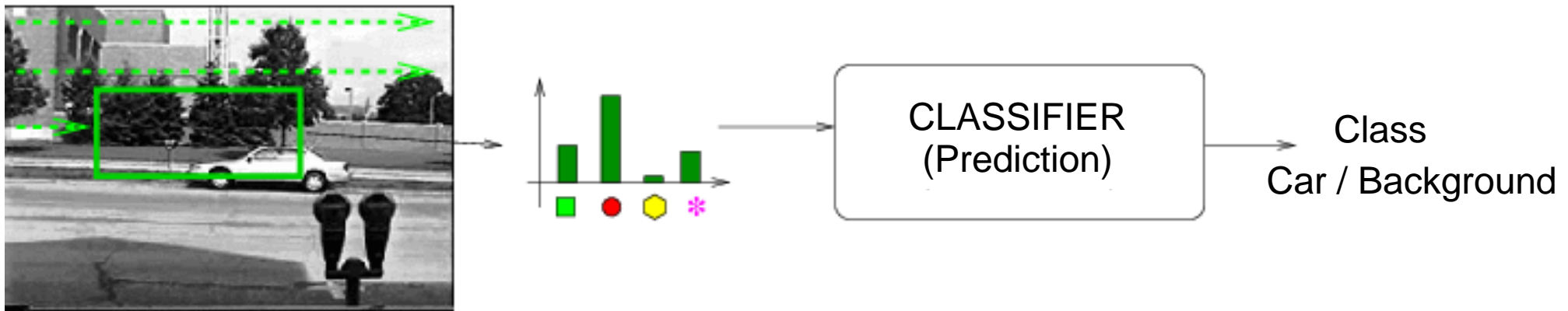
# VISUAL BAG-OF-WORDS 1: BUILDING THE CODEBOOK



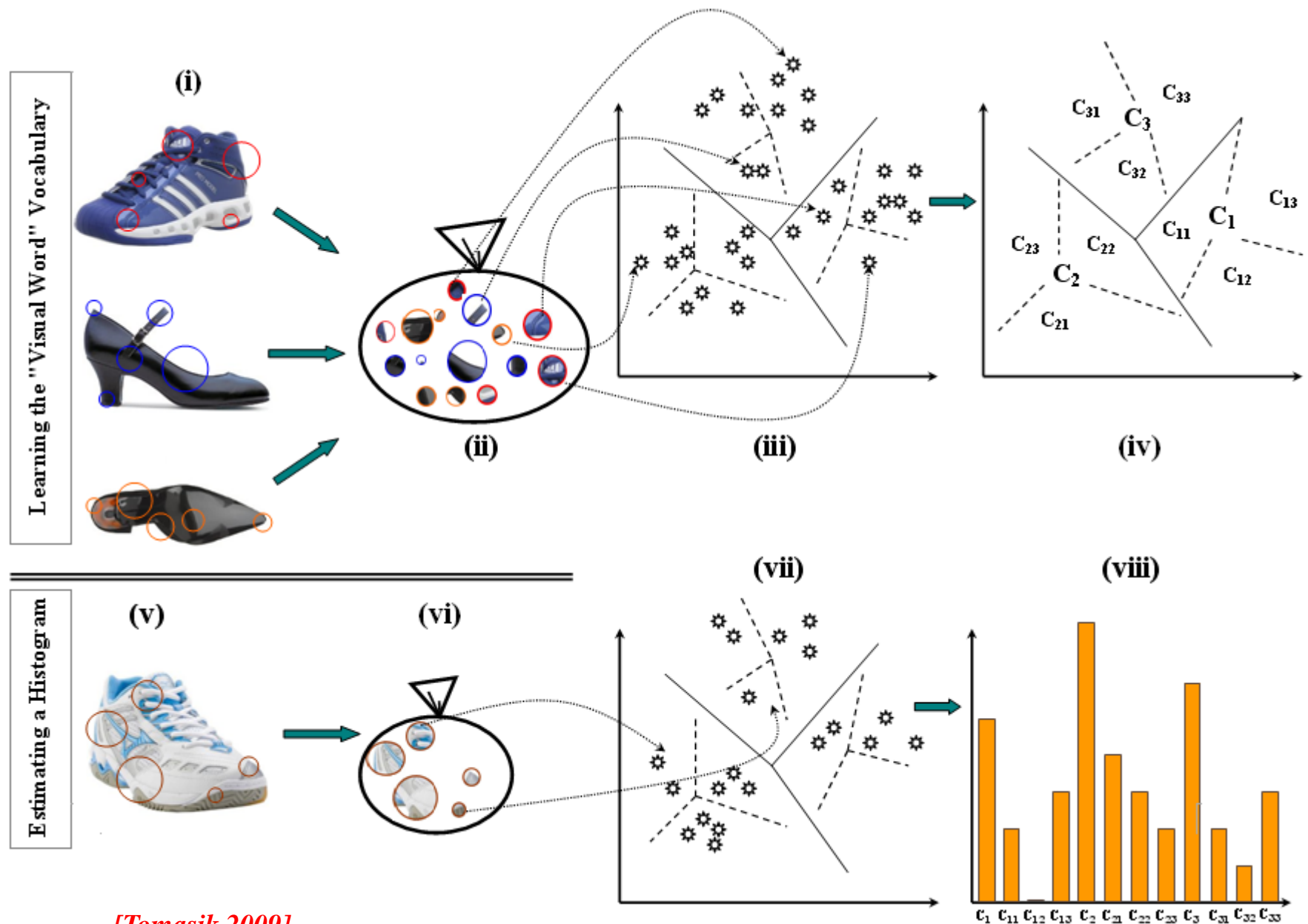
## VISUAL BAG-OF-WORDS 2: TRAINING THE CLASSIFIER



## VISUAL BAG-OF-WORDS 3: PREDICTING THE CLASS



# MULTISCALE / HIERARCHICAL VISUAL BAG-OF-WORDS



[Tomasik 2009]



# GLOBAL MATCHING: FREQUENCY BASED METHODS

$$I(x, y) = \frac{1}{wh} \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} F(u, v) e^{2j\pi(ux/w + vy/h)} \longleftrightarrow F(u, v) = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} I(x, y) e^{-2j\pi(ux/w + vy/h)}$$

$$F(u, v) = \|F(u, v)\| e^{j\phi_F(u, v)}$$

Frequency based motion estimation methods are based on the equivalence between translation and phase shift in the Fourier transform:

$$\begin{array}{ccc} I(x, y) & \xrightarrow{\text{TF}} & F(u, v) \\ I(x + \delta x, y + \delta y) & \xrightarrow{\text{TF}} & G(u, v) = F(u, v) e^{2j\pi(u\delta x/w + v\delta y/h)} \end{array}$$

