

# Concepte și aplicații în Vederea Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Cursul 7

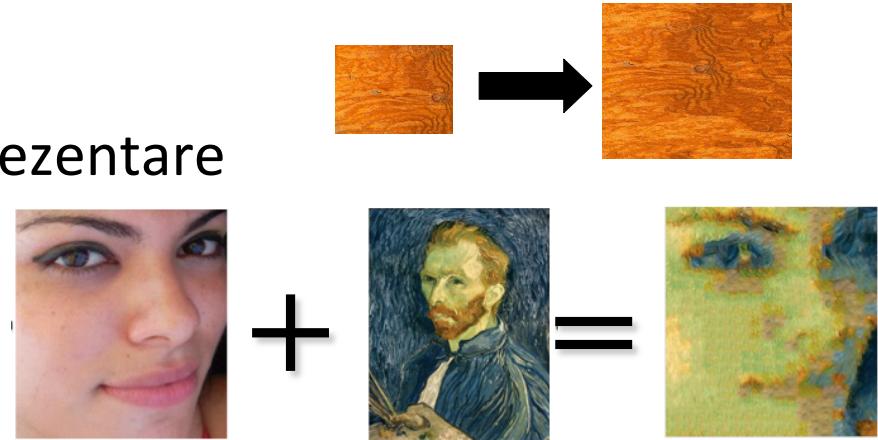
anul III, Opțional Informatică, semestrul I, 2018-2019

# Studenti ce au predat tema 2

Nr.	Nume student	Grupa
1	Dranca Constantin	334
2	Botezatu Daniel Andrei	342
3	Isaia Vlad-Lucian	343
4	ANDREI-ALEXANDRU AVRAM	
5	Andra Tomi	344
6	Radu Muntean	331
7	Uta Stefana	344
8	Pandele Maria Smaranda	331
9	Banu Robert Emanuel	342
10	Lupascu Marian	331
11	NEDELCU ANDREEA	344
12	Feraru Andrei Ionut	342
13	Belcineanu Alexandru-Ioan	344
14	Jitca David	334
15	Stancu Robert	343
16	Poesina Eduard Gabriel	334
17	Caluian Iulian	343
18	Mare Tudor	344
19	Robu Costin Stefan	332
20	Zugravu Andrei	331
21	Alexandru Banu	331
22	Teodor Ionescu	331
23	Dragos Giumanca	342
24	Valentin Calinescu	331

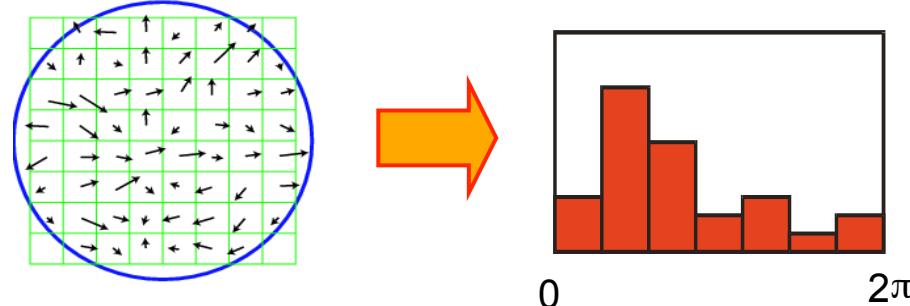
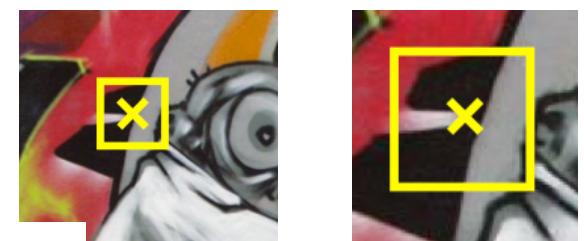
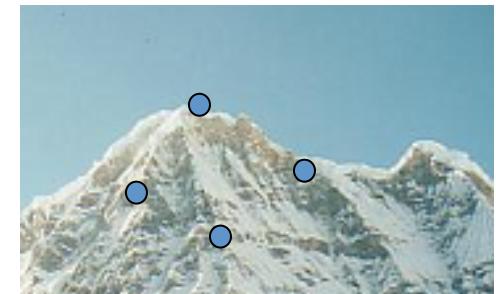
# Cursul trecut

- Textură
  - Reprezentare
  - Colecții de filtre pentru reprezentare
  - Sinteză texturii (tema 3)
  - Transferul texturii (tema 3)



# Cursul de azi

- trăsături locale: detectare + descriere
- detectorul Harris
- detectorul Harris Laplace
- descriptorul SIFT



# Trăsături locale: terminologie

- În limba engleză: local feature = interest point + region descriptor = keypoint + region descriptor
- În limba română: trăsătură locală = caracteristică locală = regiune din imagine centrată într-un punct de interes (punct cheie) + descriptorul conținutului vizual al regiunii din imagine;
- trăsătură locală = o regiune din imagine cu proprietăți speciale, în sensul că poate fi detectată dacă imaginea suferă transformări geometrice sau fotometrice (nu foarte mari)

# Trăsături locale

- trăsătură locală = o regiune din imagine cu proprietăți speciale, în sensul că poate fi detectată dacă imaginea suferă transformări geometrice sau fotometrice (nu foarte mari)

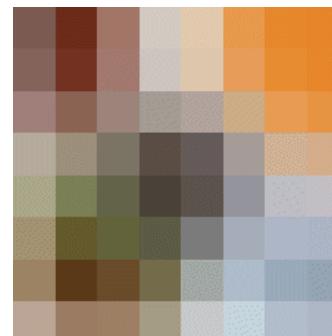
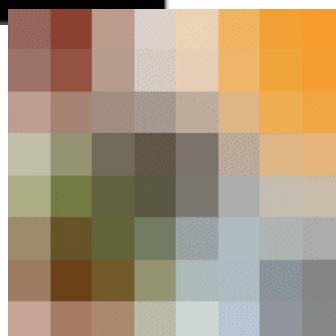
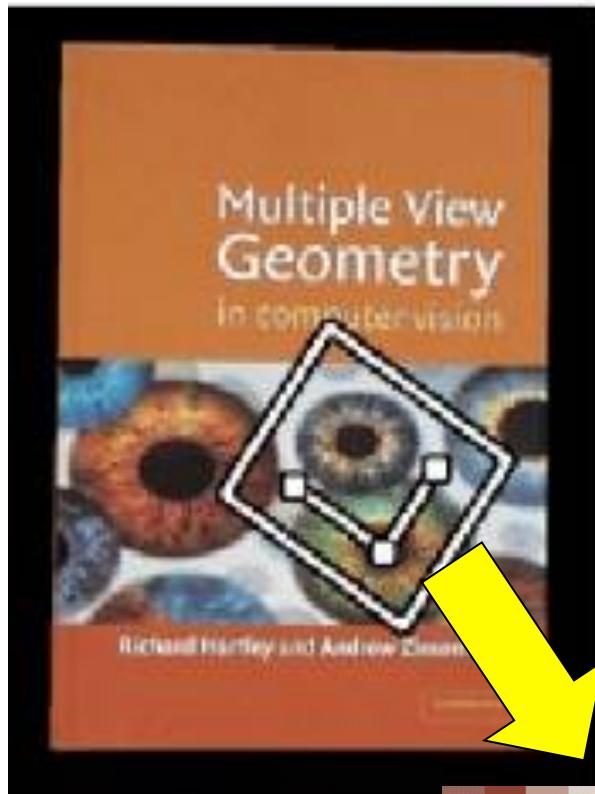


# Trăsături locale

- trăsătură locală = o regiune din imagine cu proprietăți speciale, în sensul că poate fi detectată dacă imaginea suferă transformări geometrice sau fotometrice (nu foarte mari)



# Transformări geometrice



Exemplu:  
translație +  
rotație +  
scalare

# Transformări fotometrice



# Trăsături locale

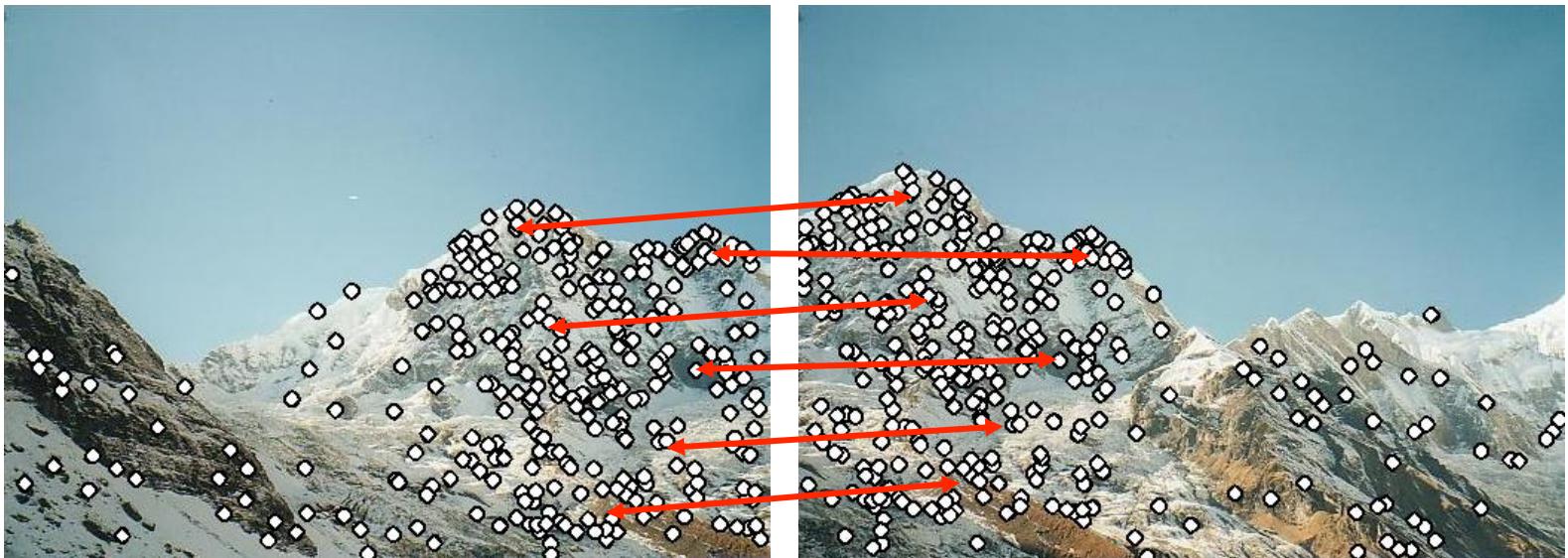
- posibilă aplicație: realizare de panorame



- avem 2 imagini ale aceleasi scene
- cum le combinăm astfel încât să obținem o imagine mai mare (o panoramă)?

# Trăsături locale

- posibilă aplicație: realizare de panorame



Pasul 1: extrage trăsături (puncte din imagini + descriptori ai conținutului vizual)

Pasul 2: găsește corespondențe între trăsăturile extrase

# Trăsături locale

- posibilă aplicație: realizare de panorame

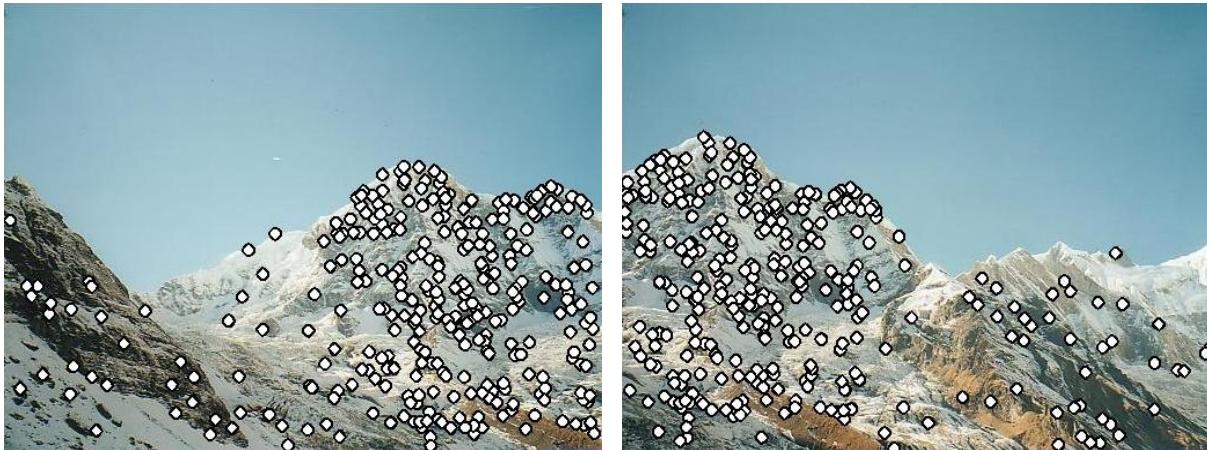


Pasul 1: extrage trăsături (puncte din imagini + descriptori ai conținutului vizual)

Pasul 2: găsește corespondențe între trăsăturile extrase

Pasul 3: aliniază imaginile (găsește homografia care aliniază imaginile)

# Ce proprietăți vrem să aibă trăsăturile locale?



- Repetabilitate
  - aceeași trăsătură locală să poată fi găsită în alte imagini obținute din imaginea inițială prin transformări geometrice (rotație, perspectivă, etc.) sau fotometrice (iluminarea scenei diferă zi/noapte)
- Unicitate
  - fiecare trăsătură locală are o descriere unică
- Reprezentare compactă a imaginii
  - mult mai puține trăsături locale decât pixeli
- Localitate
  - o trăsătură locală ocupă o portiune mică dintr-o imagine; reprezentare robustă la mascarea obiectelor, background

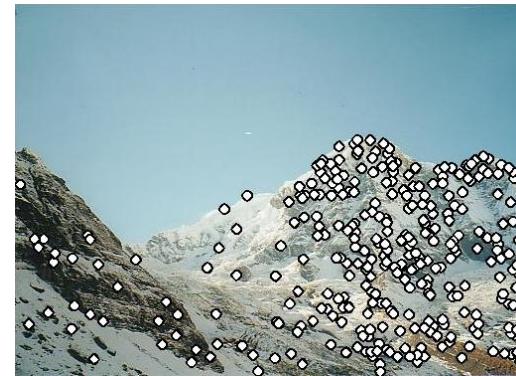
# Trăsături locale: aplicații

- alinierea imaginilor
- reconstrucție 3D
- tracking
- navigarea robotilor
- indexarea și regăsirea informației într-o colecție mare de imagini (milioane de imagini)
- recunoașterea obiectelor

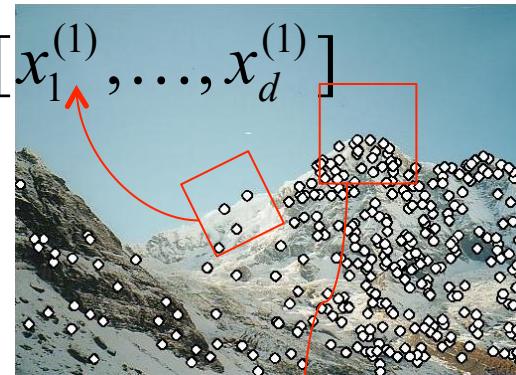


# Trăsături locale: componentele principale

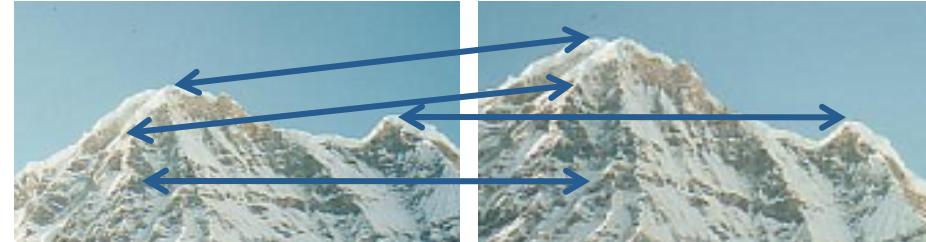
**1) Detectare:** localizează punctele de interes în imagine



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

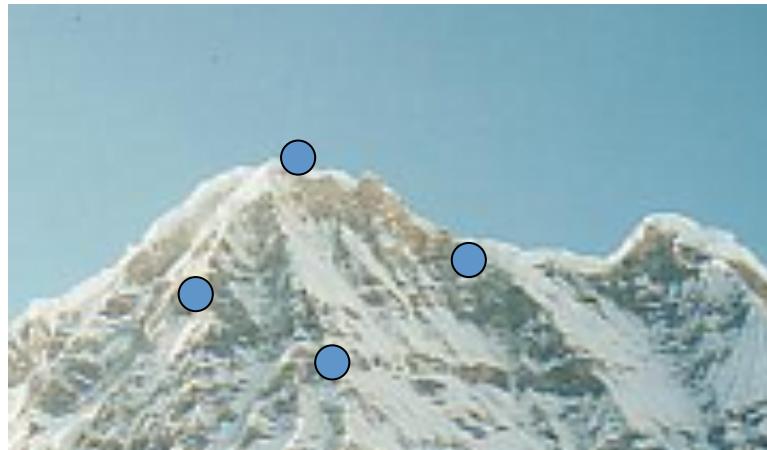


**3) Matching:** determină corespondențele dintre descriptorii dintre imagini



# Trăsături locale: repetabilitate

- vrem să găsim (mare parte din) aceleasi puncte din ambele imagini.

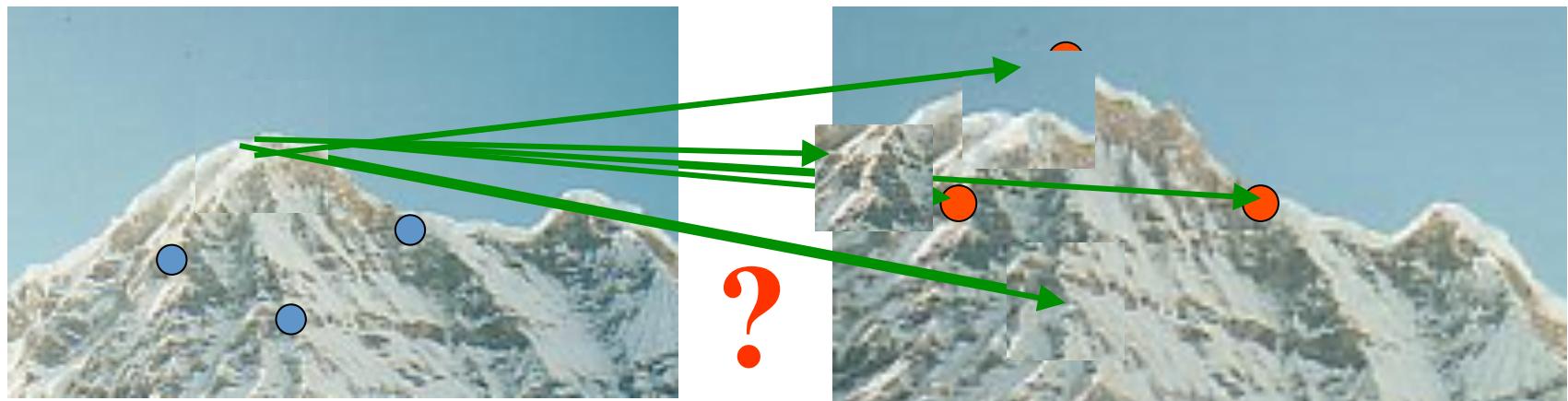


În acest caz nu am putea găsi corespondențe

- procedura de detectare a trăsăturilor locale rulează *independent* pentru fiecare imagine.

