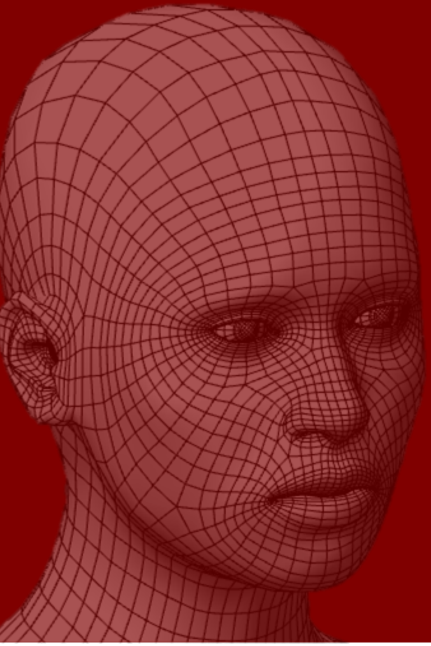


Advanced Image Processing

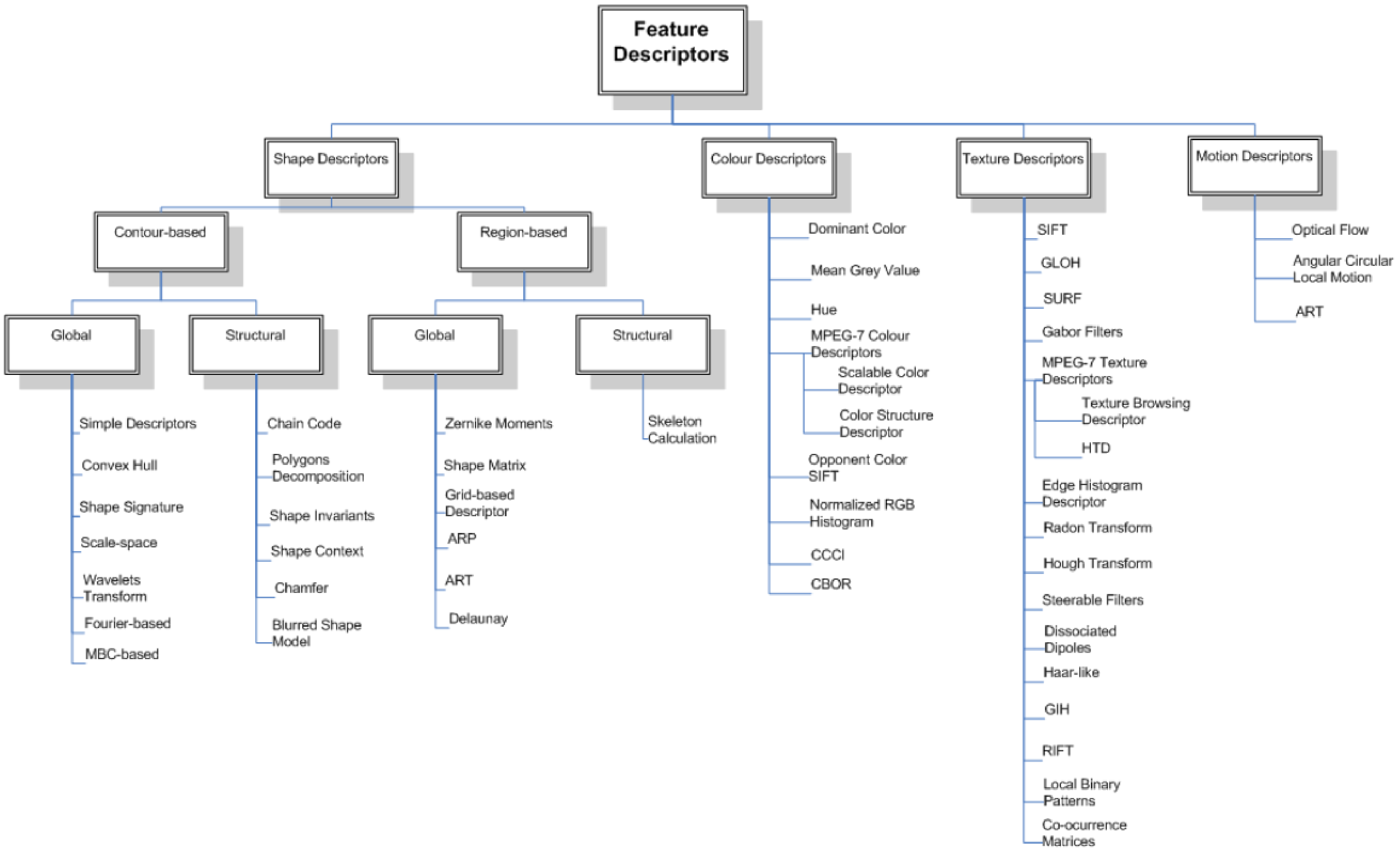
Lesson 3: Feature Detectors and Feature Descriptors: Where We are Now ?

Speaker: Alice OTHMANI, PhD
Associate professor at UPEC

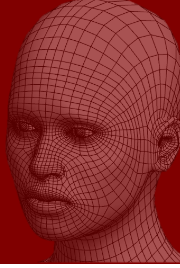
Email: alice.othmani@u-pec.fr

UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE

Feature Descriptors



Motion Descriptors



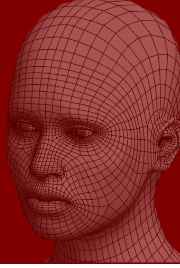
- Motion can be related to:
  - the **movement of an object in a scene** (for example, by checking its position in two consecutive frames)
  - the **movement of the device** that captures the image (such as zooming or panning of the camera).

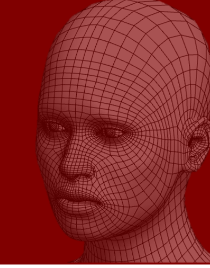
- The motion descriptor is a so-called **signature description of the motion trajectories**

UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE

Motion Descriptors

- describe motion **in video sequences**
- describe **spatio-temporal relationship** between image objects
- Applications:
  - Traffic monitoring
  - Security surveillance
  - Objects Tracking (Cell tracking, …)
  - Movement detection
  - Robot Navigation
  - etc, ..





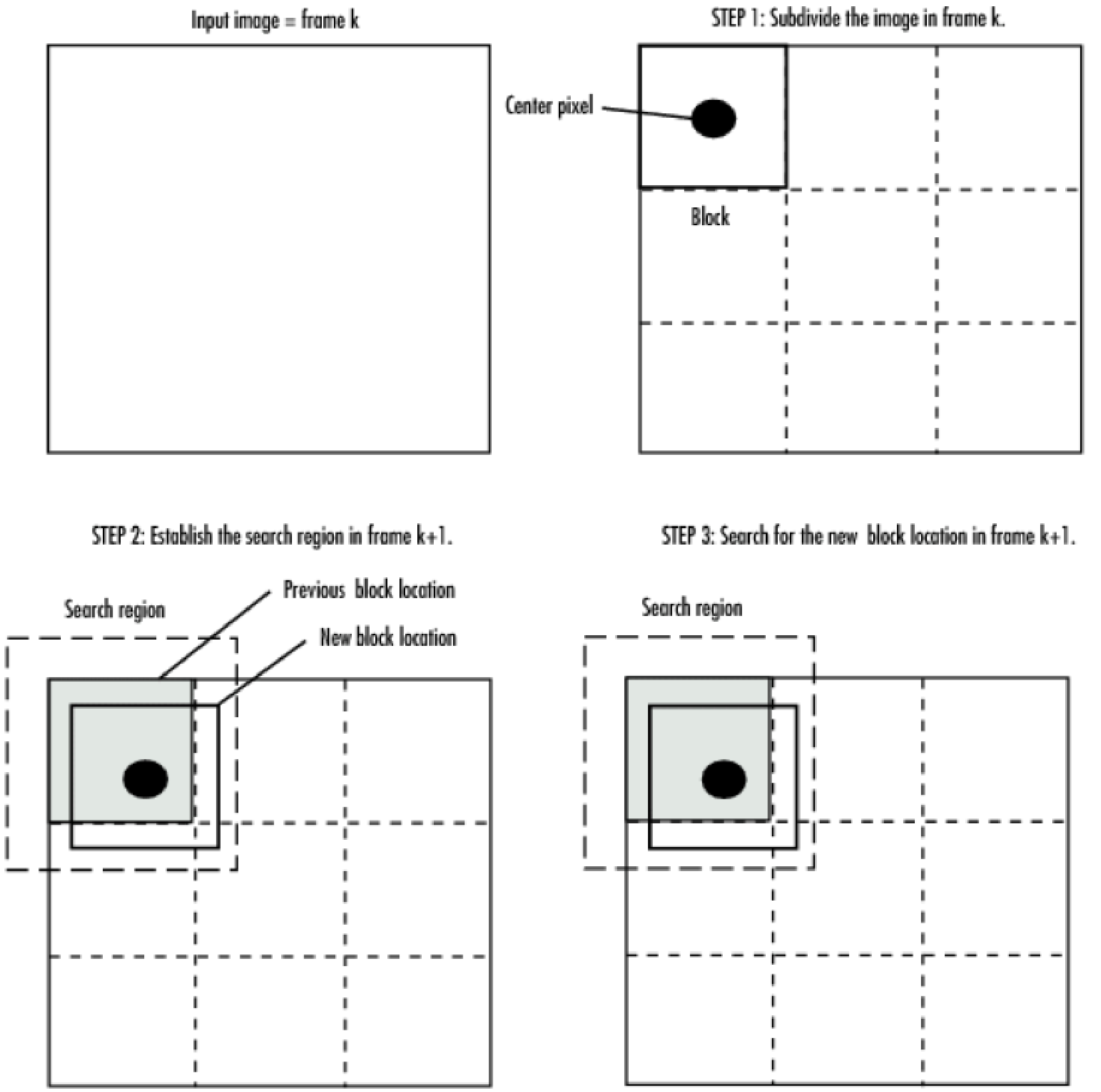
Block Matching

- Block Matching is the **simplest** algorithm for the estimation of local motion.
- It uses a **spatially constant and temporally linear motion** model over a **rectangular region** of support.
- The Block Matching module estimates motion between two images or two video frames using **"blocks" of pixels**.