And then:  $\|G(u, v)\| = \|F(u, v)\|$  and  $\phi_G(u, v) = \phi_F(u, v) + 2\pi(u\delta x/w + v\delta y/h)$

The phase shift between F and G is then:  $\Delta\phi(u, v) = 2\pi(u\delta x/w + v\delta y/h)$

Two couples  $(u, v)$  are then enough in theory to calculate  $(\delta x, \delta y)$ , but this direct method is too sensitive to noise and illumination changes.

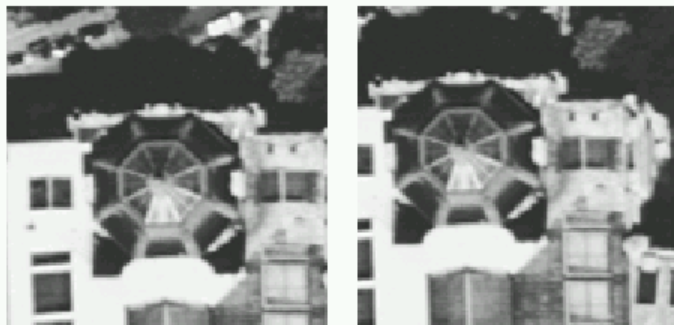
→ The *phase correlation* technique is preferred.

# PHASE CORRELATION

The phase correlation exploits a direct consequence of the translation / phase shift equivalence. If  $F$  is the FT of  $I$  and  $G$  the FT of  $I$  translated of  $(-\delta x, -\delta y)$ , then the phase shift between  $F$  and  $G$  is equal to their normalised cross power spectrum (NCPS), i.e.:

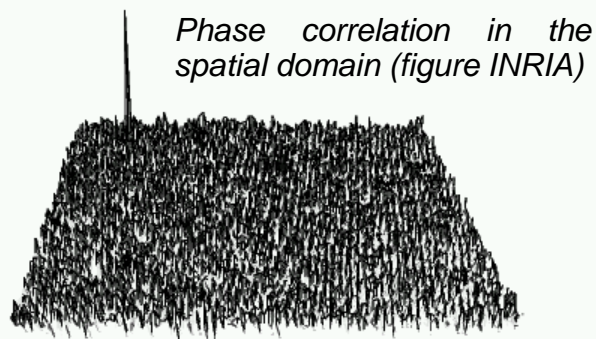
$$\frac{F^*(u, v)G(u, v)}{\|F^*(u, v)G(u, v)\|} = e^{2j\pi(u\delta x / w + v\delta y / h)}$$

The inverse FT of the NCPS is then equal to the Dirac function of the translation vector:  $\delta_{(\delta x, \delta y)}(x, y)$



(a)

(b)



(c)

The phase correlation method finally consists in:

1. Calculate the FT of  $I(x, y, t)$  and  $I(x, y, t+1)$ , say  $F_1$  and  $F_2$
2. Calculate  $\chi$  the NCPS of  $F_1$  and  $F_2$
3. Calculate  $D$  the inverse FT of  $\chi$
4. Search the position with maximum value of  $D$

Pros and Cons

- + Robust since all the frequencies contribute
- + Relatively fast thanks to the FFT
- In practice limited to a global displacement for the whole image. **Exercise:** explain why.

# A GLOBAL DESCRIPTOR: THE FOURIER-MELLIN INVARIANTS

The *Fourier-Mellin* transform allows to estimate the parameters of a similitude (*rotation and homothety*) like a *translation* vector, using a log-polar representation of the frequency space  $(u, v) \rightarrow (\theta, \log \rho)$ :

Consider  $g$  the image transformed from  $f$ , by a rotation of angle  $\alpha$ , an homothety of ratio  $\rho$ , and a translation of vector  $(x_0, y_0)$ :

$$g(x, y) = f(\sigma(\cos \alpha x + \sin \alpha y) - x_0, \sigma(-\sin \alpha x + \cos \alpha y) - y_0)$$

The magnitudes of the Fourier transforms of  $f$  and  $g$  are related as follows:

$$\|G(u, v)\| = \frac{1}{\sigma^2} \|F(\frac{1}{\sigma}(u \cos \alpha + v \sin \alpha), \frac{1}{\sigma}(-u \sin \alpha + v \cos \alpha))\|$$

meaning that the magnitude:  $\left\{ \begin{array}{l} \cdot \text{ does not depend on the translation } (x_0, y_0). \\ \cdot \text{ undergoes a rotation of angle } \alpha. \\ \cdot \text{ undergoes a scaling of ratio } 1/\sigma. \end{array} \right.$

By expressing the frequencies in polar coordinates:

$$F_p(\theta, \rho) = \|F(\rho \cos \theta, \rho \sin \theta)\|; 0 \leq \theta \leq 2\pi, 0 \leq \rho < \infty$$

$$G_p(\theta, \rho) = \|G(\rho \cos \theta, \rho \sin \theta)\|; 0 \leq \theta \leq 2\pi, 0 \leq \rho < \infty$$

we get:

$$G_p(\theta, \rho) = \frac{1}{\sigma^2} F_p\left(\theta - \alpha, \frac{\rho}{\sigma}\right)$$

Finally, by taking the logarithm of the radial coordinate:

$$r = \log \rho \quad F_{lp}(\theta, r) = F_p(\theta, \rho)$$

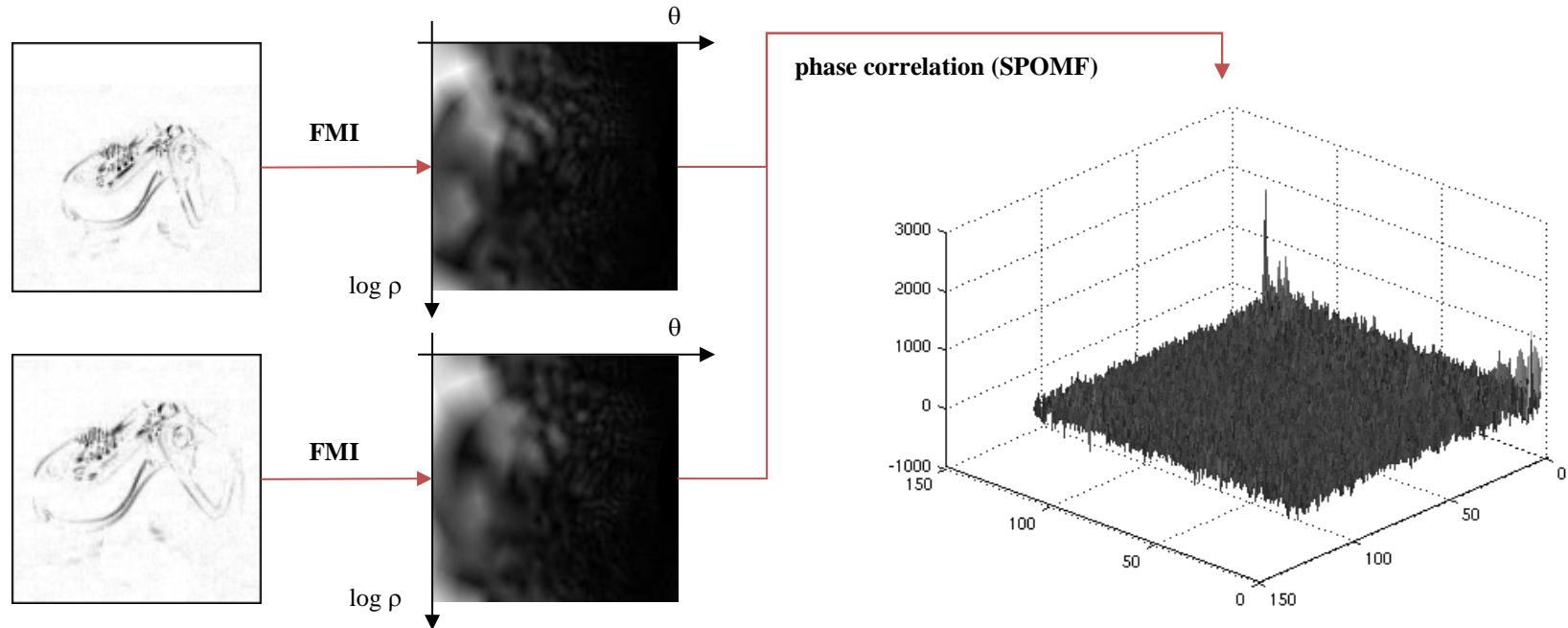
$$s = \log \sigma \quad G_{lp}(\theta, r) = G_p(\theta, \rho)$$

we get:

$$G_{lp}(\theta, r) = \frac{1}{\sigma^2} F_{lp}(\theta - \alpha, r - s)$$

Then a *similitude* in the image space corresponds to a *translation* in the space of *log-polar frequencies*.

# FOURIER-MELLIN INVARIANTS: FMI-SPOMF



Using the Fourier-Mellin transform to estimate the position of Aibo robot's head by phase correlation of Fourier-Mellin Invariants. (FMI-SPOMF: Fourier-Mellin Invariant Symmetric Phase Only Matched Filtering): **J.C. Baillie et M. Nottale** 2004.



Phase information from the original image is lost in the FMI. The FMI-SPOMF only looks for the best (rotation, homothety) that put 2 magnitude spectra in correspondence. *The translation parameters are lost, and the shape information carried by the phase is lost too!*

Also note that, like the phase correlation method, the FMI-SPOMF is used in general to estimate global transformation, since it uses contribution from the whole spectrum, which implies a large spatial scope of contributed pixels.