# Trăsături locale: unicitatea descrierii

- abilitate de a determina în mod corect corespondența dintre puncte

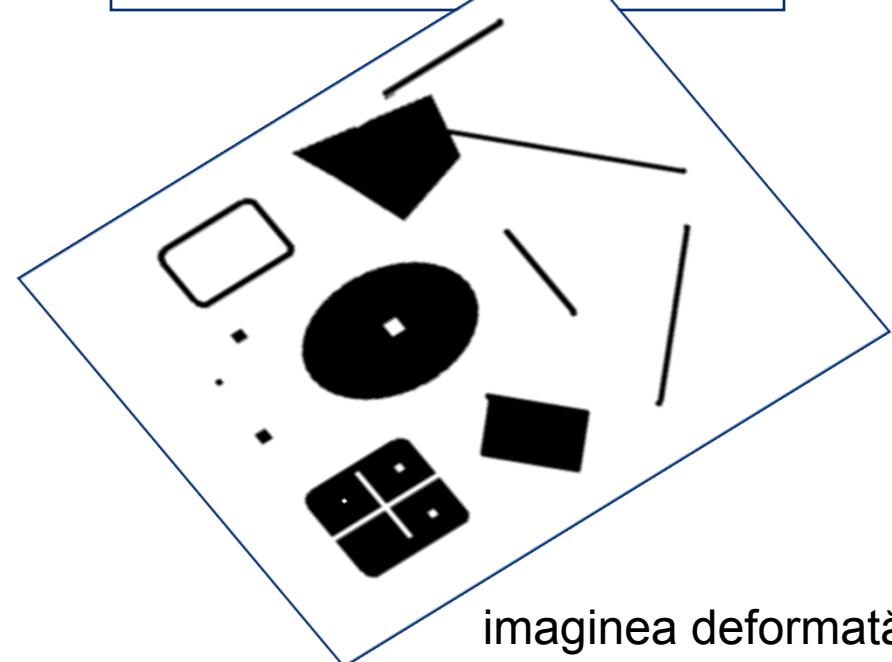
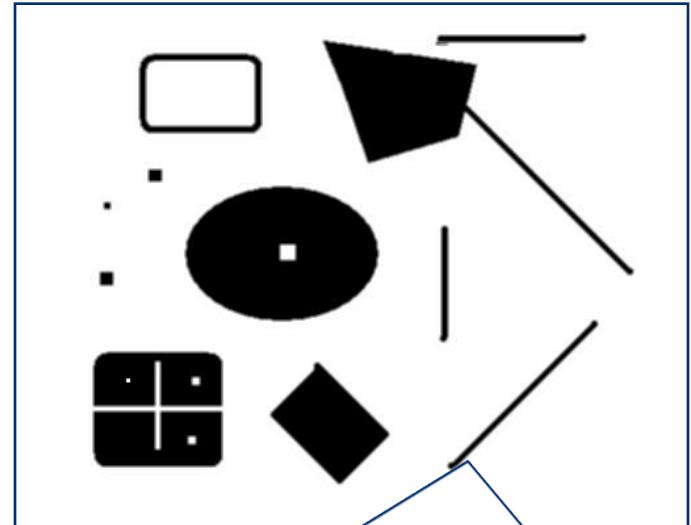


- descrierea regiunilor ocupate de trăsăturile locale trebuie să furnizeze invariантă la transformările geometrice și fotometrice dintre cele două imagini

# Trăsături locale: detectare

- trebuie să dați un click pe un punct din imagine
- plecați apoi din fața calculatorului și reveniți după ce am deformat imaginea
- trebuie să dați click pe punctul pe care l-ați ales inițial (îi găsiți corespondentul)
- ce puncte ați alege?

imaginea inițială



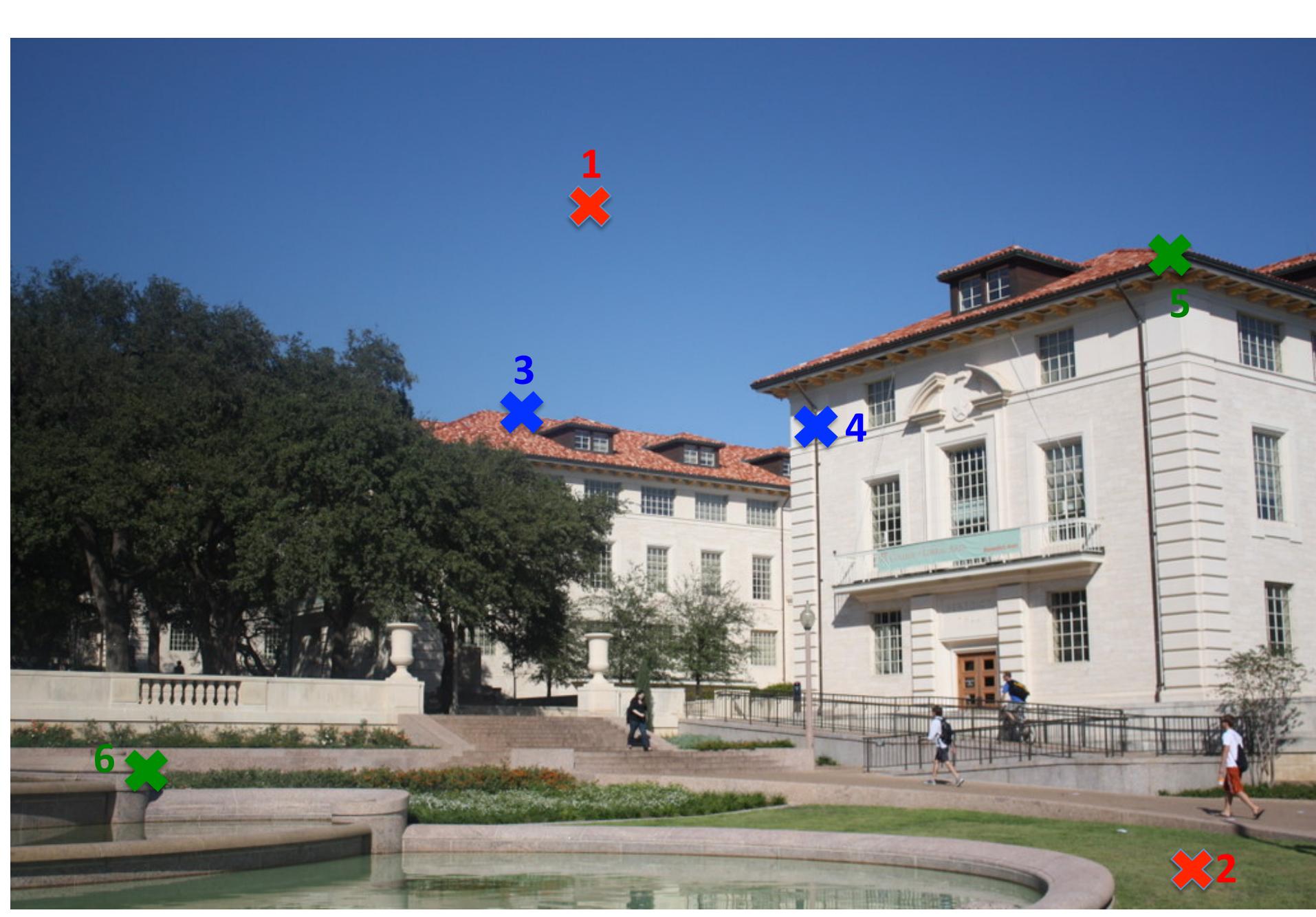
imaginea deformată

# Trăsături locale: detectare

**Detectare:** localizează punctele de interes în imagine

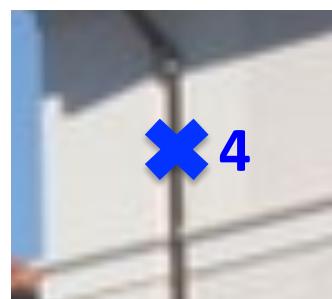
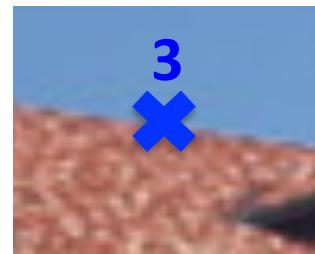
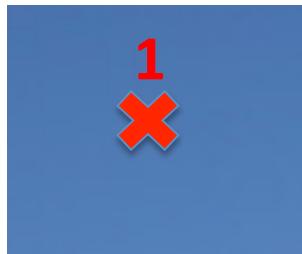


Ce puncte ați alege?



# Trăsături locale: detectare

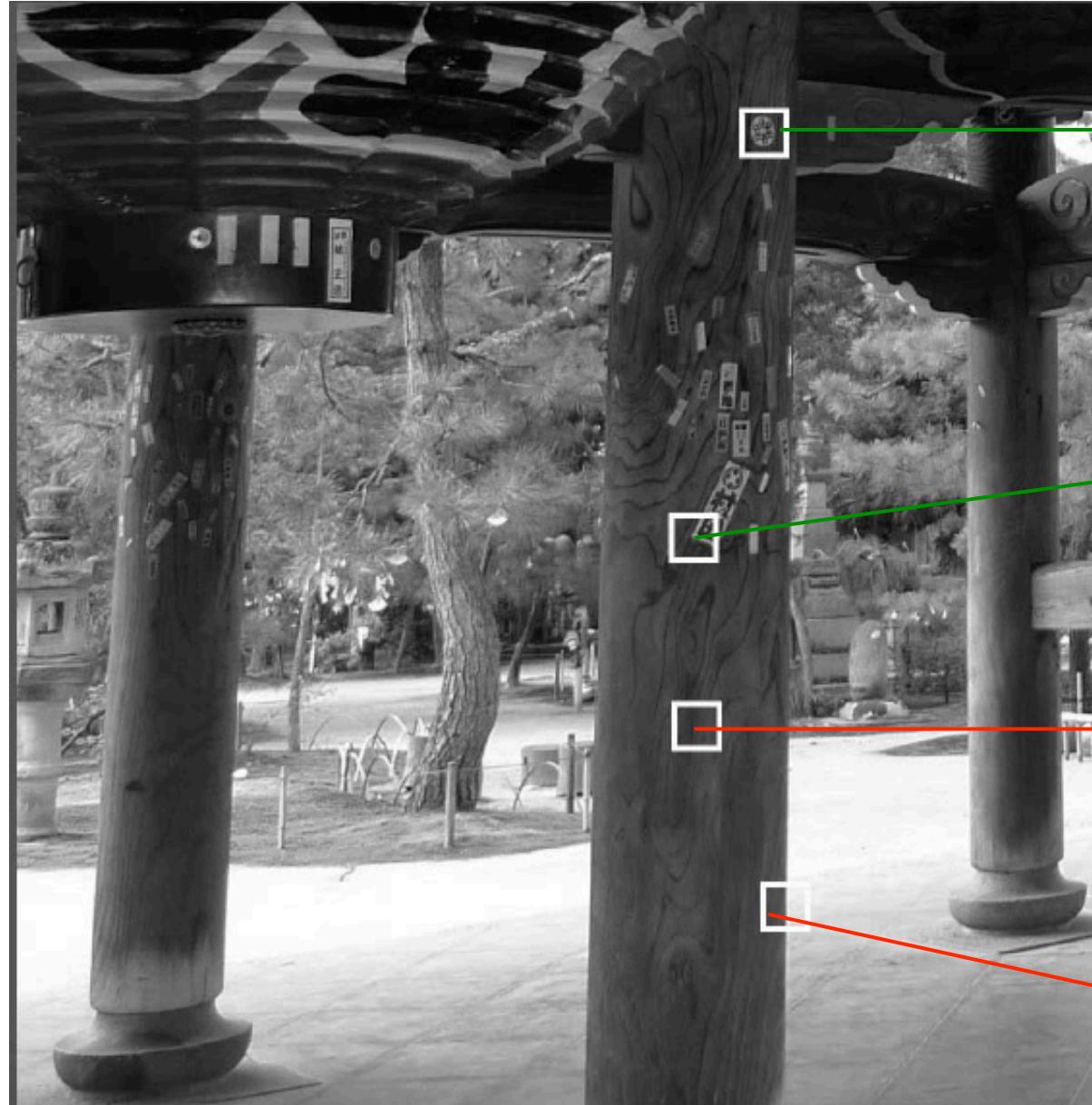
**Detectare:** localizează punctele de interes în imagine



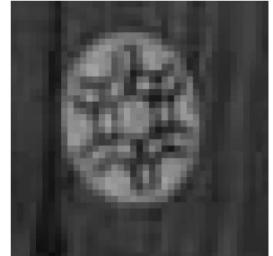
Regiuni uniforme  
ambigüe pentru  
matching

Muchii  
ambigüe pentru  
matching

Colțuri  
bune pentru  
matching



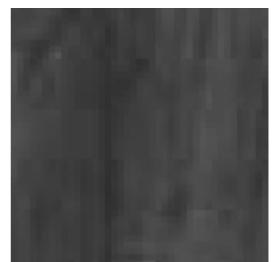
regiune  
circulară  
(blob)



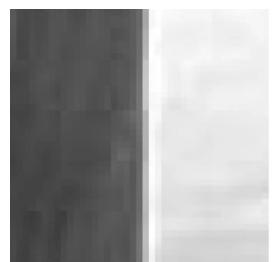
colț



regiune  
uniformă

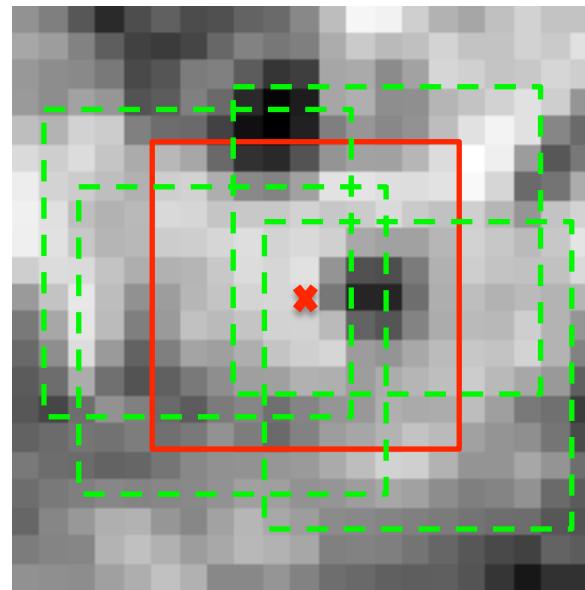


muchie



# Detectarea trăsăturilor locale

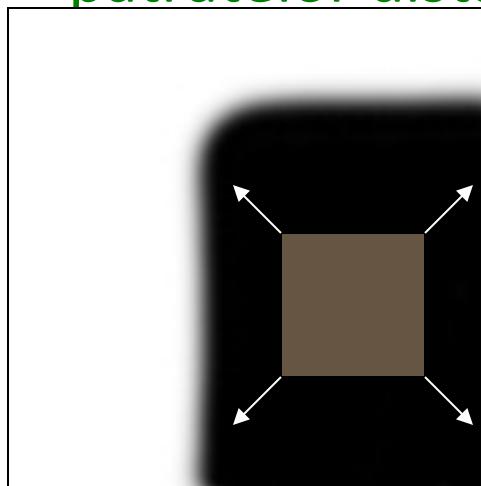
- cum formulăm matematic ce înseamnă o trăsătură locală bună?