→ Useful given its **simplicity** and **regularity** (it applies the same operations for each block of the image).

UPEC Connaissance - Action | UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE

Block Matching


Input image = frame k
STEP 1: Subdivide the image in frame k.
Center pixel
Block
STEP 2: Establish the search region in frame k+1.
Search region
Previous block location
New block location
STEP 3: Search for the new block location in frame k+1.
Search region

Horn&Schunck Optical Flow

Brightness constancy assumption

$$f(x,y,t)=f(x+dx,y+dy,t+dt)$$

$\Downarrow$ Taylor Series

$$f(x,y,t)=f(x,y,t)+\frac{\partial f}{\partial x}dx+\frac{\partial f}{\partial y}dy+\frac{\partial f}{\partial t}dt$$

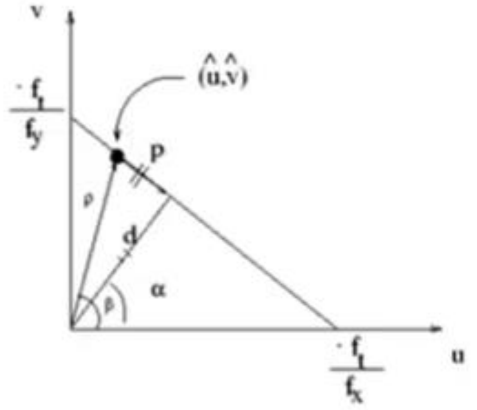
$$f_x dx+f_y dy+f_t dt=0$$

$$f_x u+f_y v+f_t=0$$

Interpretation of optical flow eq

$$f_x u + f_y v + f_t = 0$$

$$v = -\frac{f_x}{f_y}u - \frac{f_t}{f_y}$$



Equation of st.line

$$d = \frac{f_t}{\sqrt{f_x^2 + f_y^2}}$$

Optical Flow

Horn&Schunck (contd)

$$\iint \{(f_x u + f_y v + f_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\}dxdy$$

Brightness constancy

Smoothness constraint

$$\Big\downarrow \text{min}$$

$$(f_x u + f_y v + f_t) f_x + \lambda(\Delta^2 u) = 0$$
$$(f_x u + f_y v + f_t) f_y + \lambda((\Delta^2 v) = 0$$

UPEC
Connaissance - Action
UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE

Horn&Schunck (contd)

$$\iint \{(f_x u + f_y v + f_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\} dxdy$$

$\Downarrow$ min

variational calculus

$$(f_x u + f_y v + f_t) f_x + \lambda(\Delta^2 u) = 0$$
$$(f_x u + f_y v + f_t) f_y + \lambda((\Delta^2 v) = 0$$

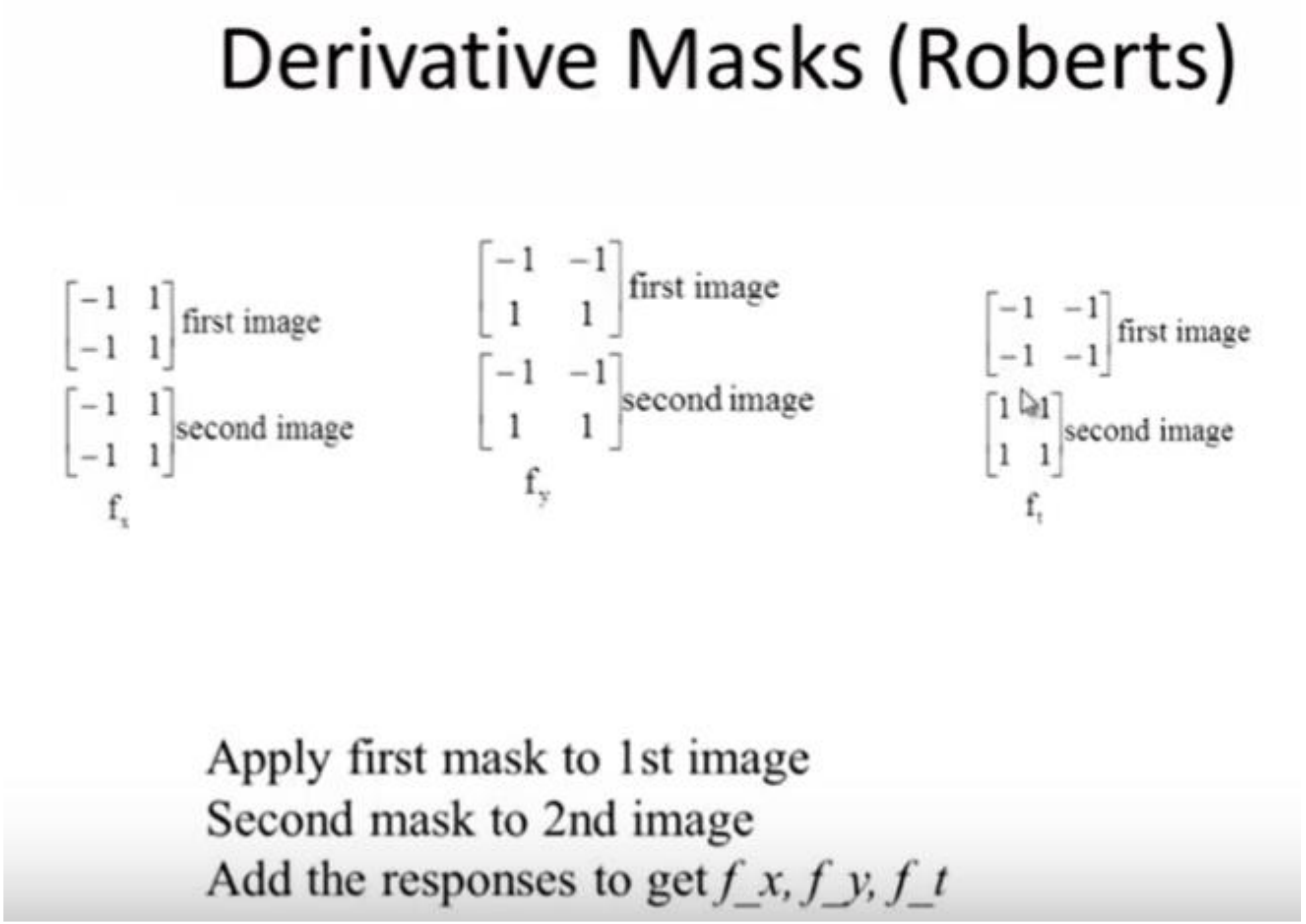
$\Downarrow$ discrete version

$$(f_x u + f_y v + f_t) f_x + \lambda(u - u_{av}) = 0$$
$$(f_x u + f_y v + f_t) f_y + \lambda((v - v_{av}) = 0$$

$$\Delta^2 u = u_{xx} + u_{yy}$$

UPEC Connaissance - Action | UNIVERSITE PARIS-EST CRÉTEIL VAL DE MARNE

Optical Flow



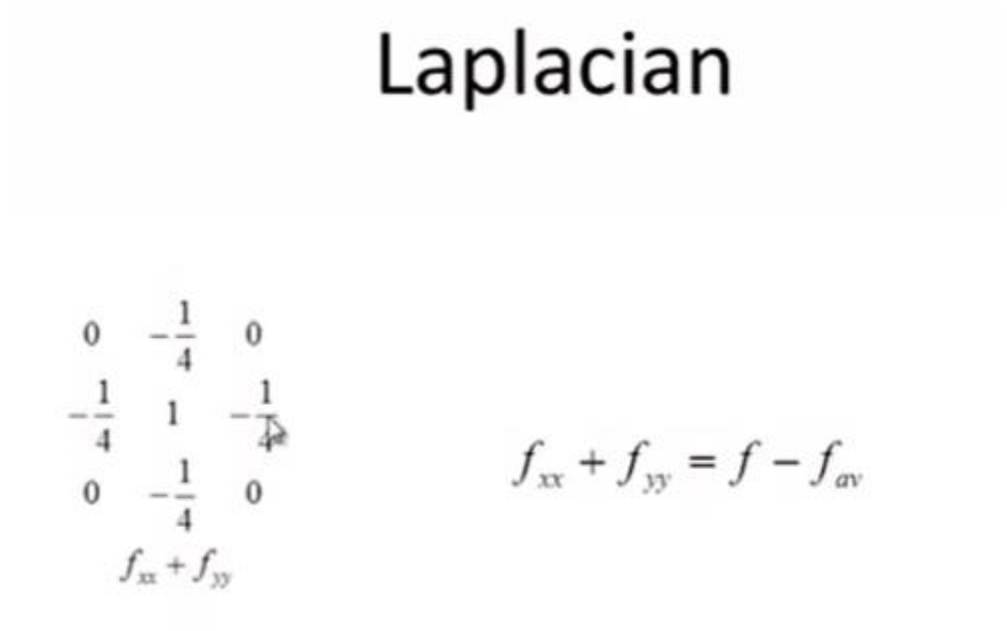
Derivative Masks (Roberts)

$$\begin{array}{ll} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} & \text{first image} \\ \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} & \text{second image} \\ f_x & \end{array} \qquad \begin{array}{ll} \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} & \text{first image} \\ \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} & \text{second image} \\ f_y & \end{array} \qquad \begin{array}{ll} \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} & \text{first image} \\ \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} & \text{second image} \\ f_t & \end{array}$$

