

Vedere Artificială (Computer Vision)

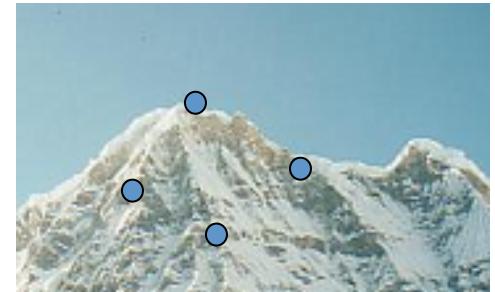
Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

anul 2, master Informatică, semestrul I, 2019-2020

Recapitulare – cursul trecut

- trăsături locale: detectare

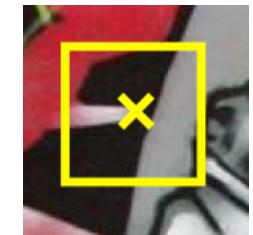
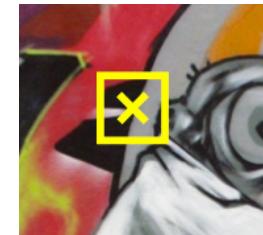


- detectorul Harris

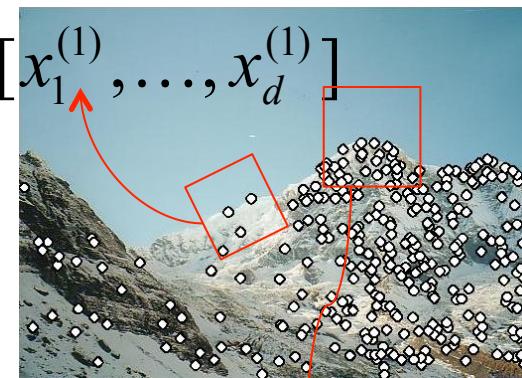


Planul cursului de azi

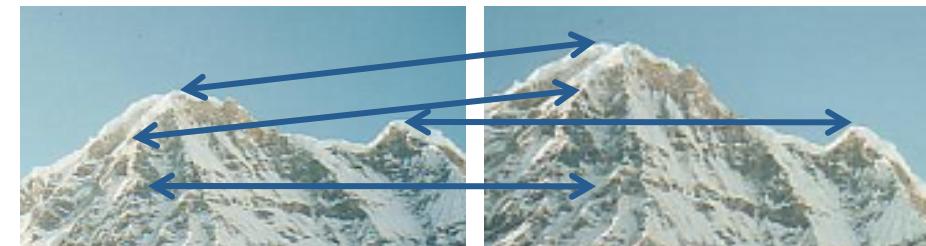
- 1) Detectare:** detectori de puncte de interes invariante la scală:
detectorul DoG, Harris-Laplace



- 2) Descriere:** descrie conținutul vizual din vecinătatea fiecărui punct de interes printr-un vector printr-un descriptor vizual (feature vector) - SIFT



- 3) Matching:** determină corespondențele dintre descriptorii dintre imagini



$$\mathbf{x}_2 = [x_1^{(2)}, \dots, x_d^{(2)}]$$

Detectorul de colțuri Harris

- 1) Calculează derivate parțiale I_x și I_y la fiecare pixel (x,y)
- 2) Calculează matricea M pentru o fereastră centrată în fiecare pixel (x,y) din imagine
- 3) Calculează funcția cornerness R la fiecare pixel (x,y)
- 4) Găsește punctele pentru care R are valori foarte mari ($>$ threshold).
- 5) Găsește maximele locale ale funcției, realizează suprimarea maximelor (non-maximum suppression)

C.Harris and M.Stephens. ["A Combined Corner and Edge Detector."](#) *Proceedings of the 4th Alvey Vision Conference*: pages 147–151, 1988.

Detectorul de colțuri Harris - rezultate



Detectorul de colțuri Harris

- 1) Calculează derivate parțiale I_x și I_y la fiecare pixel (x,y)
- 2) Calculează matricea M pentru o fereastră W centrată în fiecare pixel (x,y) din imagine
- 3) Calculează funcția cornerness R la fiecare pixel (x,y)
- 4) Găsește punctele pentru care R are valori foarte mari ($> \text{threshold}$).
- 5) Găsește maximele locale ale funcției, realizează suprimarea maximelor (non-maximum suppression)

Nu e invariant la scală -> pot rula detectorul la mai multe scale

Detectarea punctelor de interes invariante la scală

Detectorul Harris detectează colțuri la o singură scală = mărime = scale

Vrem să găsim un detector multi-scale, fiecare trăsătură locală detectată să aibă o scală caracteristică, covariantă cu zoom-ul într-o imagine. Găsirea scalei specifice conduce și la ajustarea dimensiunii ferestrei din jurul punctului detectat ca trăsătură

Cum putem detecta puncte de interes în fiecare imagine independent, astfel încât detectiile se regăsesc chiar la scale diferite?

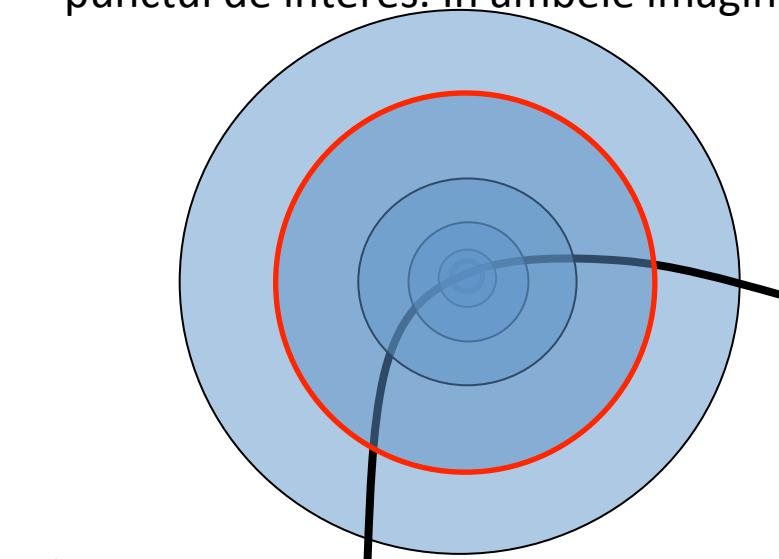


adaptat după K. Graumann

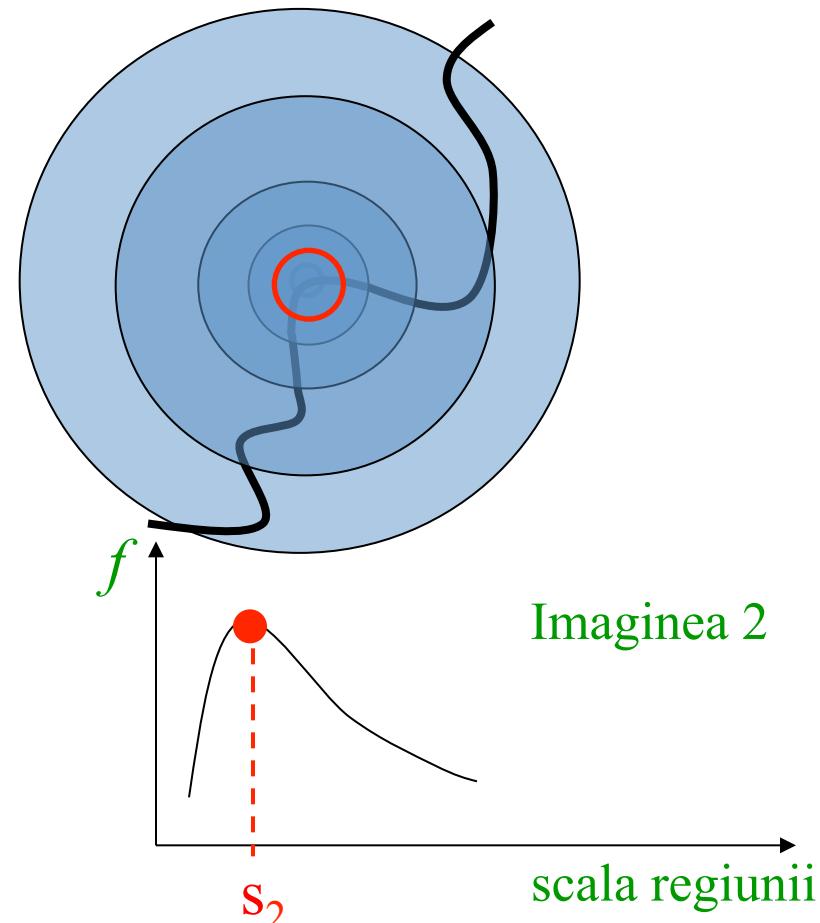
Selectia automată a scalei

Intuiție:

- găsește scala pentru care obținem un maxim local pentru o funcție care măsoară cât de bună e trăsătura găsită la un punct de interes la o anumită scală
- scala poate fi raza unei cerc centrat în punctul de interes sau latura unui pătrat centrat în punctul de interes. În ambele imagini o să găsesc aceeași regiune.



Imaginea 1



Imaginea 2

Detectare de caracteristici locale

- puncte interesante detectate în două imagini



- vreau să găsesc perechile de puncte corespondente
- am nevoie de o descriere a unei regiuni din jurul punctului
- ideal, definesc o regiune invariantă la scală: pentru două puncte corespunzătoare din două imagini de zoom diferit le corespund aceleasi regiuni

Abordarea exhaustivă

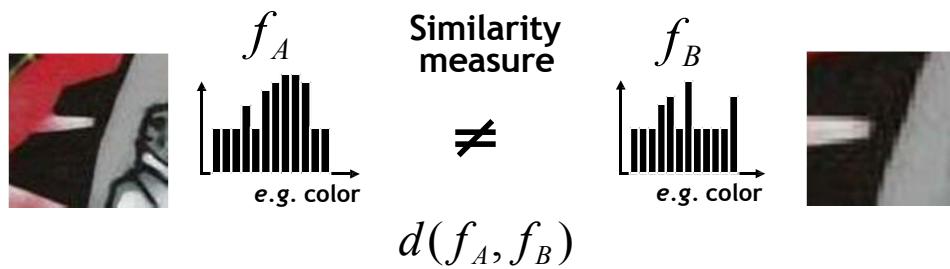
- algoritm multi-scale naiv de a compara două regiuni



- variez dimensiunea unei regiuni (cealaltă rămâne fixă), extrag din ambele regiuni descriptorul corespunzător (spre exemplu o histogramă de culori) și compar descriptorii

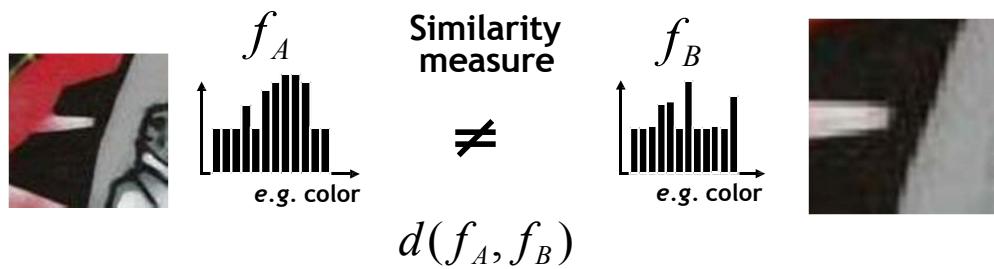
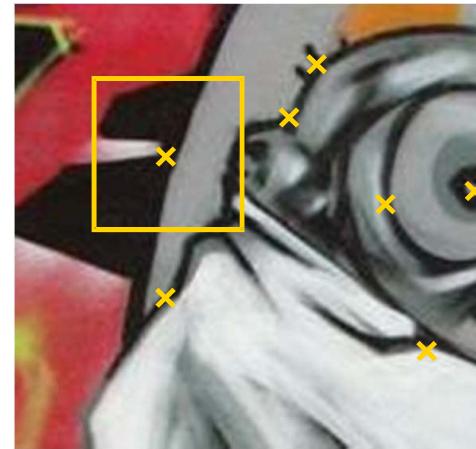
Abordarea exhaustivă

- algoritm multi-scale naiv de a compara două regiuni
- compar descriptori pentru regiuni scalate



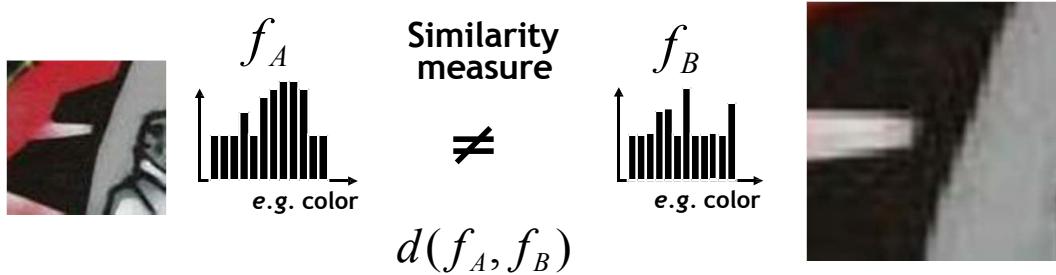
Abordarea exhaustivă

- algoritm multi-scale naiv de a compara două regiuni
- compar descriptori pentru regiuni scalate



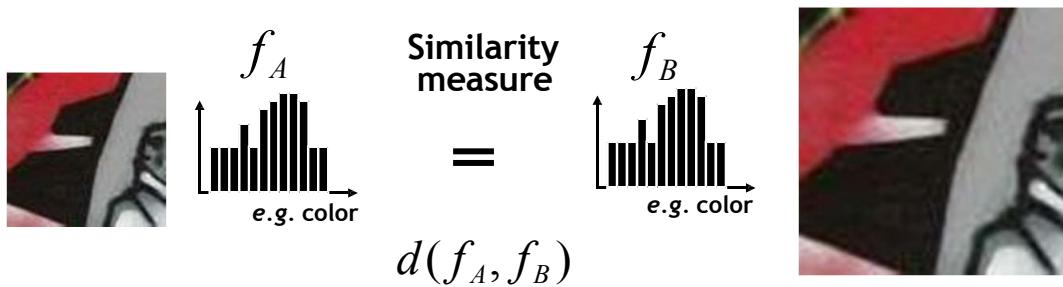
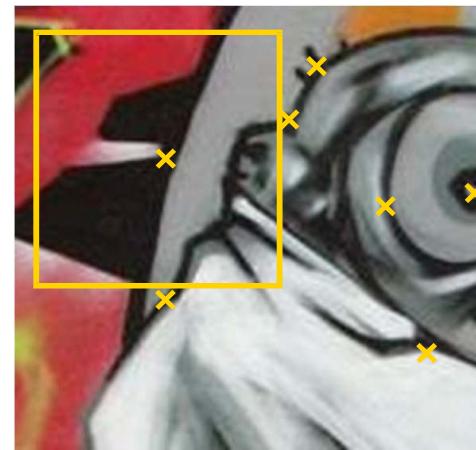
Abordarea exhaustivă

- algoritm multi-scale naiv de a compara două regiuni
- compar descriptori pentru regiuni scalate



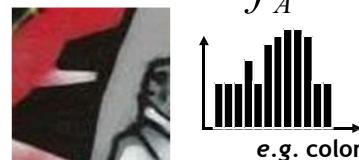
Abordarea exhaustivă

- algoritm multi-scale naiv de a compara două regiuni
- compar descriptori pentru regiuni scalate



Abordarea exhaustivă

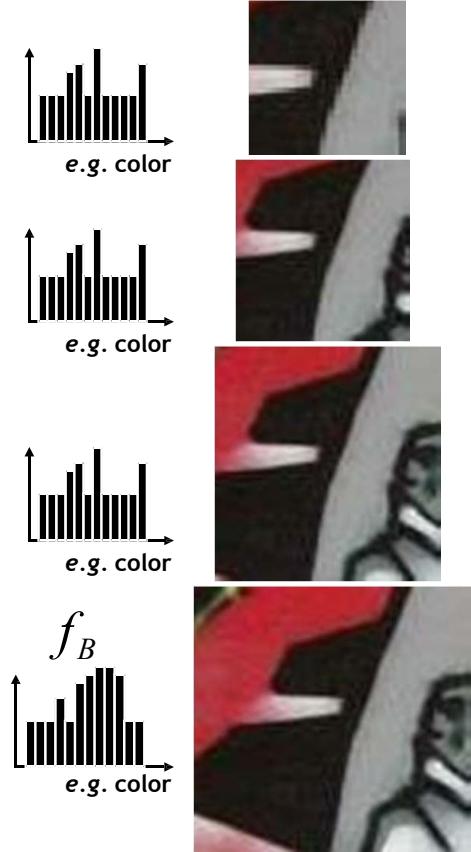
- algoritm multi-scale naiv de a compara două regiuni
- compar descriptori pentru regiuni scalate
- computațional ineficient
- pentru baze de date foarte mari nu se poate aplica pentru regăsire de obiecte



Similarity measure

=

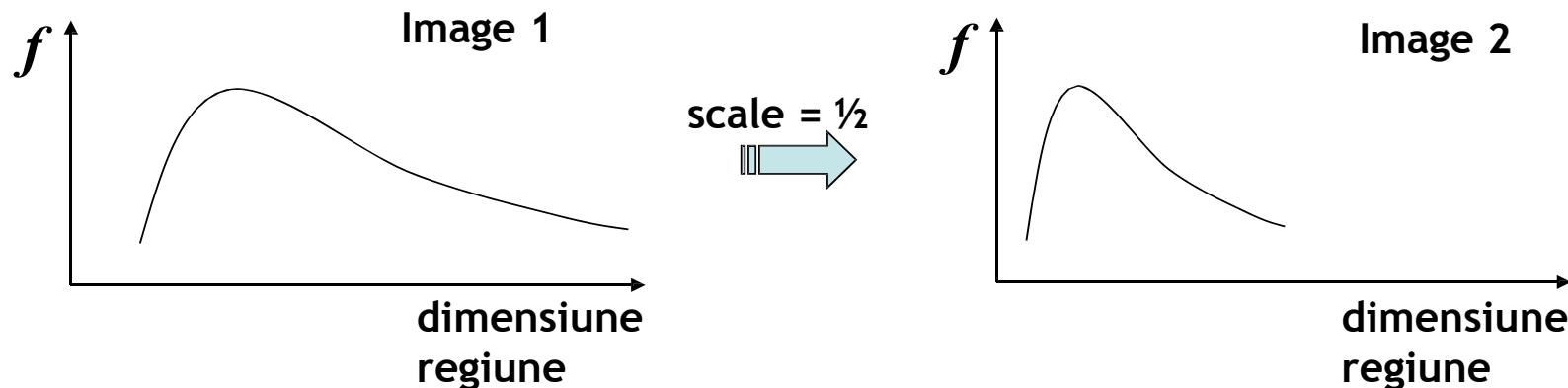
$$d(f_A, f_B)$$



Selectia automată a scalei

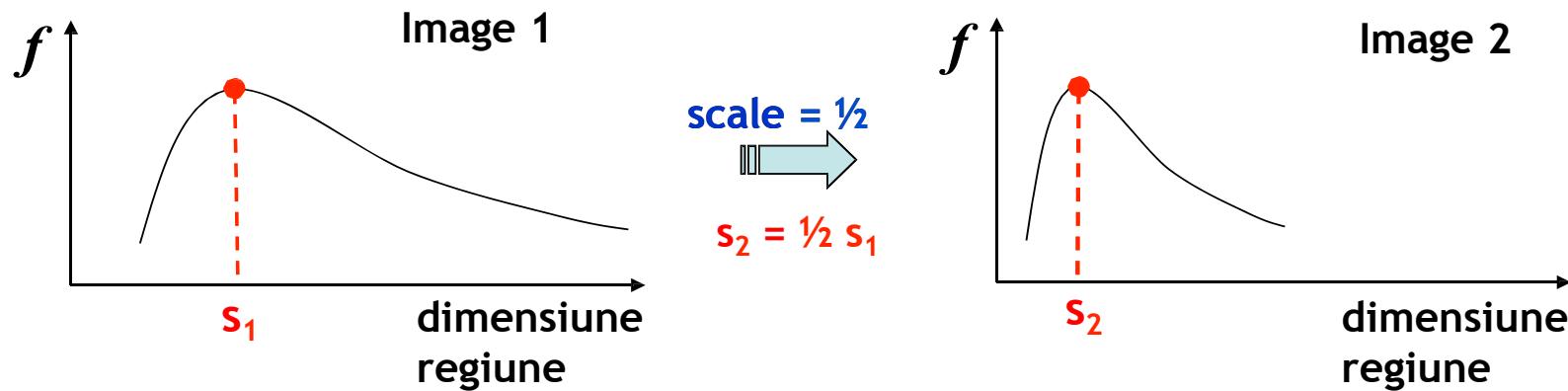
- soluție:
 - găsește o funcție pentru regiuni care este invariantă la scală
(arată la fel pentru regiuni corespondente, chiar dacă sunt la scale/mărimi diferite)

Exemplu: intensitatea medie. Pentru regiuni corespondente (chiar dacă sunt de mărimi diferite) funcția arată la fel



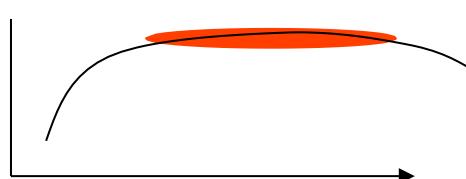
Selectia automată a scalei

- soluție:
 - găsește maximul local al acestei funcții (vrem funcții cu un singur maxim local) –dimensiunea regiunii pentru care răspunsul e maxim
 - dimensiunea regiunii covariază cu zoom-ul
 - dimensiunea regiunii este găsită în fiecare imagine independent

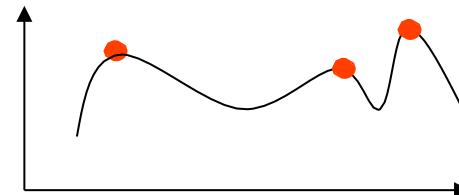


Selectia automată a scalei

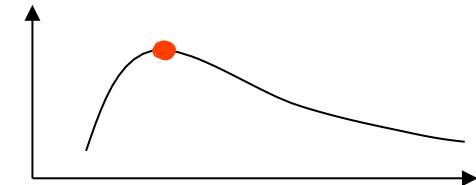
- soluție:
 - găsește o funcție pentru regiuni care este invariantă la scală (arată la fel pentru regiuni corespondente, chiar dacă sunt la scale/mărimi diferite). O funcție “bună” are un singur maxim!



dimensiune
regiune



dimensiune
regiune



dimensiune
regiune



Selectia automată a scalei



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

x, x' – coordonatele punctelor de interes în cele două imagini

σ, σ' – dimensiunile regiunilor = laturile ferestrelor centrate în x, x'

Cum să găsim scalele cele mai bune σ, σ' ?