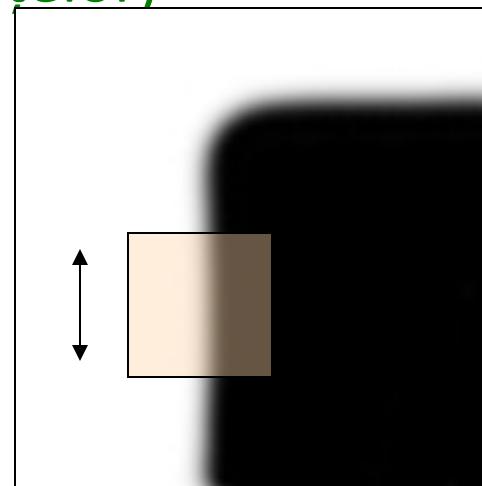
- caracterizăm punctul din mijloc cu o fereastră mică din jurul lui
- deplasarea ferestrei ***în orice direcție*** ar trebui să ducă la diferențe mari în intensitate (suma pătratelor distanțelor)

# Detectarea colțurilor

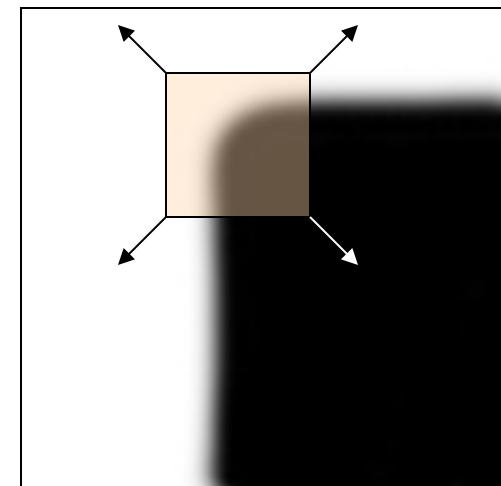
- caracterizăm punctul din mijloc cu o fereastră mică din jurul lui
- deplasare ferestrei *în orice direcție* ar trebui să ducă la diferențe mari în intensitate (suma pătratelor distantele)



regiune “flat” :  
nicio schimbare  
în toate direcțiile



“muchie”:  
nicio schimbare în  
direcția muchiei



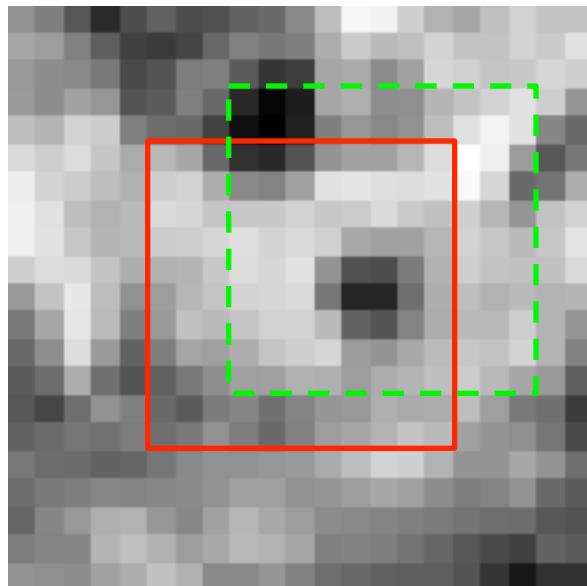
“colț”:  
schimbări  
semnificative în  
toate direcțiile

# Detectarea colțurilor

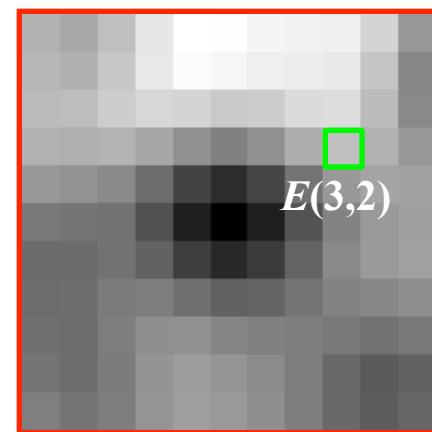
Schimbarea în înfățișare a ferestrei  $W$  la deplasarea  $[u,v]$ :

$$E(u, v) = \sum_{(x,y) \in W} (I(x + u, y + v) - I(x, y))^2$$

$I(x, y)$



$E(u, v)$



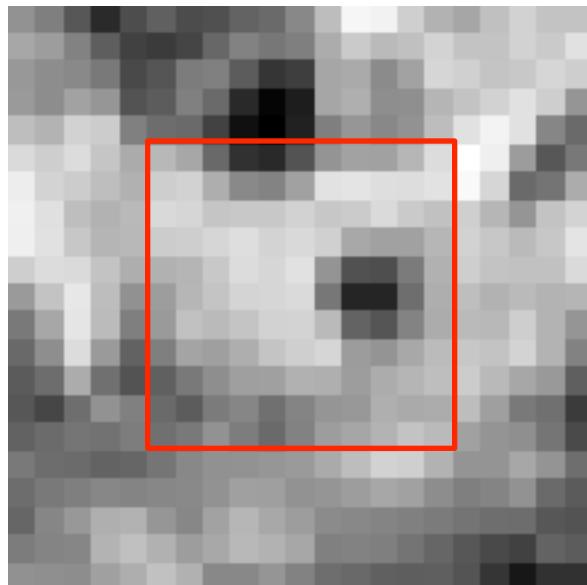
$E(3,2)$

# Detectarea colțurilor

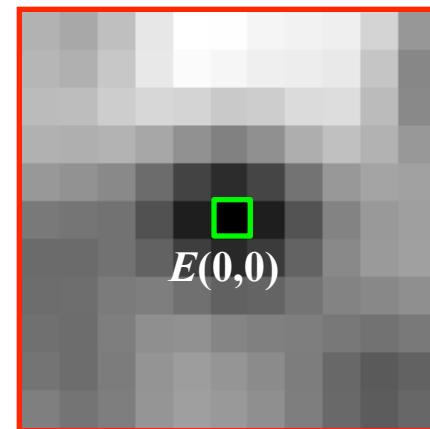
Schimbarea în înfățișare a ferestrei  $W$  la deplasarea  $[u,v]$ :

$$E(u, v) = \sum_{(x,y) \in W} (I(x + u, y + v) - I(x, y))^2$$

$$I(x, y)$$



$$E(u, v)$$



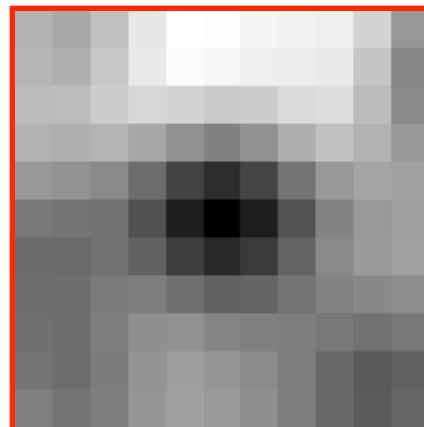
# Detectarea colțurilor

Schimbarea în înfățișare a ferestrei  $W$  la deplasarea  $[u,v]$ :

$$E(u, v) = \sum_{(x,y) \in W} (I(x + u, y + v) - I(x, y))^2$$

Vrem să studiem comportamentul funcției  $E(u,v)$  pentru mici deplasări  $[u,v]$

$$E(u, v)$$



# Detectarea colțurilor

Dezvoltarea în serie Taylor a funcțiilor de două variabile  $I(x+u,y+v)$  și  $E(u,v)$  în jurul lui  $(u,v) = (0,0)$

$$I(x + u, y + v) \approx I(x, y) + u \cdot \frac{\partial I}{\partial x}(x, y) + v \cdot \frac{\partial I}{\partial y}(x, y)$$

$$I(x + u, y + v) \approx I(x, y) + u \cdot I_x + v \cdot I_y$$

Înlocuim în expresia funcției  $E(u,v)$  pentru mici deplasări  $[u,v]$

$$E(u, v) = \sum_{(x,y) \in W} (I(x + u, y + v) - I(x, y))^2$$

$$E(u, v) \approx \sum_{(x,y) \in W} (u \cdot I_x + v \cdot I_y)^2 = \sum_{(x,y) \in W} u^2 \cdot I_x^2 + 2uv \cdot I_x \cdot I_y + v^2 \cdot I_y^2$$

# Detectarea colțurilor

$$E(u, v) \approx \sum_{(x,y) \in W} (u \cdot I_x + v \cdot I_y)^2 = \sum_{(x,y) \in W} u^2 \cdot I_x^2 + 2uv \cdot I_x \cdot I_y + v^2 \cdot I_y^2$$

$E(u,v)$  poate fi scris ca:

$$E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

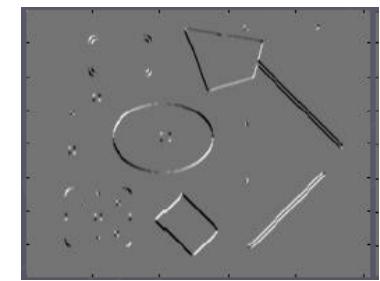
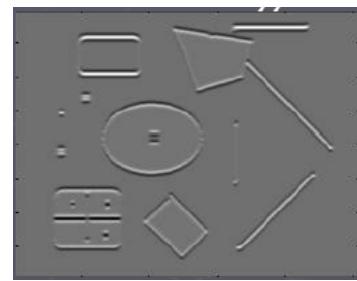
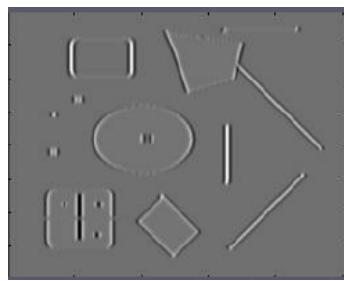
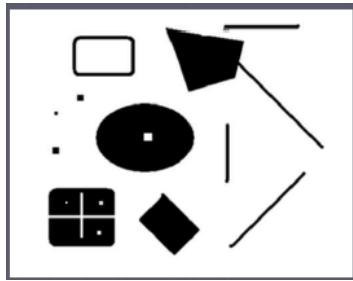
unde  $M$  este *matricea de ordin doi cu derivatele imaginii  $I_x$  și  $I_y$*

$$M = \begin{bmatrix} \sum_{x,y} I_x I_x & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y I_y \end{bmatrix}$$

# Detectarea colțurilor

$$M = \begin{bmatrix} \sum_{x,y} I_x I_x & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y I_y \end{bmatrix}$$

*M este matricea de ordin doi cu derivele imaginii  $I_x$  și  $I_y$  în vecinătatea unui punct*



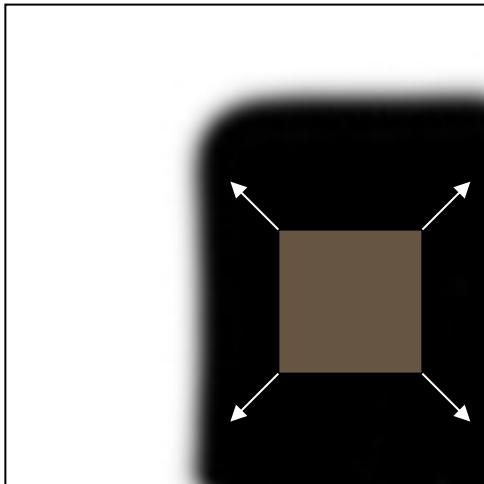
$$I_x \Leftrightarrow \frac{\partial I}{\partial x}$$

$$I_y \Leftrightarrow \frac{\partial I}{\partial y}$$

$$I_x I_y \Leftrightarrow \frac{\partial I}{\partial x} \frac{\partial I}{\partial y}$$

# Ce ne spune matricea M?

Cazul în care muchiile sunt paralele cu axele imaginii



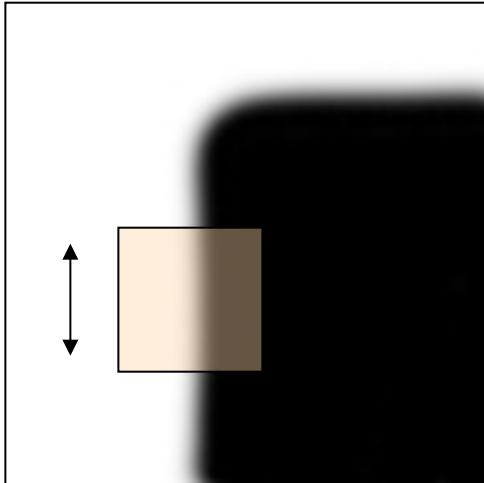
$$M = \begin{bmatrix} \sum_{x,y} I_x I_x & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y I_y \end{bmatrix}$$

regiune “flat” :  
nicio schimbare  
în toate direcțiile

$$M \approx \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Valorile proprii ale lui  $M = \text{soluțiile ecuației } \det(M - \lambda I_2) = 0$  sunt  $\lambda_1 \approx \lambda_2 \approx 0$

# Ce ne spune matricea M?



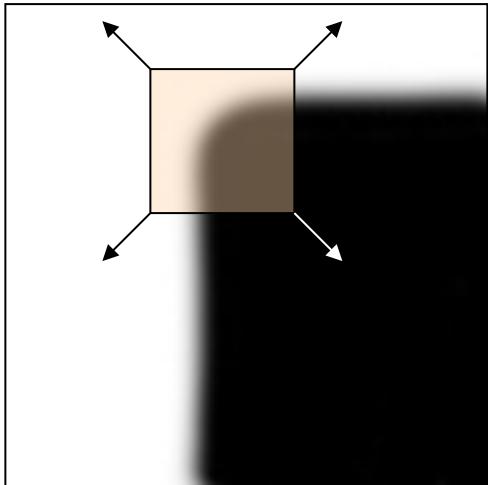
$$M = \begin{bmatrix} \sum_{x,y} I_x I_x & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y I_y \end{bmatrix}$$

“muchie”:  
nicio schimbare în  
direcția muchiei

$$M \approx \begin{bmatrix} \lambda & 0 \\ 0 & 0 \end{bmatrix}$$

Valorile proprii ale lui  $M = \text{soluțiile ecuației } \det(M - \lambda I_2) = 0$  sunt  $\lambda_1 \gg 0, \lambda_2 \approx 0$

# Ce ne spune matricea M?



$$M = \begin{bmatrix} \sum_{x,y} I_x I_x & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y I_y \end{bmatrix}$$

“colț” aliniat cu axele:  
schimbări semnificative  
în toate direcțiile

$$M \approx \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Valorile proprii ale lui  $M = \text{soluțiile ecuației } \det(M - \lambda I_2) = 0$  sunt  $\lambda_1, \lambda_2 \gg 0$

# Ce ne spune matricea M?

$$M = \begin{bmatrix} \sum_{x,y} I_x I_x & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y I_y \end{bmatrix} \quad M \approx \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

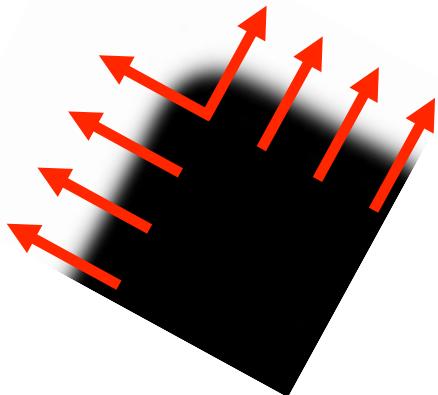
Valorile proprii ale lui  $M$  = soluțiile ecuației  $\det(M - \lambda I_2) = 0$  sunt  $\lambda_1, \lambda_2$

Caută punctele unde ambele valori proprii  $\lambda$  sunt mari.

Dacă o valoare proprie  $\lambda$  este aproape de 0, atunci punctul **NU** este un colț.

Ce se întâmplă dacă colțul nu este aliniat cu axele imaginii, Ox și Oy?

# Ce ne spune matricea M?



$$M = \begin{bmatrix} \sum_{x,y} I_x I_x & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y I_y \end{bmatrix}$$

Întrucât  $M$  e simetrică avem:

$$M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T \quad Mx_i = \lambda_i x_i$$

Valorile proprii ale lui  $M$  codifică cantitatea schimbării în intensitate în direcțiile date de vectorii proprii ai lui  $M$

# Vectori proprii și valori proprii

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Valorile proprii ale lui  $M$  = soluțiile ecuației  $\det(M - \lambda I_2) = 0$

$$\begin{aligned}|A - \lambda I| &= \left| \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = \begin{vmatrix} 2 - \lambda & 1 \\ 1 & 2 - \lambda \end{vmatrix}, \\ &= 3 - 4\lambda + \lambda^2.\end{aligned}$$

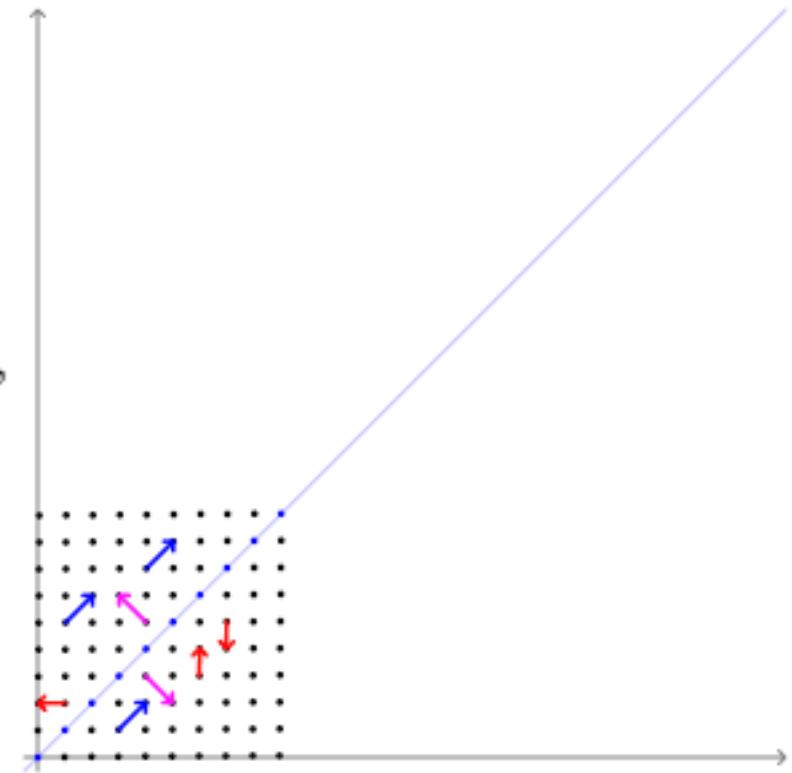
$$\lambda_1 = 1, \lambda_2 = 3$$

Vectorii proprii ai lui  $A$

$$v_{\lambda=1} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad v_{\lambda=3} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

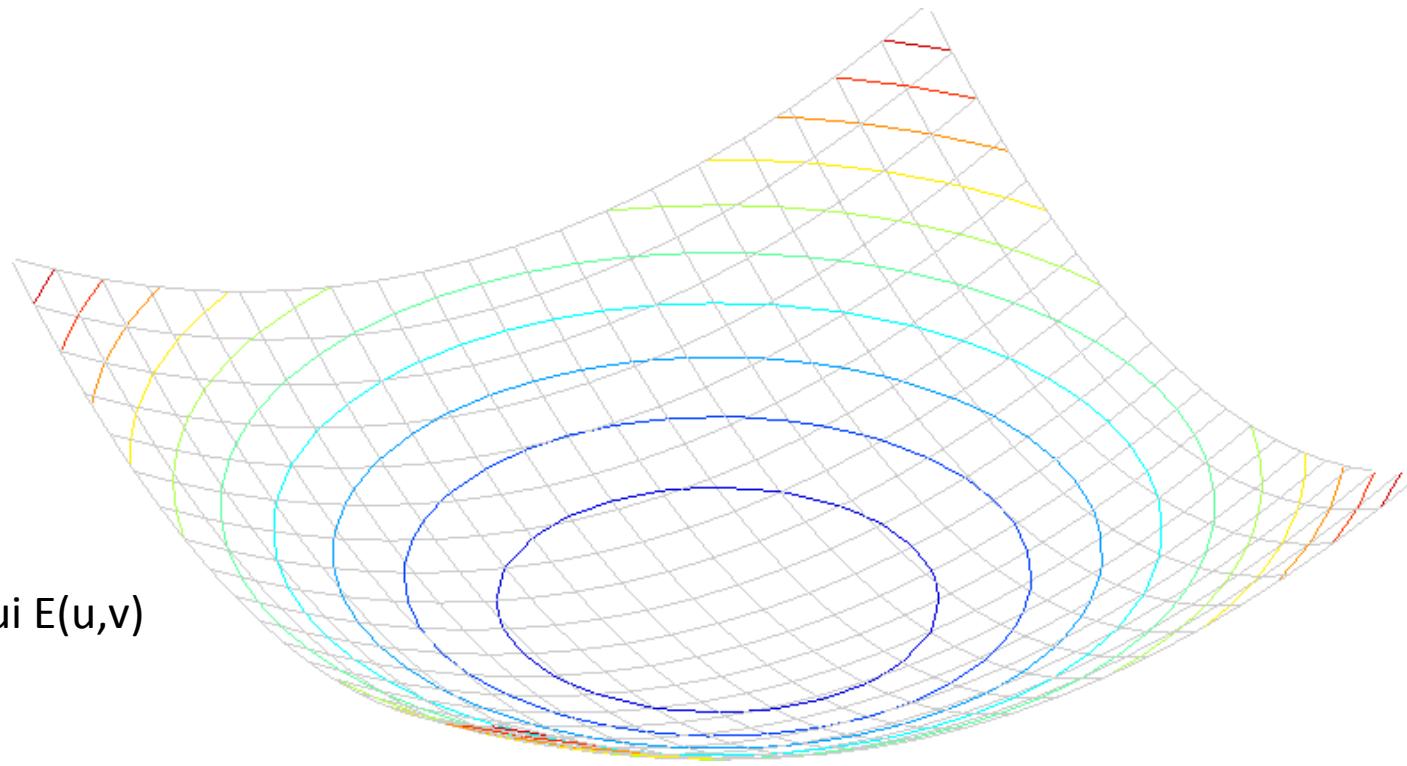
Vectorii proprii ai lui  $A$ :