Apply first mask to 1st image
Second mask to 2nd image
Add the responses to get *f_x, f_y, f_t*

UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE

Optical Flow



Laplacian

$$\begin{matrix} 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 \end{matrix}$$

$f_{xx} + f_{yy}$

$$f_{xx} + f_{yy} = f - f_{av}$$

Horn&Schunck (contd)

$$\iint \{(f_x u + f_y v + f_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\} dx dy$$

min

$$(f_x u + f_y v + f_t) f_x + \lambda(\Delta^2 u) = 0$$
$$(f_x u + f_y v + f_t) f_y + \lambda((\Delta^2 v) = 0$$

variational calculus

$$u = u_{av} - f_x \frac{P}{D}$$
$$v = v_{av} - f_y \frac{P}{D}$$

discrete version

$$(f_x u + f_y v + f_t) f_x + \lambda(u - u_{av}) = 0$$
$$(f_x u + f_y v + f_t) f_y + \lambda((v - v_{av}) = 0$$

$$P = f_x u_{av} + f_y v_{av} + f_t$$
$$D = \lambda + f_x^2 + f_y^2$$

$$\Delta^2 u = u_{xx} + u_{yy}$$

UPEC UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE
Connaissance - Action

Algorithm-1

- k=0
- Initialize u^K v^K
- Repeat until some error measure is satisfied (converges)

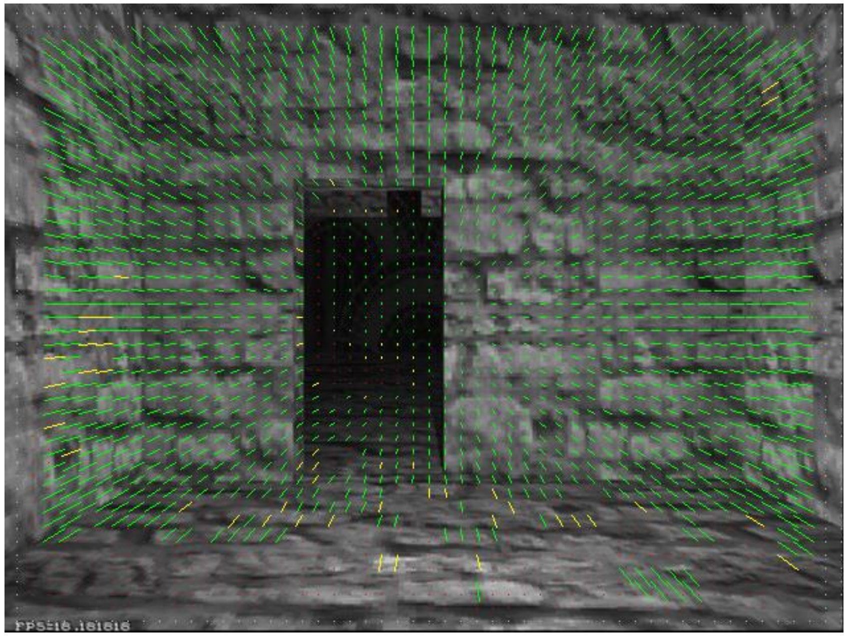
$$u = u_{av} - f_x \frac{P}{D}$$
$$v = v_{av} - f_y \frac{P}{D}$$

$$P = f_x u_{av} + f_y v_{av} + f_t$$
$$D = \lambda + f_x^2 + f_y^2$$

Optical Flow

Descriptor: The descriptor will consist on the motion vectors that will describe the relative movement between the two consecutive images.

FPS=18.181818



Optical Flow Lucas-Kanade method

We have seen an assumption before, that all the neighbouring pixels will have similar motion. Lucas-Kanade method takes a 3x3 patch around the point. So all the 9 points have the same motion. So now our problem becomes solving 9 equations with two unknown variables which is over-determined

- Optical flow eq

$$f_x u + f_y v = -f_t$$

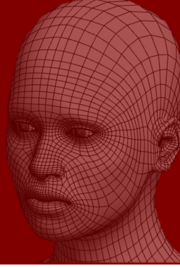
- Consider 3 by 3 window

$$\begin{aligned} f_{x1}u + f_{y1}v &= -f_{t1} \\ &\vdots \\ f_{x9}u + f_{y9}v &= -f_{t9} \end{aligned}$$

$$\begin{bmatrix} f_{x1} & f_{y1} \\ \vdots & \vdots \\ \\ f_{x9} & f_{y9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_{t1} \\ \vdots \\ \\ -f_{t9} \end{bmatrix}$$

$$\mathbf{Au} = \mathbf{f}_t$$

UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE



Optical Flow Lucas-Kanade method

$$\mathbf{A}\mathbf{u} = \mathbf{f}_t$$

$$\mathbf{A}^T\mathbf{A}\mathbf{u} = \mathbf{A}^T\mathbf{f}_t$$

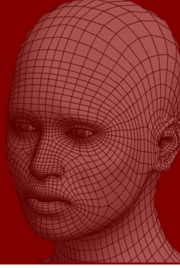
$$\mathbf{u} = \boxed{(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}}^T\mathbf{f}_t$$

Pseudo Inverse

$$\Big\uparrow$$

$$\min \sum_i (f_{xi}u + f_{yi}v + f_t)^2$$

Least Squares Fit



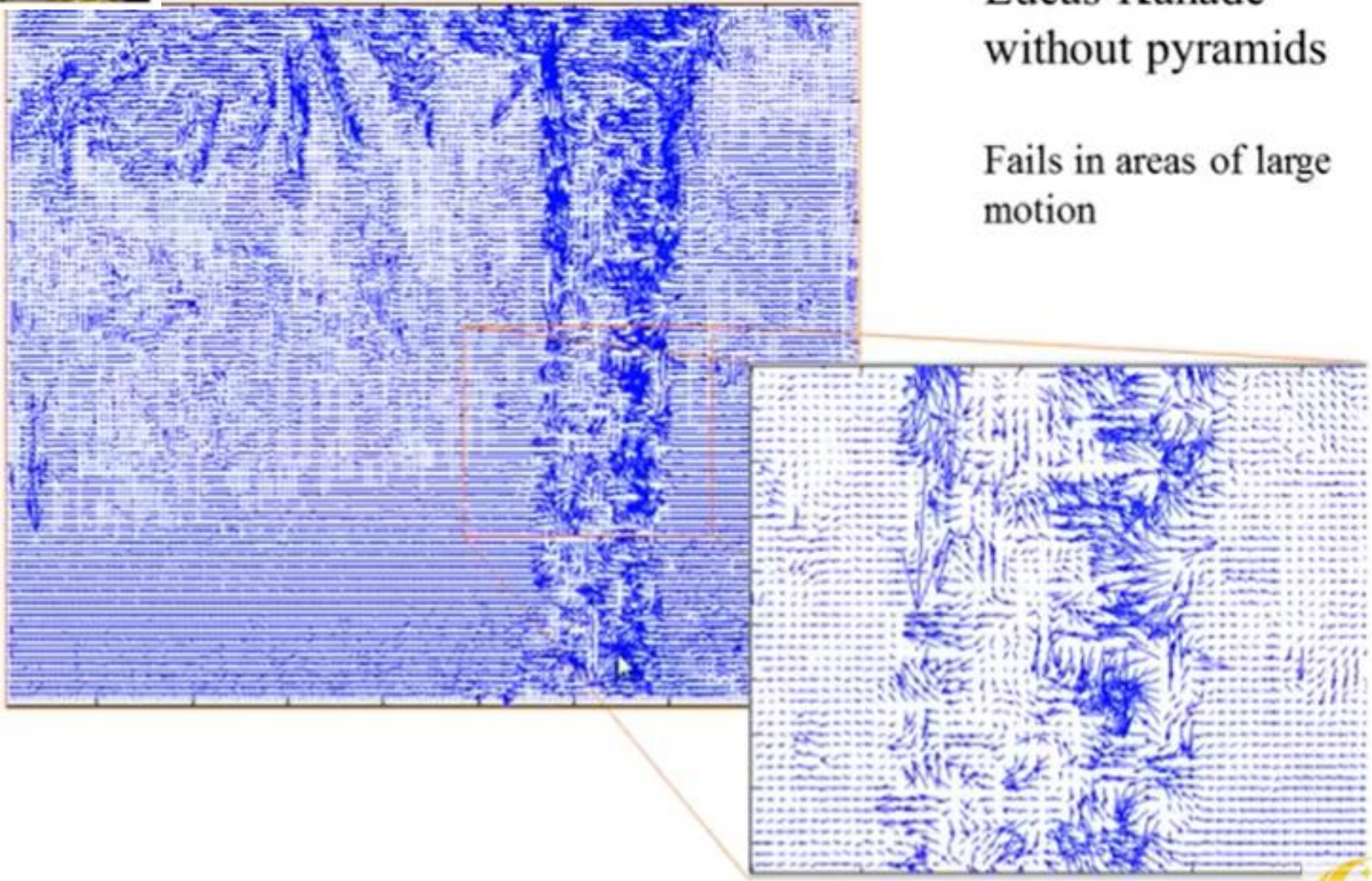
Optical Flow Lucas-Kanade method

Lucas & Kanade

$$u = \frac{-\sum f_{yi}^2 \sum f_{xi} f_{ti} + \sum f_{xi} f_{yi} \sum f_{yi} f_{ti}}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi} f_{yi})^2}$$

$$v = \frac{\sum f_{xi} f_{ti} \sum f_{xi} f_{yi} - \sum f_{xi}^2 \sum f_{yi} f_{ti}}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi} f_{yi})^2}$$