Selectia automată a scalei



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

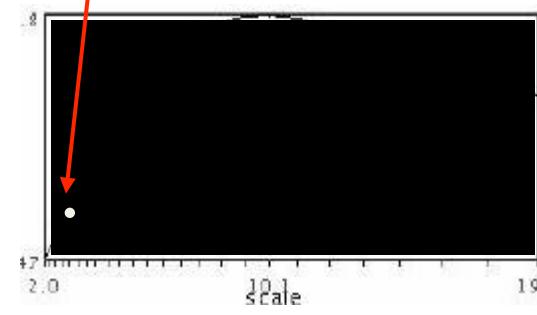
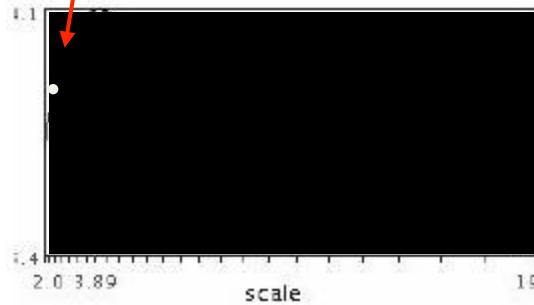
x, x' – coordonatele punctelor de interes în cele două imagini

σ, σ' – dimensiunile regiunilor = laturile ferestrelor centrate în x, x'

Cum să găsim scalele cele mai bune σ, σ' ?

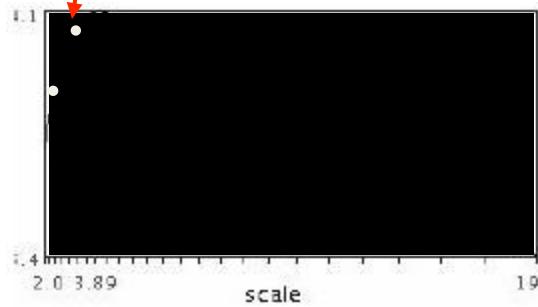
Selectia automată a scalei

- valoarea funcției f pentru fiecare scală

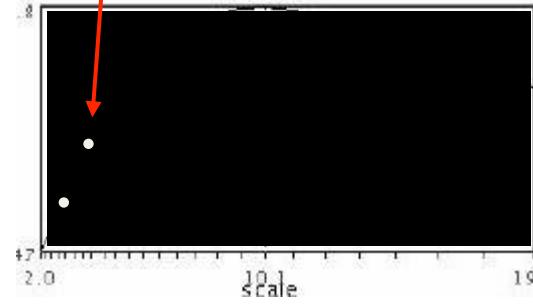


Selectia automată a scalei

- valoarea funcției f pentru fiecare scală



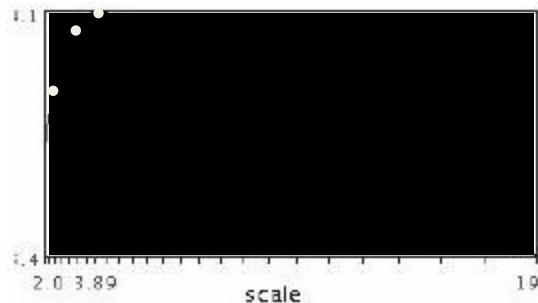
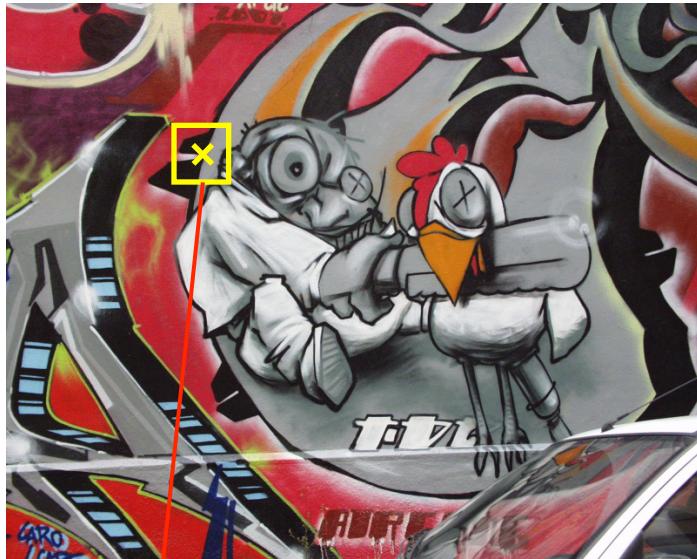
$$f(I_{i_1 \dots i_m}(x, \sigma))$$



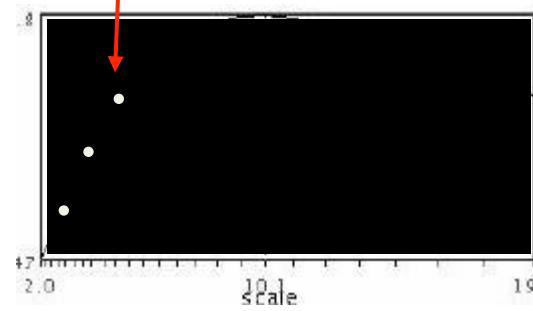
$$f(I_{i_1 \dots i_m}(x', \sigma))$$

Selectia automată a scalei

- valoarea funcției f pentru fiecare scală



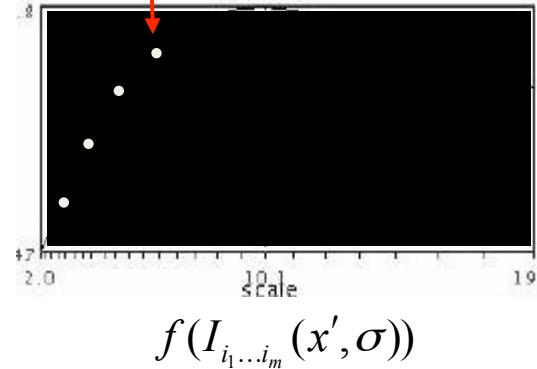
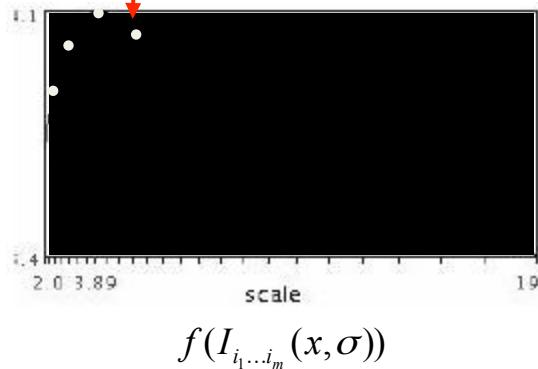
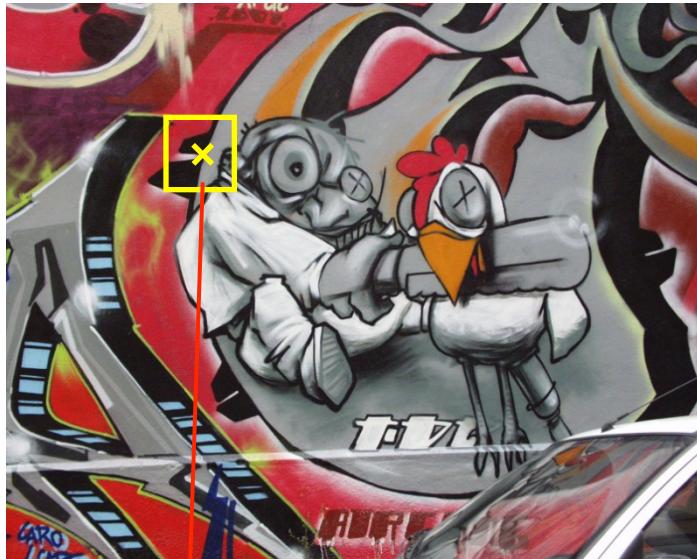
$$f(I_{i_1 \dots i_m}(x, \sigma))$$



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

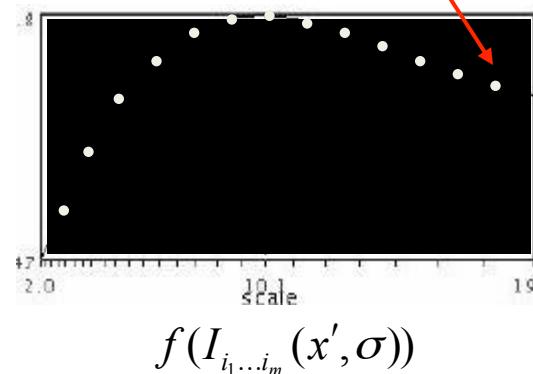
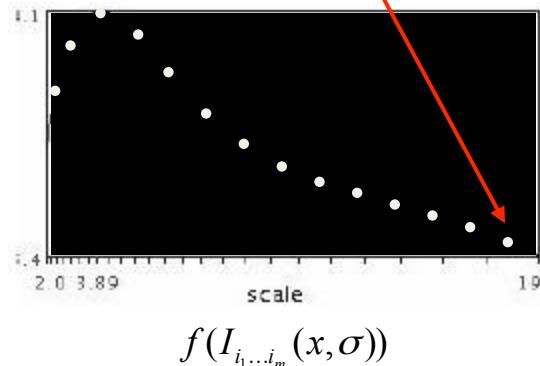
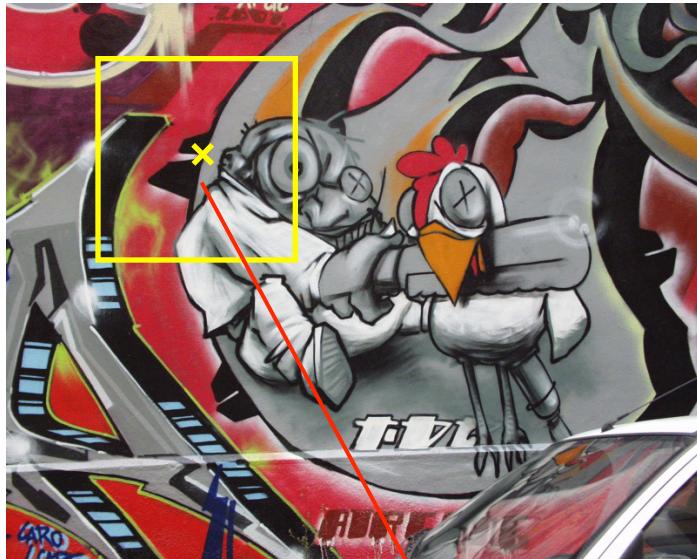
Selectia automată a scalei

- valoarea funcției f pentru fiecare scală



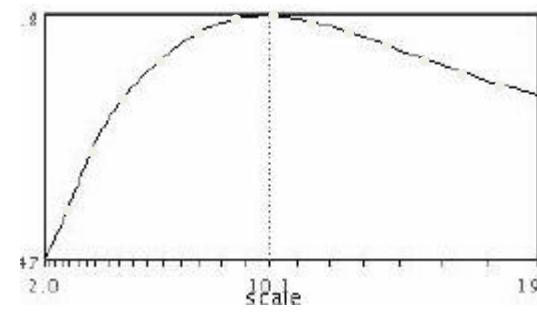
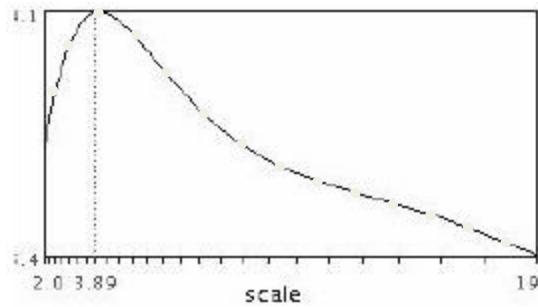
Selectia automată a scalei

- valoarea funcției f pentru fiecare scală



Selectia automată a scalei

- valoarea funcției f pentru fiecare scală



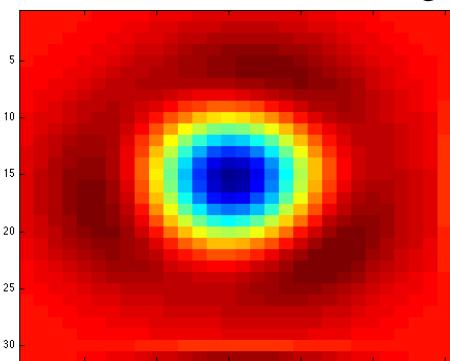
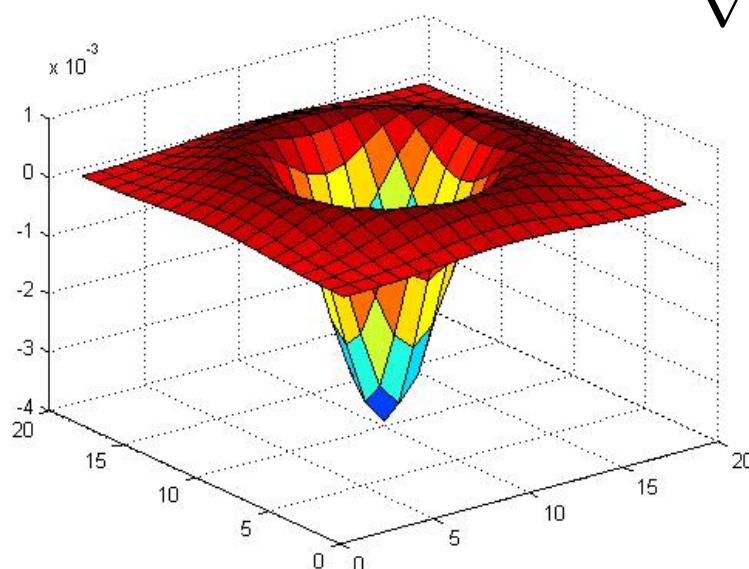
Cum găsim o astfel de funcție f cu proprietățile dorite?

Slide adaptat după K. Graumann

Selectia automată a scalei

Folosim $f = \text{filtrul Laplacian}$

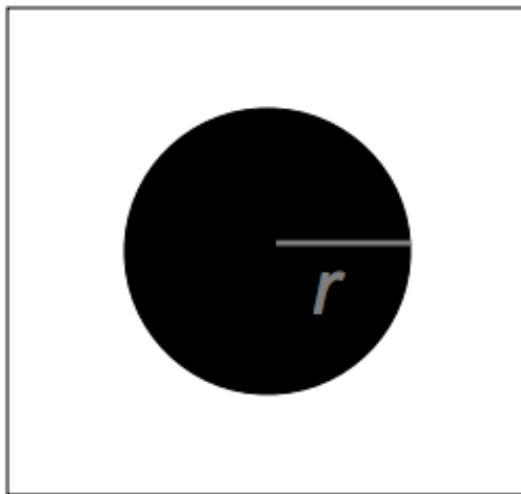
$$\nabla^2 h = \frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = L_{xx}(\sigma) + L_{yy}(\sigma)$$



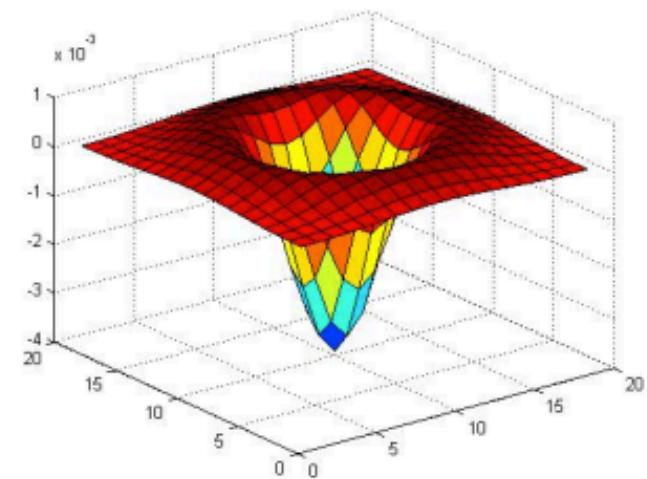
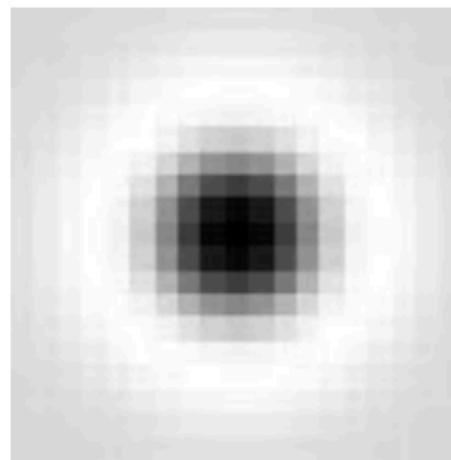
Magnitudinea răspunsului imaginii filtrate cu filtrul Laplacian atinge valoarea maximă pentru centrul regiunii când scala filtrului Laplacian corespunde cu scala optimă a regiunii de detectat = regiune circulară (blob)

Selectia automată a scalei

Pentru ce scală a filtrului Laplacian obținem un răspuns maxim pentru un cerc de rază r ?



image

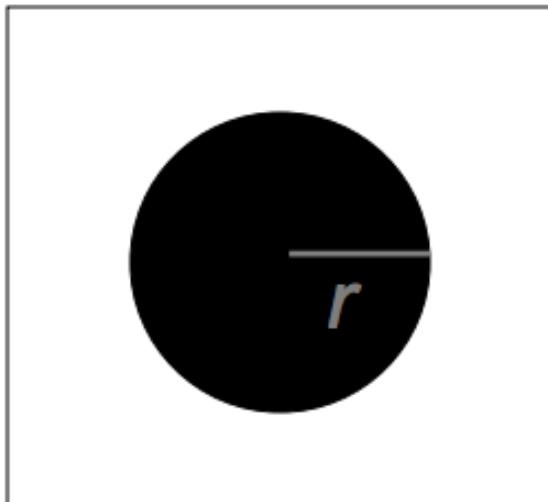


Laplacian

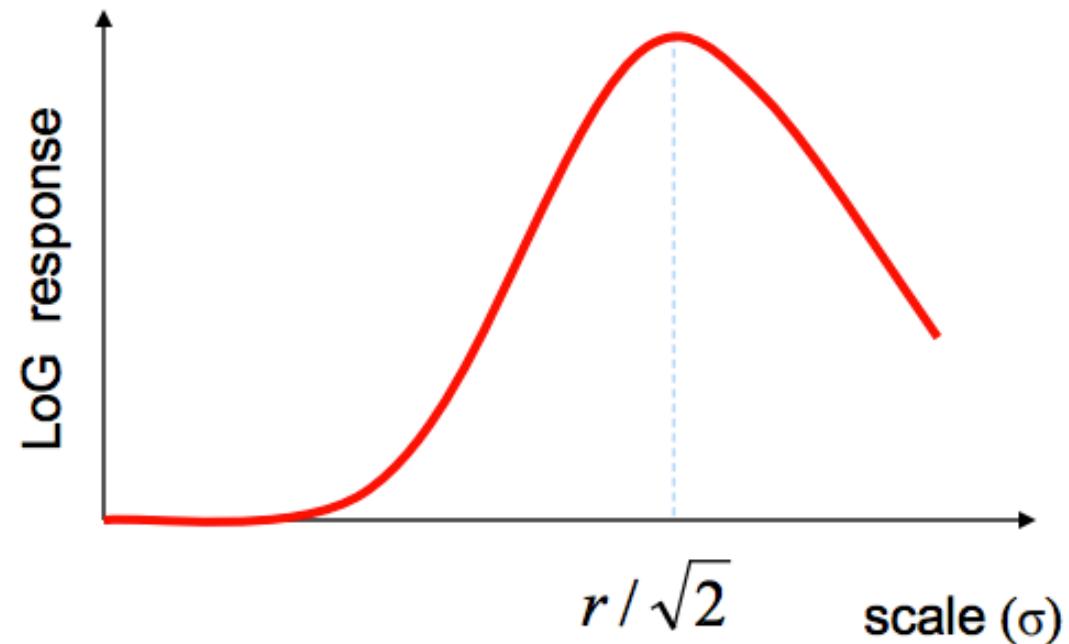
Selectia automată a scalei

Pentru ce scală a filtrului Laplacian obținem un răspuns maxim pentru un cerc de rază r ?