- rămân paraleli după transformarea  $A$
- lungimea lor se modifică conform valorilor proprii



[https://en.wikipedia.org/wiki/Eigenvalues\\_and\\_eigenvectors](https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors)

# Interpretarea lui $E(u,v)$



Graficul lui  $E(u,v)$

Considerăm o secțiune orizontală al lui  $E(u, v)$ :  $[u \ v] M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

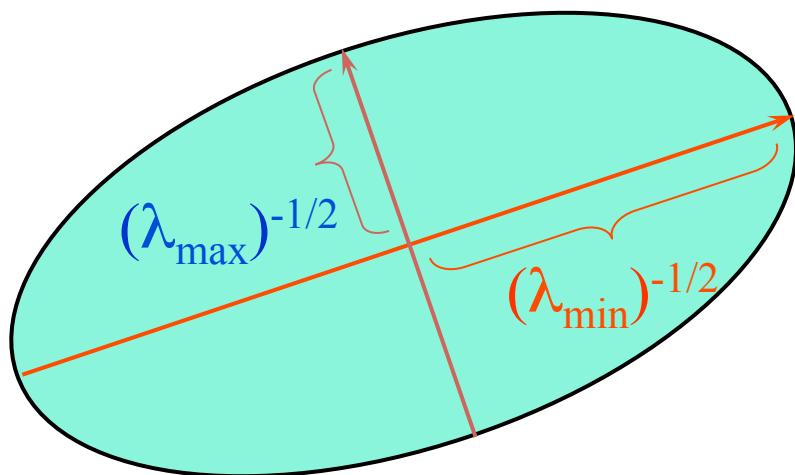
Obținem ecuația unei elipse

# Interpretarea lui $E(u,v)$

Considerăm o secțiune orizontală al ui  $E(u, v)$ :  $[u \ v] M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

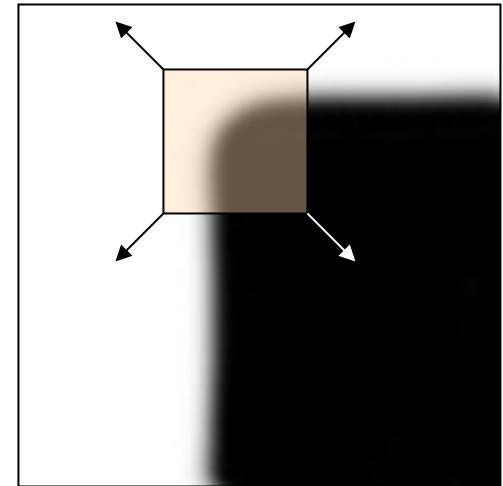
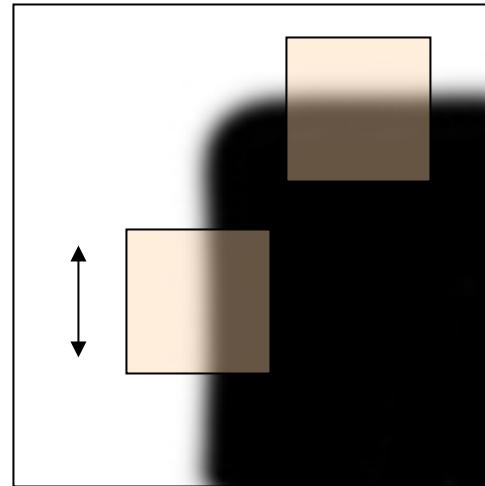
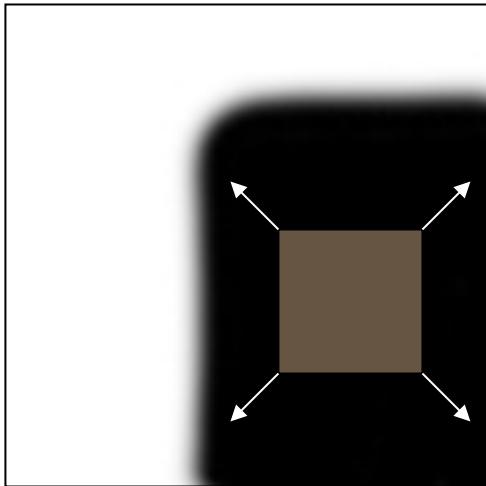
Obținem ecuația unei elipse

Lungimile axelor elipsei sunt determinate de valorile proprii, orientarea elipsei e dată de X



$$M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T$$

# Funcția cornerness



“uniform”

$\lambda_1$  și  $\lambda_2$  sunt mici;

“muchie”:

$\lambda_1 \gg \lambda_2$

$\lambda_2 \gg \lambda_1$

“colț”:

$\lambda_1$  și  $\lambda_2$  sunt mari,  
 $\lambda_1 \sim \lambda_2$ ;

Funcția cornerness R măsoară cât e mult un punct este “colț”.

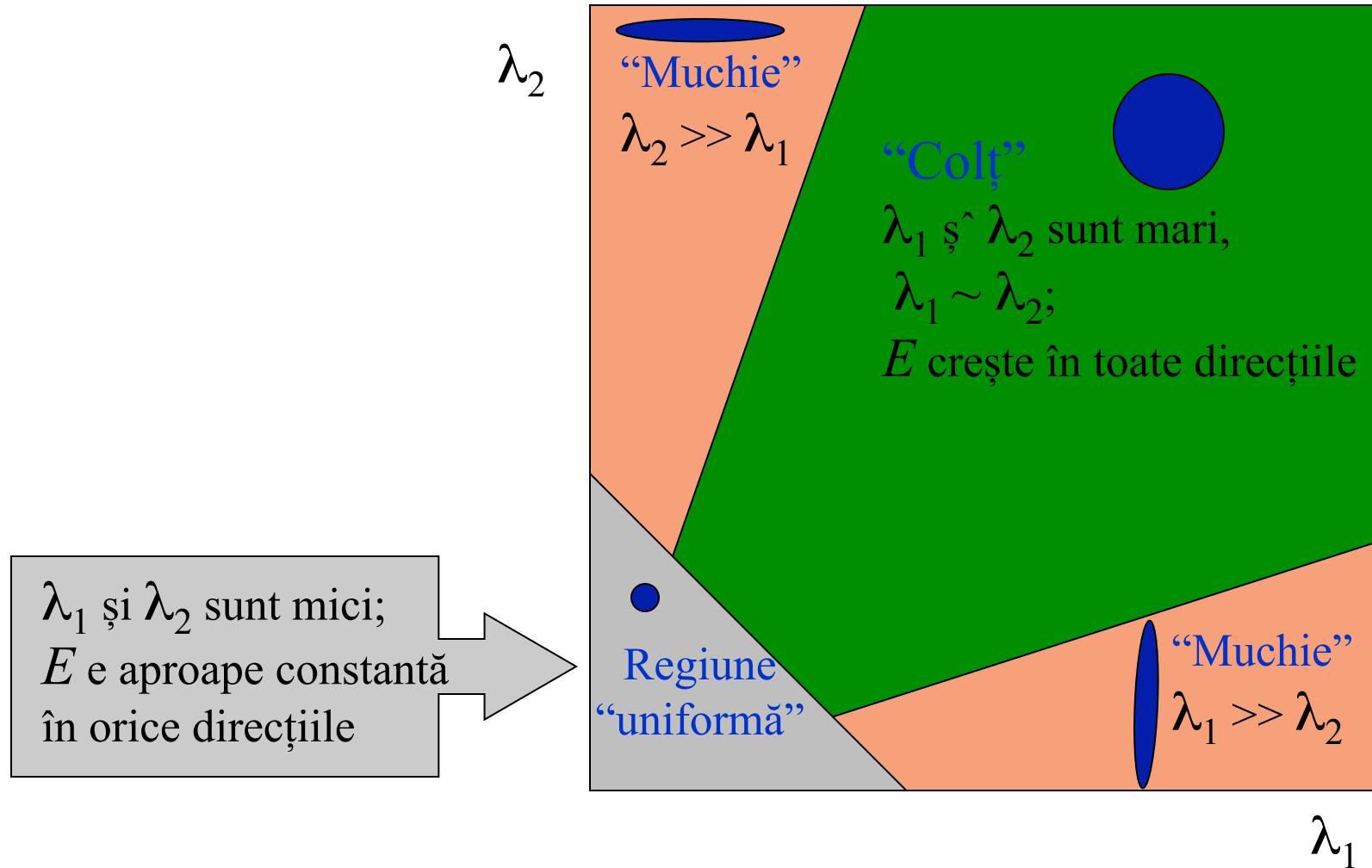
Posibile formulări:

$$R = \frac{\lambda_1 \cdot \lambda_2}{\lambda_1 + \lambda_1}$$

$$R = \det(M) - \alpha \cdot \text{trace}(M)^2 = \lambda_1 \cdot \lambda_2 - \alpha * (\lambda_1 + \lambda_2)^2$$

# Interpretarea valorilor proprii

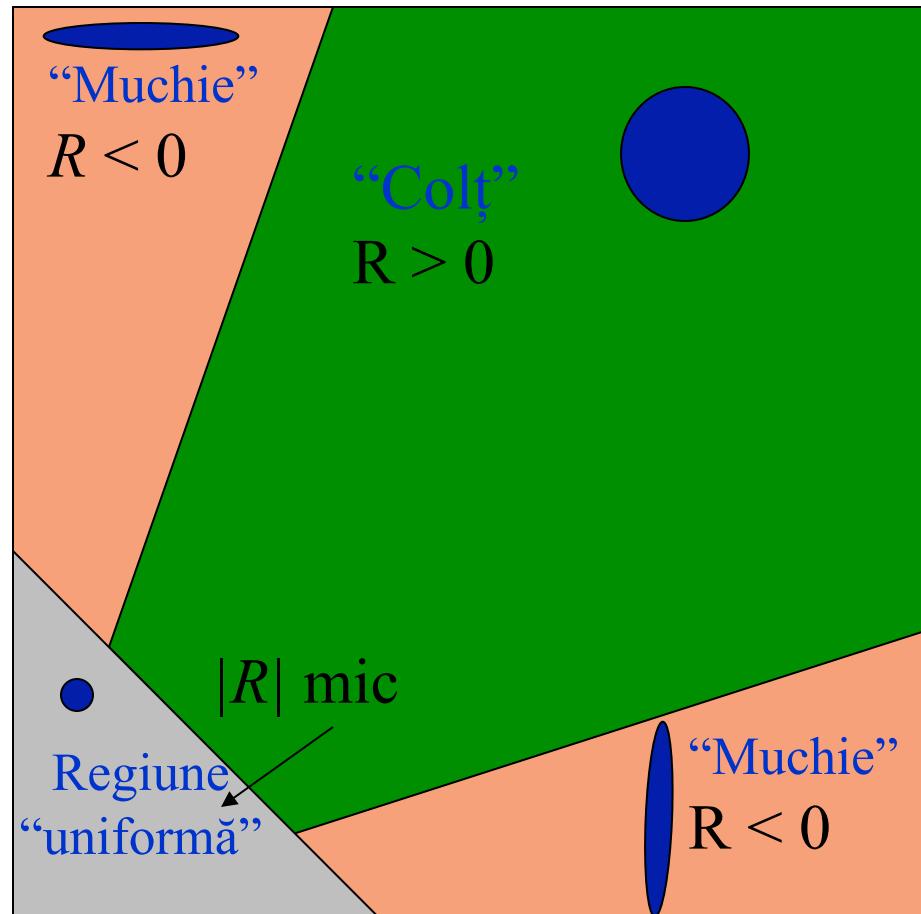
Clasificarea punctelor imaginii pe baza valorilor proprii ale lui  $M$ :



# Valori ale funcției cornerness R

$$R = \det(M) - \alpha \cdot \text{trace}(M)^2 = \lambda_1 \cdot \lambda_2 - \alpha * (\lambda_1 + \lambda_2)^2$$

$\alpha$ : constant (0.04 - 0.06)



# 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)

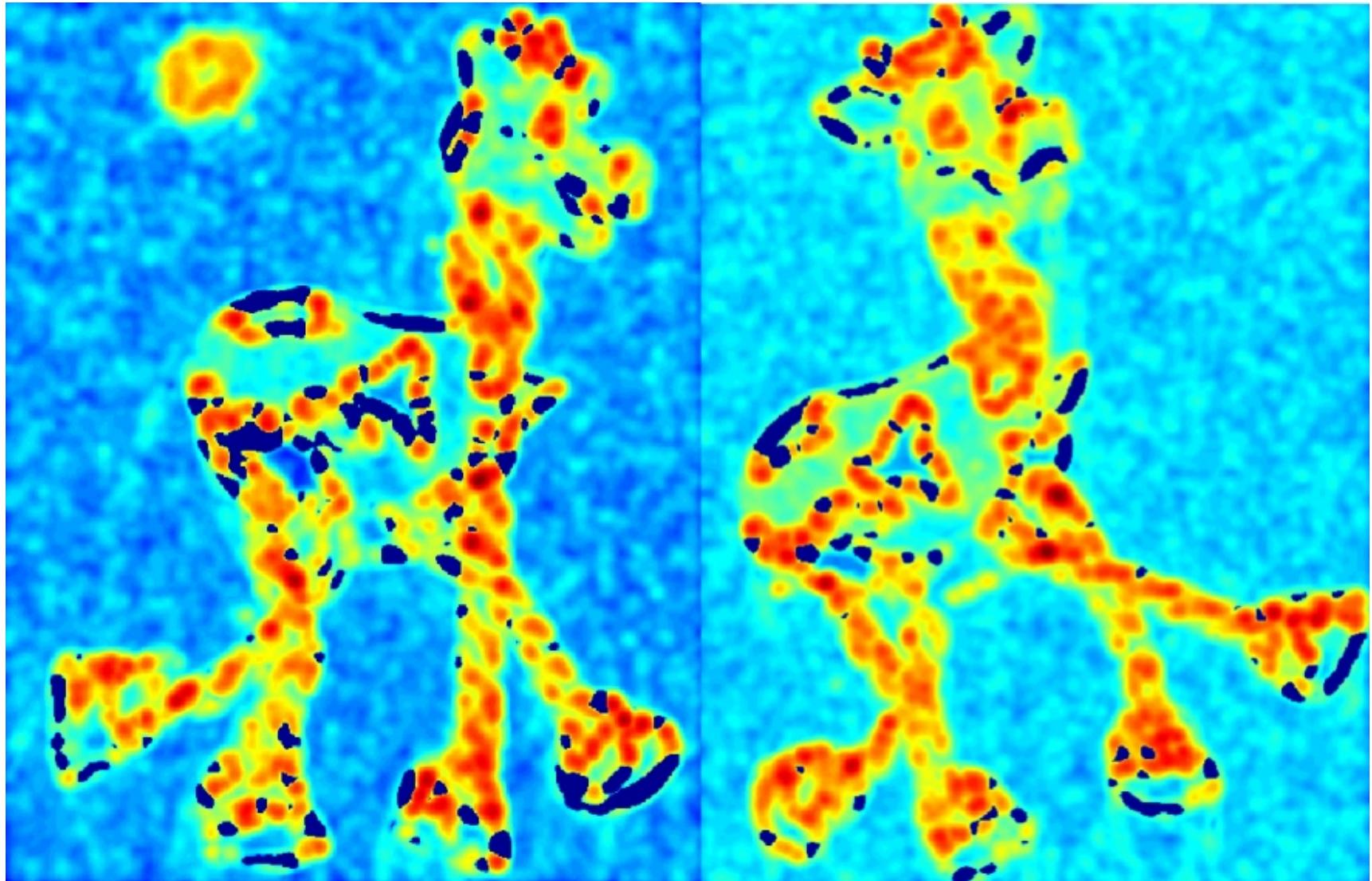
C.Harris and M.Stephens. ["A Combined Corner and Edge Detector."](#) *Proceedings of the 4th Alvey Vision Conference*: pages 147–151, 1988.