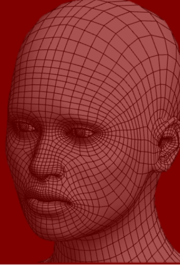
Optical Flow Lucas-Kanade method



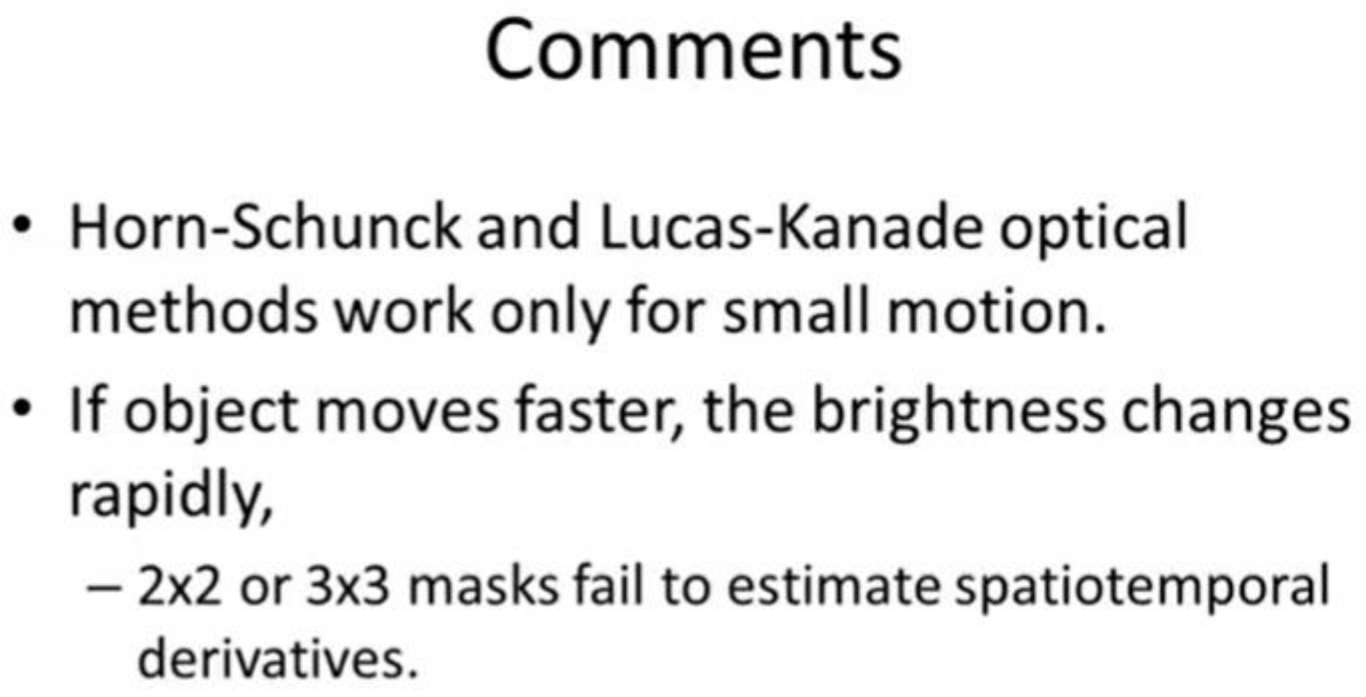
Lucas-Kanade without pyramids

Fails in areas of large motion

UPEC Connaissance - Action



Optical Flow Lucas-Kanade method



Comments

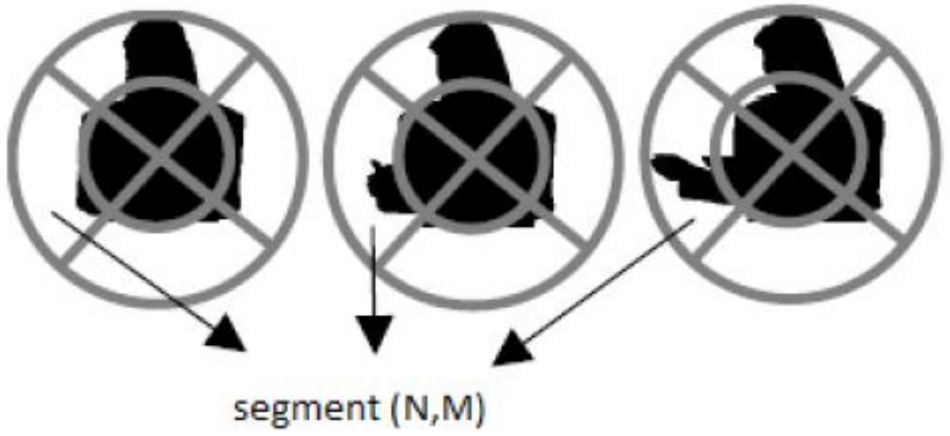
- Horn-Schunck and Lucas-Kanade optical methods work only for small motion.
- If object moves faster, the brightness changes rapidly,
  – 2x2 or 3x3 masks fail to estimate spatiotemporal derivatives.

Angular Circular Motion

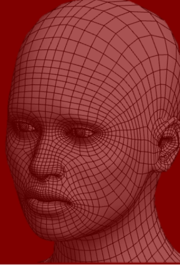
- divide the mask of a shape of a video object in M angular and N circular segments

- use the variation of the pixels that belong to each segment in order to describe local movement.

Descriptor: will consist of, for each segment, the variation of the intensity value of the pixels that belong to the concrete segment.



Feature Detection



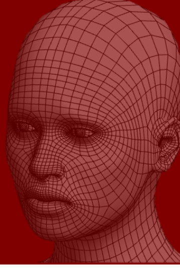
- **Feature Detection** is a compulsory step to do in order to obtain **Feature Descriptors.**

- Features Descriptors **locates points and regions;**
- They are generally capable of reproducing similar performance that the human would provide in locating elemental features in images.

- Features can be defined as singular visual traits, associated to the visual primitives that constitute an object, such as **edges, corners or lines** among others.

UPEC Connaissance - Action
UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE

Feature Detection



- **Feature Detection** is often a **low-level** image processing operation.
  - It is usually performed as the first operation on an image, and it examines every pixel to assess the presence of a feature at that pixel.

- There are several Feature Detection methods and we have divided them into four main groups: **Edge Detectors**, **Corner Detectors**, **Blob Detectors** and **Region Detectors.**

Feature Detection

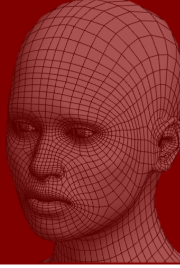
- **Edges** are the points in a digital image at which the **image brightness changes sharply** or more formally, has discontinuities.

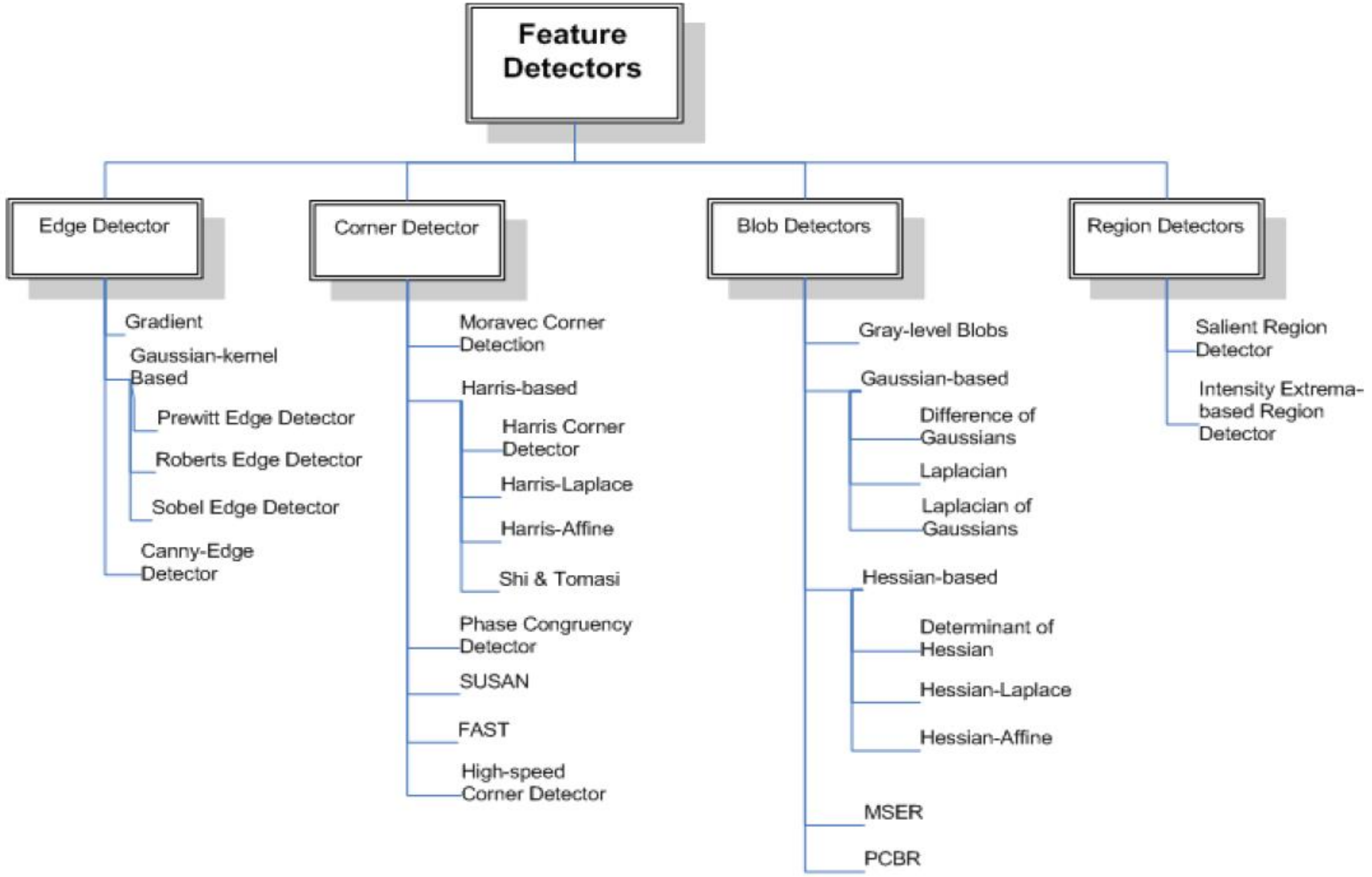
- A **corner** can be defined as the intersection of two edges or as a point for which there are two dominant and different edge directions in a local neighbourhood.