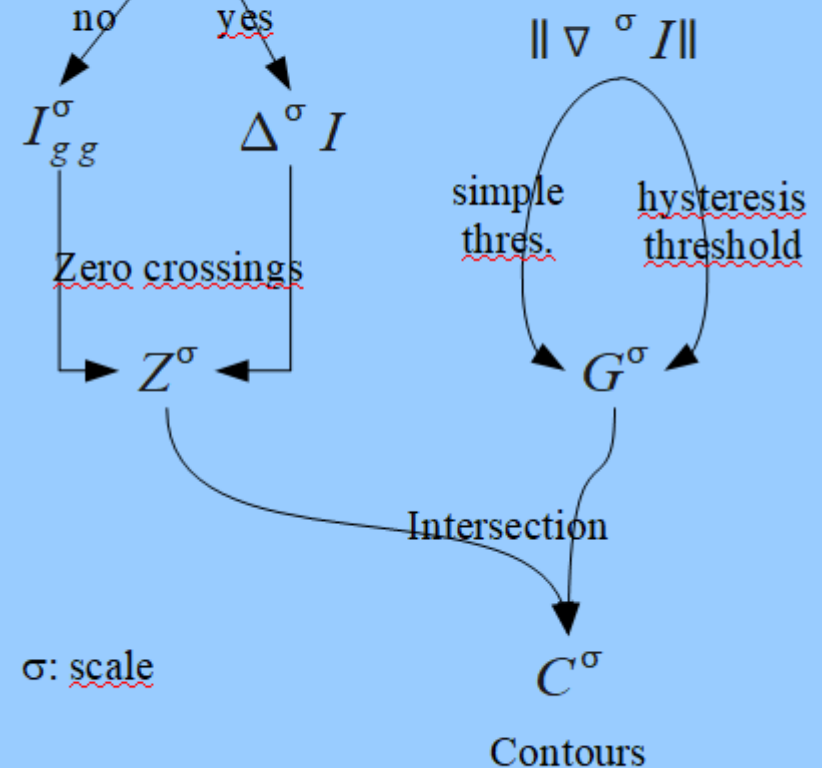
# CONCLUSIONS: MULTISCALE DERIVATIVES AND CONTOURS

## MULTISCALE DERIVATIVES

- Derivative estimated at a given scale (variance of the Gaussian)
- Order 1, Gradient: Contrast, Direction...
- Order 2, Hessian: Curvature, Contrast, Direction...
- Continuum from the local (geometry) to the global (statistics).

## CONTOURS

Neglected curvature?



### DETECTORS AND DESCRIPTORS

**Detector:** reduce the data support → repeatable *and*/vs representative.

- Corners: Maxima of curvature, Harris, FAST...
- Blobs: Determinant of Hessian, SIFT, SURF...

**Descriptor:** data representation → invariant *and*/vs discriminant.

- Differential invariants: colour (intensity), contrast, Laplacian,...
- Histograms of contrast-invariant features: direction, curvature,...

**Local:** geometrical → contour, curvature, corner, blob...

**Global:** statistical → histogram, magnitude / phase spectrum...

In between: **multiscale analysis** → continuum...

## REFERENCES

- **C. Harris & M. Stephens 1988** « *A combined corner and edge detector* » *Alvey Vision Conference* pp 147-151
- **A.P. Witkin 1983** « *Scale-space filtering* » *8th Int. Joint Conf. On Artificial Intelligence*, vol.2, pp1019-1022.
- **D.G. Lowe 2004** « *Distinctive Image Features from Scale-Invariant Keypoints* » *International Journal of Computer Vision* 60(2) pp 91-110
- **C. Schmid, R. Mohr & C. Bauckhage 2000** « *Evaluation of Interest Point Detectors* » *Int. Journal of Computer Vision* 37(2) pp 151-172
- **C. Schmid & R. Mohr 1997** « *Local grayvalue invariants for image retrieval* » *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(5) pp 530-534
- **E. Rosten & T. Drummond** “Fusing points and lines for high performance tracking” *Int. Conf. on Computer Vision (ICCV 2005)*, 1508—1511, **2005**.
- **H. Bay, T. Tuytelaars & L. Van Gool** “SURF: Speeded up robust features”, *Computer Vision and Image Understanding*, 110 (3), June, **2008**, 346-359

## REFERENCES

- **A. Hyvriinen, J. Hurri & P.O. Hoyer** « *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision* », Springer Publishing Company, 2009
- **N. Dalal & B. Triggs 2005** « *Histogram of oriented gradients for human detection* », Int. Conf. Of Computer Vision and Pattern recognition (CVPR), 2005
- **G. Csurka, C.R. Dance, L. Fan, J. Willamowski & C. Bray**, "Visual categorization with bags of keypoints", In Workshop on Statistical Learning in Computer Vision, ECCV, 2004.
- **B. Tomasik, P. Thiha & D. Turnbull 2009** « *Tagging products using image classification* », SIGIR 2009.
- **H. Foroosh, J. Zerubia & M. Berthod 2002** « *Extension of phase correlation to subpixel registration* » IEEE Transactions on Image Processing 11(3) pp 188-200
- **Q. Chen, M. Defrise & F. Deconinck 1994** « *Symmetric Phase-Only Matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition* » IEEE Transactions on Pattern Analysis and Machine Intelligence 16(12) pp 1156-1168