# Detectorul de colțuri Harris: pași



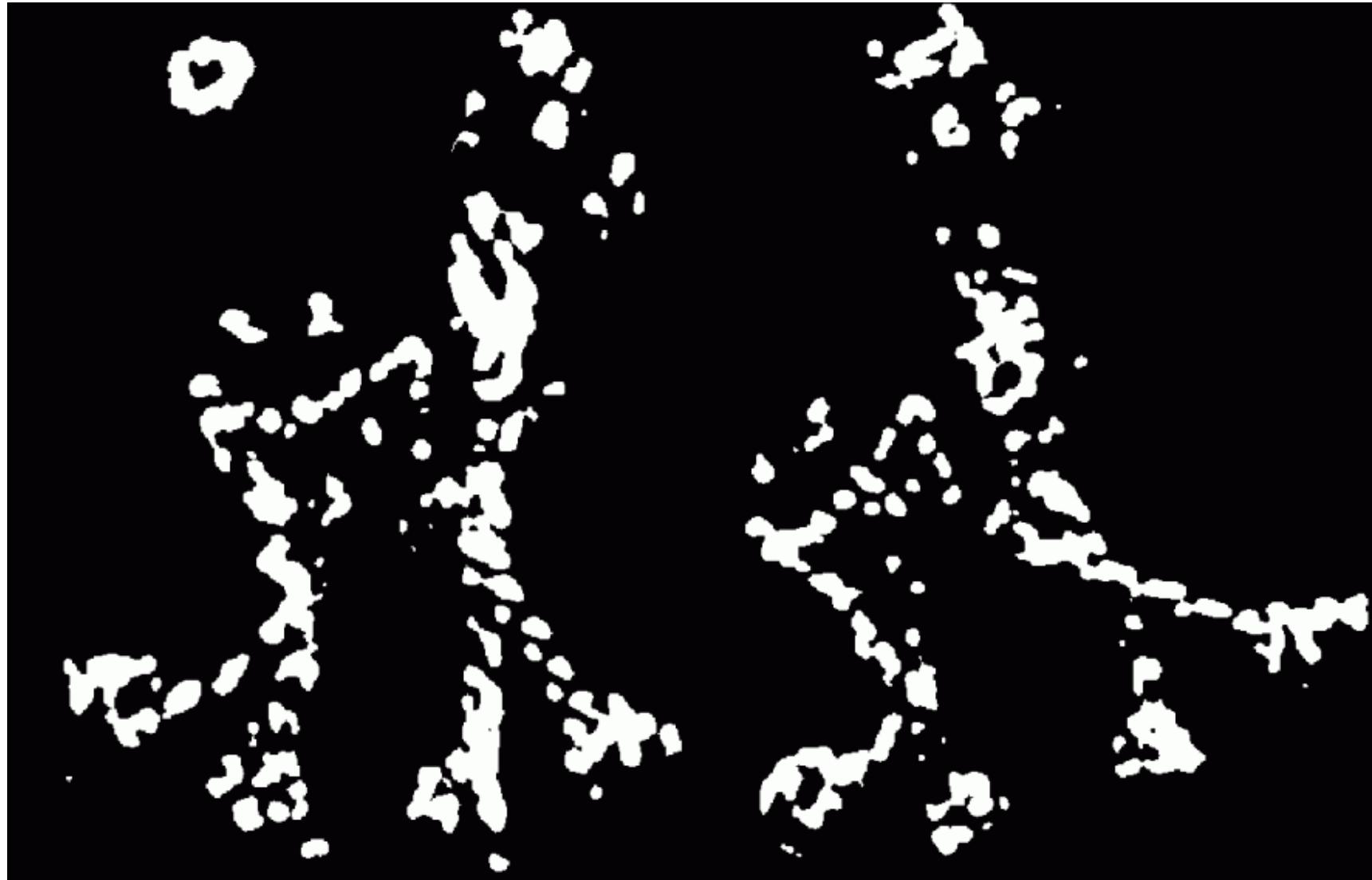
# Detectorul de colțuri Harris: pași

Calculul funcției cornerness R



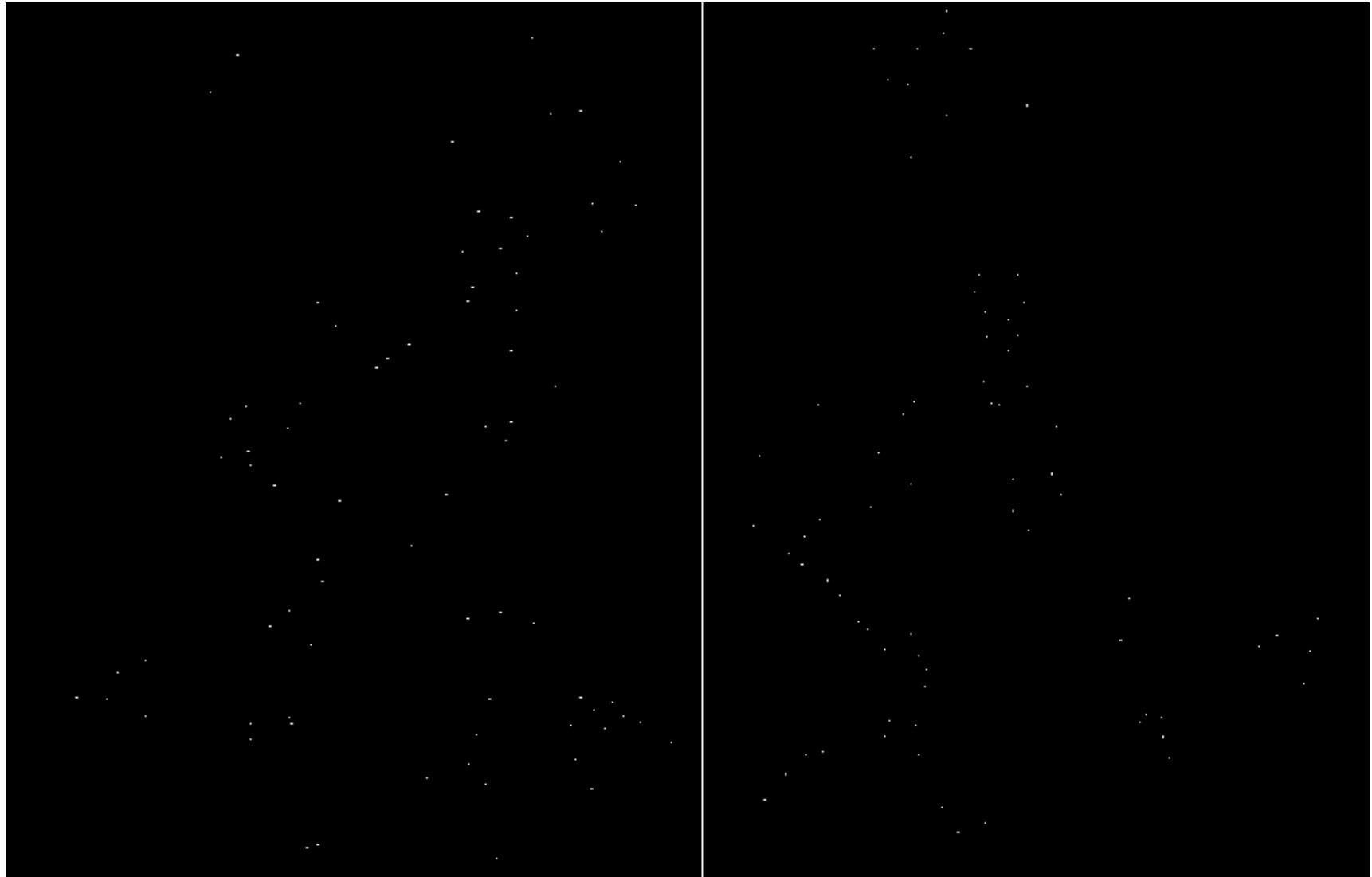
# Detectorul de colțuri Harris: pași

Găsește pixelii  $(x,y)$  din imagine cu  $R(x,y) > \text{threshold}$



# Detectorul de colțuri Harris: pași

Păstrează pixelii  $(x,y)$  din imagine cu  $R(x,y)$  maxim local

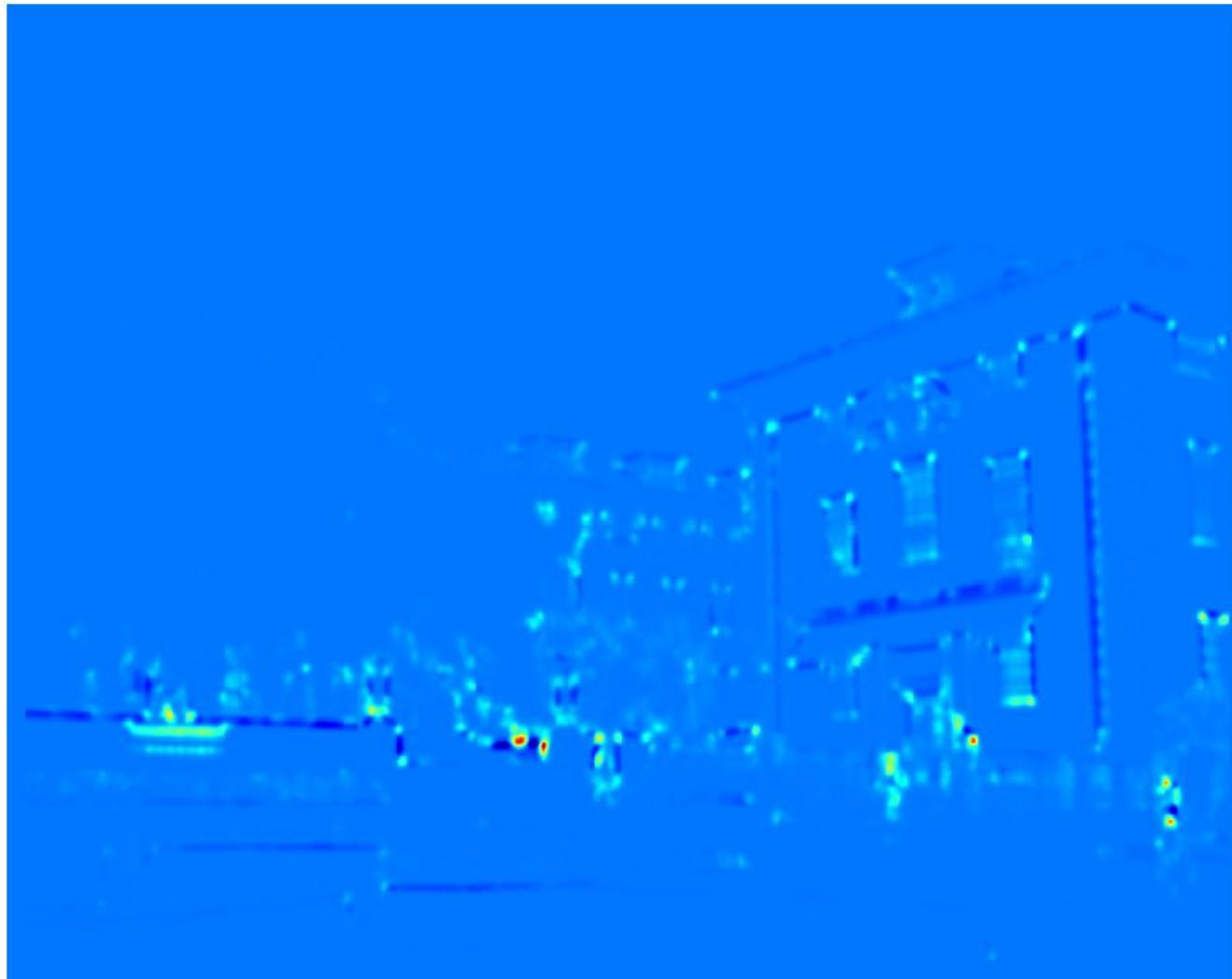


# Exemple pentru detectorul Harris



# Exemple pentru detectorul Harris

Calculul funcției cornerness R



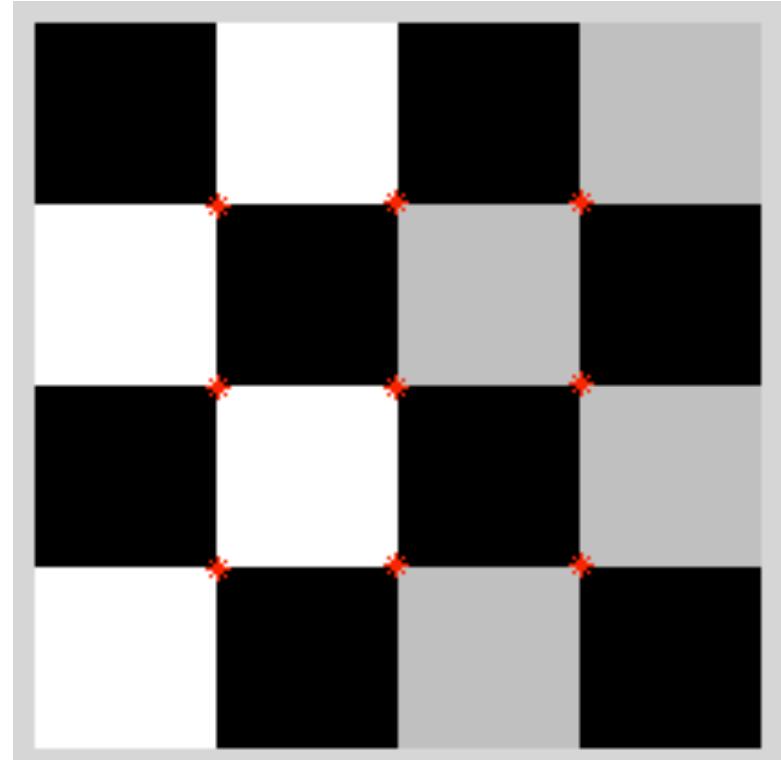
# Exemple pentru detectorul Harris



# Exemple pentru detectorul Harris

Matlab: functia `corner`

```
I = checkerboard(50,2,2);
corners = corner(I);
imshow(I);
hold on
plot(corners(:,1),corners(:,2),'r*');
```



# Invariantă și covariantă

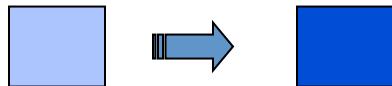
Vrem să găsim pixeli din imagini care sunt colțuri care să fie **invariante** la transformări fotometrice și **covariante** la transformări geometrice

**Invariantă:** imaginea e transformată fotometric și poziția colțului nu se schimbă

**Covariantă:** dacă avem două versiuni ale aceleasi imagini obținute prin transformări geometrice (translație, rotație, transformare afină, perspectivă), colțurile ar trebui detectate în poziții corespondente.

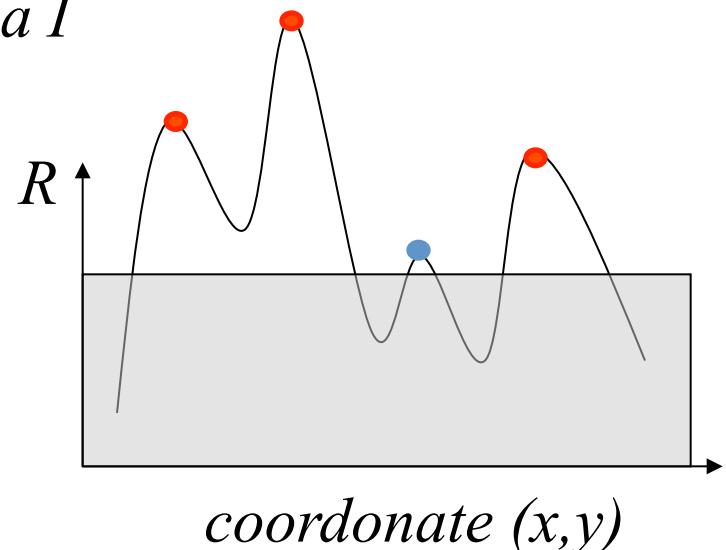
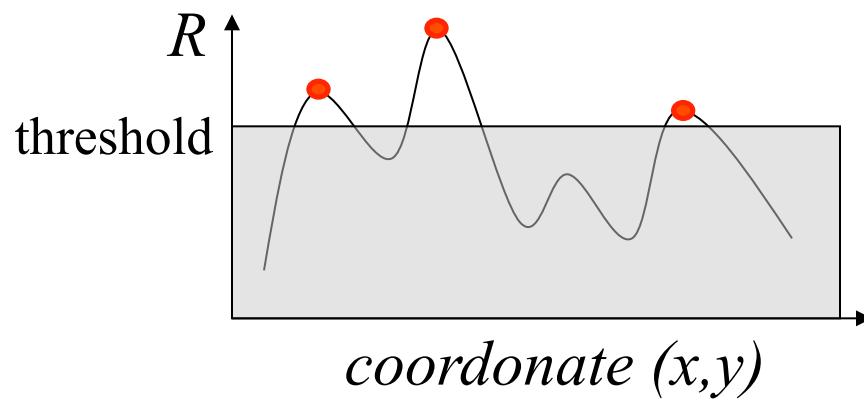


# Transformări fotometrice: schimbări afine de intensitate



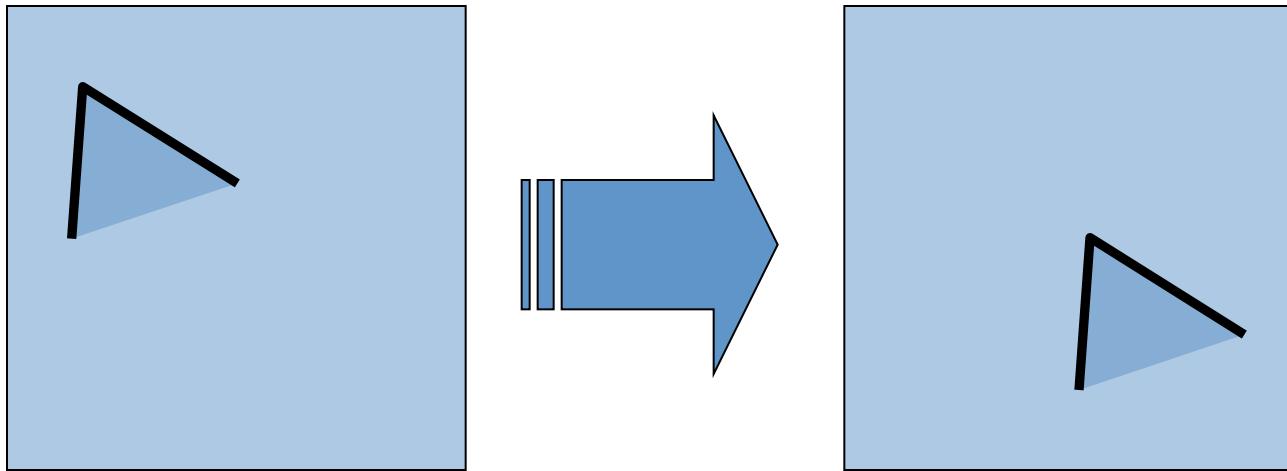
$$I \rightarrow a I + b$$

- folosim numai derivate => invariантă la adunare/scădere în  $I \rightarrow I + b$
- înmulțire cu scalar:  $I \rightarrow a I$