- **Blobs** are regions in the image that are either brighter or darker than the surrounding.

- A **Region of Interest** (**ROI**) as an area of the image with **uniform distribution** that is either limited by contours or a significant change in some of its characteristics such as colour or texture.

Feature Detection



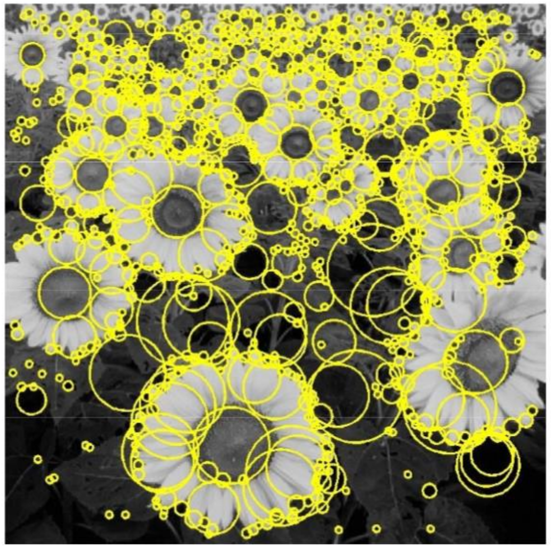


Blob Detectors

Blob detection methods are aimed at detecting regions in a digital image that differ in properties, such as **brightness** or **color**, compared to surrounding regions.

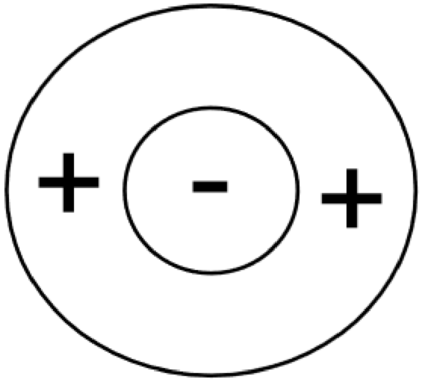
- **Gaussian-based Detectors:**

  Gaussian functions are used as smoothing kernels for generating multi-scale representations.



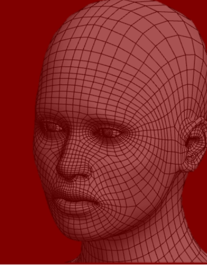
What is a good blob detector?

- A filter that has versions at **multiple scales.**
- The biggest response should be when the filter has the **same location and scale** as the blob.



When does this have biggest response ?

- When inside is as dark as possible
- And outside is as light as possible.
- Ie, a **dark spot.**
- Note, this locates position and scale.



Blob Detectors

Difference of Gaussians :

- This is not a Blob Detector by itself but an algorithm that consists of the subtraction of one blurred version of an original grey level image from another less blurred version of the original.

  → **feature enhancement algorithm**

The kernel's calculation

$$f(x,y,\sigma) = \frac{1}{2\pi\sigma^2} exp(-\frac{(x^2+y^2)}{2\sigma^2}) - \frac{1}{2\pi K^2\sigma^2} exp(-\frac{(x^2+y^2)}{2\sigma_2^2})$$

- The blurred images are obtained by convolving the original image with Gaussian kernels using different standard deviations.

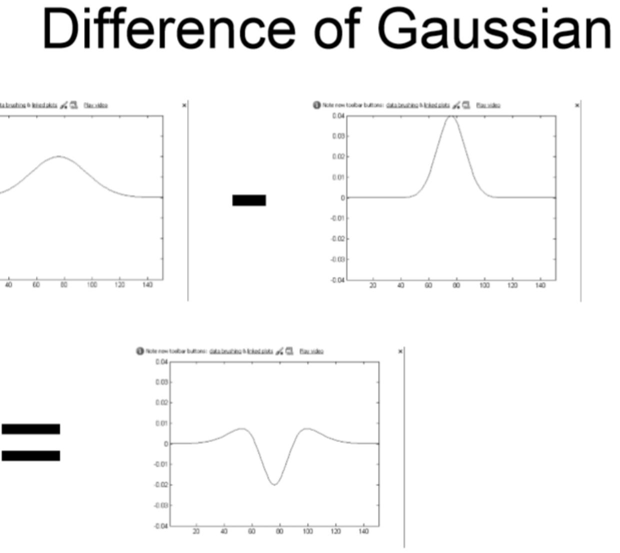
- Blurring an image by using a Gaussian kernel only **suppresses high-frequency spatial information** and that by subtracting one image from the other one helps to **preserve spatial information** that lies in the range of frequencies preserved, acting like a **band-pass filter**.

Difference of Gaussian

1. **Filter with Gaussian at different scales**
   This is done by just repeatedly filtering with the same Gaussian.

2. **Subtract image filtered at one scale with image filtered at previous scale.**

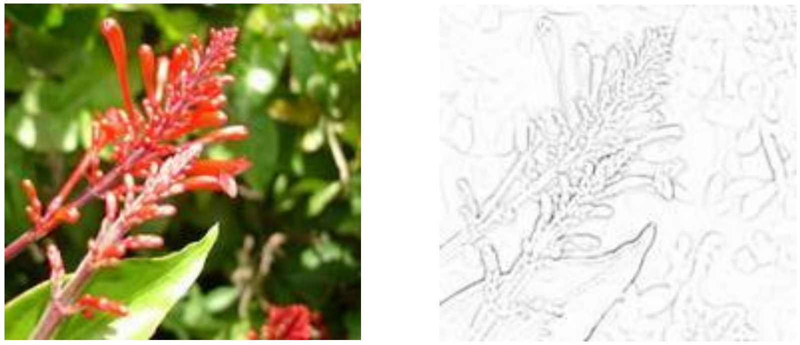
3. **Look for local extrema**
   A pixel is bigger (smaller) than all eight neighbors, and all nine neighboring pixels at neighboring scales.



UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE
Connaissance - Action

Blob Detectors

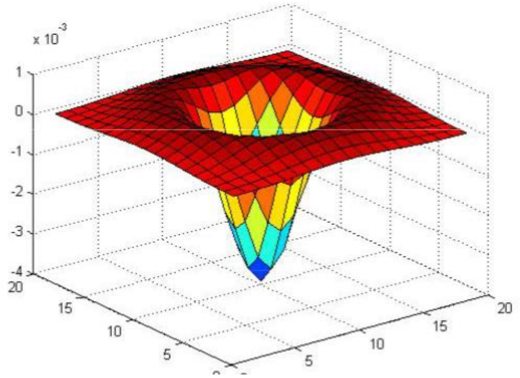
- The difference of Gaussians can be used to **increase the visibility of edges.**

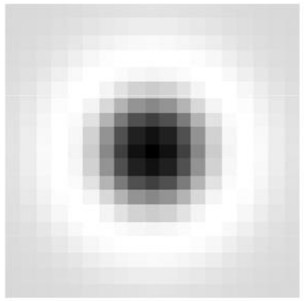


- DoG removes high frequency details that often include random noise → it can be considered as one of the **most suitable for processing images with a high degree of noise**.
- A major drawback to the application of the algorithm is an inherent reduction in overall image contrast produced by the operation.
- **Differences of Gaussians have also been used for blob detection as part of the *SIFT algorithm*.**

Laplacian of Gaussians

- One of the first and also **most common Blob Detectors** is based on the Laplacian of the Gaussian (LoG).
- It combines the use of two operators (Laplacian and Gaussian) to improve Blob Detection.
- Circularly symmetric operator for blob detection in 2D



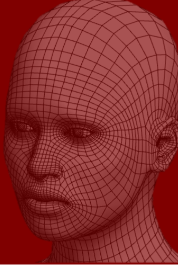


Scale-normalized: $\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$

(Slides from Lazebnik)

UPEC Connaissance - Action
UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE

Blob detection using LoG



Blob detection : The Laplacian of the Gaussian (LoG)

g is a gaussian kernel at a scale t

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

f is an image and * is a convolution

$$L(x,y;t) = g(x,y,t) * f(x,y)$$

applying the laplacian operator

$$\nabla^2 L = L_{xx} + L_{yy}$$

scale-normalized Laplacian operator

$$\nabla^2_{norm} L = t\,(L_{xx} + L_{yy})$$

Blobs is the local maxima/minima of the LoG

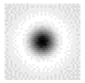
$$(\hat{x}, \hat{y}; \hat{t}) = \text{argmaxminlocal}_{(x,y;t)}((\nabla^2_{norm} L)(x,y;t)) .$$