Scală caracteristică este $\sigma = r / \sqrt{2}$

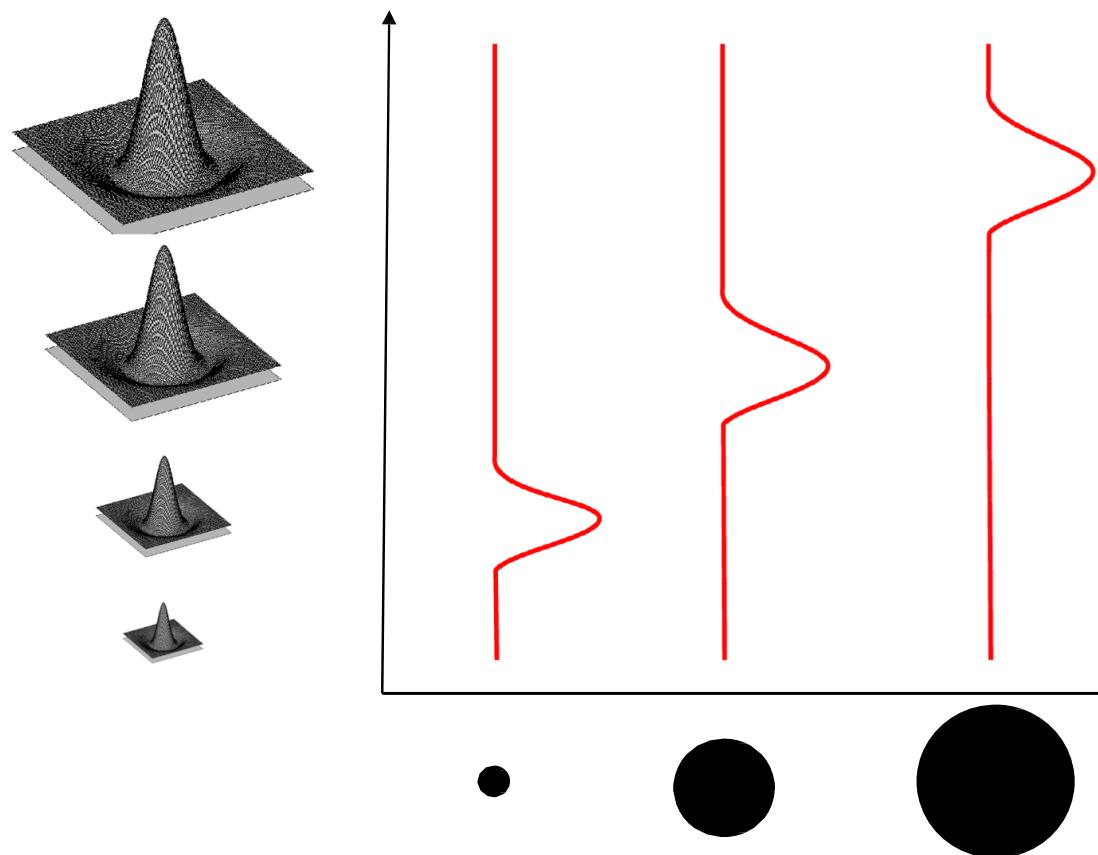


image



Selectia automată a scalei

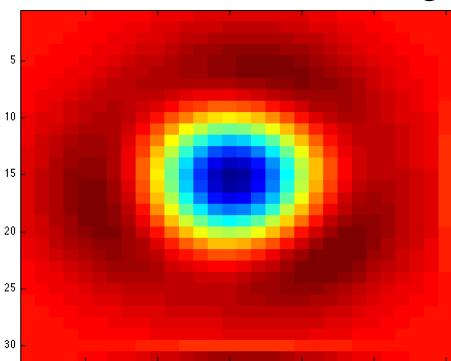
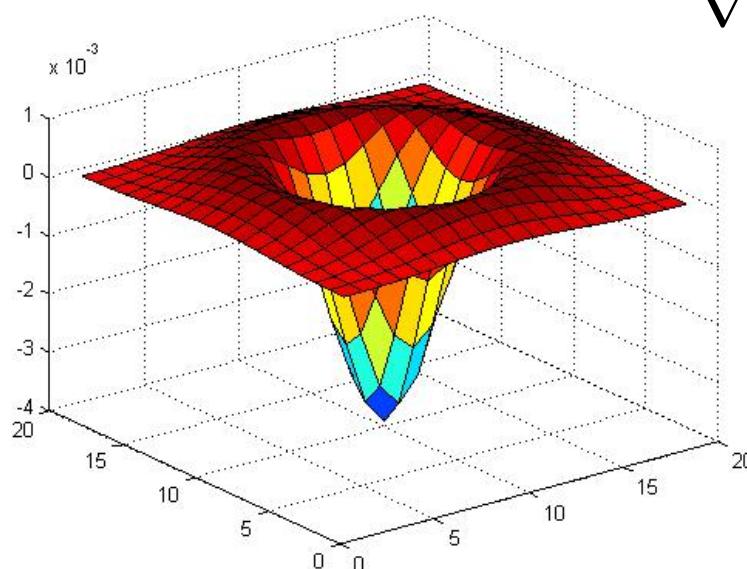
Folosim $f =$ filtrul Laplacian = detector de “blob-uri”



Selectia automată a scalei

Folosim $f = \text{filtrul Laplacian}$

$$\nabla^2 h = \frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = L_{xx}(\sigma) + L_{yy}(\sigma)$$



Magnitudinea răspunsului imaginii filtrate cu filtrul Laplacian atinge valoarea maximă pentru centrul regiunii când scala filtrului Laplacian corespunde cu scala optimă a regiunii de detectat = regiune circulară (blob)

Cursul 5: Derivata filtrelor Gaussiane - 2D

$$d_y \otimes (h \otimes I) = (d_y \otimes h) \otimes I$$

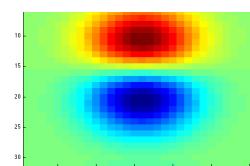
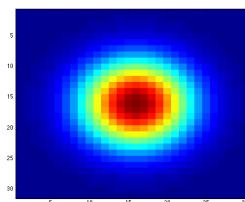
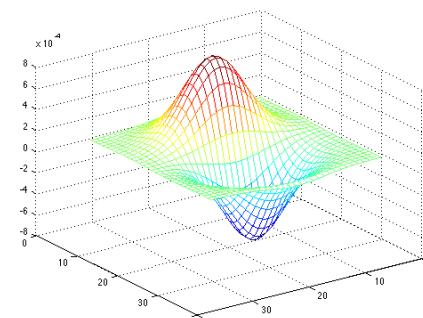
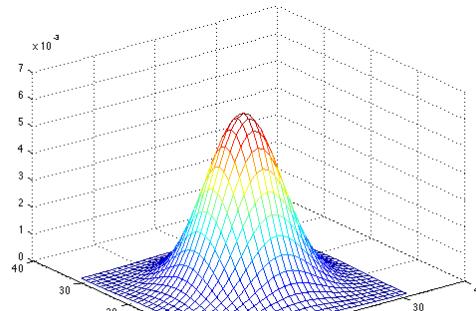
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$



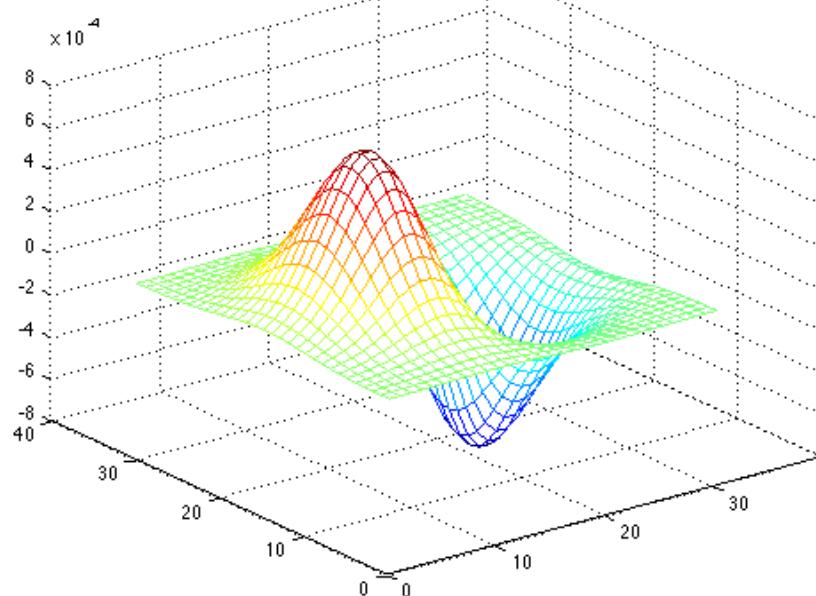
$$\begin{bmatrix} 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0219 & 0.0983 & 0.1621 & 0.0983 & 0.0219 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \end{bmatrix}$$



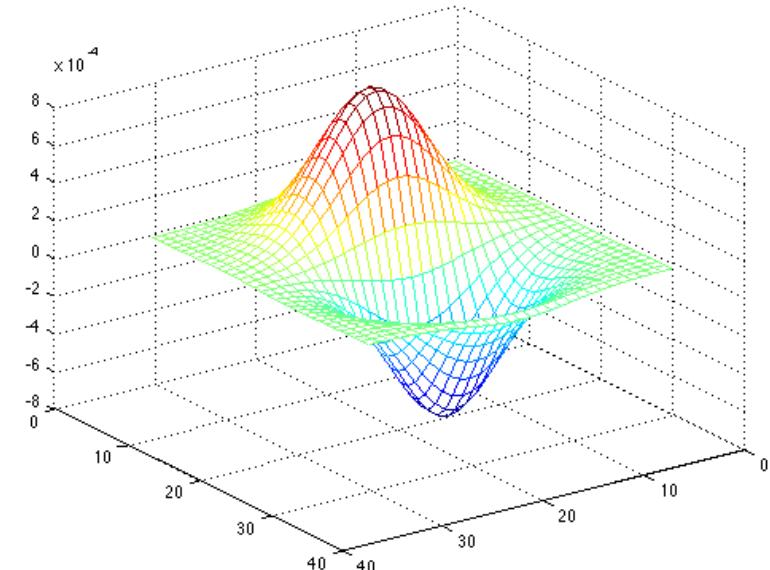
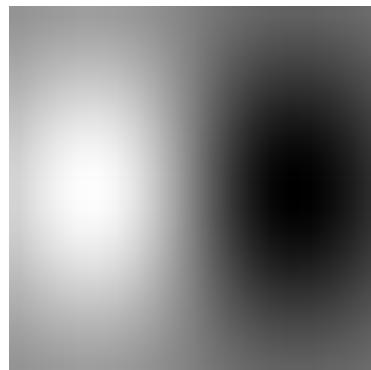
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$



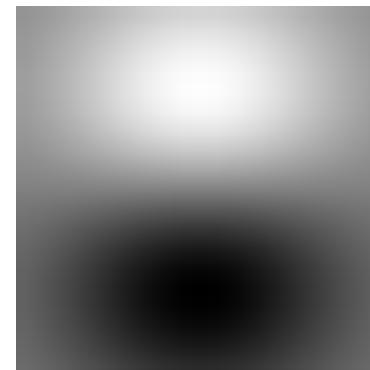
Cursul 5: Derivata filtrelor Gaussiane - 2D



direcția x

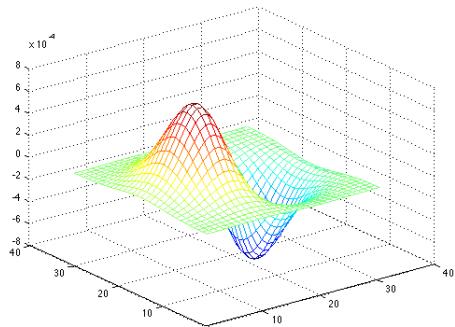


direcția y

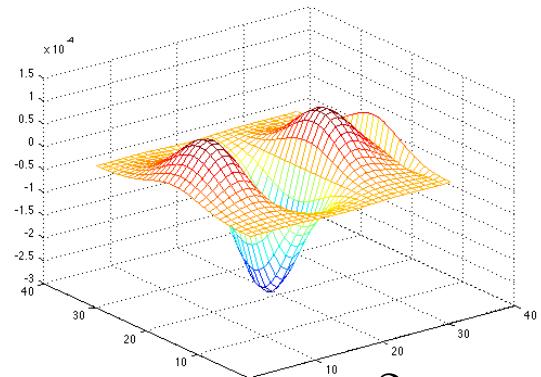


Cursul 5: Derivata a doua a filtrelor Gaussiane

0	0	0
0	-1	1
0	0	0



=

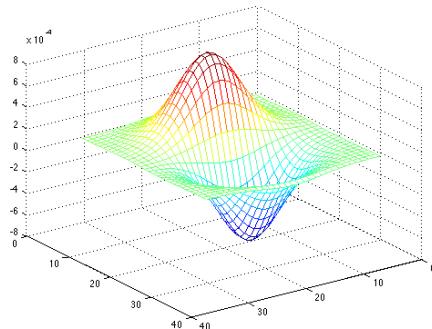


direcția x

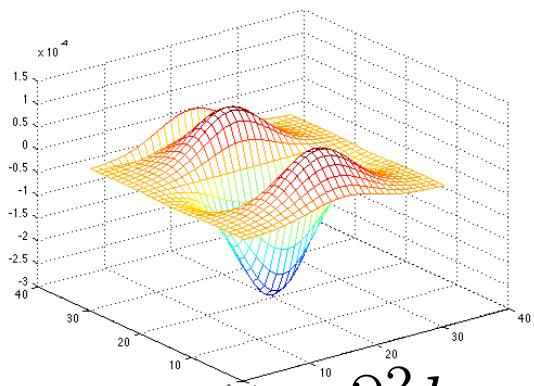
$$\frac{\partial h}{\partial x}$$

$$\frac{\partial^2 h}{\partial x^2}$$

0	0	0
0	-1	0
0	1	0



=



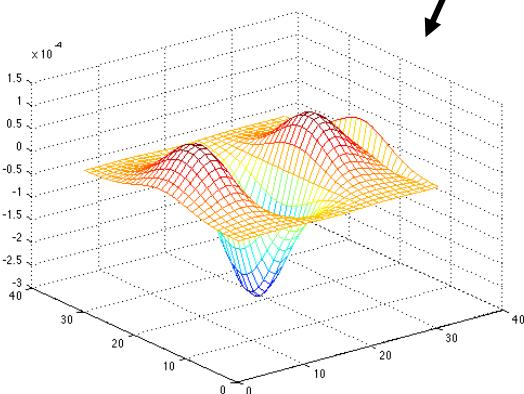
direcția y

$$\frac{\partial h}{\partial y}$$

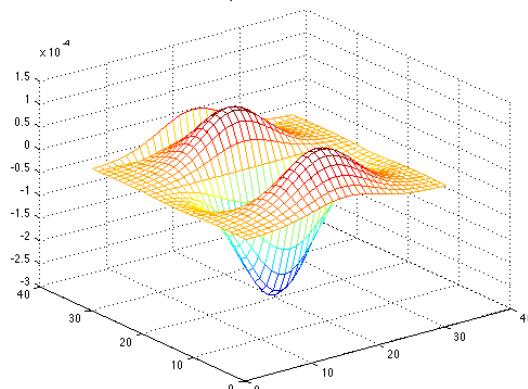
$$\frac{\partial^2 h}{\partial y^2}$$

Cursul 5: Laplacian-ul unei funcții Gaussiene – 2D

$$\nabla^2 h = \frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2}$$

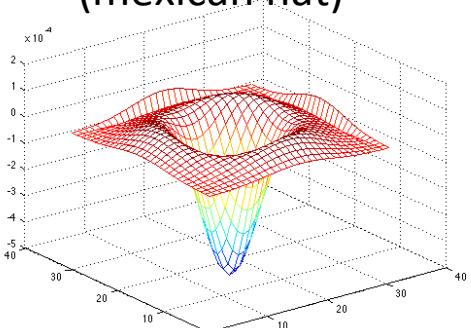


+



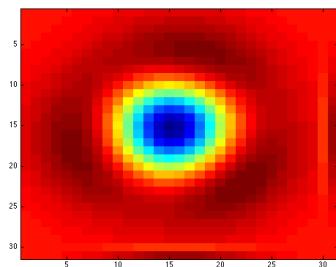
=

Forma generală a
unui filtru Laplacian
(mexican hat)



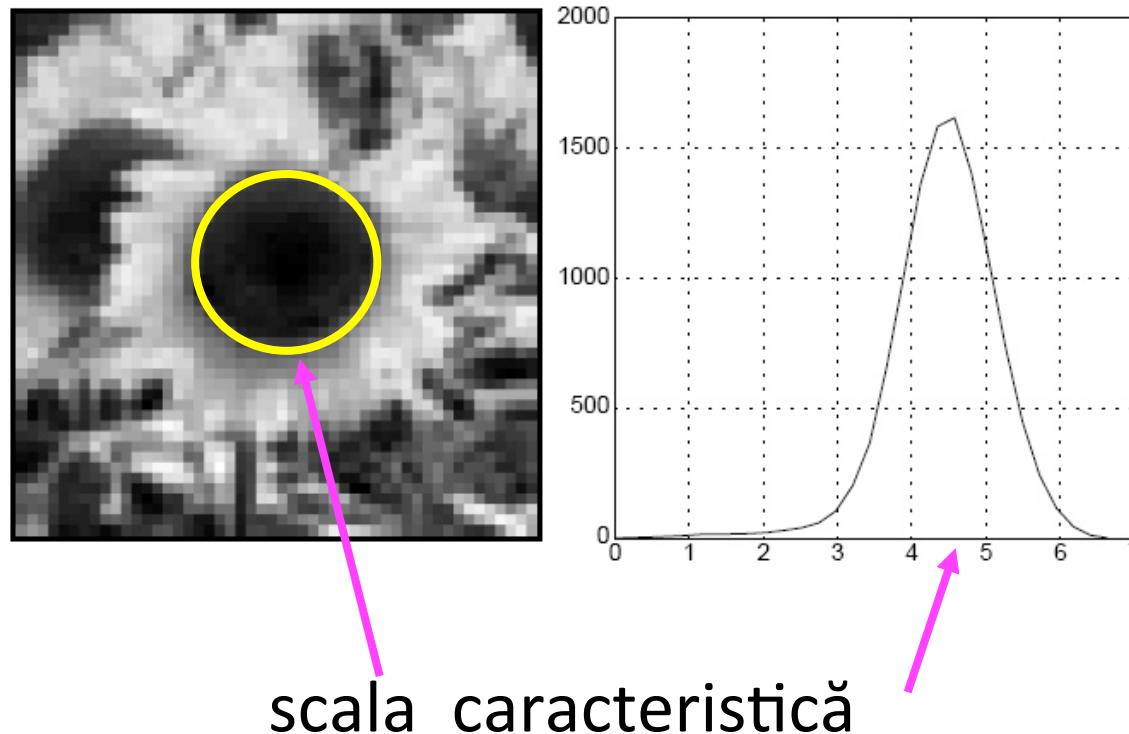
Exemplu particular
de filtru Laplacian

0	1	0
1	-4	1
0	1	0



Scala caracteristică

- definim scala caracteristică a unui punct de interes ca fiind scala pentru care se obține răspunsul maxim (în modul) în imaginea filtrată cu filtrul Laplacian parametrizat de această scală



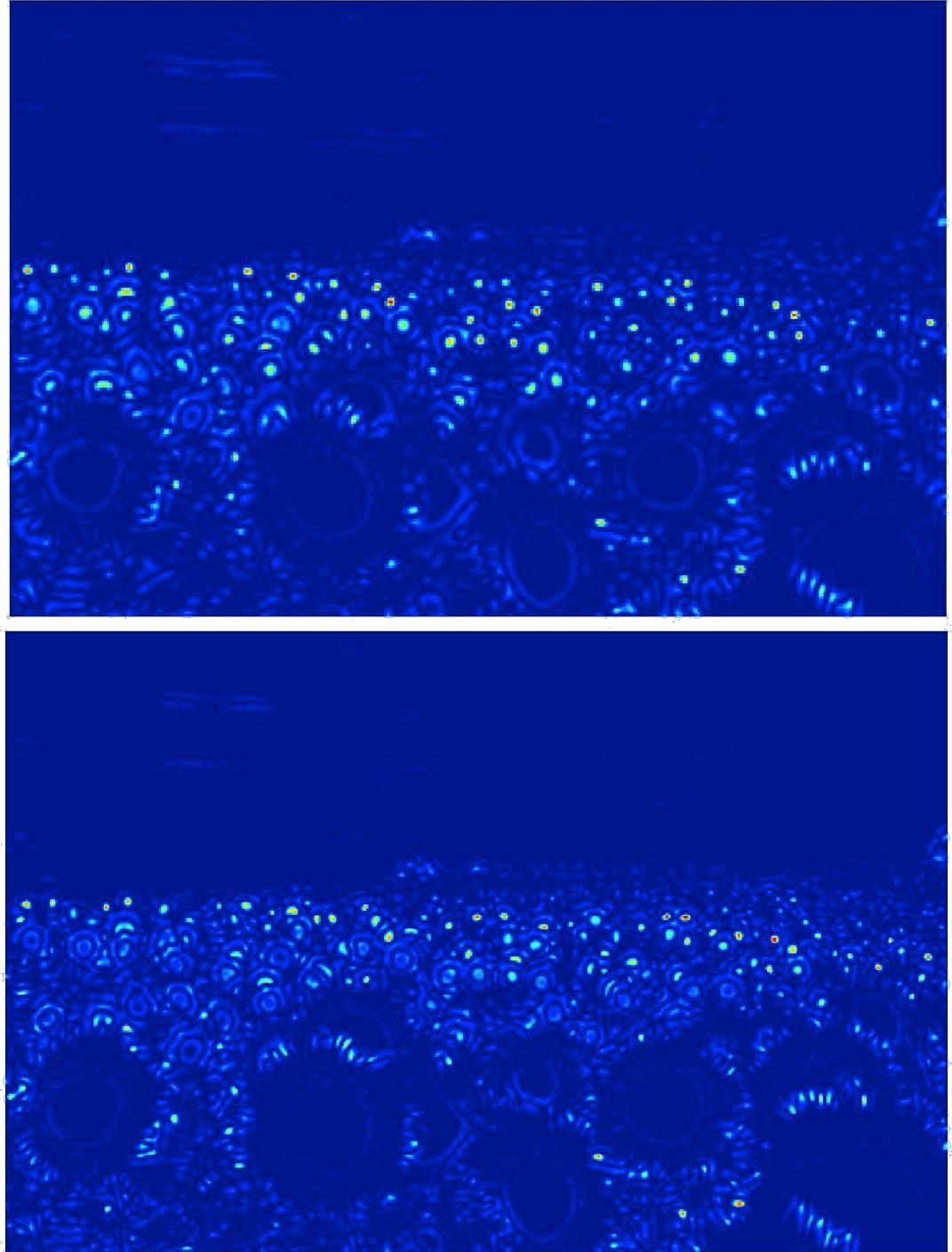
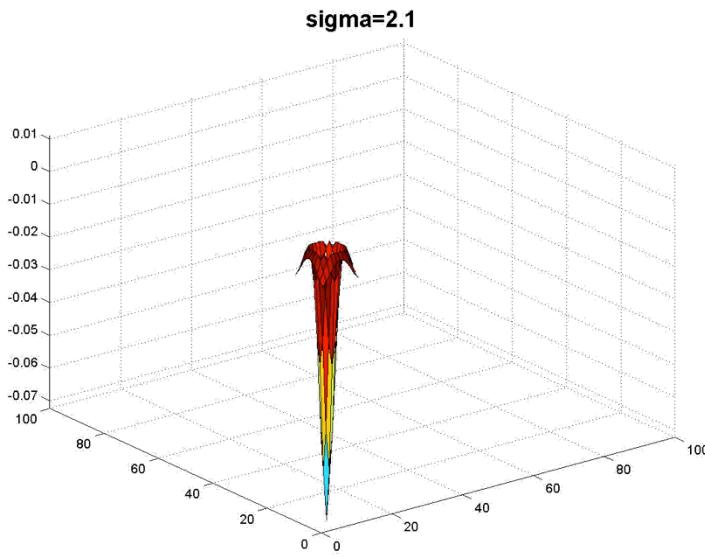
Exemplu

Imagine cu
dimensiunile $\frac{3}{4}$
din cea inițială



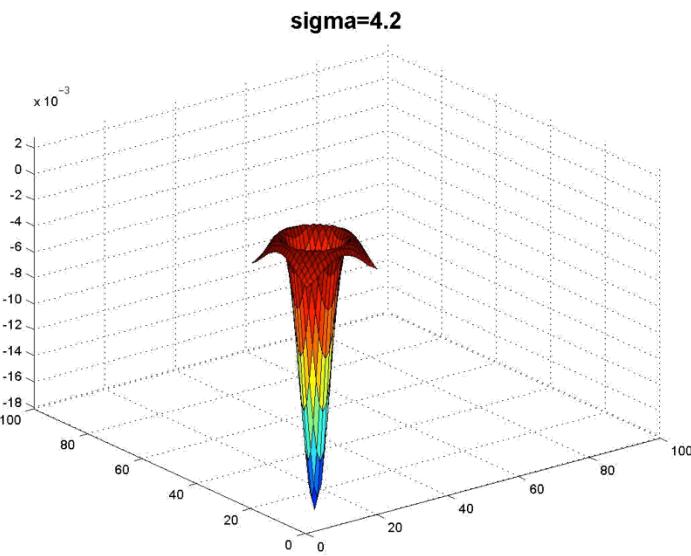
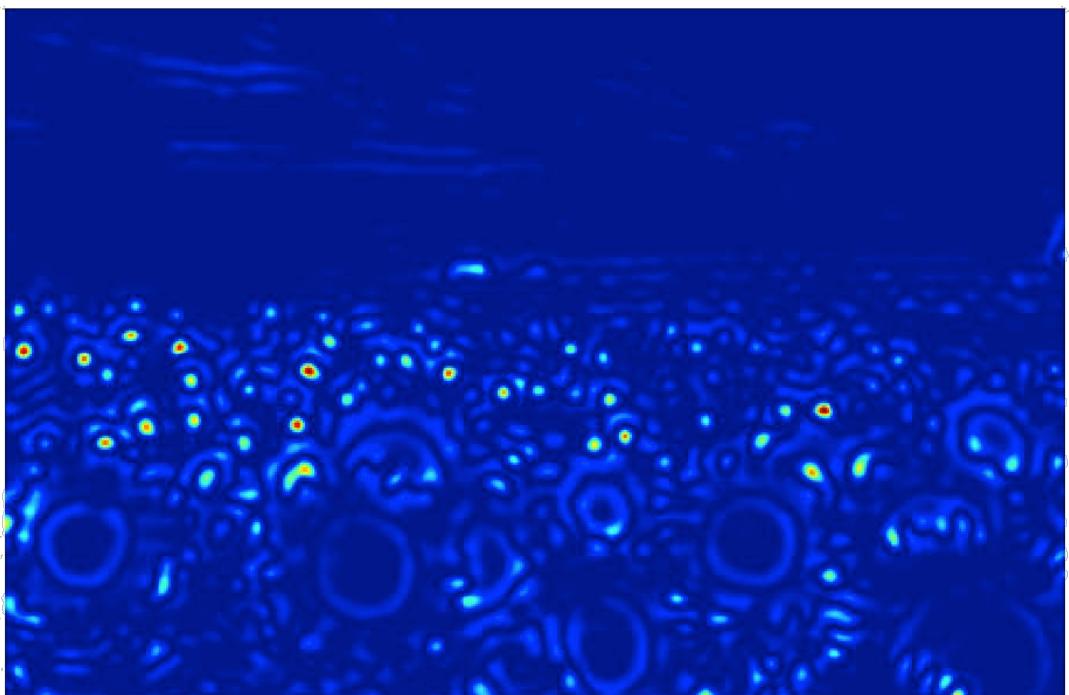
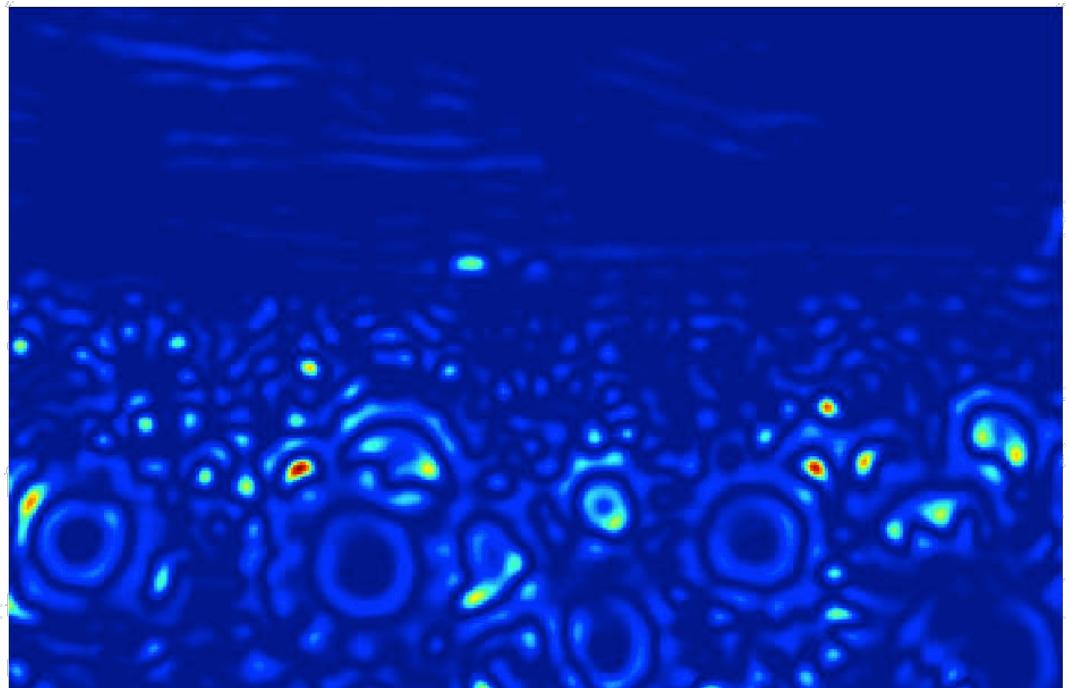
Exemplu

Imagine cu
dimensiunile $\frac{3}{4}$
din cea inițială



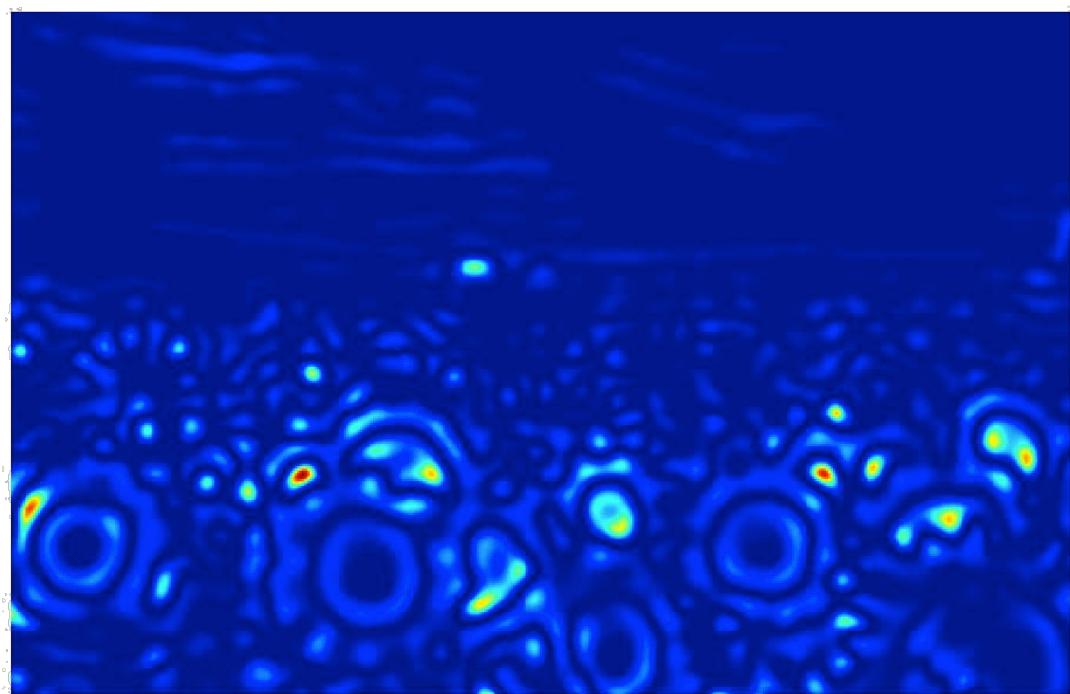
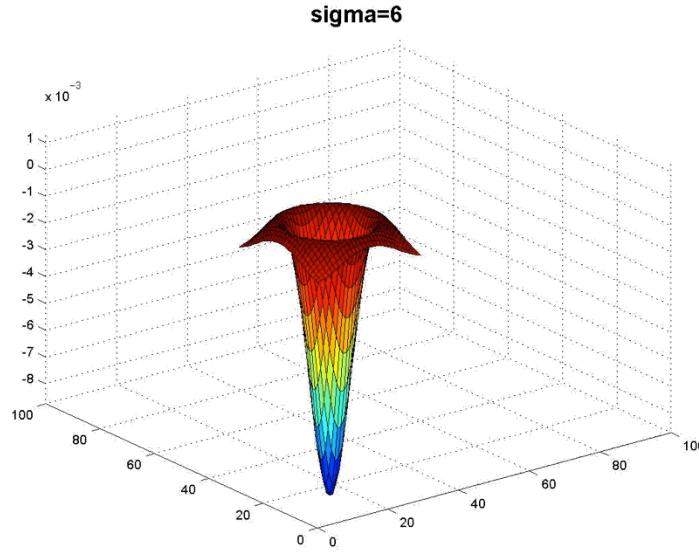
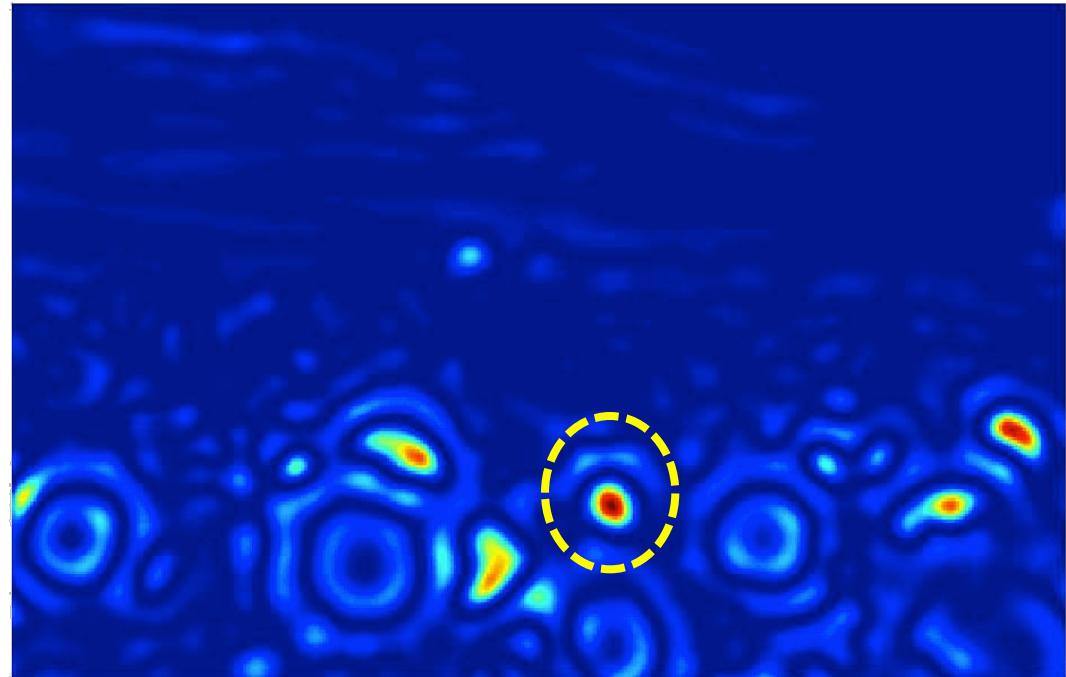
Exemplu

Imagine cu
dimensiunile $\frac{3}{4}$
din cea inițială



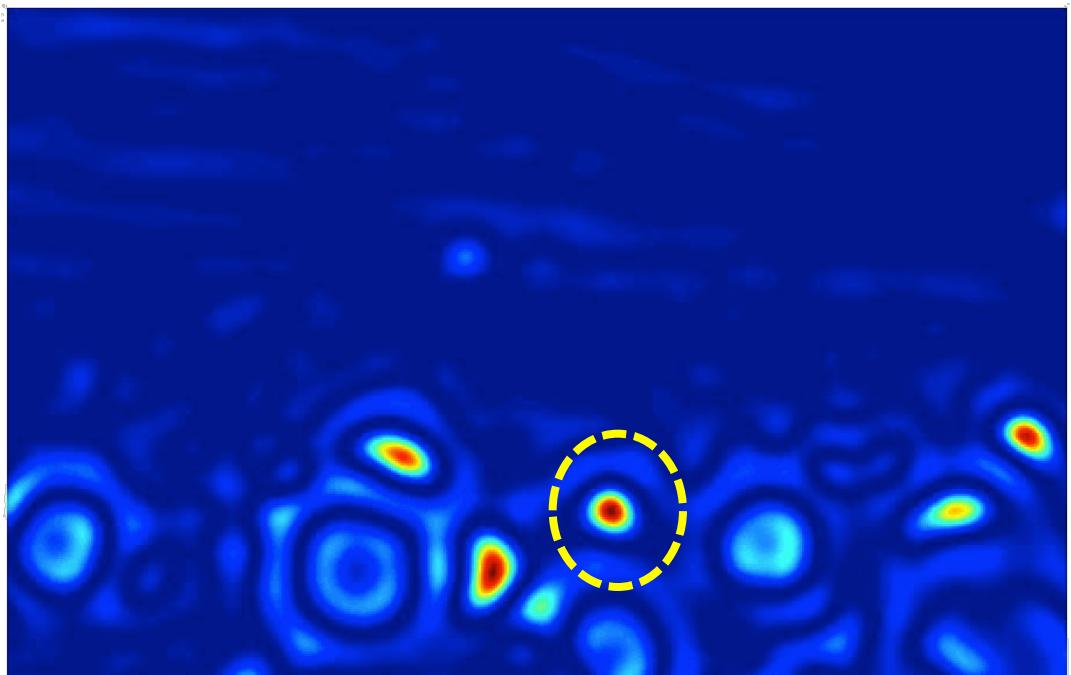
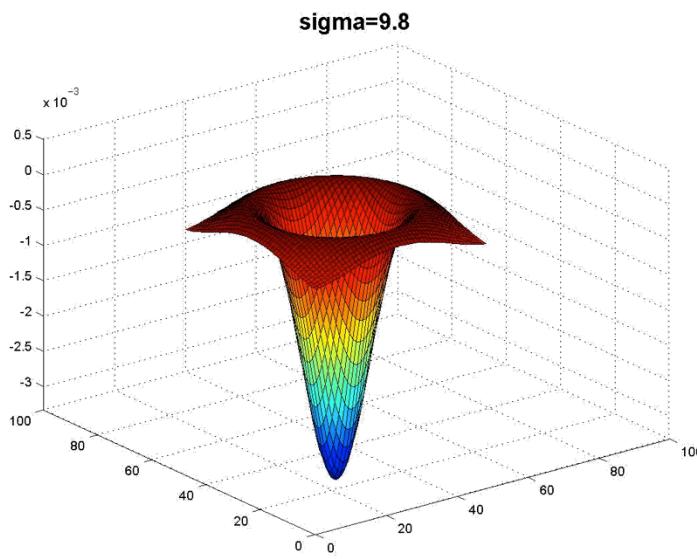
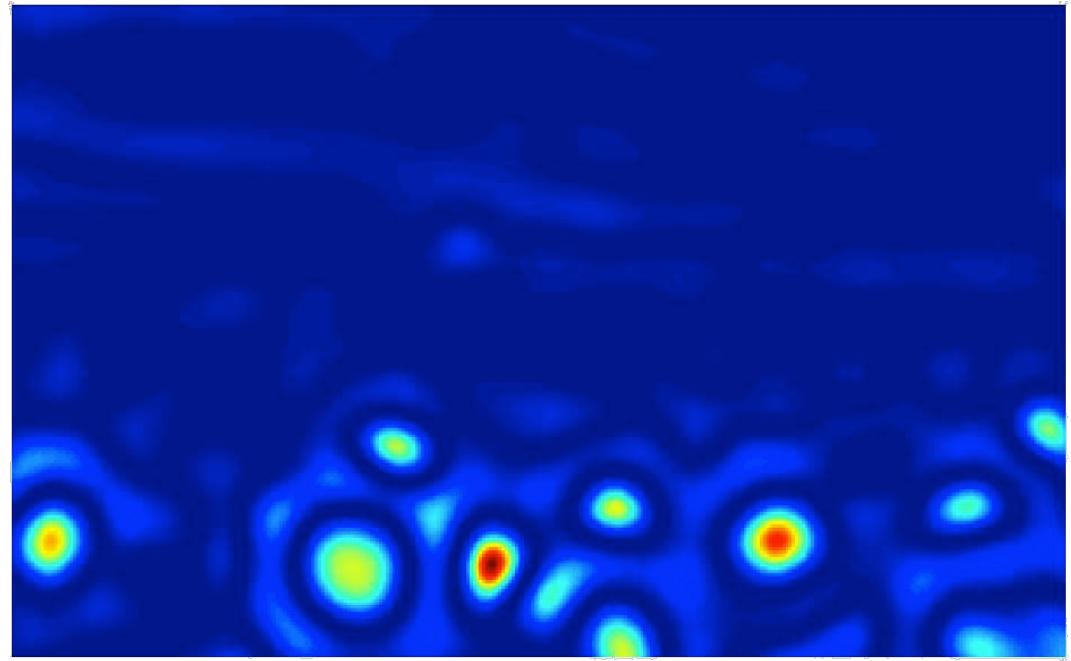
Exemplu

Imagine cu
dimensiunile $\frac{3}{4}$
din cea inițială



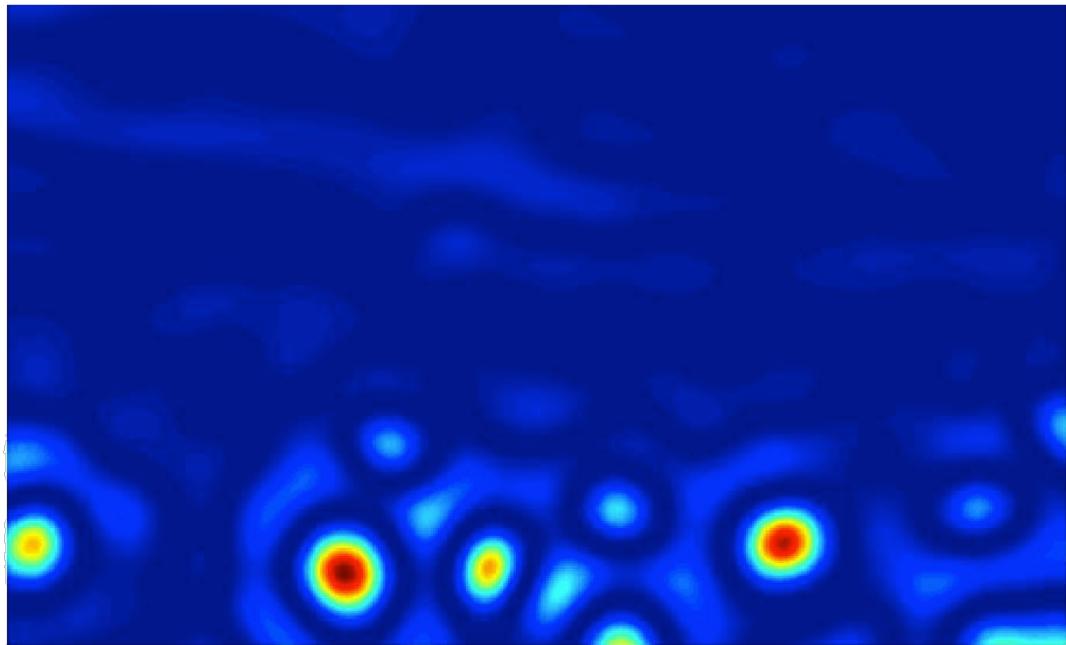
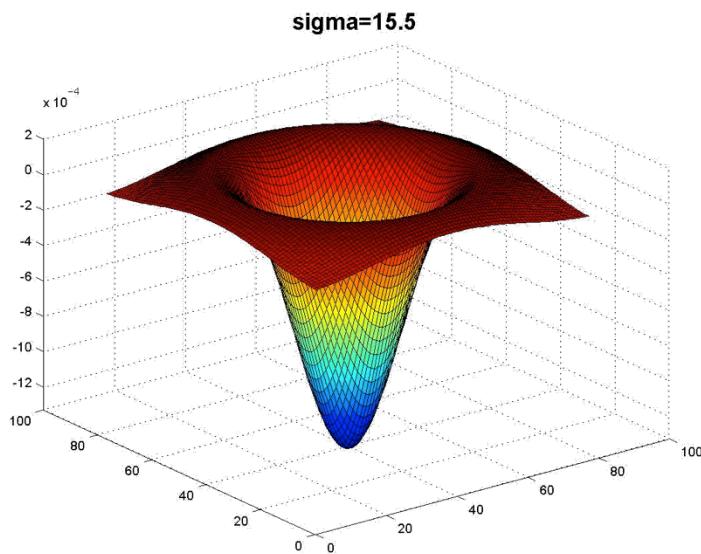
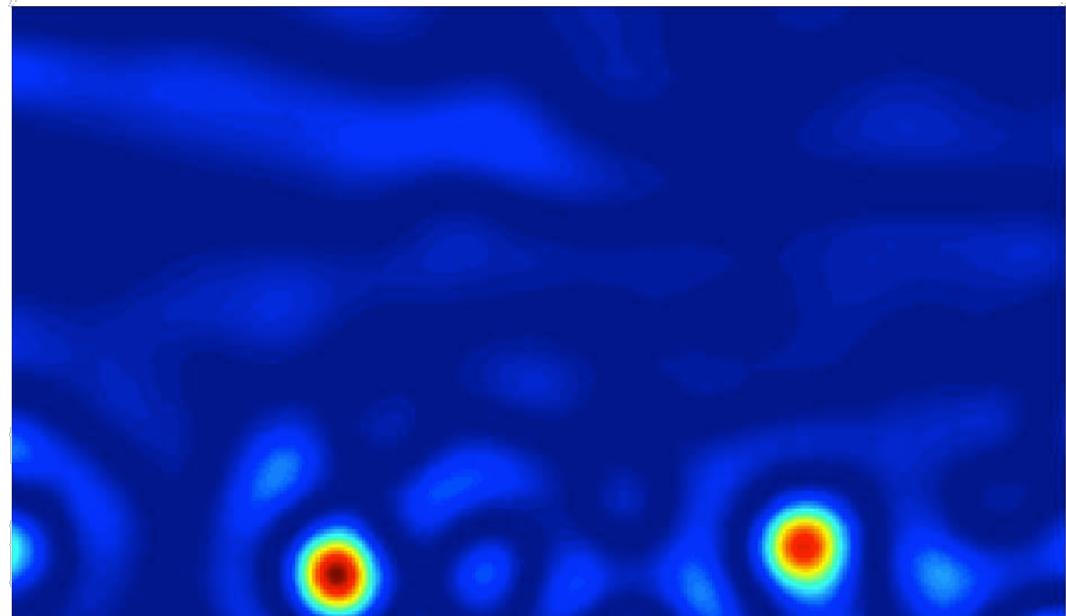
Exemplu

Imagine cu
dimensiunile $\frac{3}{4}$
din cea inițială



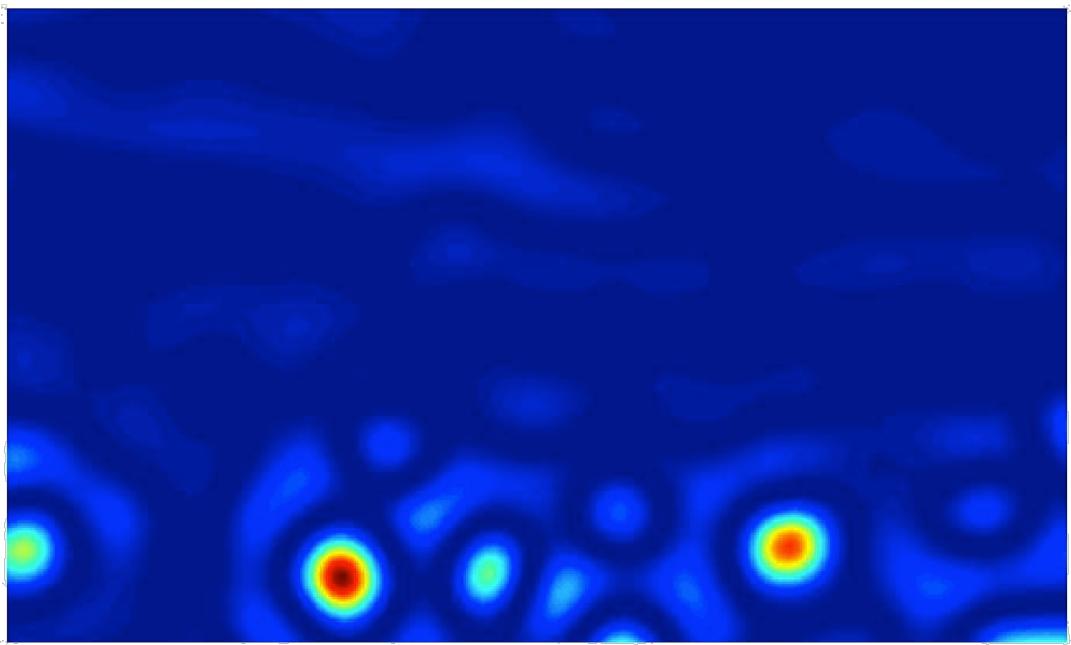
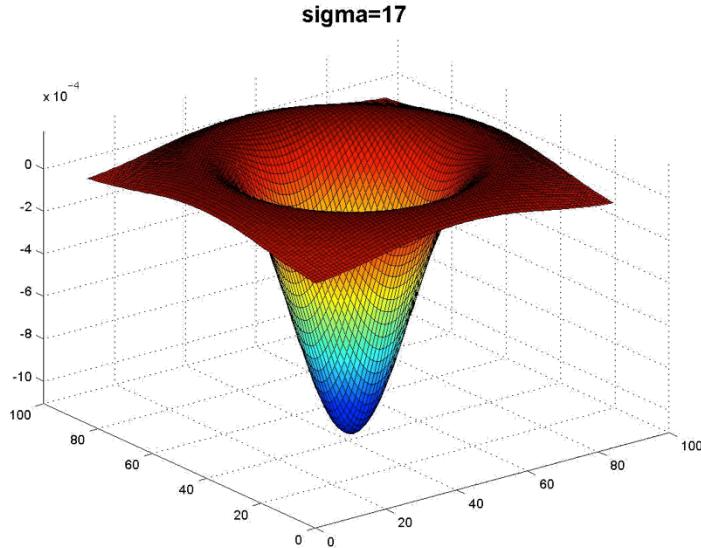
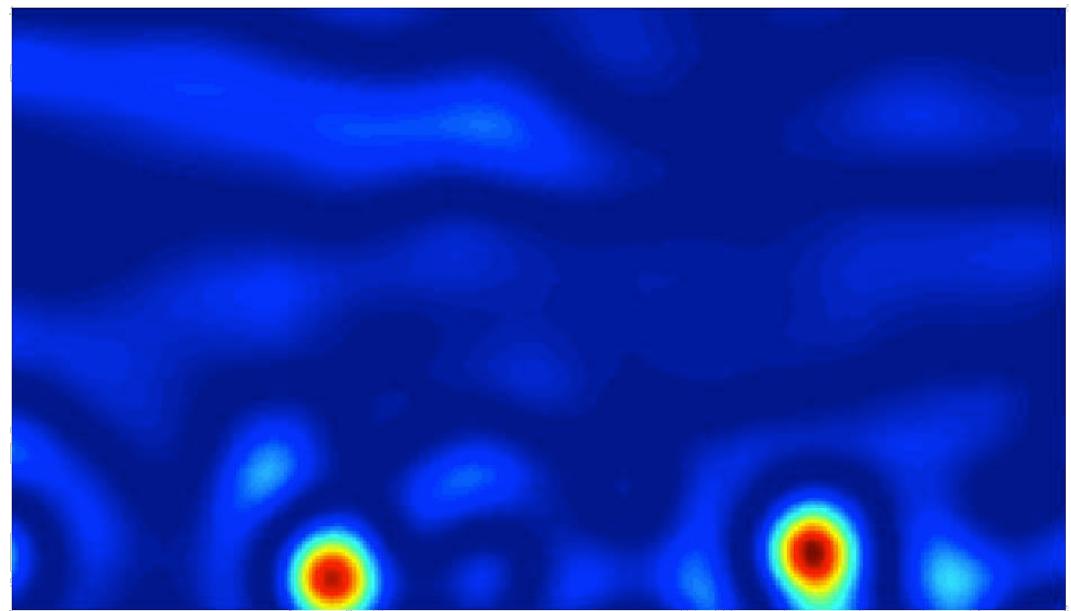
Exemplu

Imagine cu
dimensiunile $\frac{3}{4}$
din cea inițială



Exemplu

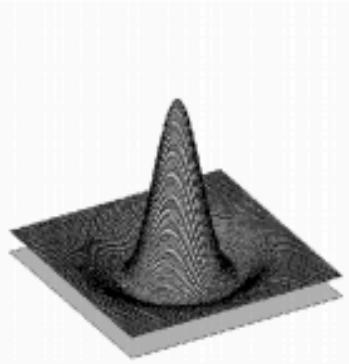
Imagine cu
dimensiunile $\frac{3}{4}$
din cea inițială



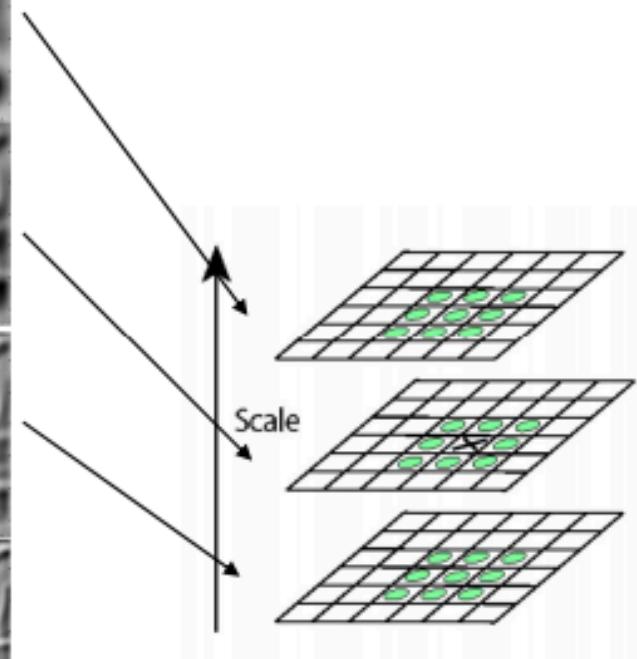
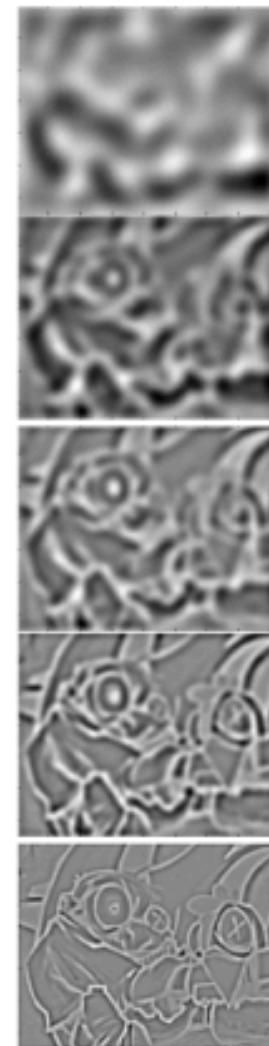
Detectorul LoG

(Laplacian of Gaussian)

Punctele de interes sunt
maxime locale în poziție
și scală



$$L_{xx}(\sigma) + L_{yy}(\sigma)$$
$$\sigma^5$$
$$\sigma^4$$
$$\sigma^3$$
$$\sigma^2$$
$$\sigma$$



⇒ Lista punctelor de
interes (x, y, s)

Detectorul LoG

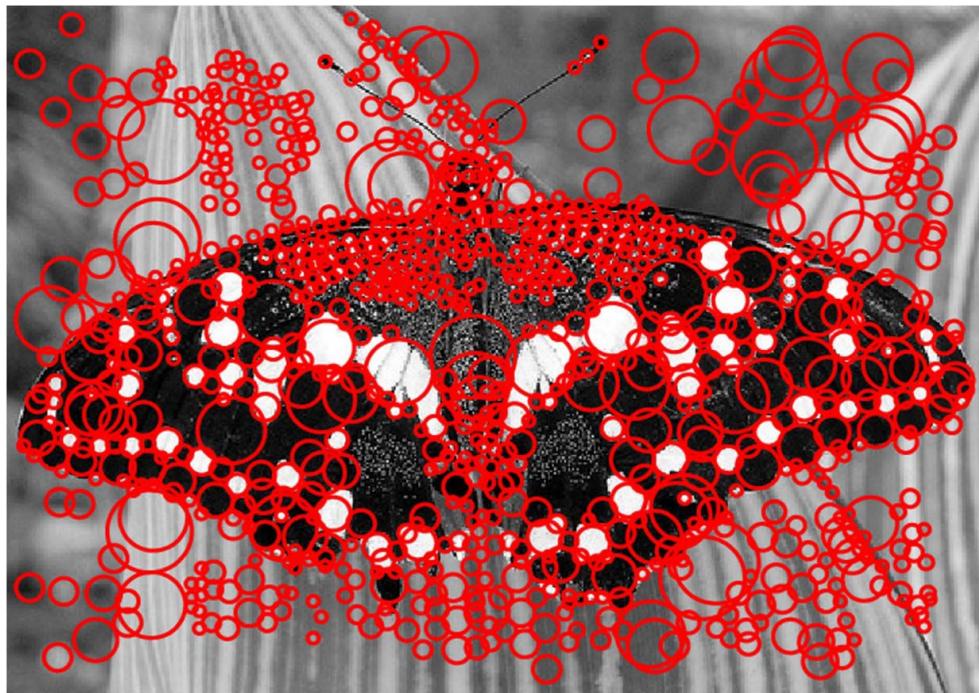


Detectorul LoG



sigma = 11.9912

Detectorul LoG (blob detector)



Diferențele a două Gaussiene (DoG)

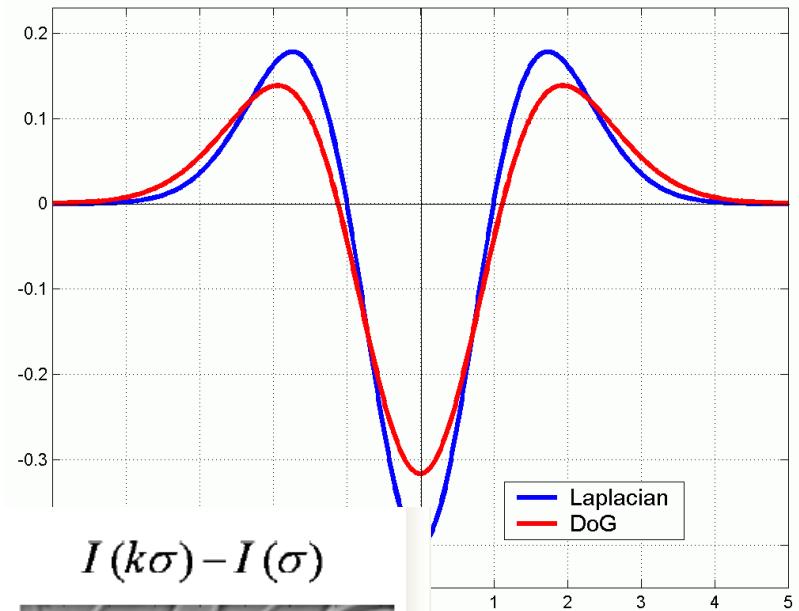
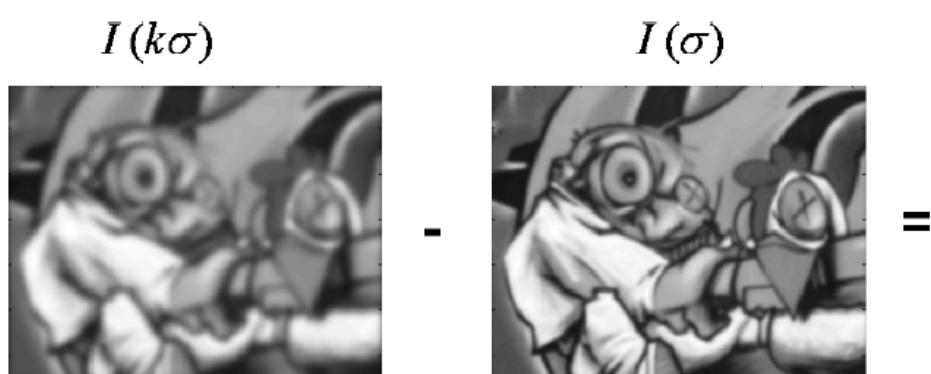
- putem aproxima Laplacianul cu diferențele a două filtre Gaussiene

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)



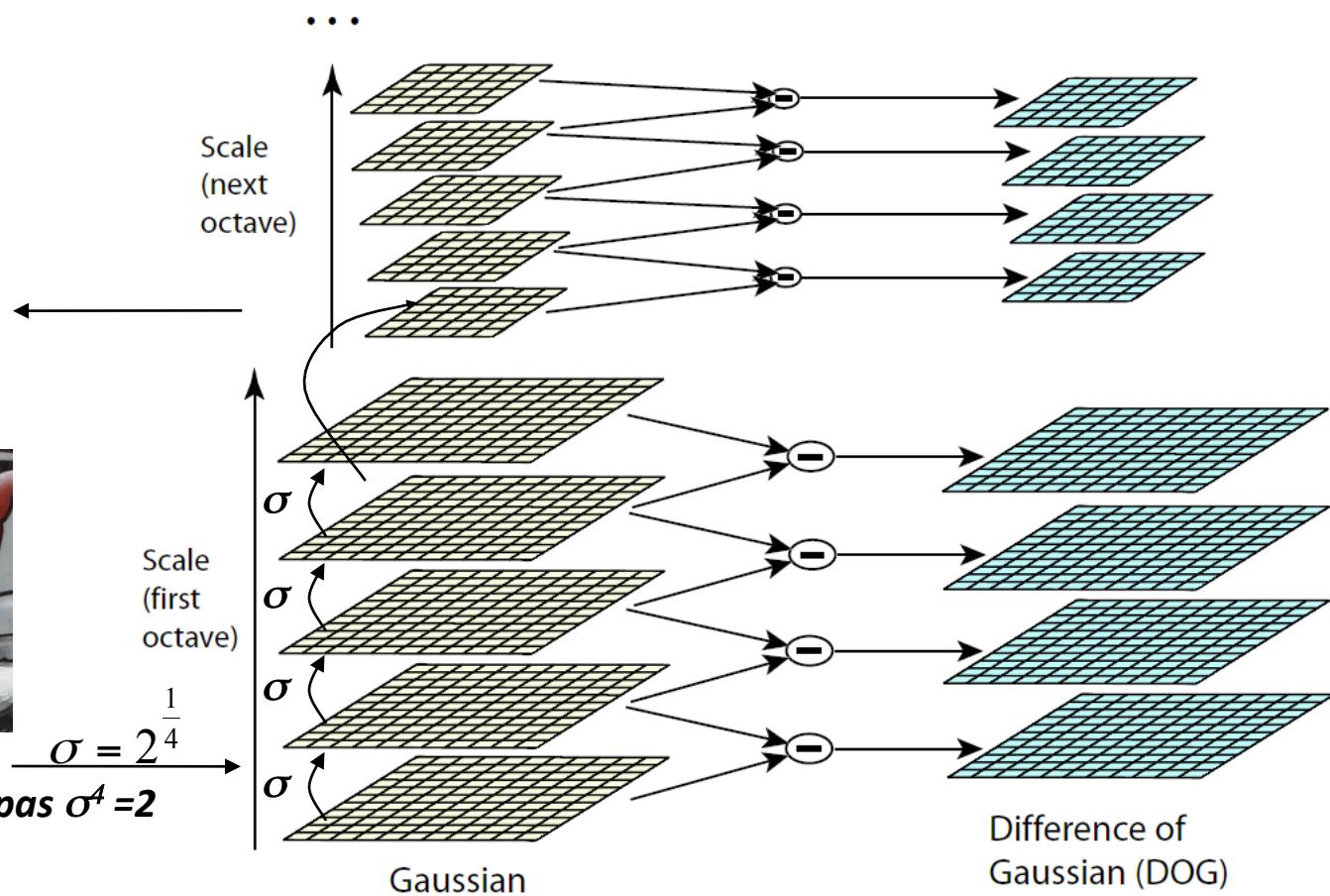
Diferențele a două filtre Gaussiene (DoG)

- construcția unei piramide Gaussiene: imaginea scalată cu factor 2
- pentru fiecare octavă calculez răspunsurile DoG în spațiul 3D (x, y, σ)



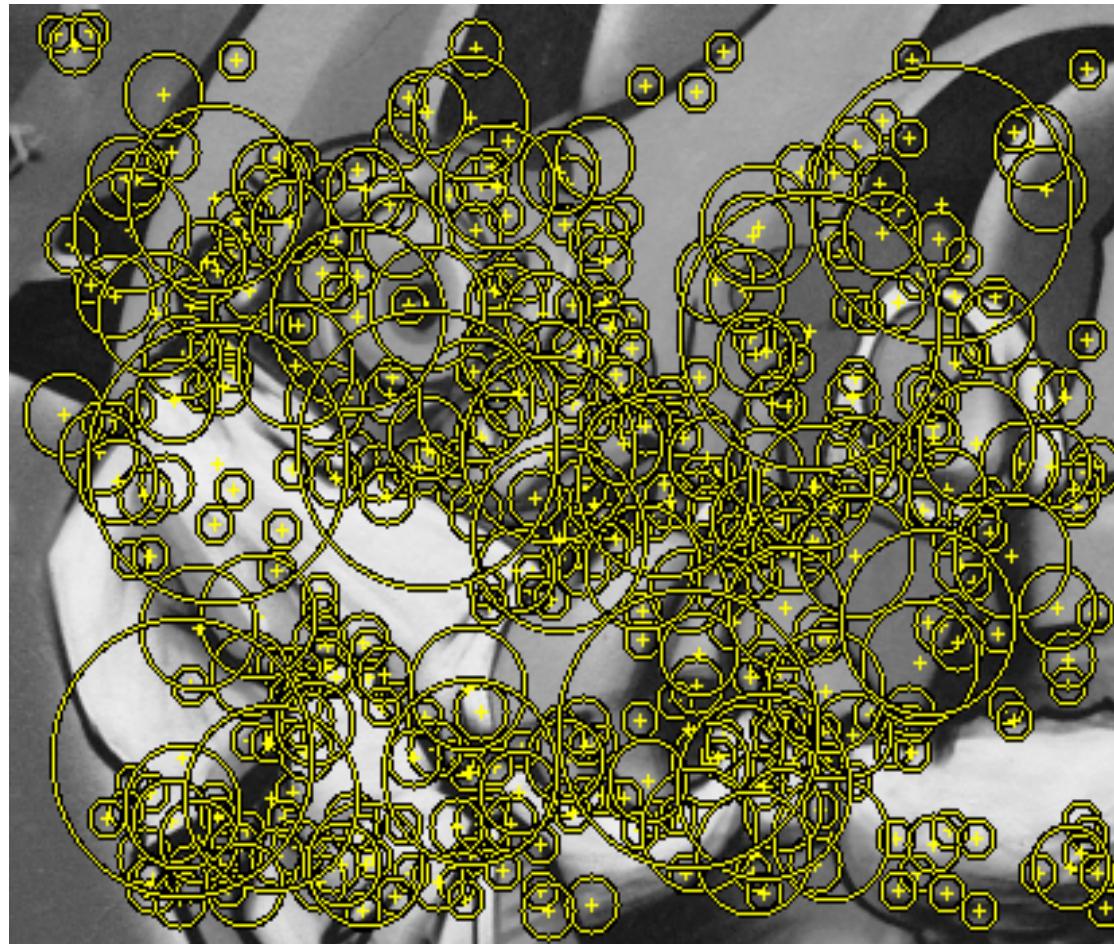
Imagine inițială

$$\frac{\sigma = 2^{\frac{1}{4}}}{\text{pas } \sigma^4 = 2}$$



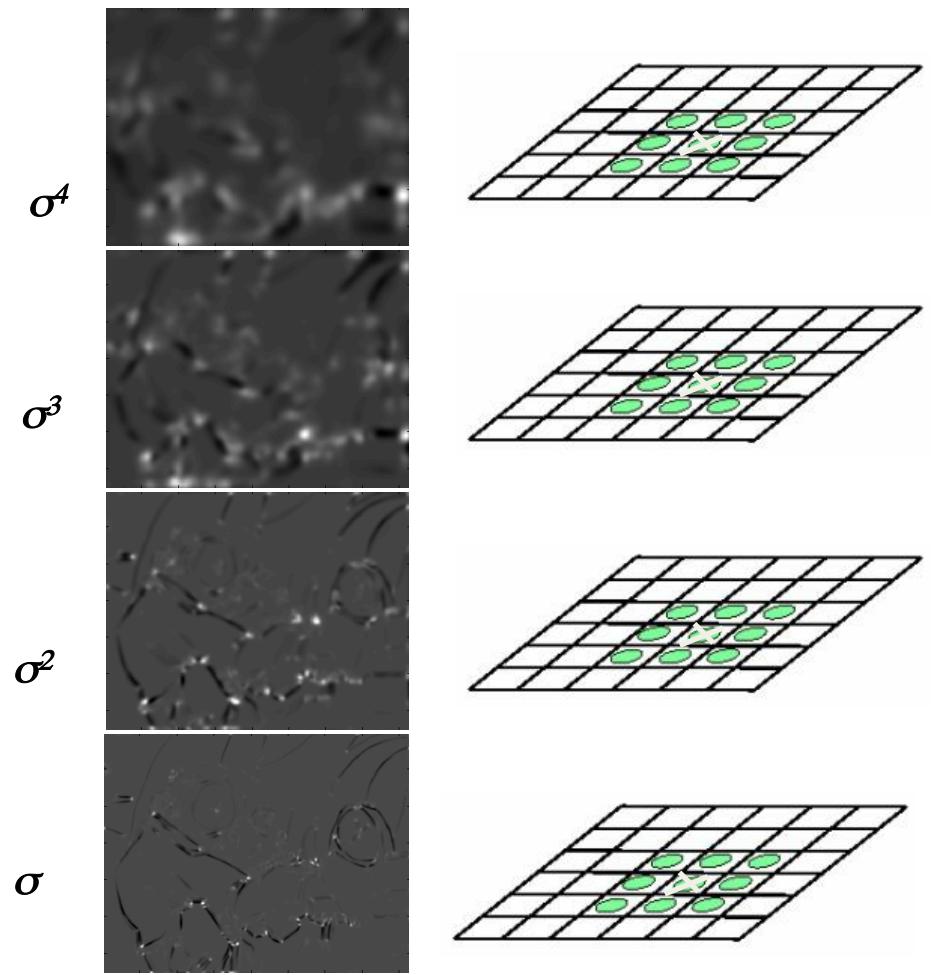
Difference of
Gaussian (DOG)

Diferențele a două filtre Gaussiene (DoG)



Detectorul Harris-Laplace

1. Inițializare: detectorul Harris aplicat pentru mai multe scale



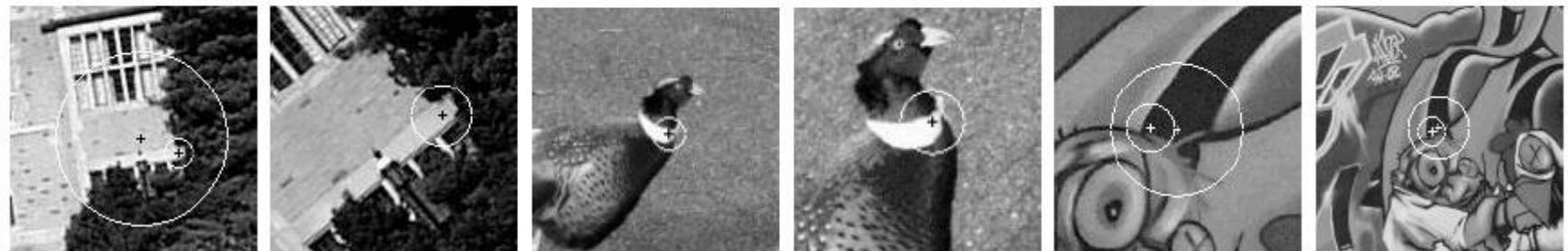
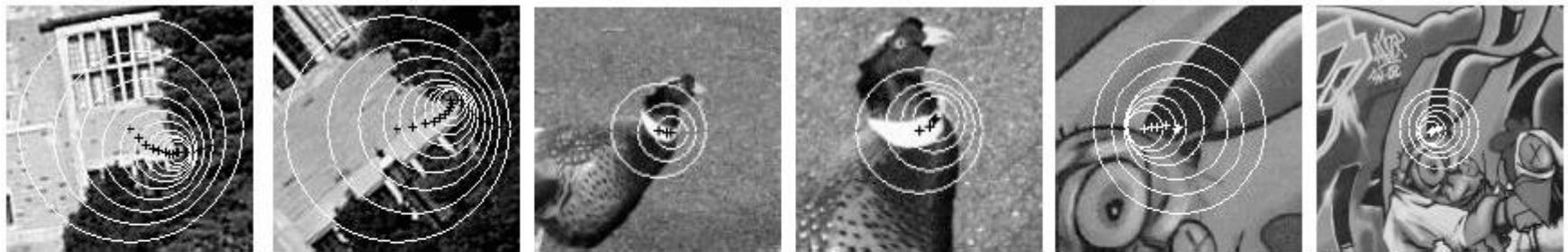
Funcția cornerness

Detectare maxime locale

Detectorul Harris-Laplace

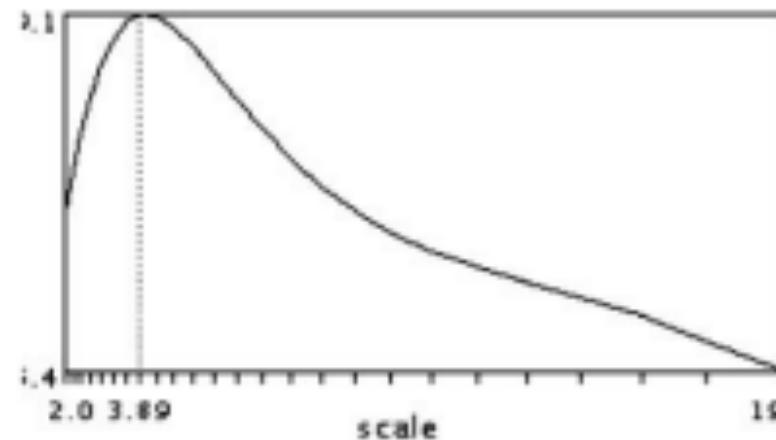
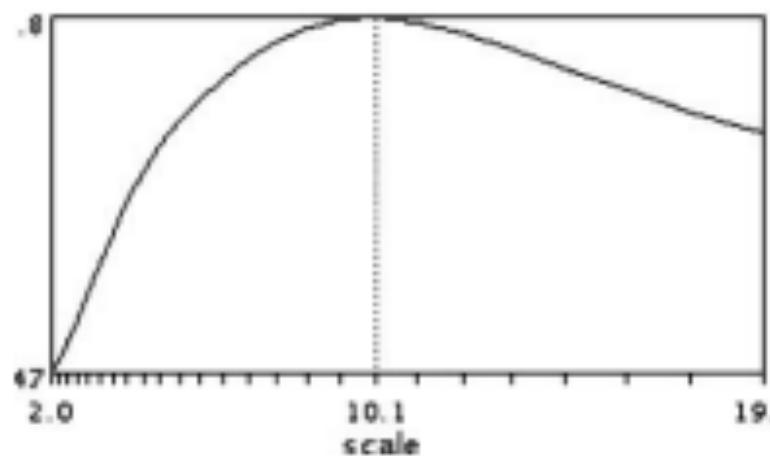
1. Inițializare: detectorul Harris aplicat pentru mai multe scale
2. Selectez scala pe baza filtrului Laplacian
(aceeași procedură și pentru detectorul Hessian-Laplace)

Puncte selectate de detectorul Harris



Puncte selectate de detectorul Harris-Laplace

Exemplu Harris-Laplace



Comparație între detectoare de puncte de interes

Table 7.1 Overview of feature detectors.

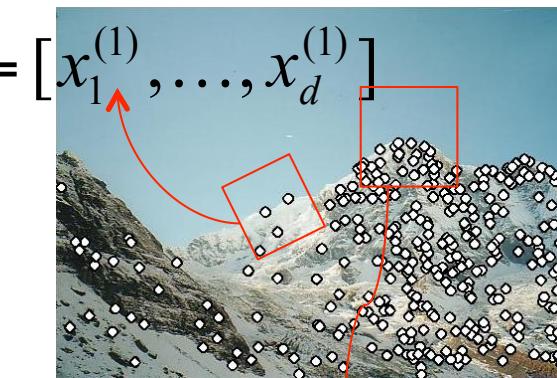
Feature Detector	Corner	Blob	Region	Rotation invariant	Scale invariant	Affine invariant	Repeatability	Localization accuracy	Robustness	Efficiency
Harris	✓			✓			+++	+++	+++	++
Hessian		✓		✓			++	++	++	+
SUSAN	✓			✓			++	++	++	+++
Harris-Laplace	✓	(✓)		✓	✓		+++	+++	++	+
Hessian-Laplace	(✓)	✓		✓	✓		+++	+++	+++	+
DoG	(✓)	✓		✓	✓		++	++	++	++
SURF	(✓)	✓		✓	✓		++	++	++	+++
Harris-Affine	✓	(✓)		✓	✓	✓	+++	+++	++	++
Hessian-Affine	(✓)	✓		✓	✓	✓	+++	+++	+++	++
Salient Regions	(✓)	✓		✓	✓	(✓)	+	+	++	+
Edge-based	✓			✓	✓	✓	+++	+++	+	+
MSER		✓		✓	✓	✓	+++	+++	++	+++
Intensity-based		✓		✓	✓	✓	++	++	++	++
Superpixels		✓		✓	(✓)	(✓)	+	+	+	+

Trăsături locale: componentele principale

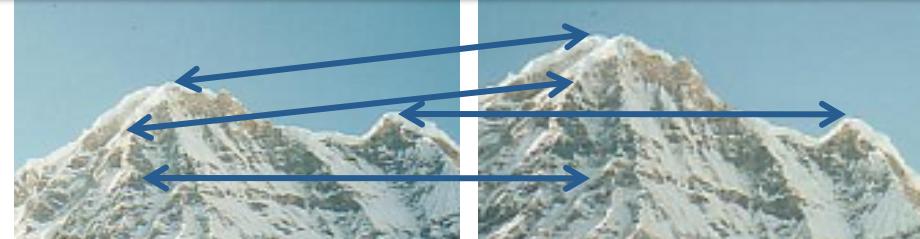
1) Detectare: localizează punctele de interes în imagine



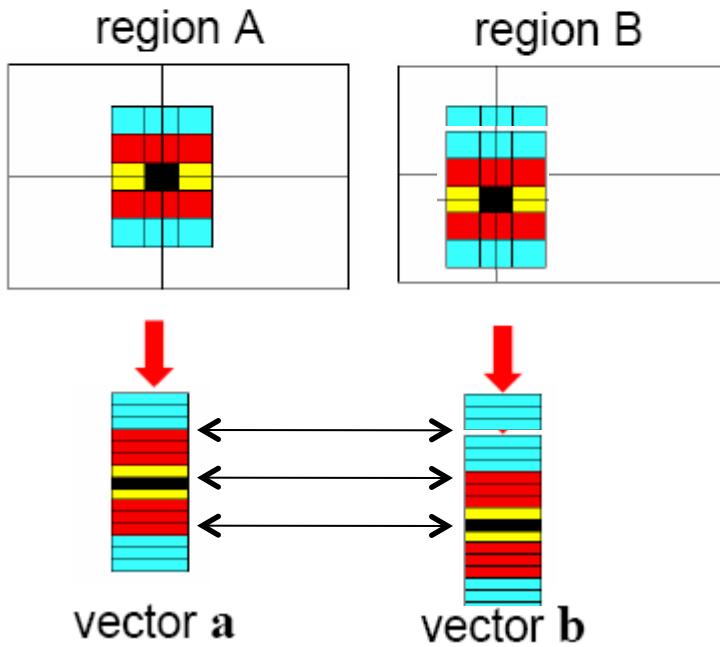
2) Descriere: descrie conținutul vizual din vecinătatea fiecărui punct de interes printr-un vector printr-un descriptor vizual (feature vector)



3) Matching: determină corespondențele dintre descriptorii dintre imagini



Descriptori pe baza intensităților



Cel mai simplu mod de a descrie vecinătatea unui punct de interes (= trăsătură locală) este de include intensitățile din acea regiune în descriptor.

Această alegere este foarte sensitivă la mici translații, rotații.



David Lowe

FOLLOW

Computer Science Dept., [University of British Columbia](#)

Verified email at cs.ubc.ca - [Homepage](#)

Computer Vision Object Recognition

TITLE

CITED BY

YEAR

[Distinctive image features from scale-invariant keypoints](#)

54091

2004

DG Lowe

International journal of computer vision 60 (2), 91-110

[Object recognition from local scale-invariant features](#)

18041

1999

DG Lowe

International Conference on Computer Vision, 1999, 1150-1157

[Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration.](#)

2876

2009

M Muja, DG Lowe

VISAPP (1) 2, 331-340

[Automatic panoramic image stitching using invariant features](#)

2427

2007

M Brown, DG Lowe

International Journal of Computer Vision 74 (1), 59-73

Distinctive Image Features from Scale-Invariant Keypoints

Abstract

This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through least-squares solution for consistent pose parameters. This approach to recognition can robustly identify objects among clutter and occlusion while achieving near real-time performance.

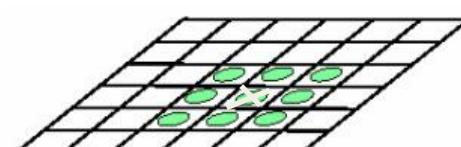
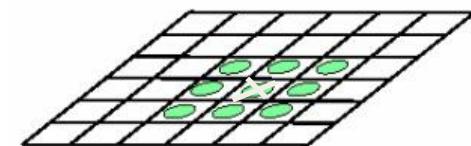
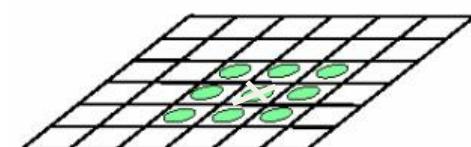
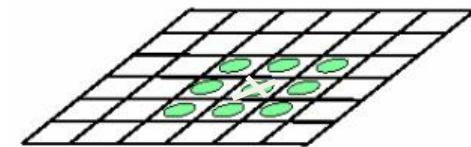
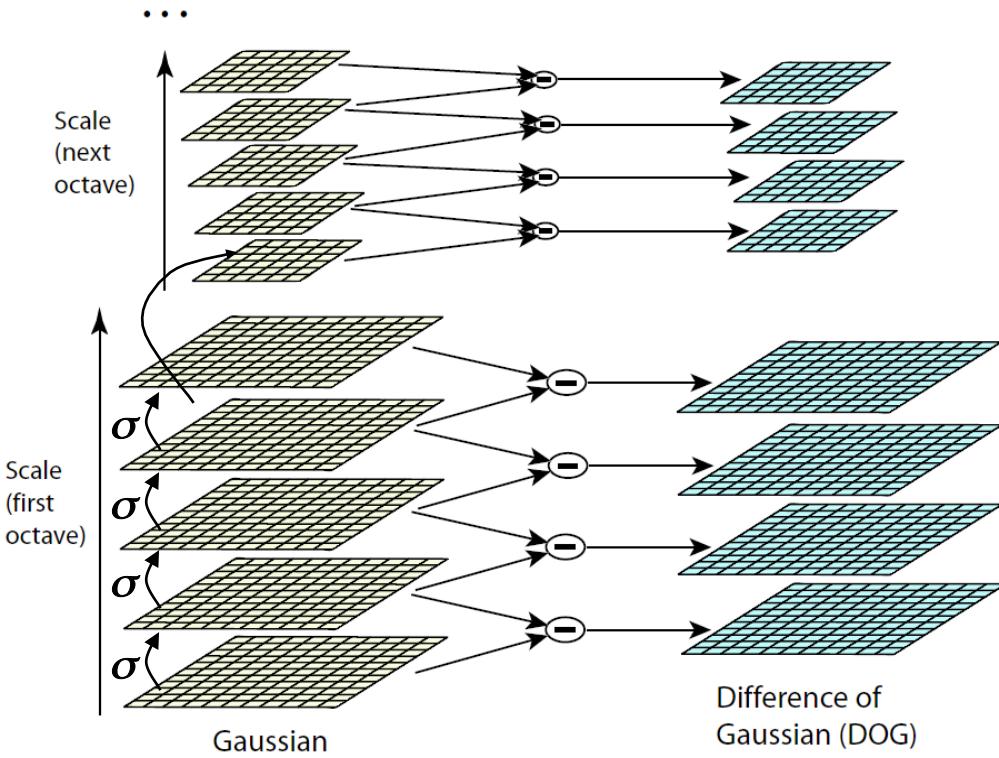
Distinctive Image Features from Scale-Invariant Keypoints

1. **Scale-space extrema detection:** The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.
2. **Keypoint localization:** At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.
3. **Orientation assignment:** One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.
4. **Keypoint descriptor:** The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

This approach has been named the Scale Invariant Feature Transform (SIFT), as it transforms image data into scale-invariant coordinates relative to local features.

Descriptorul SIFT [Lowe 2004]

1. Rulează detectorul DoG
 - găsește maximele în poziție/scală



Detectare maxime locale

Descriptorul SIFT [Lowe 2004]

1. Rulează detectorul DoG

- găsește maximele în poziție/scală
- elimină punctele cu contrast mic + punctele de pe muchii

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad \frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

Folosește matricea Hessiană

Elimină punctele cu raportul între
valorile proprii = $r > 10$

Detectare puncte de interes cu DoG

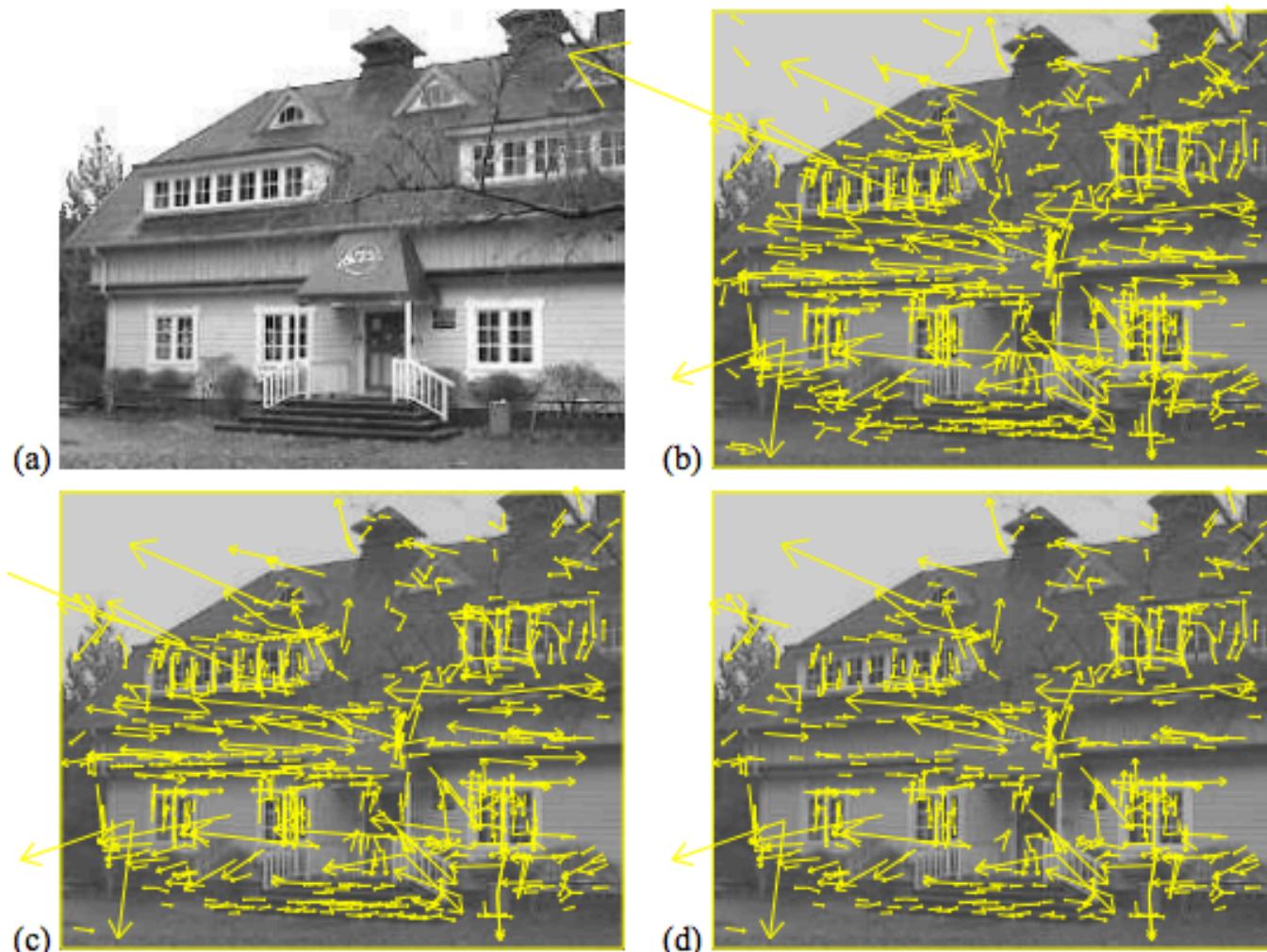


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

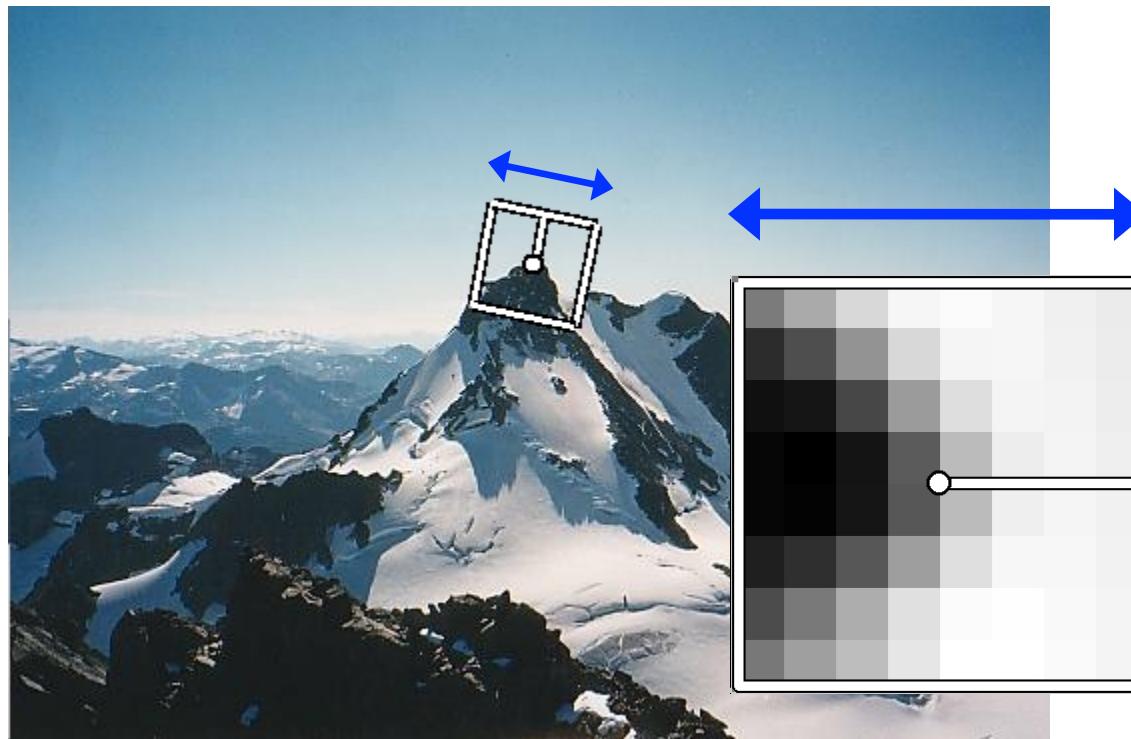
Descriptorul SIFT [Lowe 2004]

1. Rulează detectorul DoG
 - găsește maximele în poziție/scală
 - elimină punctele cu contrast mic + punctele de pe muchii
2. Găsește principalele orientări
 - grupează orientările pixelilor într-o histogramă cu 36 de intervale
 - pondere dată de magnitudinea gradientului + distanța către centrul intervalului
 - returnează orientările care se încadrează în 80% din valoarea maximă
 - în jur de 15% din puncte sunt dublate cu orientări multiple

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

Orientarea canonica



- rotim patch-ul (regiunea) pe baza orientării dominante
- obțin un patch cu orientare canonica

Repetabilitate (1)

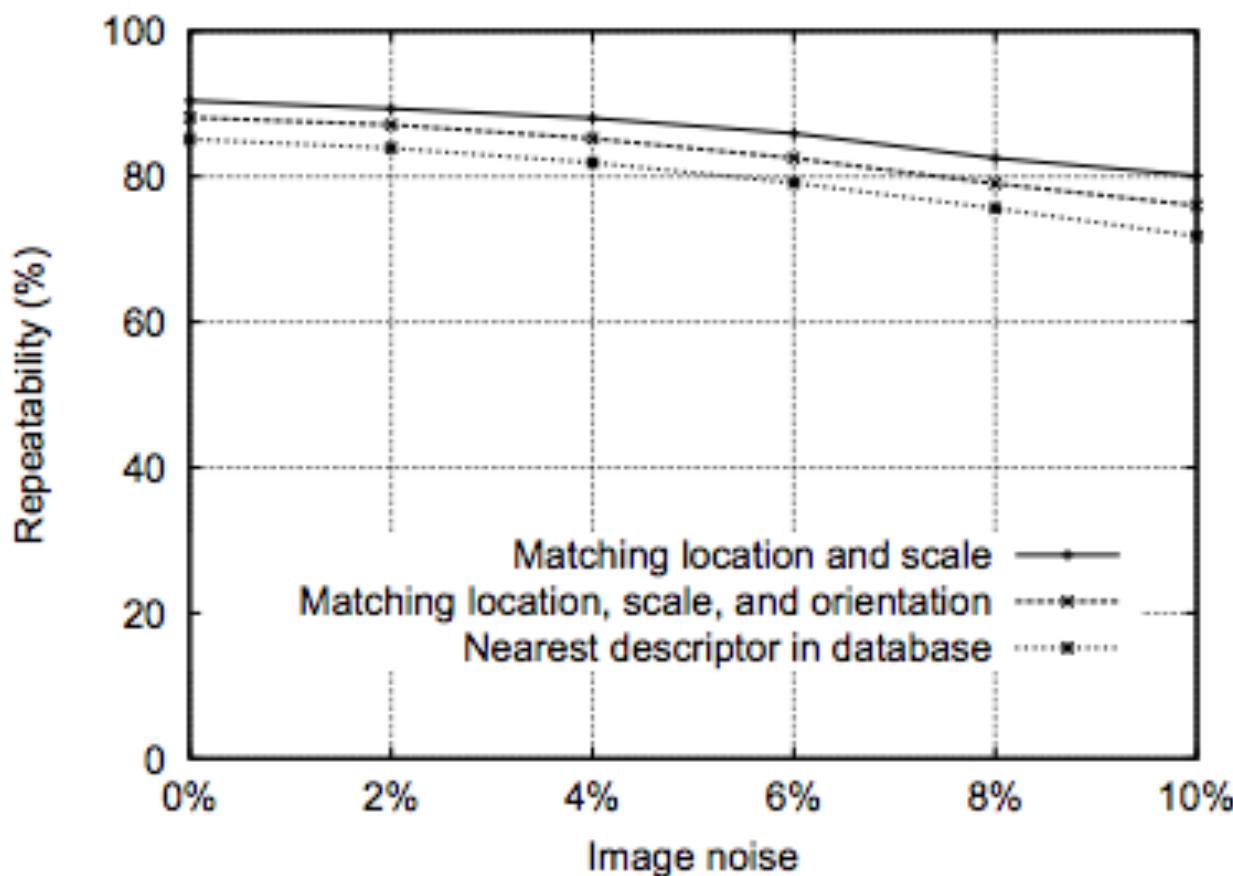


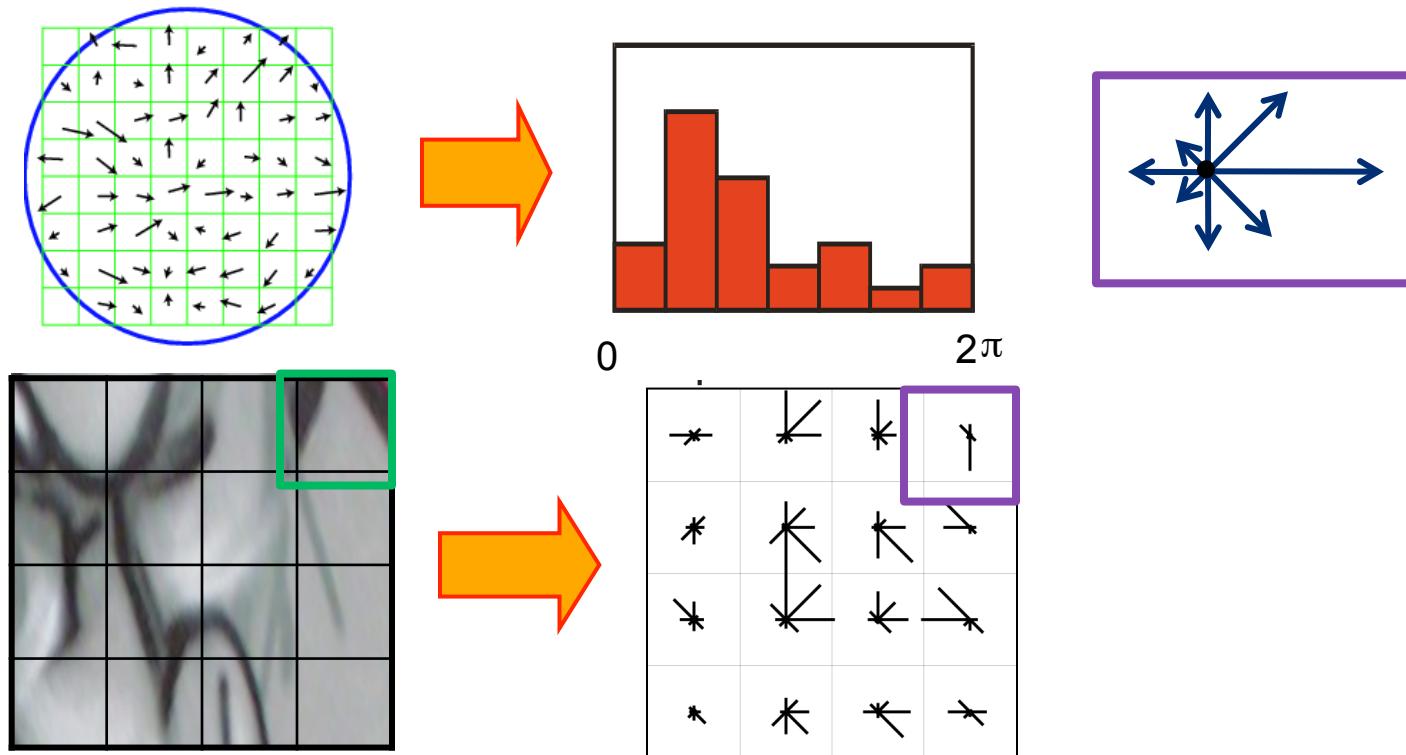
Figure 6: The top line in the graph shows the percent of keypoint locations and scales that are repeatably detected as a function of pixel noise. The second line shows the repeatability after also requiring agreement in orientation. The bottom line shows the final percent of descriptors correctly matched to a large database.

Descriptorul SIFT [Lowe 2004]

1. Rulează detectorul DoG
 - găsește maximele în poziție/scală
 - elimină punctele cu contrast mic + punctele de pe muchii
2. Găsește principalele orientări
 - grupează orientările pixelilor într-o histogramă cu 36 de intervale
 - pondere dată de magnitudinea gradientului + distanța către centrul intervalului
 - returnează orientările care se încadrează în 80% din valoarea maximă
 - în jur de 15% din puncte sunt dublate cu orientări multiple
3. Pentru fiecare $(x,y,scală,orientare)$, calculează descriptorul:
 - împarte regiunea de 16×16 pixeli în 4×4 blocuri = 16 blocuri
 - calculează pentru fiecare bloc o histogramă de orientări ale pixelilor (intervalul $0-360^0$ împărțit în 8 intervale)
 - valoare maximă a gradientilor la 0.2 și apoi re-normalizare
 - descriptorul SIFT = 16 blocuri = 16 histograme x 8 valori = dimensiune 128

Descriptorul SIFT [Lowe 2004]

- împarte regiunea de 16×16 pixeli în 4×4 blocuri = 16 blocuri
- calculează pentru fiecare bloc o histogramă de orientări ale pixelilor (intervalul $0-360^0$ împărțit în 8 intervale)
- descriptorul SIFT = 16 blocuri = 16 histograme \times 8 valori = dimensiune 128



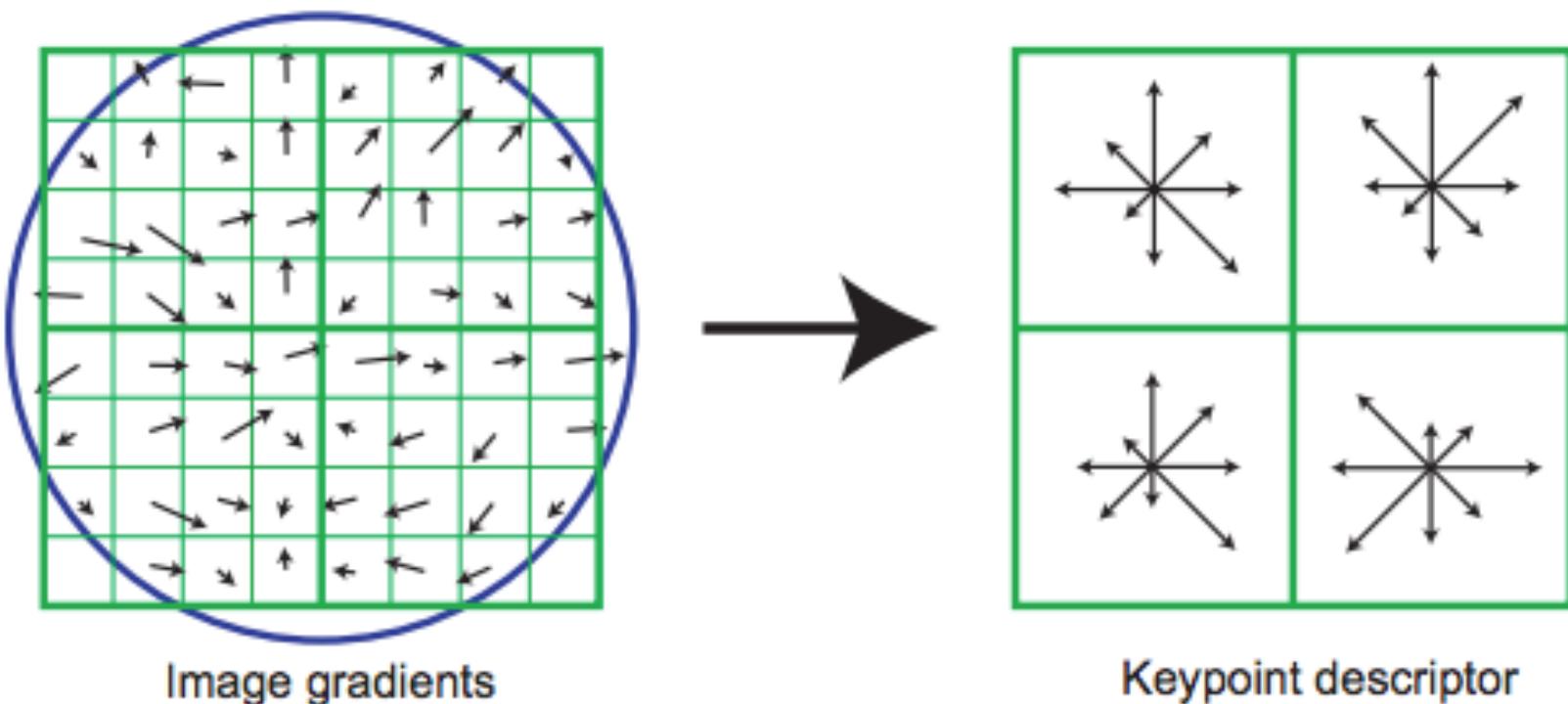


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4×4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2×2 descriptor array computed from an 8×8 set of samples, whereas the experiments in this paper use 4×4 descriptors computed from a 16×16 sample array.

Alegerea numărului de blocuri

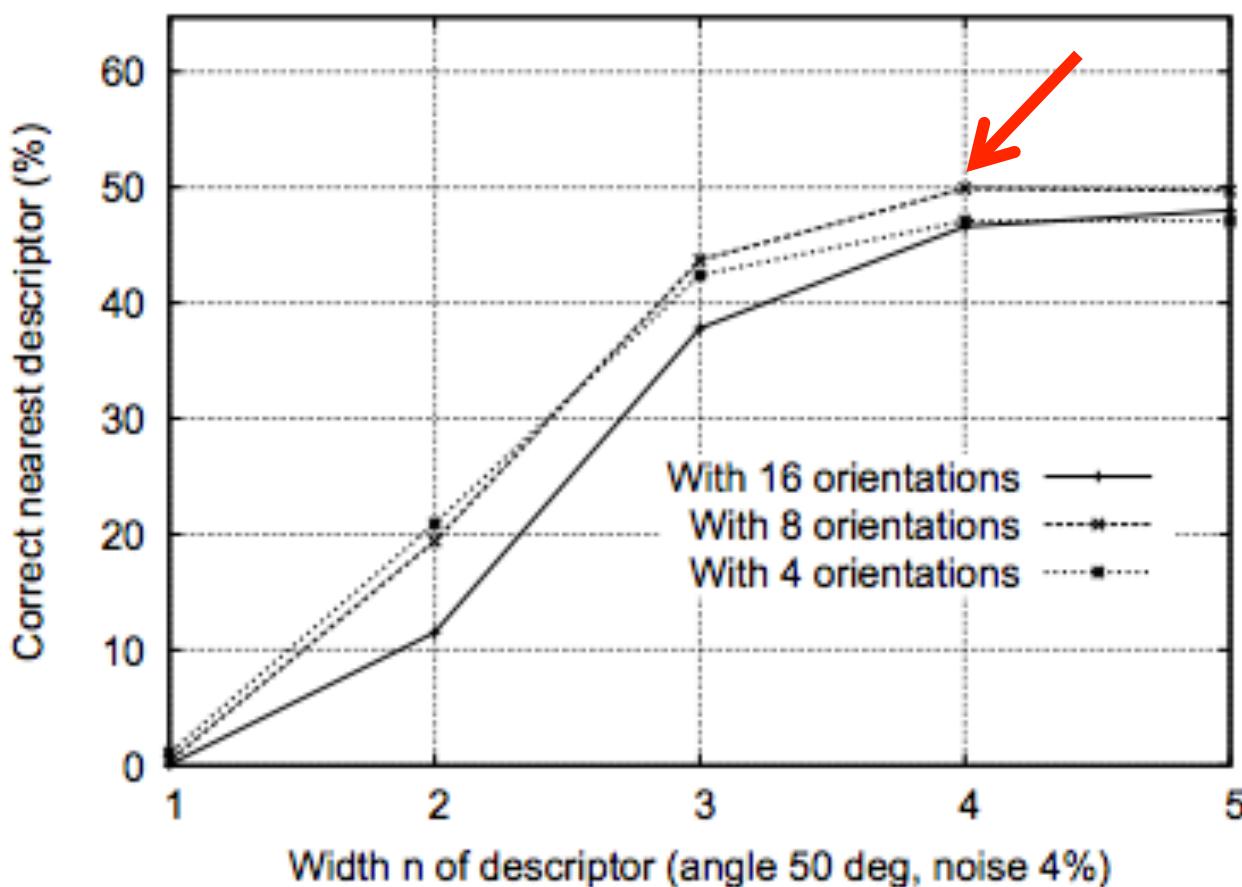


Figure 8: This graph shows the percent of keypoints giving the correct match to a database of 40,000 keypoints as a function of width of the $n \times n$ keypoint descriptor and the number of orientations in each histogram. The graph is computed for images with affine viewpoint change of 50 degrees and addition of 4% noise.

Repetabilitate (2)

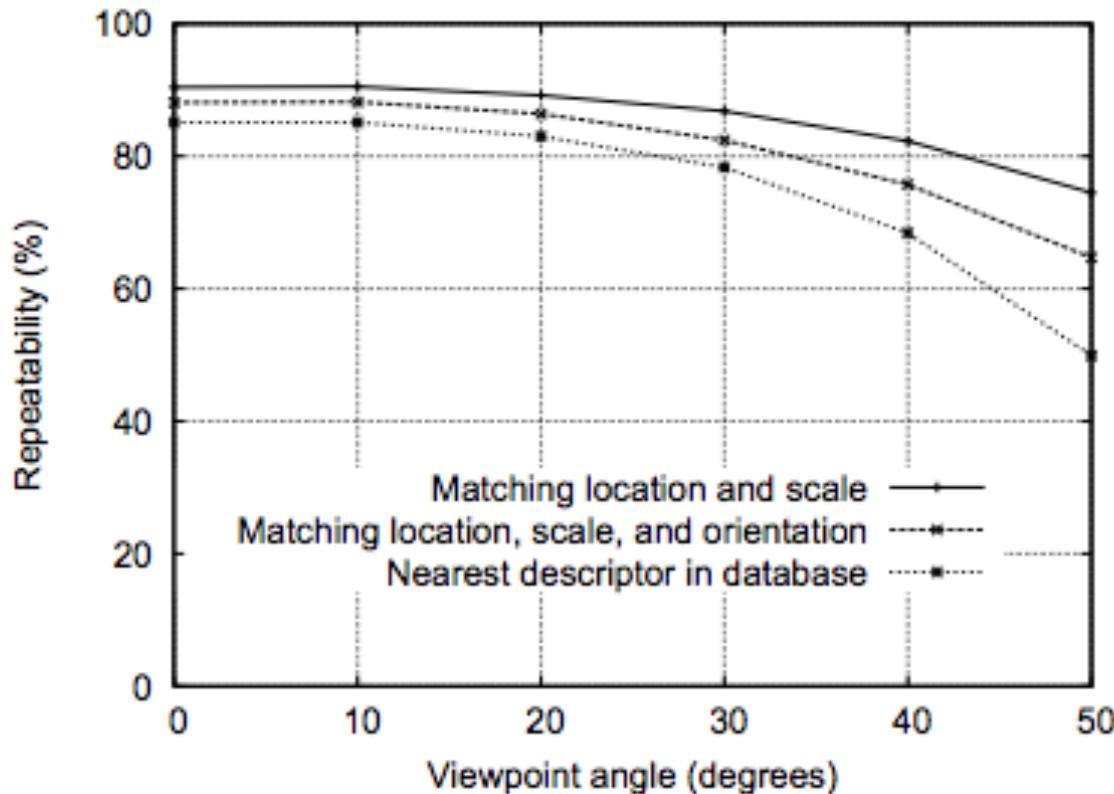


Figure 9: This graph shows the stability of detection for keypoint location, orientation, and final matching to a database as a function of affine distortion. The degree of affine distortion is expressed in terms of the equivalent viewpoint rotation in depth for a planar surface.

Descriptorul SIFT [Lowe 2004]

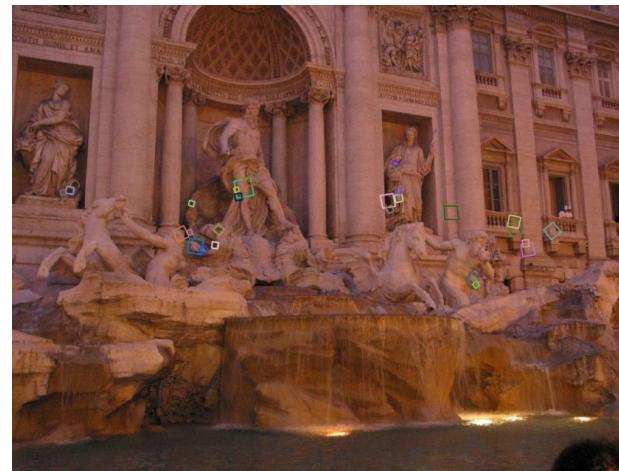


Puncte de interes cu
scalele și orientările
asociate
(selectie aleatoare de 50
de puncte)

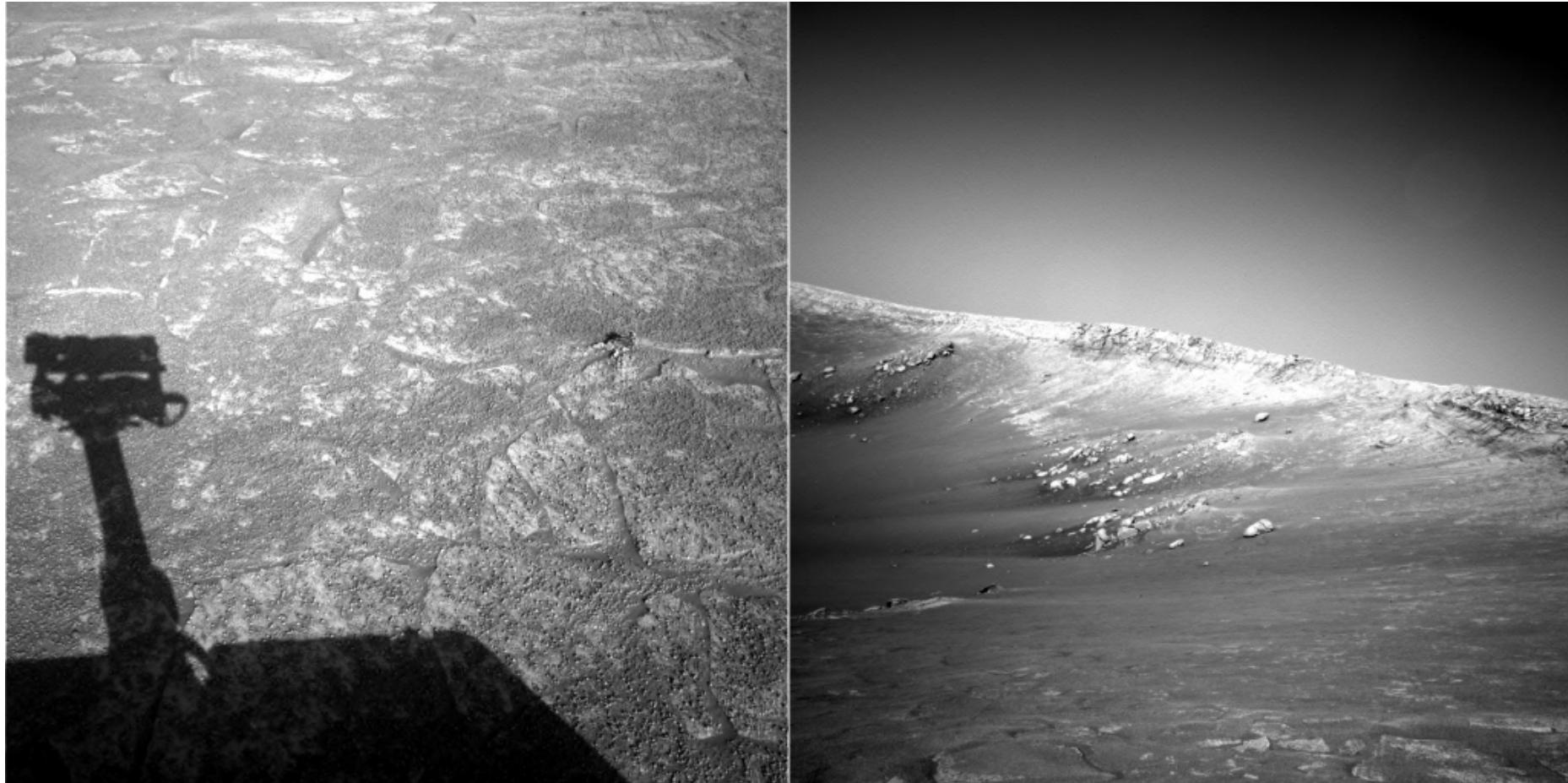
Descriptori SIFT

Descriptorul SIFT [Lowe 2004]

- foarte robust pentru matching (găsire de corespondențe)
- robust la schimbări mari ale unghiului din care a fost luată imaginea
- robust la schimbări mari în iluminarea scenei
 - chiar și condiții de zi vs condiții de noapte
- foarte rapid – poate rula în timp real
- cod disponibil online

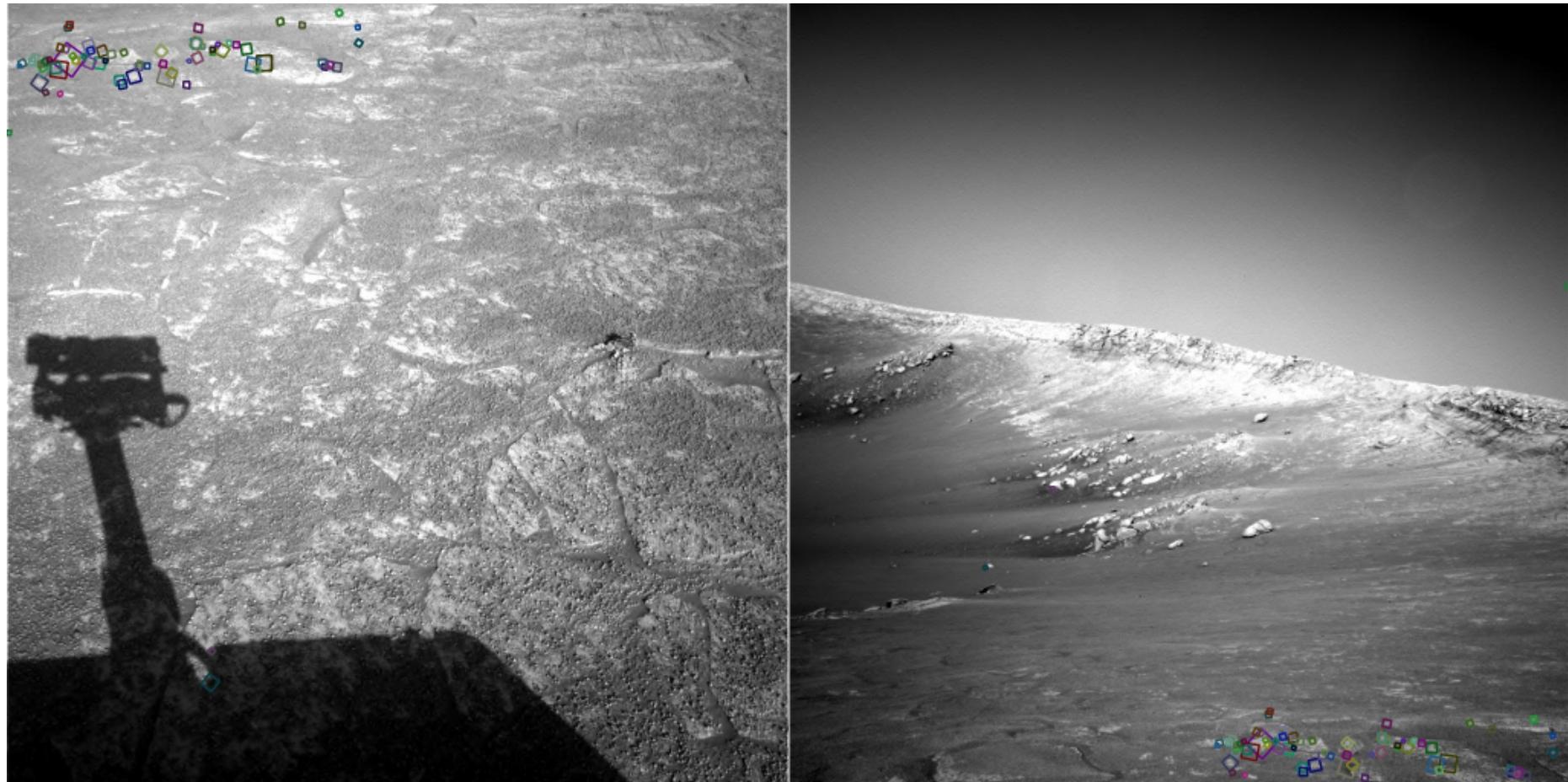


Exemplu



NASA Mars Rover images

Exemplu

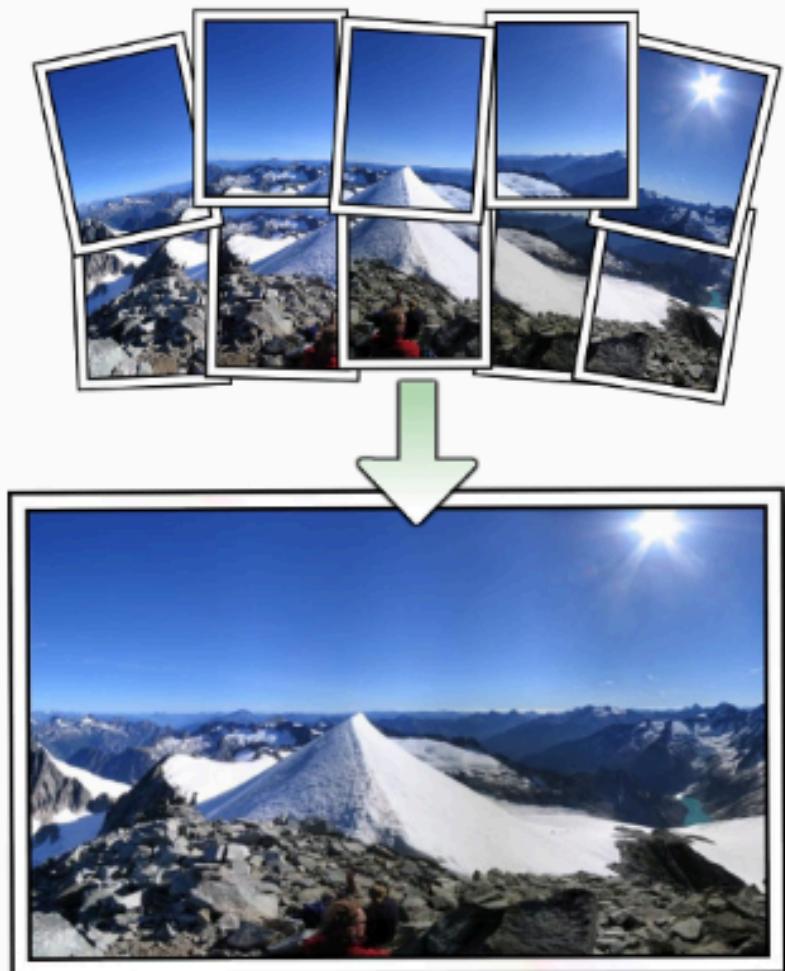


NASA Mars Rover images

AUTOSTITCH

[Windows](#) | [Mac OSX](#) | [Licensing](#) | [FAQ](#) | [Publications](#)

AutoStitch: a new dimension in automatic image stitching



What if your camera could see everything you can see?

The human visual system has a field of view of around 135 x 200 degrees, but a typical camera has a field of view of only 35 x 50 degrees. Panoramic image mosaicing works by taking lots of pictures from an ordinary camera, and stitching them together to form a composite image with a much larger field of view.

AutoStitch takes a step forward in panoramic image stitching by automatically recognising matching images. This allows entire panoramas to be constructed with no user input whatsoever. AutoStitch is incredibly simple to use! Just select a set of photos, and AutoStitch does the rest: digital photos in, panoramas out.

Mobile versions are no longer available, sorry! Please try the free demo instead

The AutoStitch Process

AutoStitch works from unordered collections of images, automatically finding matches between images using the [SIFT](#) algorithm. It then robustly aligns all images and uses advanced blending algorithms to form seamless panoramas (see below). For more details, see our [research papers](#).



25 of 57 images aligned



All 57 images aligned



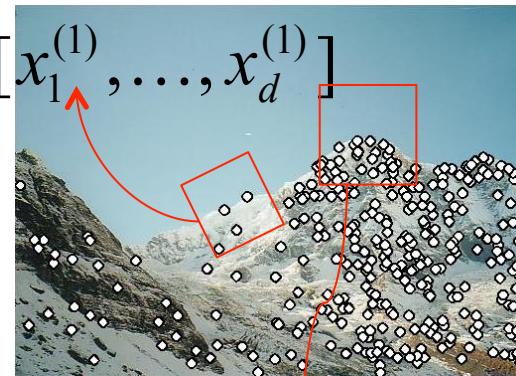
Final Result

Trăsături locale: componentele principale

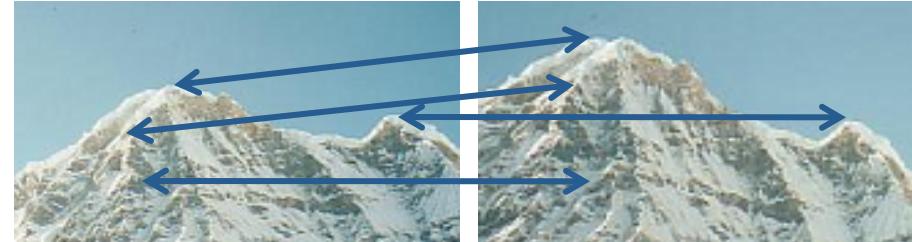
1) Detectare: localizează punctele de interes în imagine



2) Descriere: descrie conținutul vizual din vecinătatea fiecărui punct de interes printr-un vector printr-un descriptor vizual (feature vector)



3) Matching: determină corespondențele dintre descriptorii dintre imagini



Corespondențe pentru trăsături locale



Corespondențe pentru trăsături locale



Pentru găsirea de trăsături locale corespondente, găsește patch-uri similare în infățișare – calculează o distanță
Abordare simplă: k nearest neighbors, cu o distanță maximă admisă

Corespondențe ambigu



Imagine 1



Imagine 2

Pentru ce distanțe avem o corespondență bună?

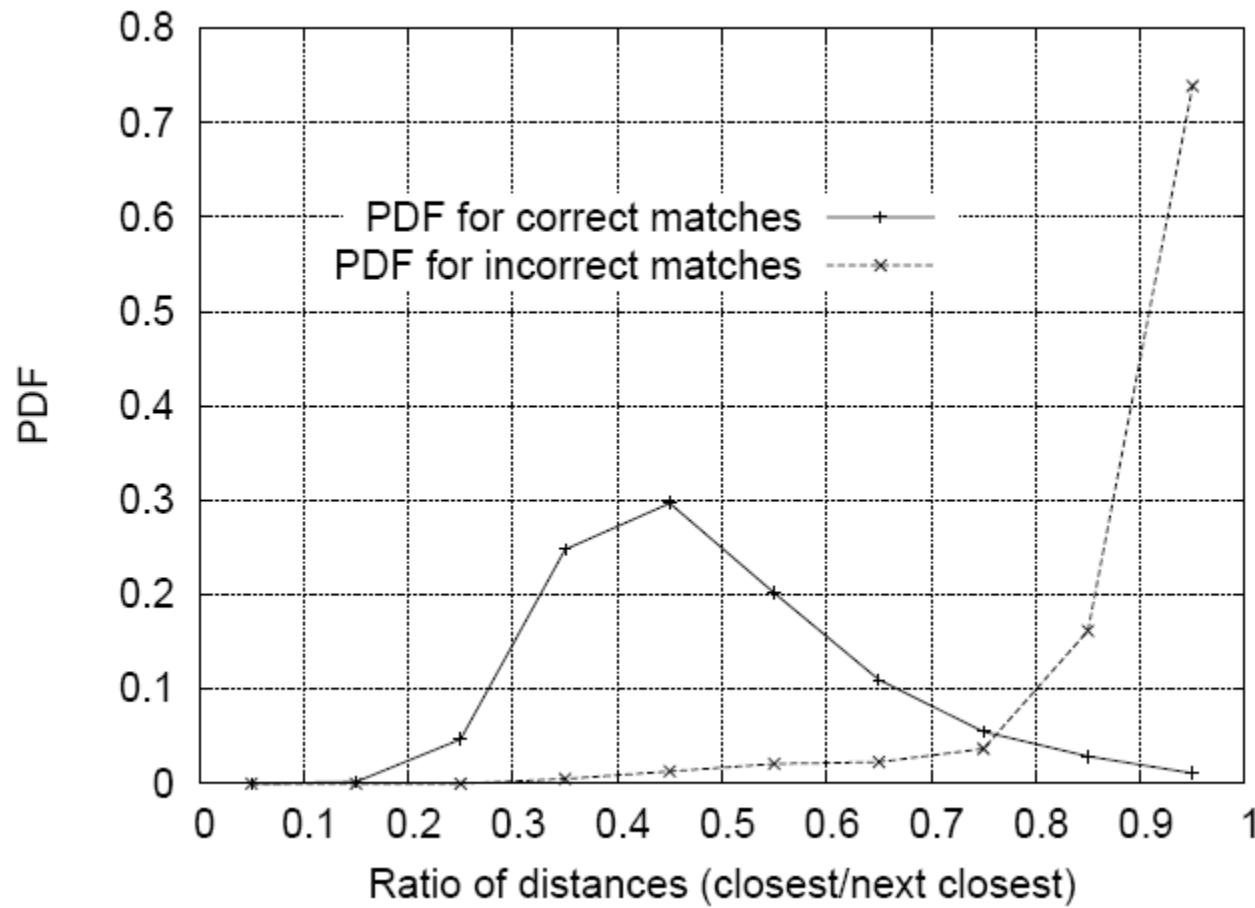
Pentru găsirea de corespondențe robuste, considerăm raportul:
distanța față de first NN / distanța față de 2nd NN

Dacă **raportul e mic**, totul **pare ok** (nu există ambiguități)

Dacă **raportul e mare**, nu **pare ok** (pot exista ambiguități)

Corespondențe între descriptori SIFT

- Nearest neighbor (distanța Euclidiană)
- Raportul dintre distanța față de primul vecin și distanța față de al doilea vecin



Corespondențe pe baza descriptorilor SIFT

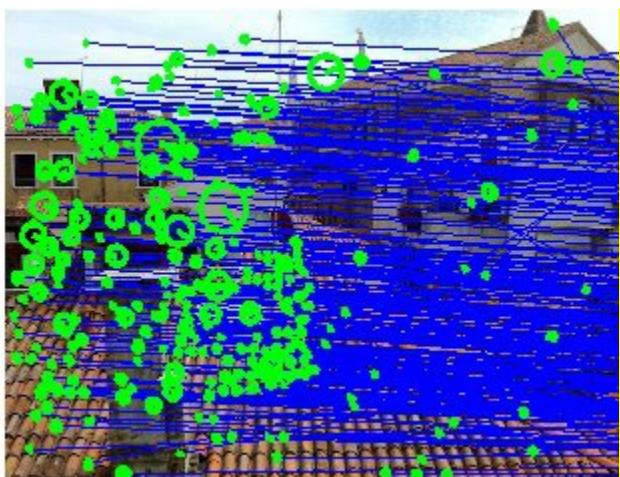
img1



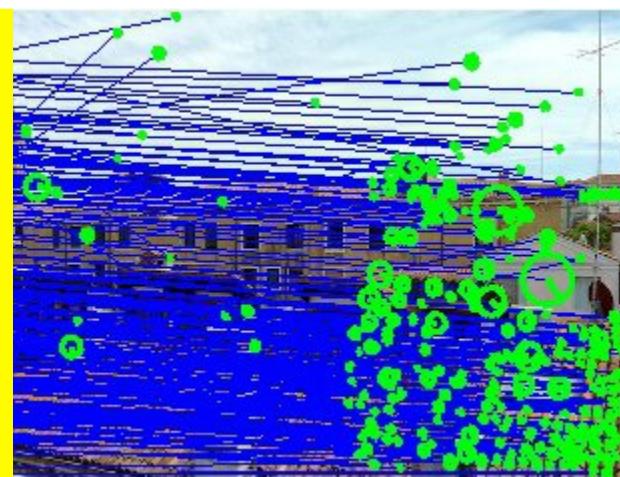
img2



img1



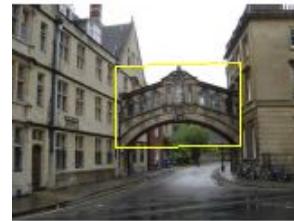
img2



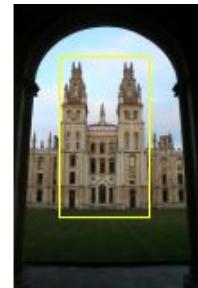
Recunoaștere de obiecte sau scene specifice



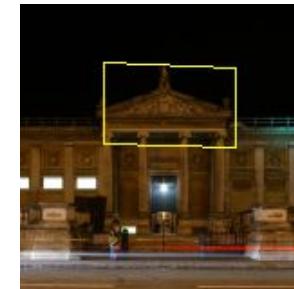
Scale



Viewpoint



Lighting



Occlusion