*Parțial invariантă la schimbări affine de intensitate*

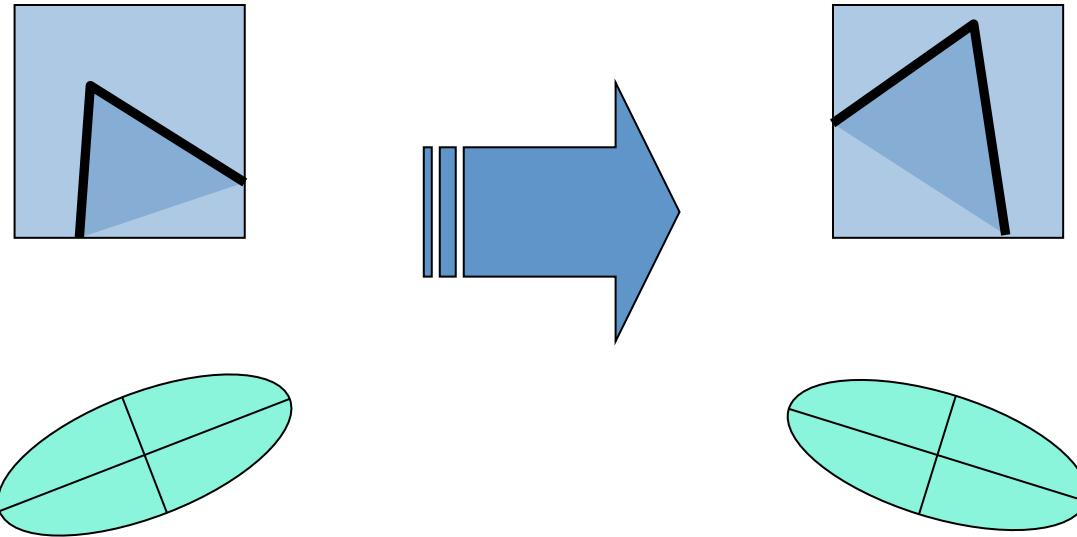
# Transformări geometrice: translații



- derivatele sunt covariante la translație

Pozitia colțului este covariantă la translații

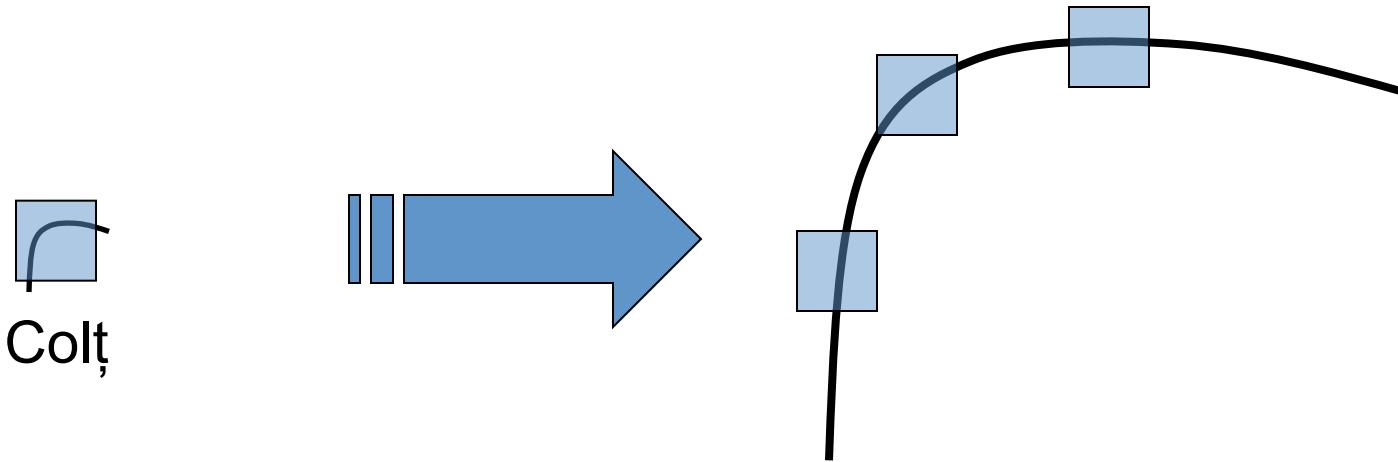
# Transformări geometrice: rotații



Elipsa determinată de  $M$  se rotește însă forma sa (dată de valorile proprii) rămâne aceeași

Pozitia colțului este covariantă la translații

# Transformări geometrice: scalare a imaginii (mărire, micsorare)



Toate punctele  
sunt clasificate ca  
fiind **muchii**

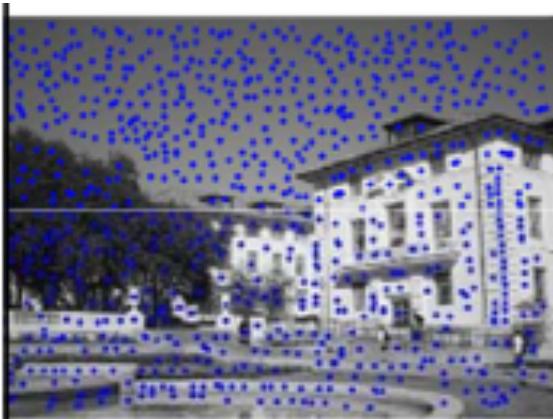
Pozitia colțului nu este covariantă la scalarea imaginilor

# Detectorul Harris

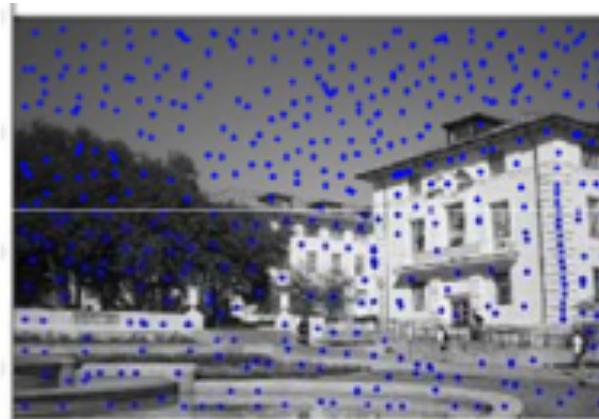
Detectorul Harris detectează colțuri la o singură scală = mărime = scale

Vrem să găsim un detector multi-scale, fiecare trăsătură locală detectată să aibă o scală caracteristică, covariantă cu zoom-ul într-o imagine.

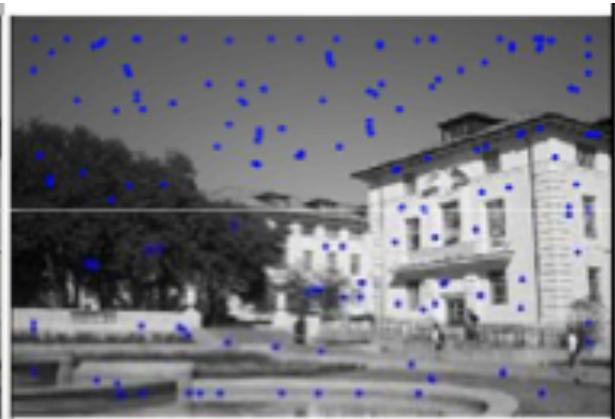
Găsirea scalei specifice conduce și la ajustarea dimensiunii ferestrei din jurul punctului detectat ca trăsătură



$W = 10 \times 10$  pixeli



$W = 15 \times 15$  pixeli



$W = 30 \times 30$  pixeli

# Detectarea punctelor de interes invariante la scală

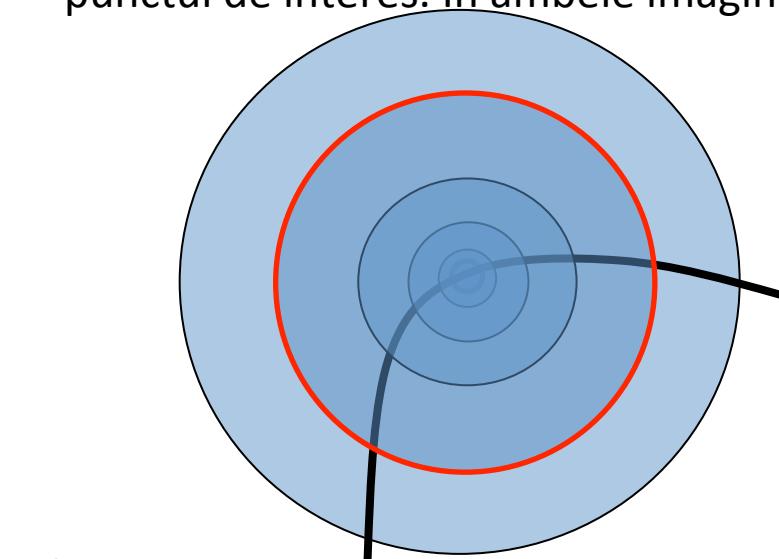
Cum putem detecta puncte de interes în fiecare imagine independent, astfel încât detecțiile se regăsesc chiar la scale diferite?



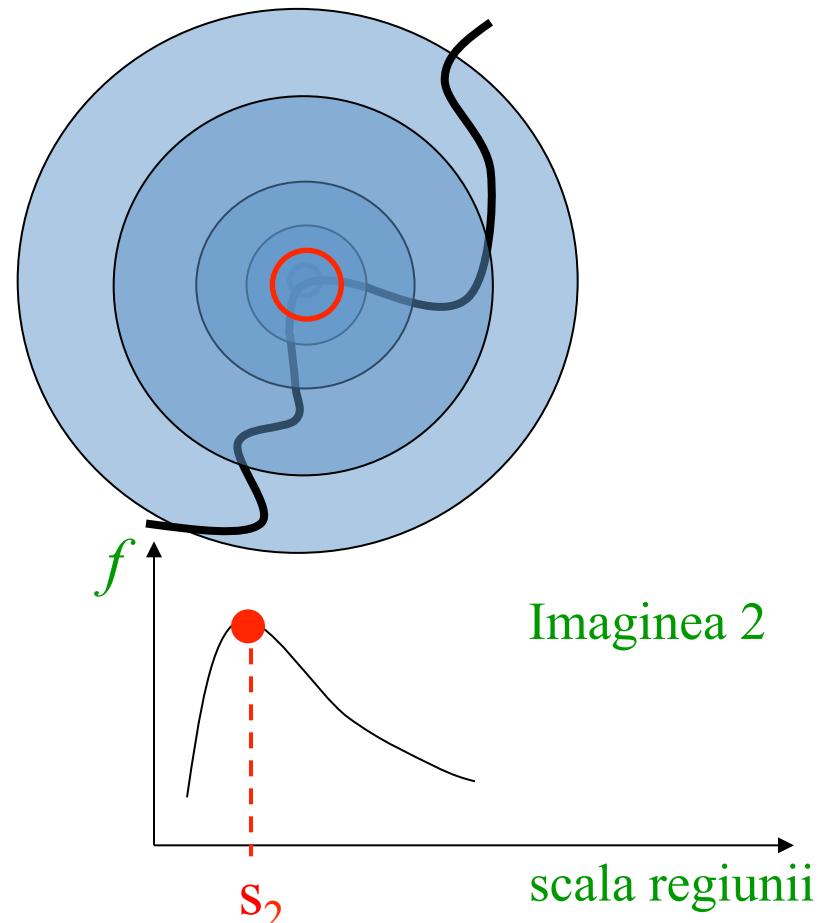
# Selectia automată a scalei

## 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

# 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  
 $\sigma, \sigma'$  – dimensiunile laturilor ferestrelor centrate în  $x, x'$

Cum să găsim scalele cele mai bune  $\sigma, \sigma'$  ?

# Selectia automată a scalei



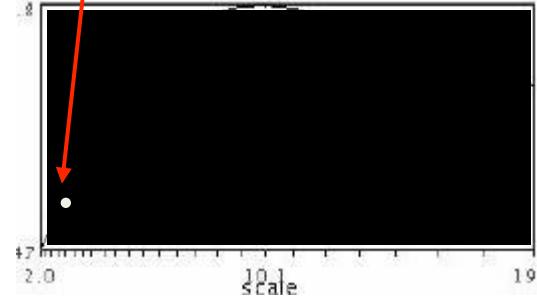
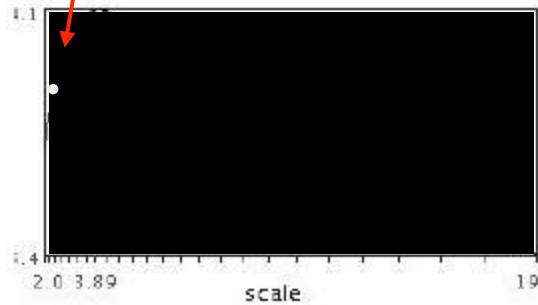
$$f(I_{i_1 \dots i_m}(x, \sigma)) = f(I_{i_1 \dots i_m}(x', \sigma'))$$

$x, x'$  – coordonatele punctelor de interes în cele două imagini  
 $\sigma, \sigma'$  – dimensiunile laturilor ferestrelor centrate în  $x, x'$

Cum să găsim scalele cele mai bune  $\sigma, \sigma'$  ?

# Selectia automată a scalei

- valoarea funcției  $f$  pentru fiecare scală

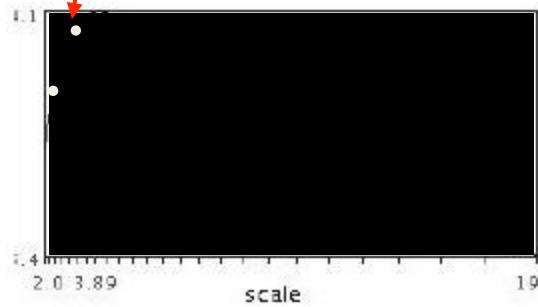


K. Grauman, B. Leibe

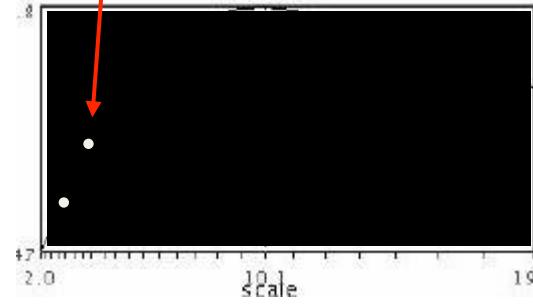
Slide adaptat după K. Graumann

# Selectia automată a scalei

- valoarea funcției  $f$  pentru fiecare scală



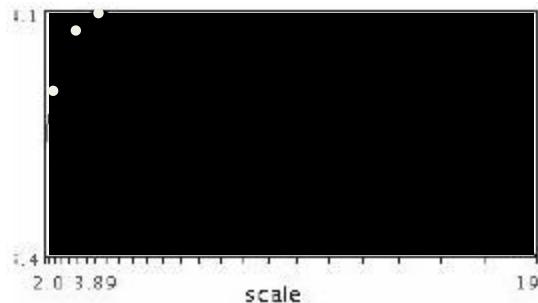
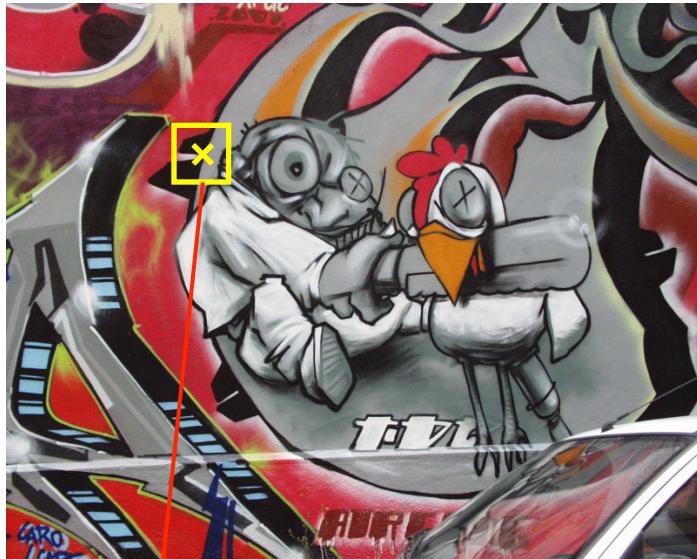
$$f(I_{i_1 \dots i_m}(x, \sigma))$$



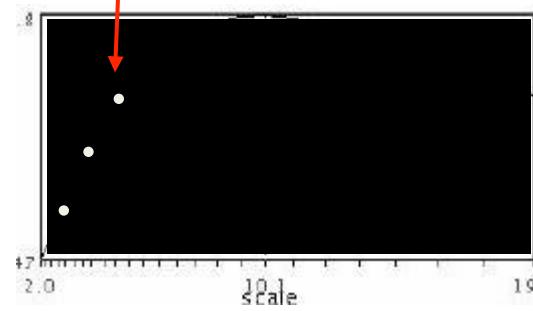
$$f(I_{i_1 \dots i_m}(x', \sigma))$$

# Selectia automată a scalei

- valoarea funcției  $f$  pentru fiecare scală



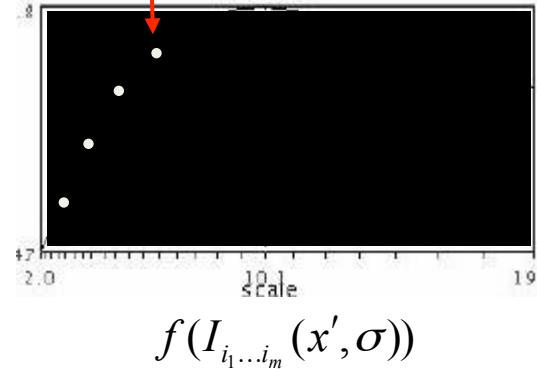
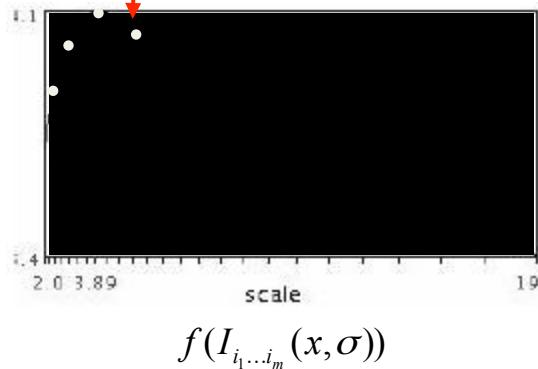
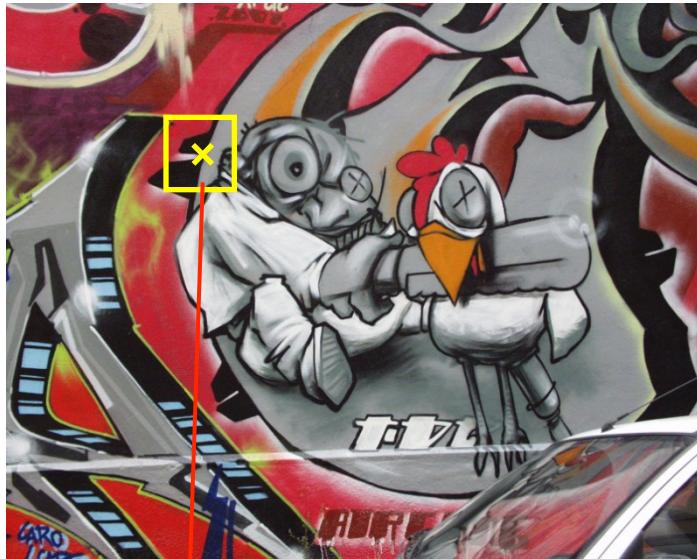
$$f(I_{i_1 \dots i_m}(x, \sigma))$$



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

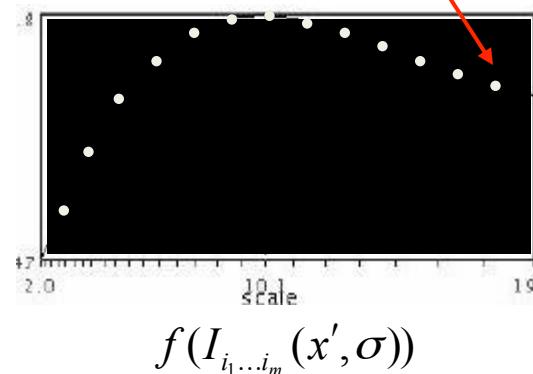
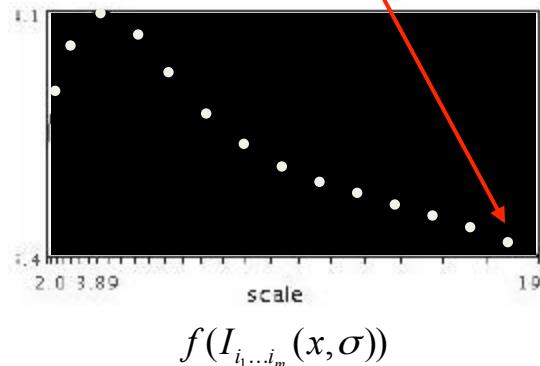
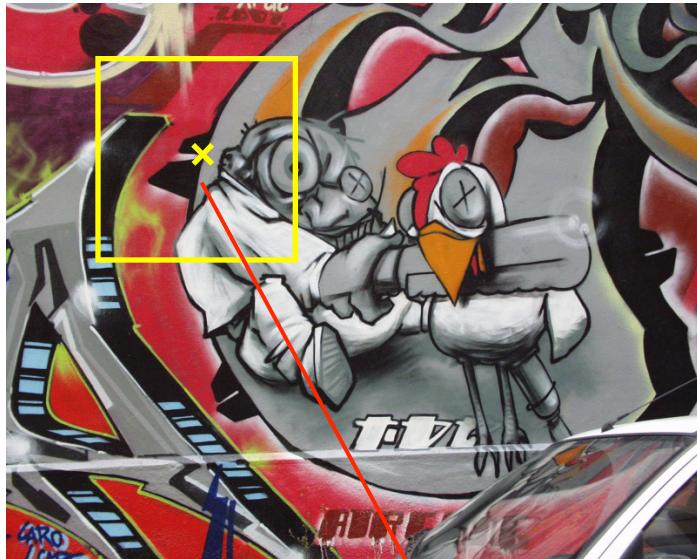
# Selectia automată a scalei

- valoarea funcției  $f$  pentru fiecare scală



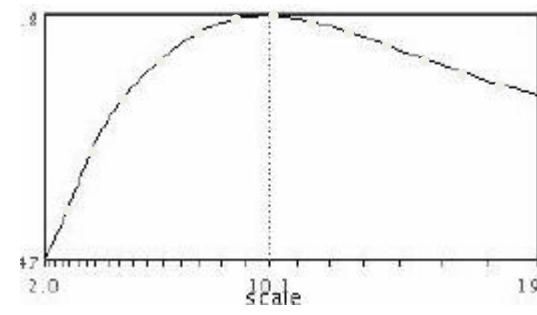
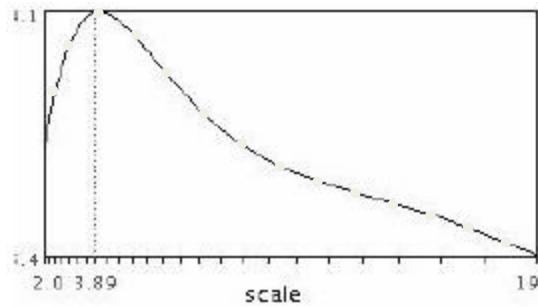
# Selectia automată a scalei

- valoarea funcției  $f$  pentru fiecare scală



# Selectia automată a scalei

- valoarea funcției  $f$  pentru fiecare scală

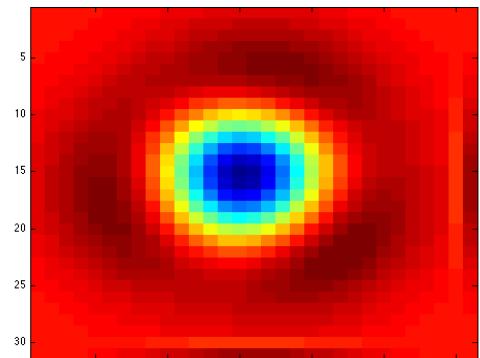
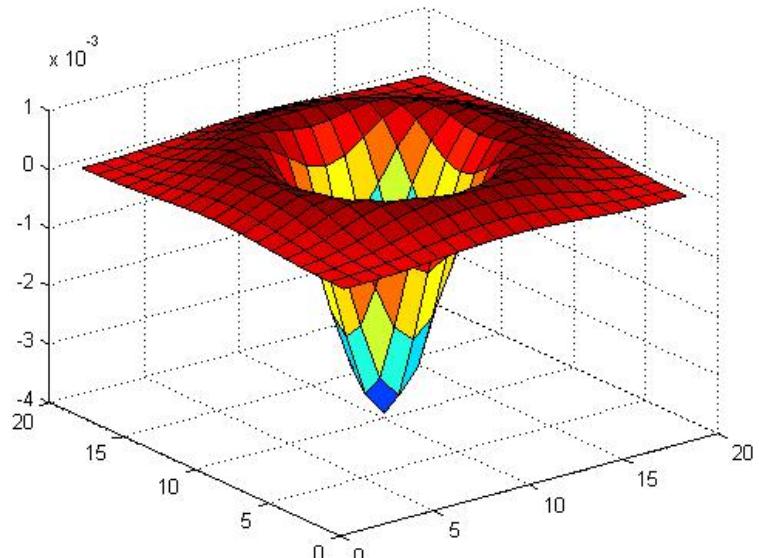


Cum găsim o astfel de funcție  $f$  cu proprietățile dorite?

Slide adaptat după K. Graumann

# Selectia automată a scalei

Folosim  $f = \text{filtrul Laplacian}$

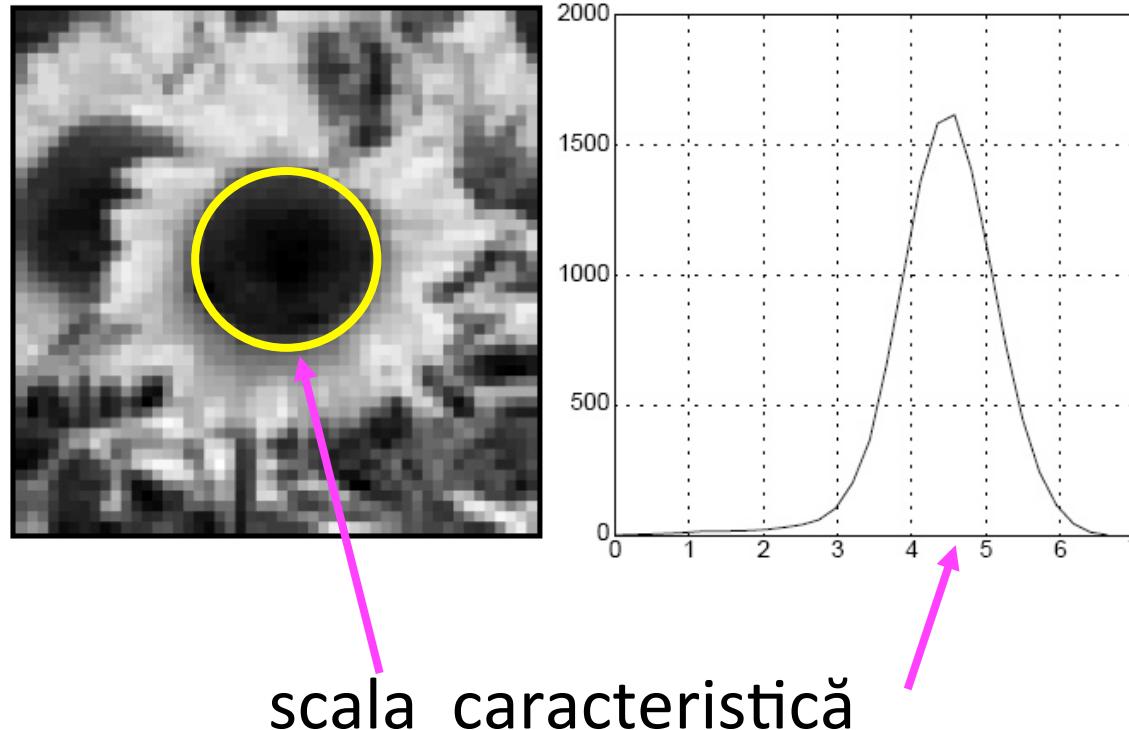


$$\nabla^2 h = \frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2}$$

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

# Scala caracteristică

- definim scala caracteristică a unui punct de interes ca fiind scala pentru care se obține valoarea maximă în imaginea filtrată cu filtrul Laplacian



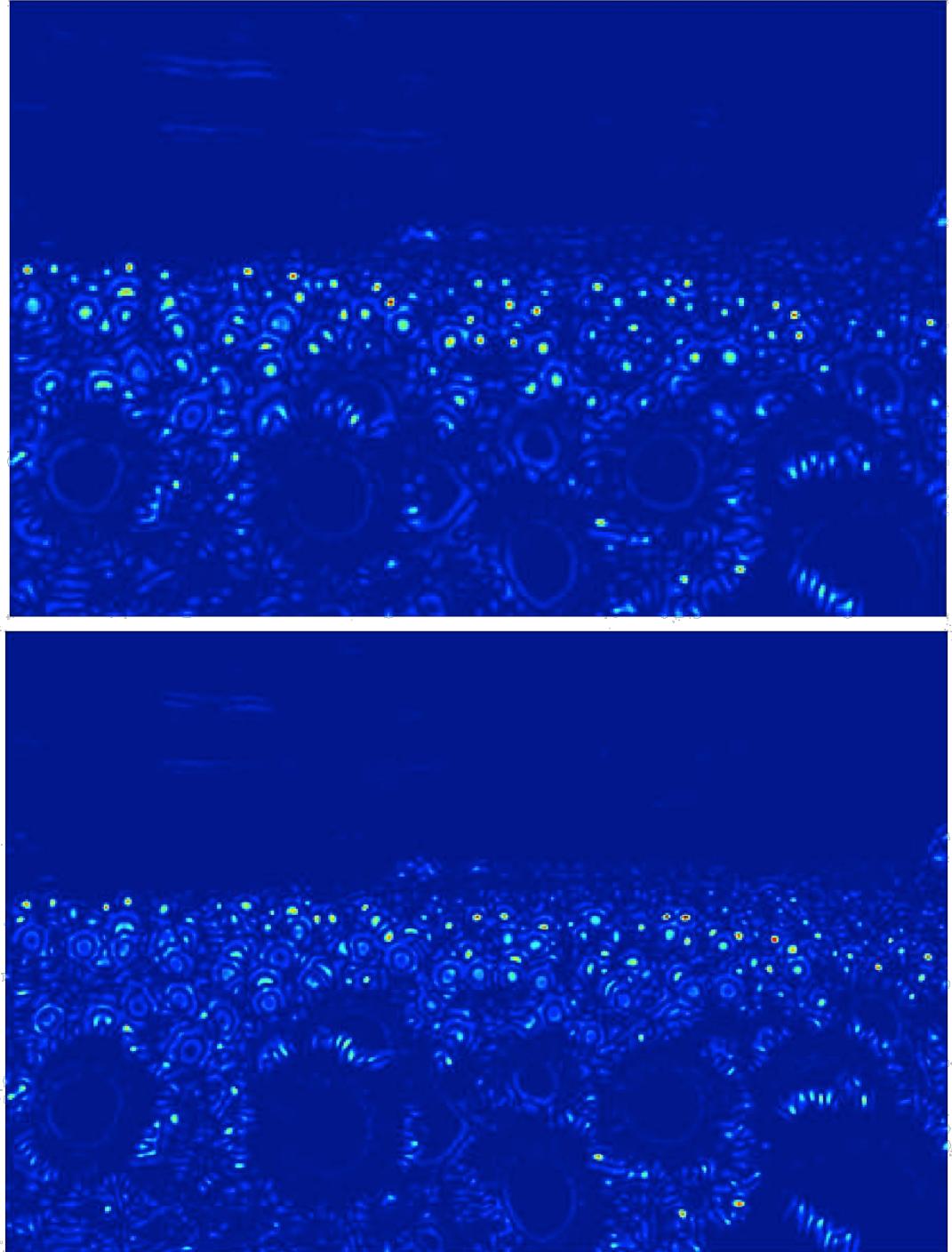
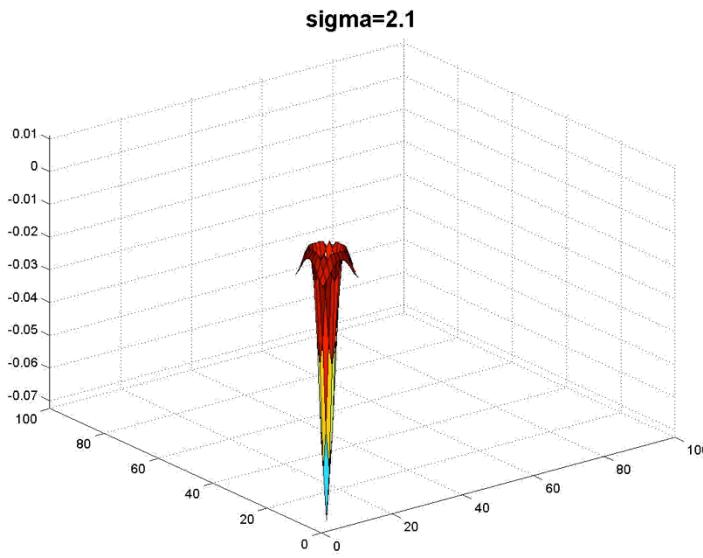
# Exemplu

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



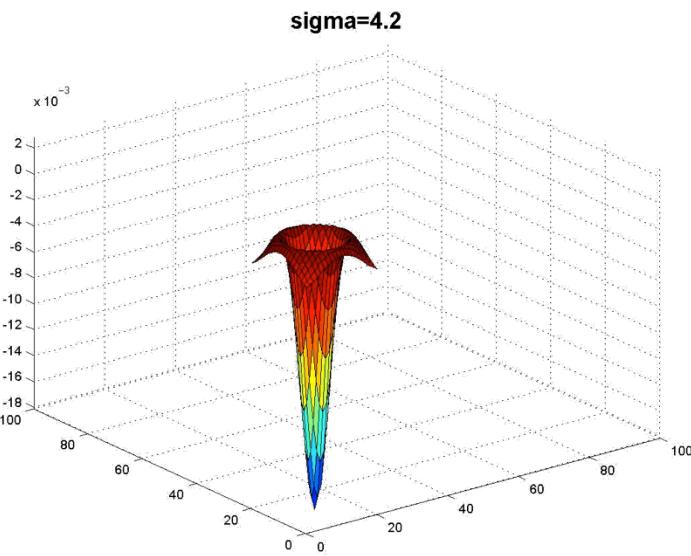
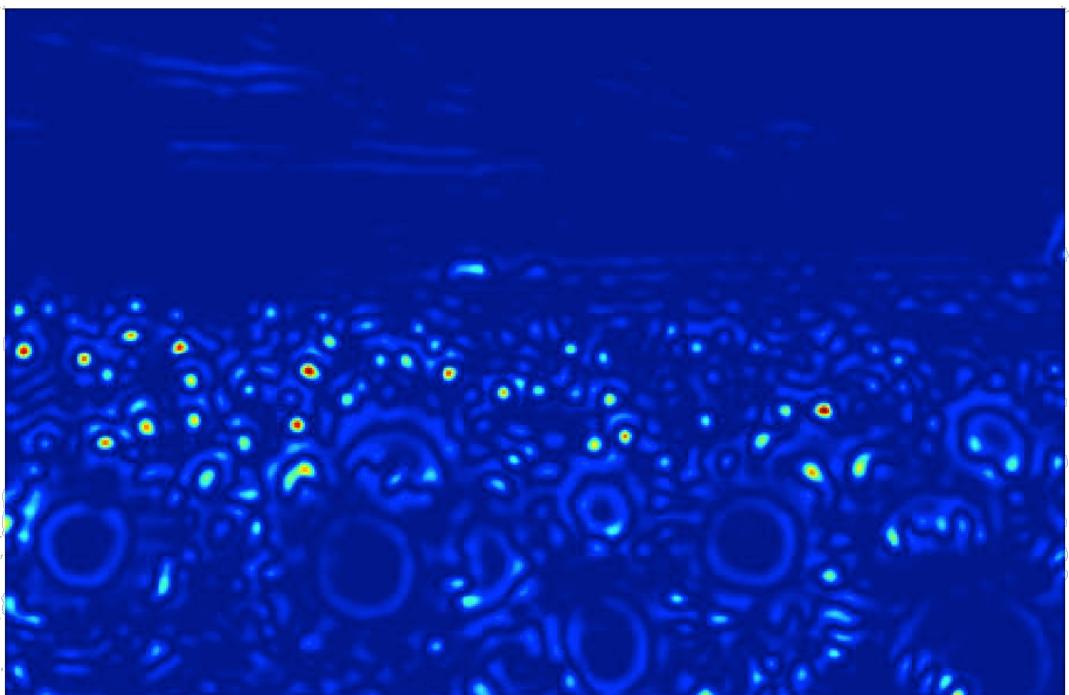
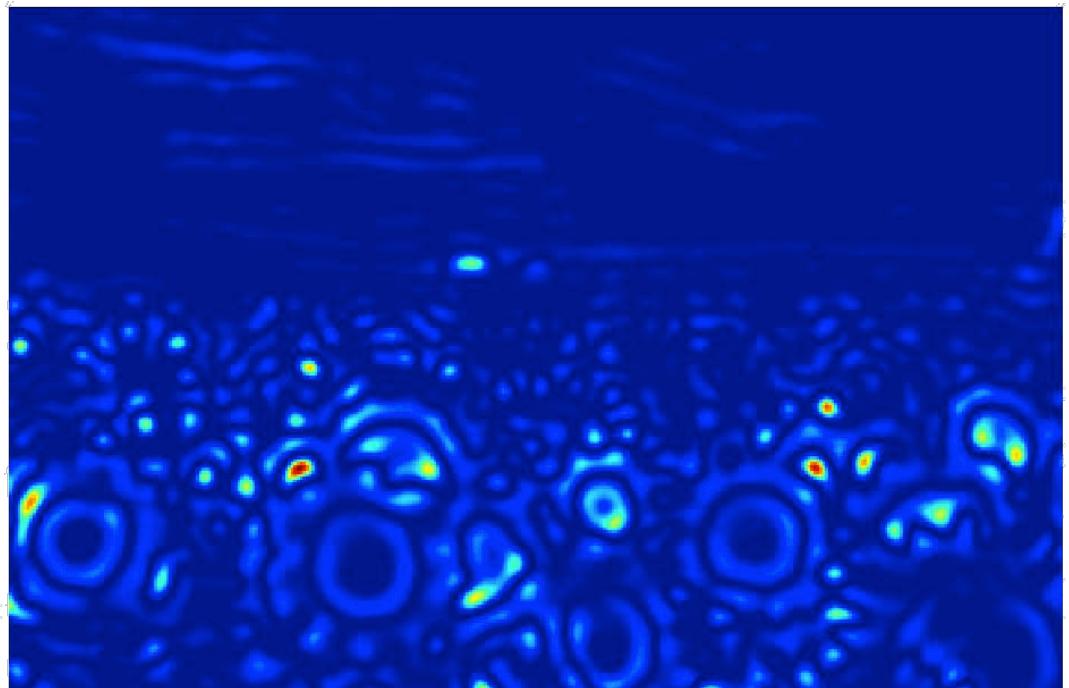
# Exemplu

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



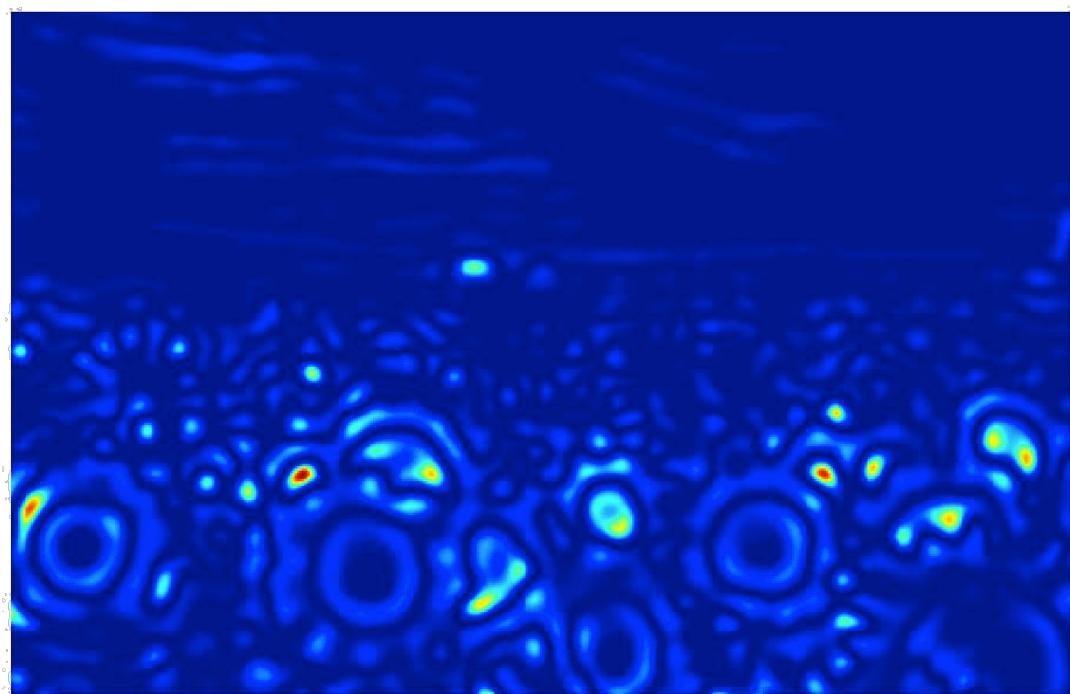
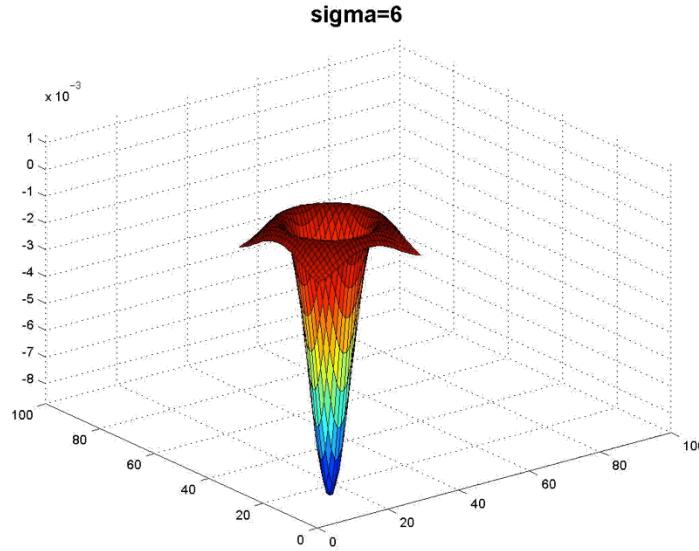
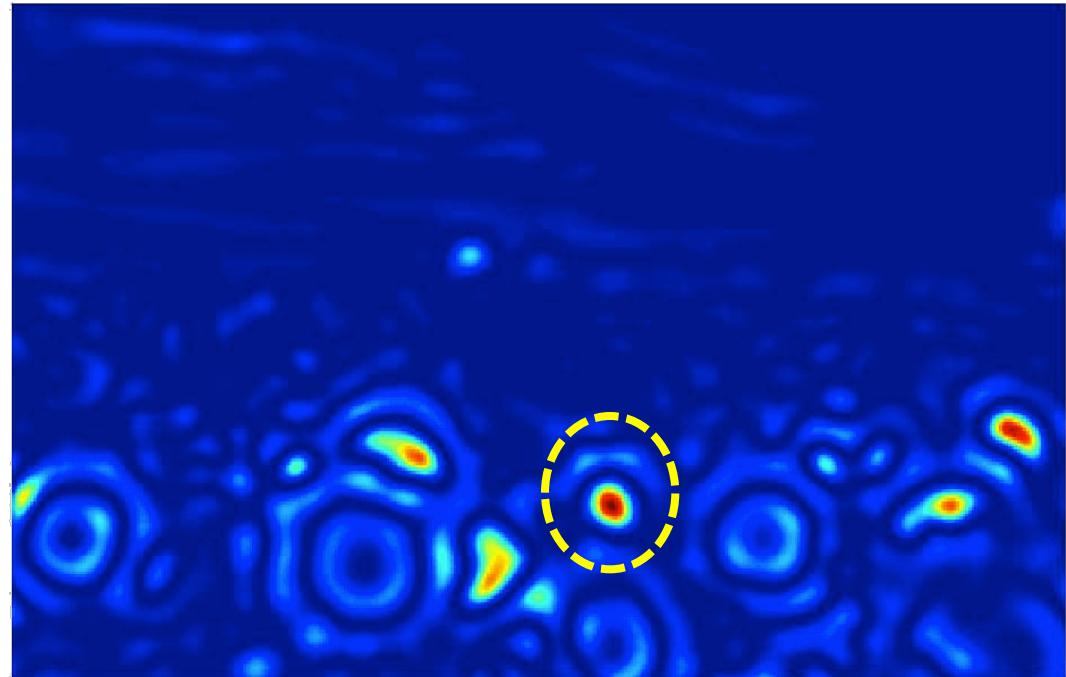
# Exemplu

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



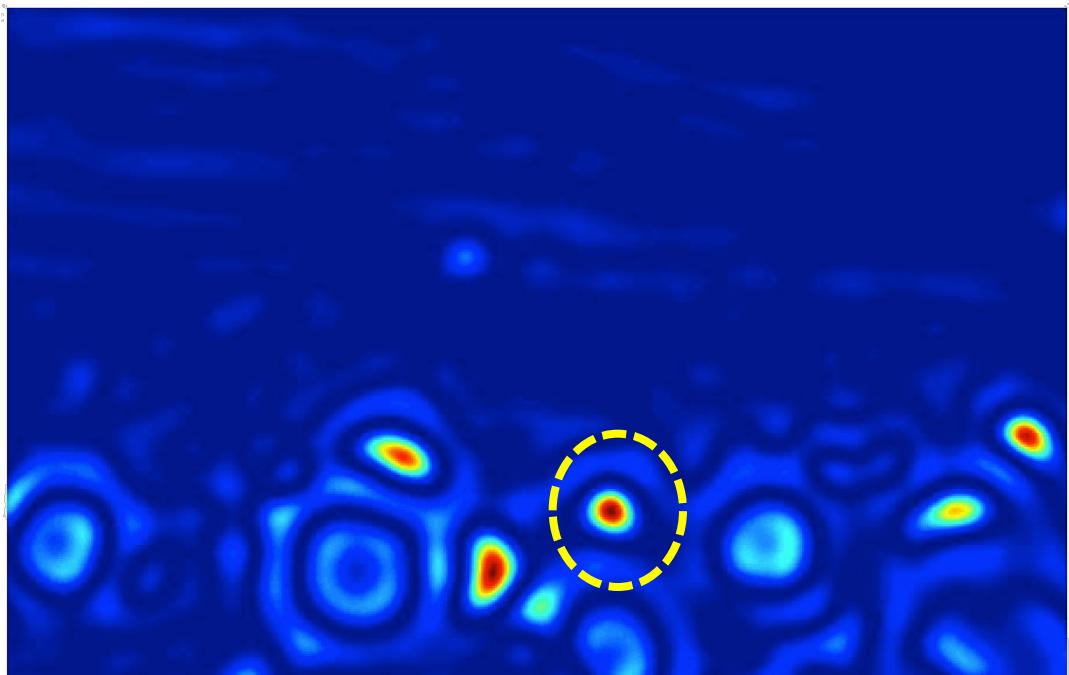
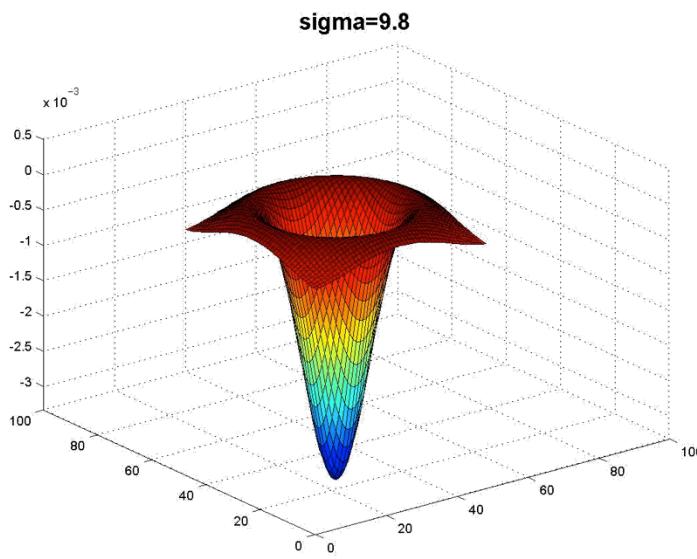
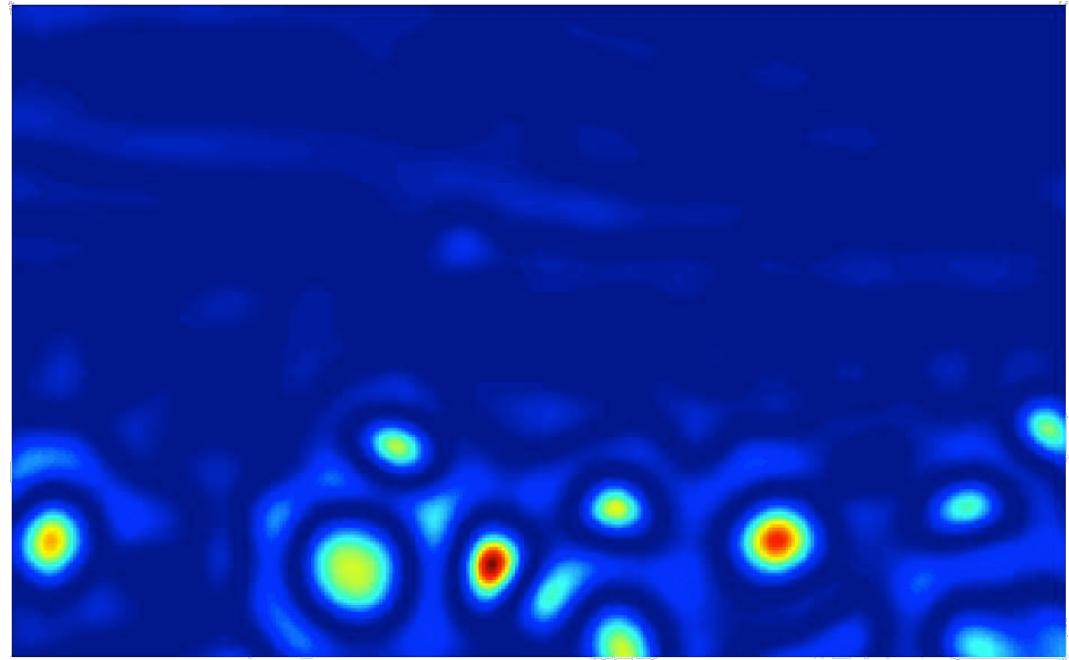
# Exemplu

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



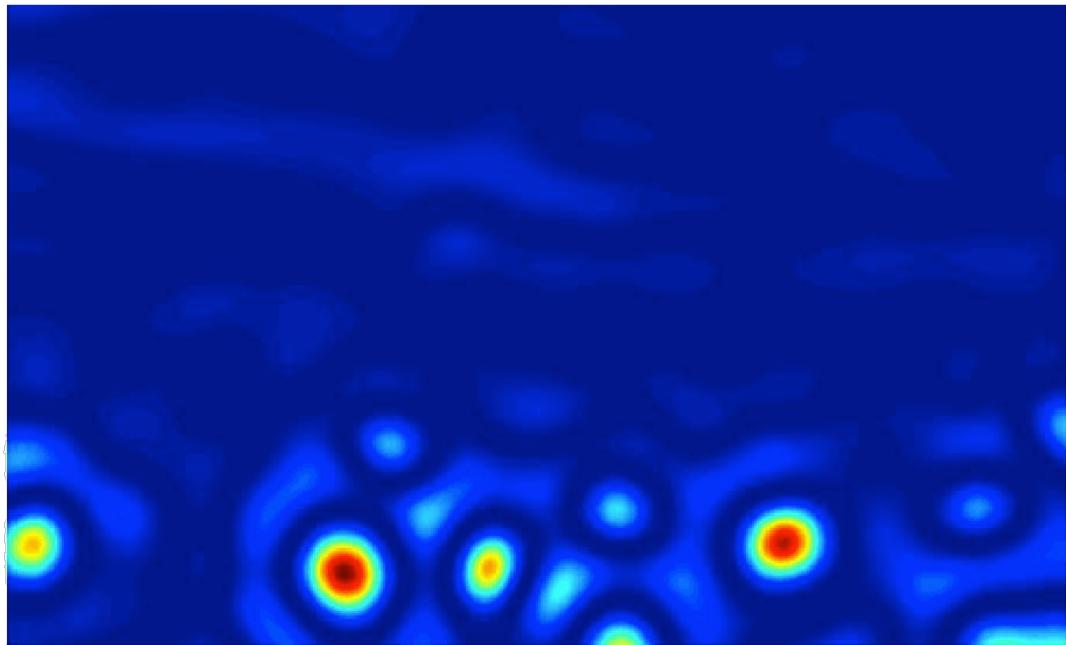
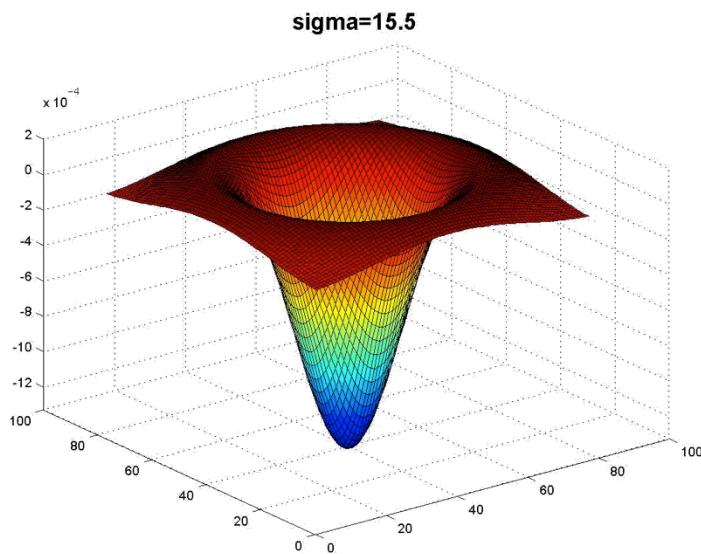
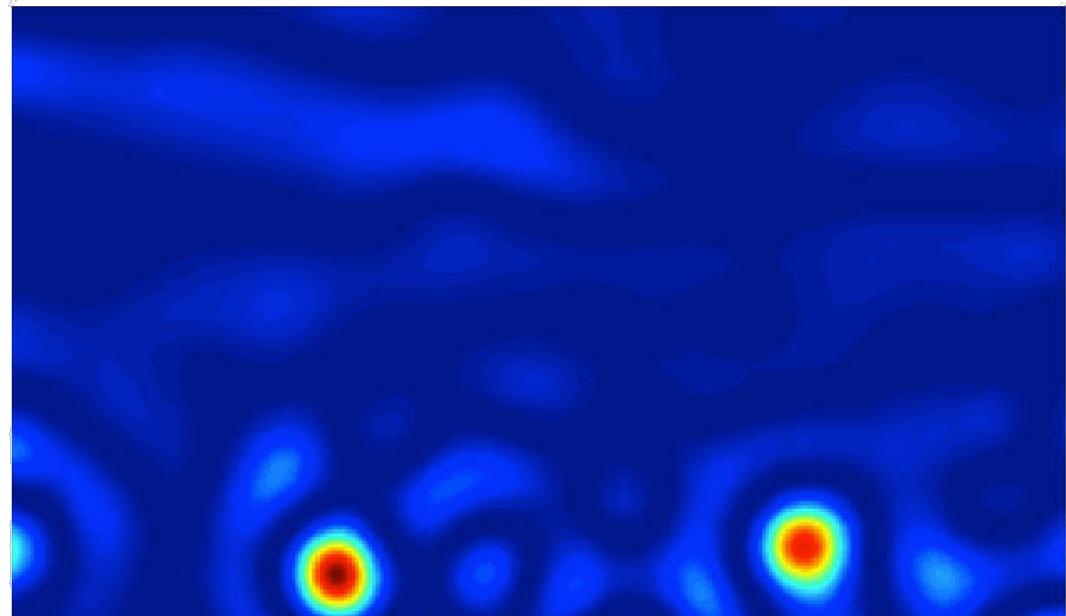
# Exemplu

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