UPEC Connaissance - Action | UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE

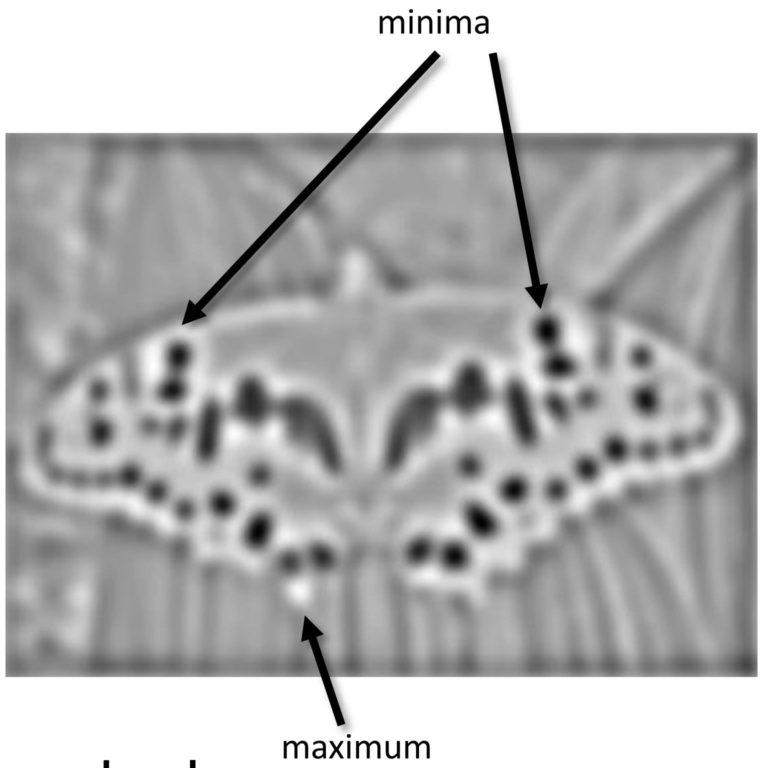
Laplacian of Gaussian

- "Blob" detector

$*$



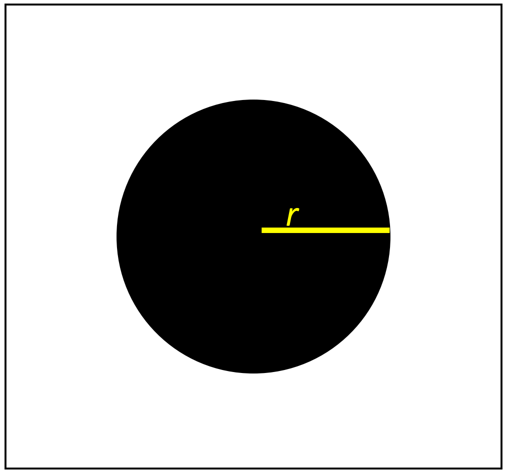
$=$



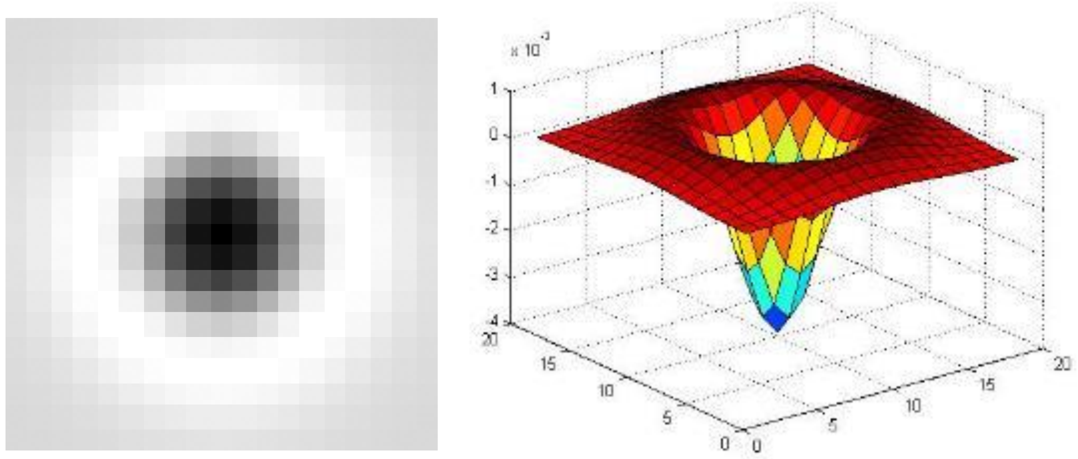
- Find maxima *and minima* of LoG operator in space and scale

Scale selection

- At what scale does the Laplacian achieve a maximum response for a binary circle of radius r?



image



Laplacian

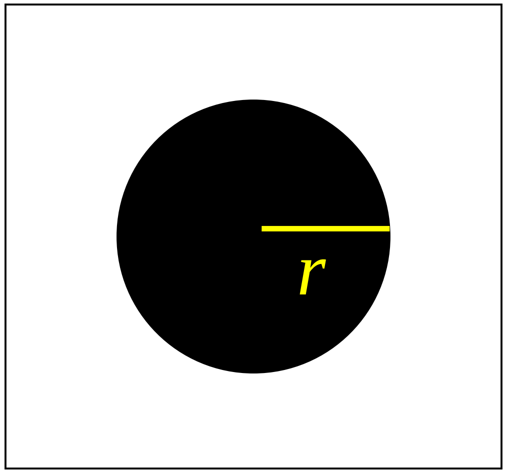
Scale selection

- The 2D Laplacian is given by

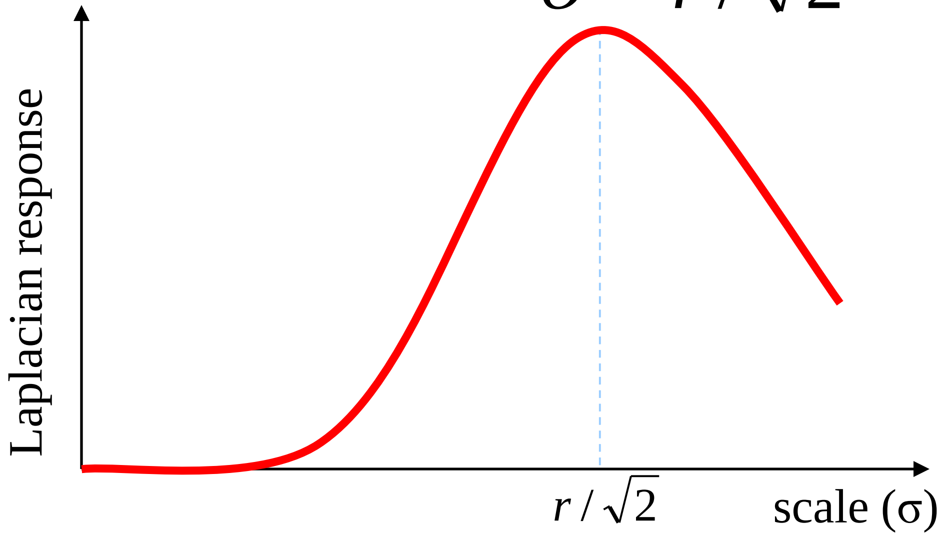
$$(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2+y^2)/2\sigma^2} \text{ (up to scale)}$$

- For a binary circle of radius r, the Laplacian achieves a maximum at

$$\sigma = r / \sqrt{2}$$

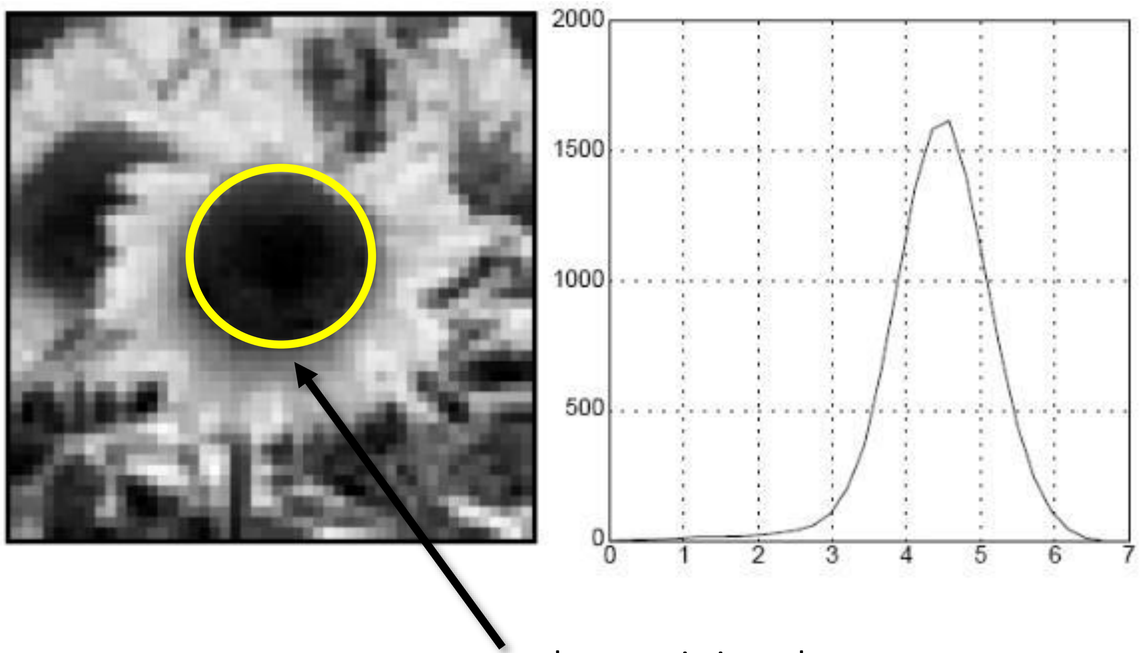


image



Characteristic scale

- We define the characteristic scale as the scale that produces peak of Laplacian response



characteristic scale

T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* **30** (2): pp 77--116.

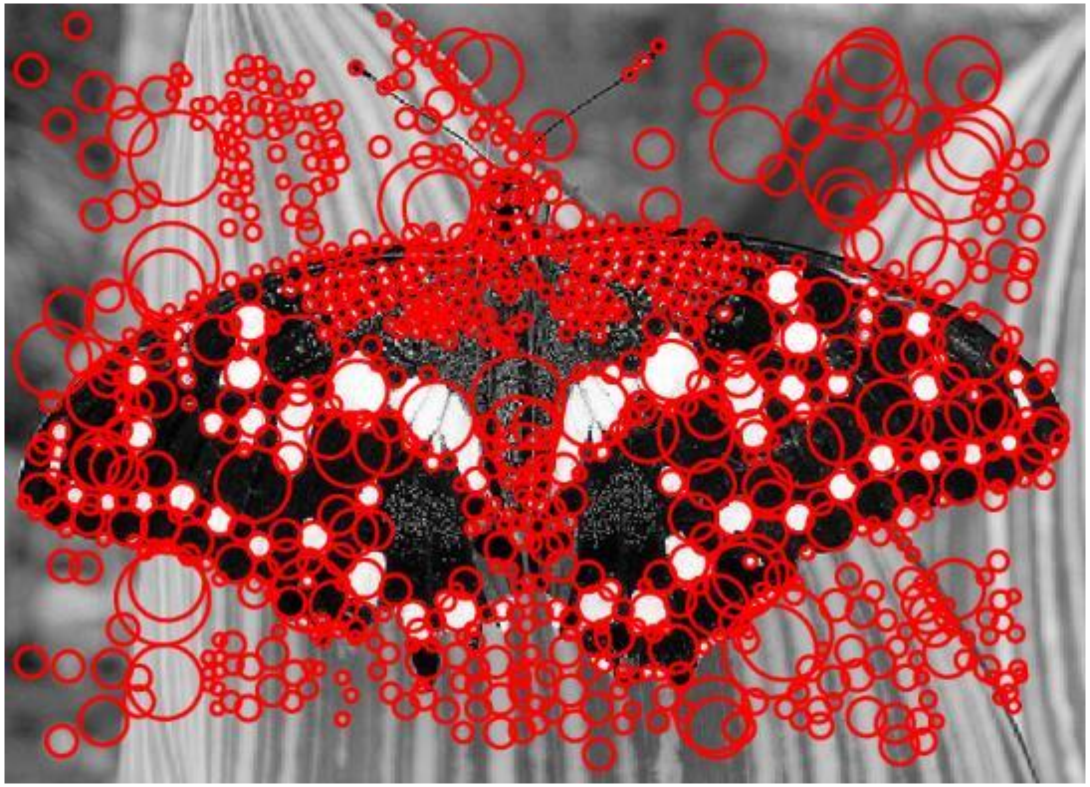
Scale-space blob detector: Example

Scale-space blob detector: Example

sigma = 11.9912

UNIVERSITE
PARIS-EST CRÉTEIL
VAL DE MARNE

Scale-space blob detector: Example



Hessian-based Detectors

- The Hessian matrix is the square matrix of second-order partial derivatives of a function.
- It describes the **local curvature** of a function of many variables.

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_n} \\[2.2ex] \dfrac{\partial^2 f}{\partial x_2\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\,\partial x_n} \\[2.2ex] \vdots & \vdots & \ddots & \vdots \\[2.2ex] \dfrac{\partial^2 f}{\partial x_n\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

Hessian-based Detectors

- The scale-normalized determinant of the Hessian

$$\det H_{norm} L = t^2 (L_{xx} L_{yy} - L_{xy}^2)$$

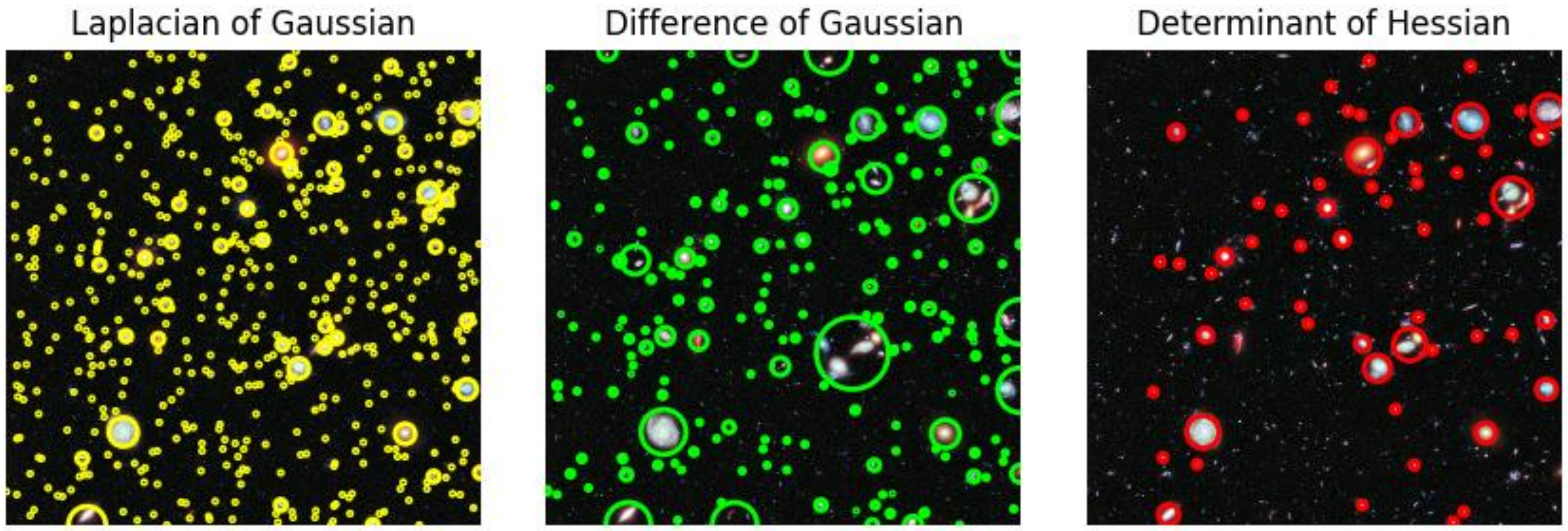
- The scale-space maxima are the blobs

$$(\hat{x}, \hat{y}; \hat{t}) = \operatorname{argmaxlocal}_{(x,y;t)} ((\det H_{norm} L)(x, y; t))$$

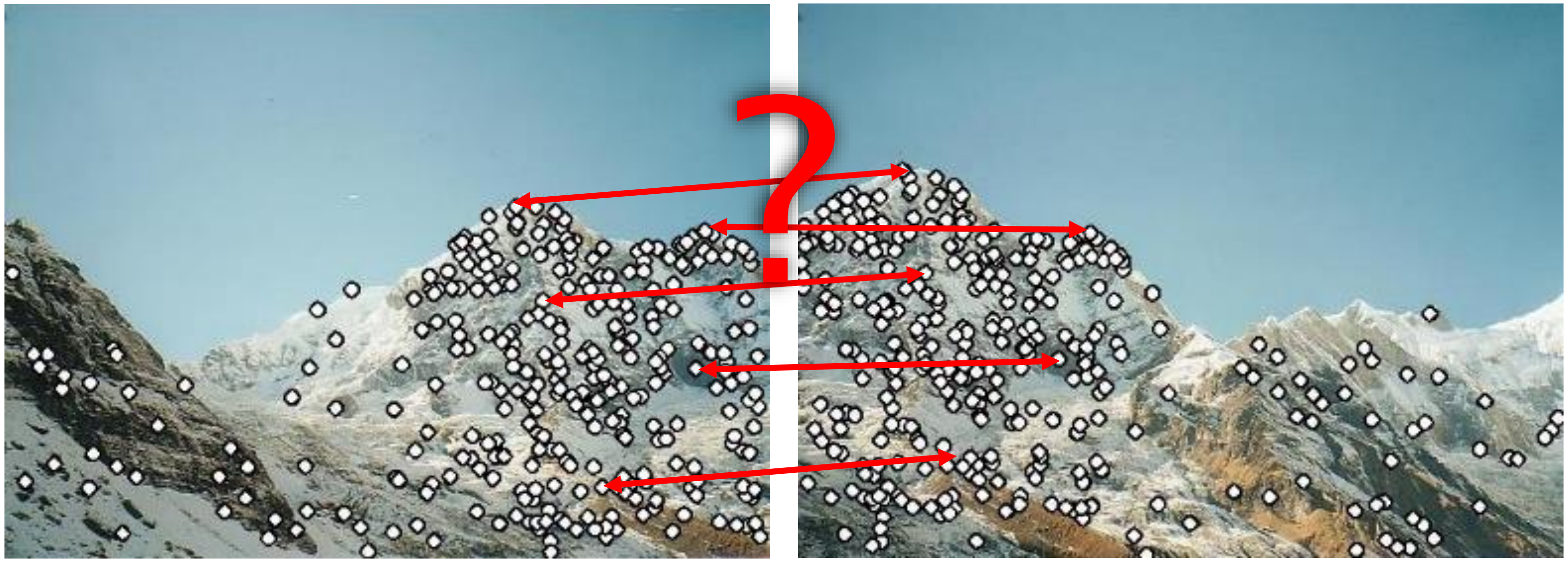
UPEC Connaissance - Action UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE

Python implementation : Scikit-image

http://scikit-image.org/docs/dev/auto_examples/features_detection/plot_blob.html



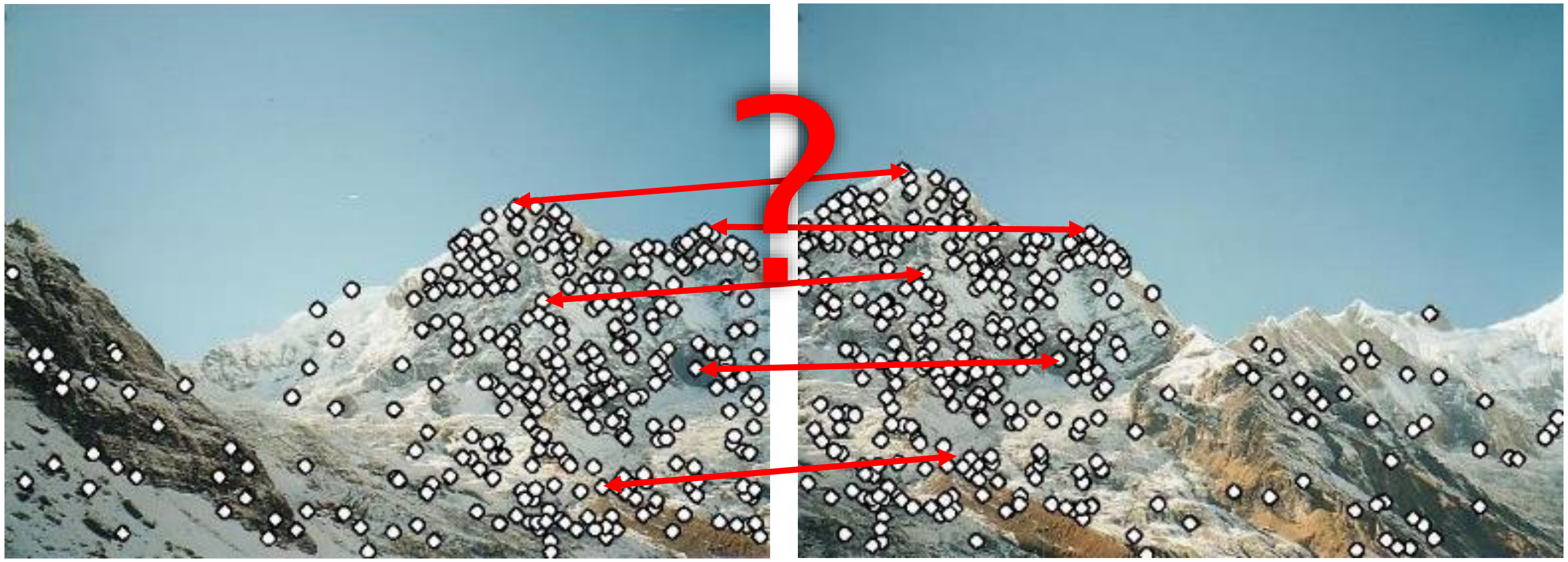
We know how to detect good points
Next question: How to match them?



Answer: Come up with a *descriptor* for each point, find similar descriptors between the two images

Feature descriptors

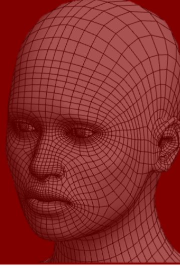
We know how to detect good points
Next question: How to match them?



Lots of possibilities (this is a popular research area)

- Simple option: match square windows around the point
- State of the art approach: SIFT
  - David Lowe, UBC http://www.cs.ubc.ca/~lowe/keypoints/

Invariance vs. discriminability

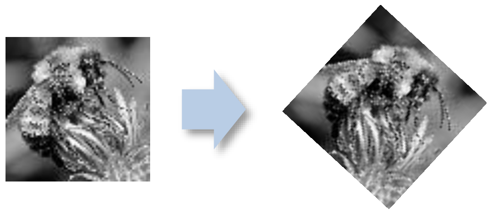


- **Invariance:**
  - Descriptor shouldn’t change even if image is transformed
- **Discriminability:**
  - Descriptor should be highly unique for each point

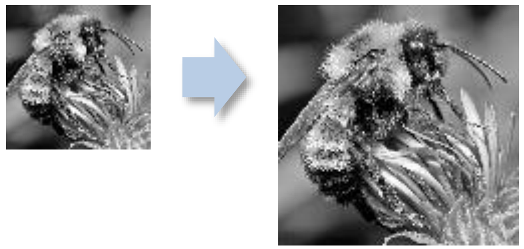
Image transformations

- ❑ **Geometric**

  Rotation

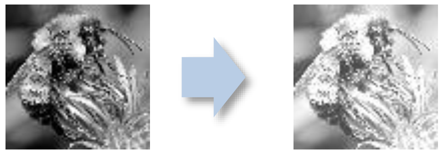


  Scale



- ❑ **Photometric**

  Intensity change

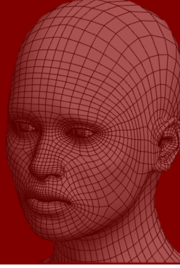


Maximally Stable Extremal Regions (MSER)

- MSER regions are connected areas characterized by almost uniform intensity, surrounded by contrasting background.
- They are constructed through a process of trying multiple thresholds.
- The selected regions are those that maintain unchanged shapes over a large set of thresholds.

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, *22*(10), 761-767.

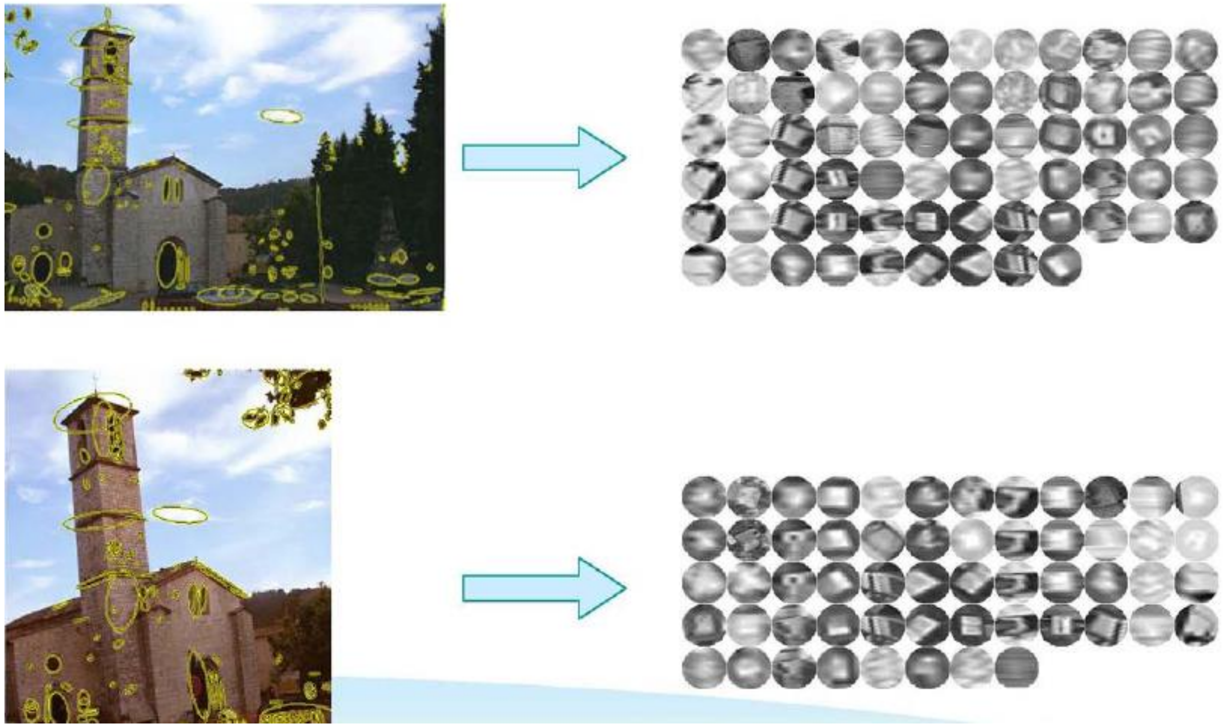
Maximally Stable Extremal Regions (MSER)

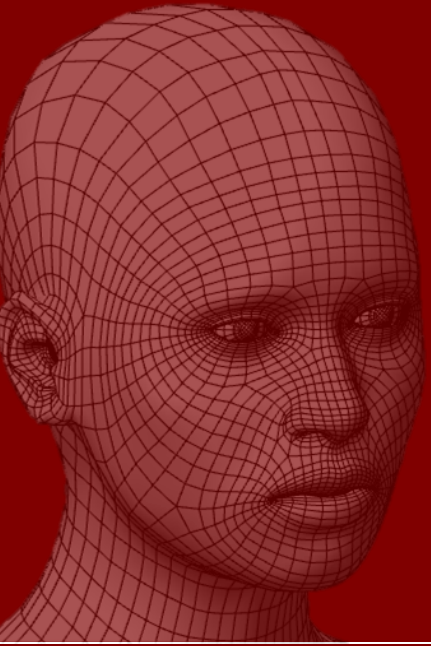


- For each threshold, compute the connected binary regions.
- Compute a function, area A(i), at each threshold value i.
- Analyze this function for each potential region to determine those that persist with similar function value over multiple thresholds.

UPEC Connaissance - Action | UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE

Examples of MSER Regions





Thank you for your attention

UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE