LAPI – Laboratorul de Analiza şi Prelucrarea Imaginilor

Universitatea POLITEHNICA din Bucureşti

Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei

Interfață Vizuală Om-Mașină
Analiza și recunoașterea gesturilor

Dr.ing. Ionuț Mironică

Cuprins curs

- Introducere modelare gesturi
- Modelarea gesturilor mâinii
  - Algoritmi de detecție
  - Algoritmi pentru detecția de obiecte și urmărirea traiectoriei
  - Clasificarea gesturilor

III. Urmărirea traiectoriei

- **Introducere**
- **Utilizarea punctelor de interes – descriptori locali**
- **Potrivire de șablon**
- **Mean-Shift / Cam-Shift**

III. Urmărirea traiectoriei

Introducere

- Detecție a obiectelor de interes (descriptori locali)
- Algoritmi pentru urmărirea traiectoriei

III. Urmărirea traiectoriei

Utilizarea punctelor de interes

Căutarea cu puncte de interes

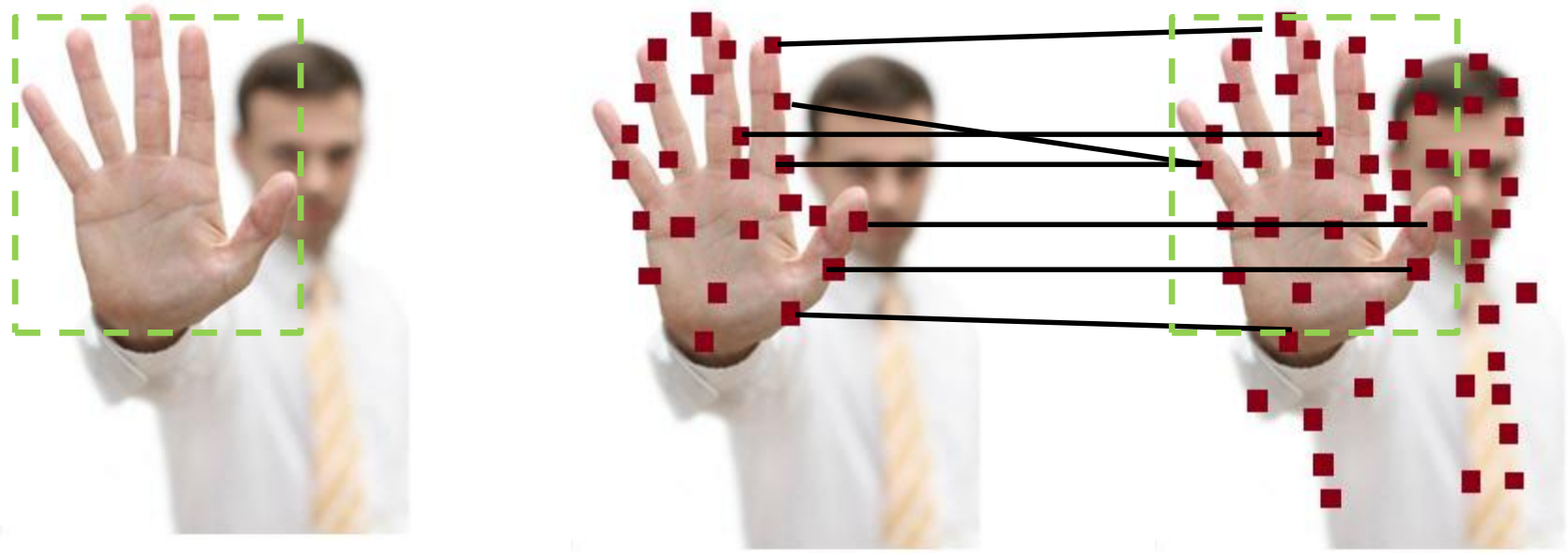
- Extragerea de puncte de interes reprezintă unul dintre algoritmii cei mai utilizați în computer vision;

- Aplicațiile acestora includ recunoașterea obiectelor, cartografiere, modelare 3D, recunoaștere gesturi, algoritmi de urmărire a traiectoriei obiectelor, creare de panorame etc.

III. Urmărirea traiectoriei

Utilizarea punctelor de interes

Căutarea cu puncte de interes



III. Descriptori locali

Utilizarea punctelor de interes

Căutarea cu puncte de interes

Algoritm general

(1) se detectează regiunile în care se extrag punctele de interes;

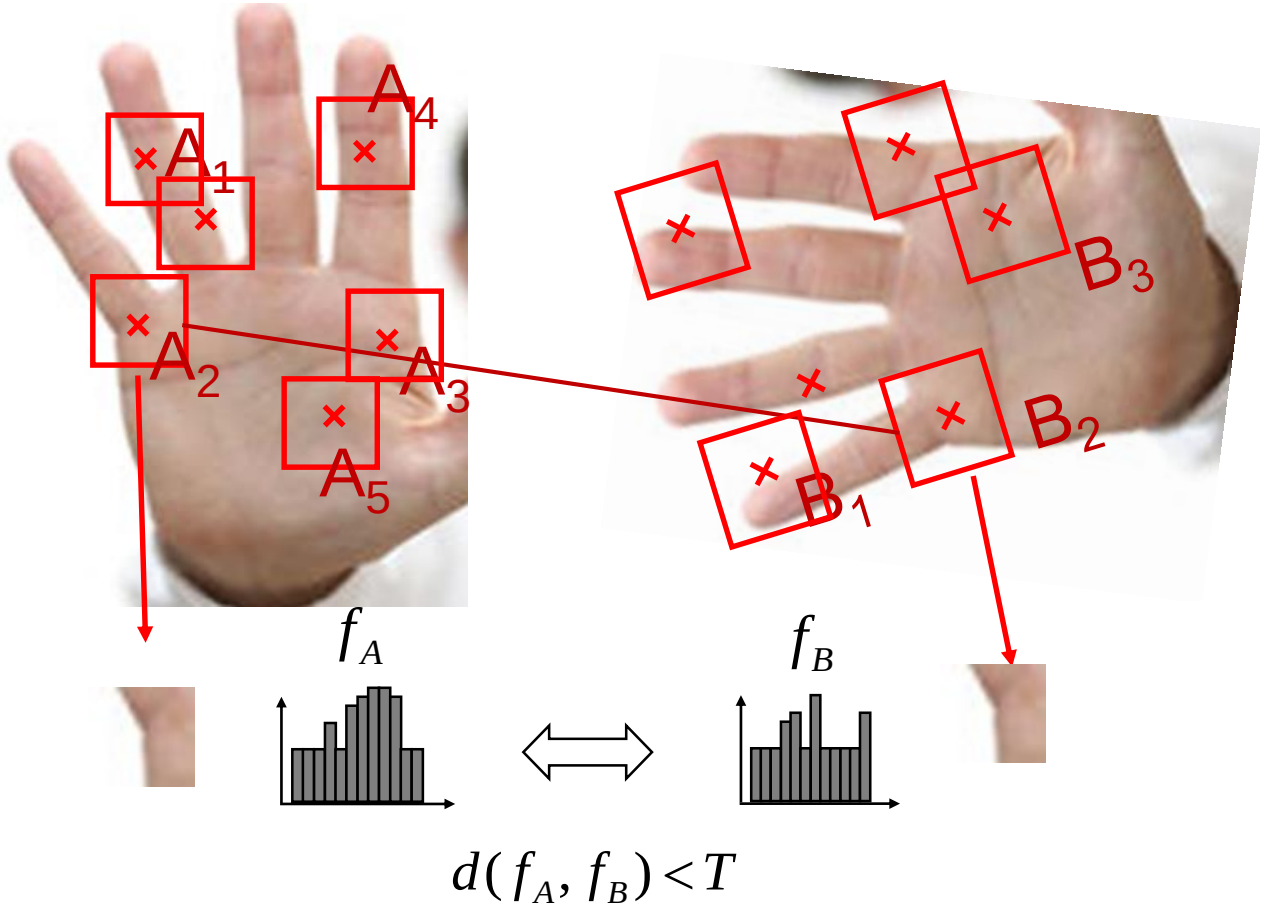
(2) pentru fiecare punct de interes se definește o regiune în jurul acestuia. Pentru fiecare regiune se calculează un descriptor;

(3) se calculează o distanță între lista de puncte de interes din șablonul de antrenare și celelalte cadre.

III. Descriptori locali

Utilizarea punctelor de interes

Căutarea cu puncte de interes – algoritm general



III. Descriptori locali

Utilizarea punctelor de interes

Căutarea cu puncte de interes

Algoritm general

(1) se detectează regiunile în care se extrag punctele de interes;

(2) pentru fiecare punct de interes se definește o regiune în jurul acestuia. Pentru fiecare regiune se calculează un descriptor;

(3) se calculează o distanță între lista de puncte de interes din șablonul de antrenare și celelalte cadre.

III. Descriptori locali

Utilizarea punctelor de interes

Extragerea spațială a punctelor cheie

Obiective:

- Repetitivitate – regiunile extrase să se găsească și în alte regiuni:
  - invariant la translație, rotație, schimbări de scală;
  - invariant la ocluziuni;
  - variații de iluminare.
- Localizare precisă;
- Conținut relevant.

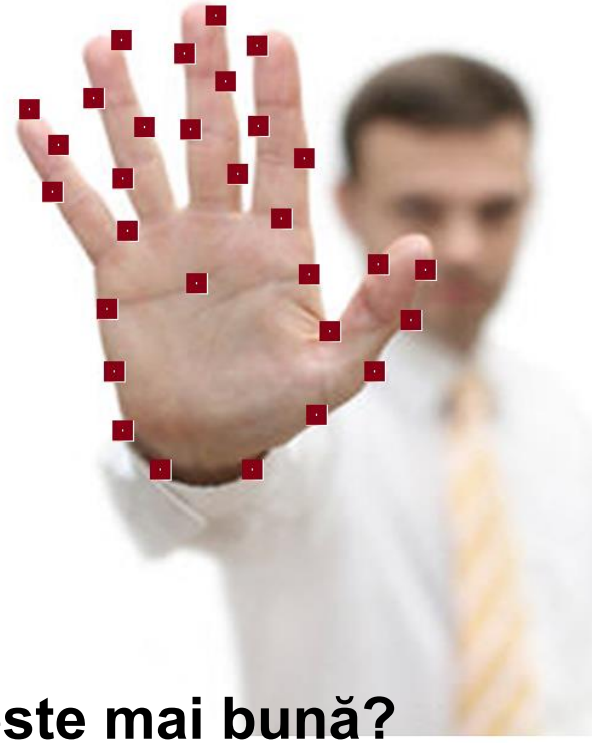
III. Descriptori locali

Utilizarea punctelor de interes

Extragerea spațială a punctelor cheie

Extragere densă

Extragere selectivă



Care variantă este mai bună?

III. Descriptori locali

Utilizarea punctelor de interes

Extragerea spațială a punctelor cheie – discuție

Compromisuri

Localizare

Mai multe puncte

Robust la apariția de ocluziuni

Mai multă distinctivitate

Localizare precisă

Descriere

Mai robust

Este mai robust la anumite variații
Maximizează probabilitatea de împerechere

Mai selectiv

Minimizează împerecherile greșite

III. Descriptori locali

Utilizarea punctelor de interes

Extragerea spațială a punctelor cheie – discuție

- Nu există o metodă general valabilă mai bună decât cealaltă. Ambele abordări pot fi eficiente în contexte diferite;

- Pentru algoritmul de extracție densă regiunile cu mai puțin contrast contribuie în mod egal în reprezentarea imaginii; Legătura spațială dintre punctele de interes este mai bine păstrată;

- Pe de altă parte, extragerea densă nu poate atinge același grad de distinctivitate.

III. Descriptori locali

Utilizarea punctelor de interes

Extragerea spațială a punctelor cheie – discuție

- Un algoritm de extracție densă a punctelor cheie poate obține performanțe superioare în cazul în care informația de fundal este foarte importantă. (ex: în competiția Pascal, există 20 de clase care sunt dependente de context: avioanele apar de obicei în imagini cu nori, animalele sunt prezente într-un spațiu natural, iar obiectele de mobilier sunt localizate în interiorul unor camere).

- La extracția densă, calculul poziției punctelor cheie este mult mai rapidă, însă numărul de descriptori extras este mult mai ridicat, ceea ce reduce timpul câștigat pentru extracție.

III. Descriptori locali

Utilizarea punctelor de interes

Căutarea cu puncte de interes

Algoritm general

(1) se detectează regiunile în care se extrag punctele de interes;

(2) pentru fiecare punct de interes se definește o regiune în jurul acestuia. Pentru fiecare regiune se calculează un descriptor;

(3) se calculează o distanță între lista de puncte de interes din șablonul de antrenare și celelalte cadre.

III. Descriptori locali

Utilizarea punctelor de interes

Căutarea cu puncte de interes

Dimensiune regiune

Mai mică (10x10)

- Robust la apariția de ocluziuni;
- Deoarece suprafața este mai mică vor fi generate mai multe puncte de interes.

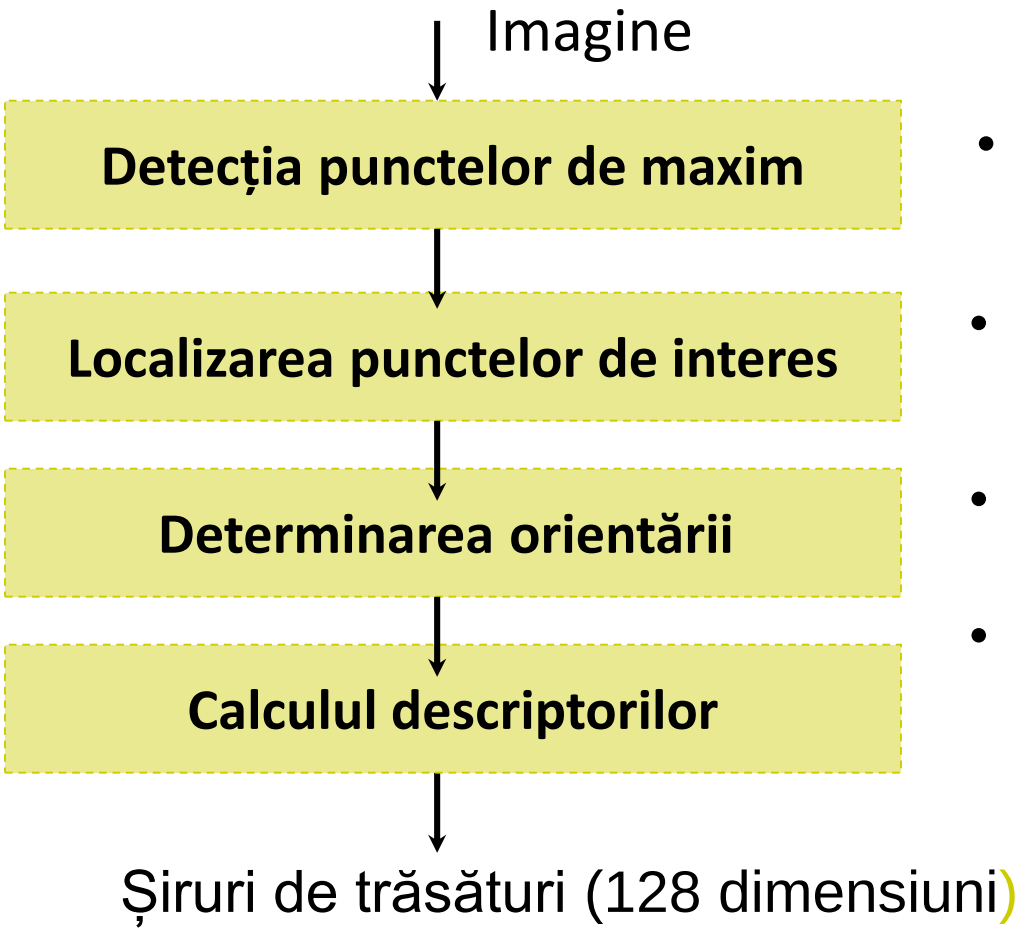
Mai mare (40x40)

- Mai expus la erori (preluare din alte obiecte);
- Descriere mai precisă și particulară.

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SIFT (Scale-invariant feature transform)



- identificarea potențialelor puncte de interes;
- localizarea candidaților și eliminarea punctelor duplicate;
- identificarea orientării dominate;
- construcția unui descriptor bazat pe histograma de gradienți din vecinătatea punctelor de interes.

[D. Lowe '04]

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SIFT

Cum se pot detecta locații care sunt invariante la schimbări de scală ale imaginii?

Soluție: utilizarea funcției de diferențe de gausiene (DoG).

- Pentru o imagine *I(x,y)*, fie o reprezentare liniară a acesteia:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

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

- Se caută valorile de minim/maxim local pentru o serie de diferențe de gausiene:

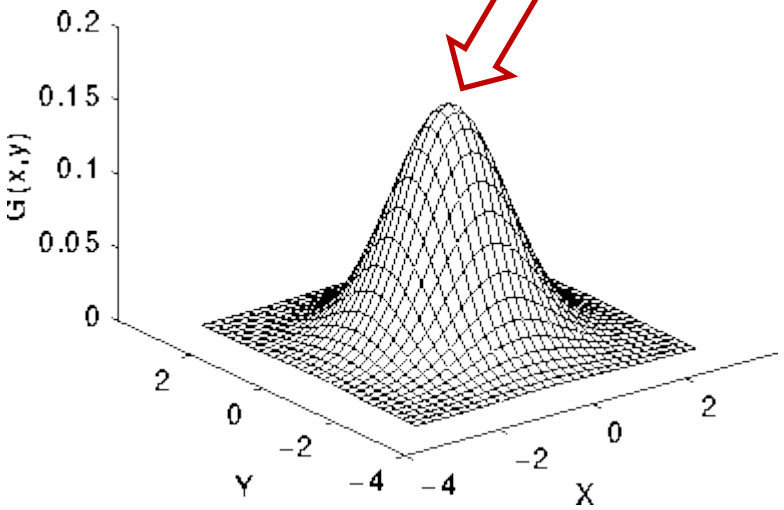
$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$$
$$= L(x, y, k\sigma) - L(x, y, \sigma)$$

III. Descriptori locali

Filtru Gaussian - reamintire

$\sigma = 1$, W = 5

0.00296902	0.01330621	0.02193823	0.01330621	0.00296902
0.01330621	0.0596343	0.09832033	0.0596343	0.01330621
0.02193823	0.09832033	0.16210282	0.09832033	0.02193823
0.01330621	0.0596343	0.09832033	0.0596343	0.01330621
0.00296902	0.01330621	0.02193823	0.01330621	0.00296902



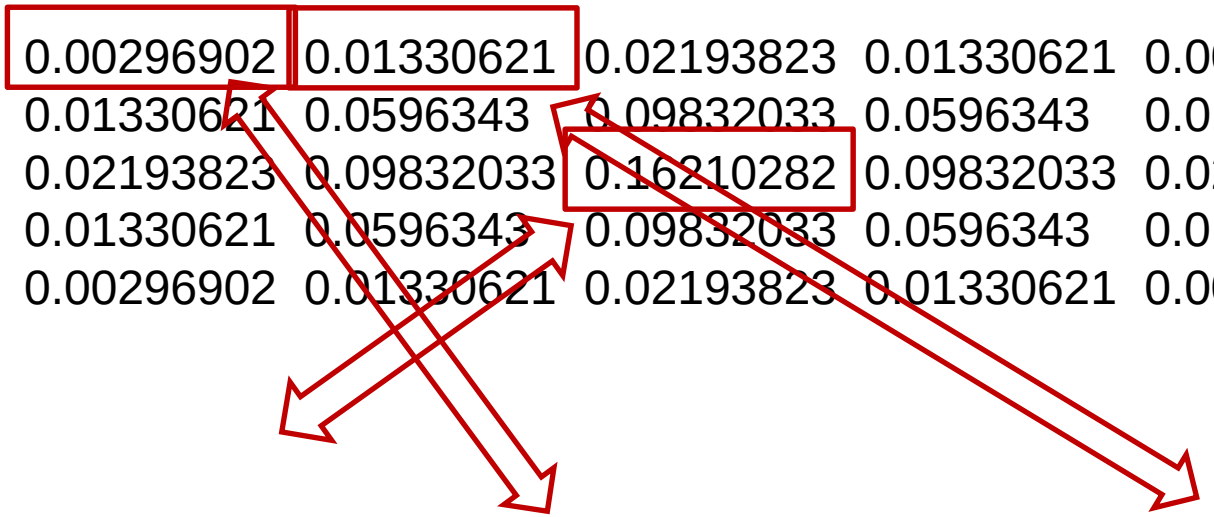
[https://github.com/imironica/IVOM-Demo/blob/master/IVOM_Demo/KeypointsDetector/compute_gaussian_filter.py]

III. Descriptori locali

Filtru Gaussian - reamintire

σ = 1, W = 5

0.00296902	0.01330621	0.02193823	0.01330621	0.00296902
0.01330621	0.0596343	0.09832033	0.0596343	0.01330621
0.02193823	0.09832033	0.16210282	0.09832033	0.02193823
0.01330621	0.0596343	0.09832033	0.0596343	0.01330621
0.00296902	0.01330621	0.02193823	0.01330621	0.00296902



$$G(0,0,1) = 0.00296902 \cdot I(-2,-2) + 0.01330621 \cdot I(-2,-1) + 0.02193823 \cdot I(-2,0) \ldots$$

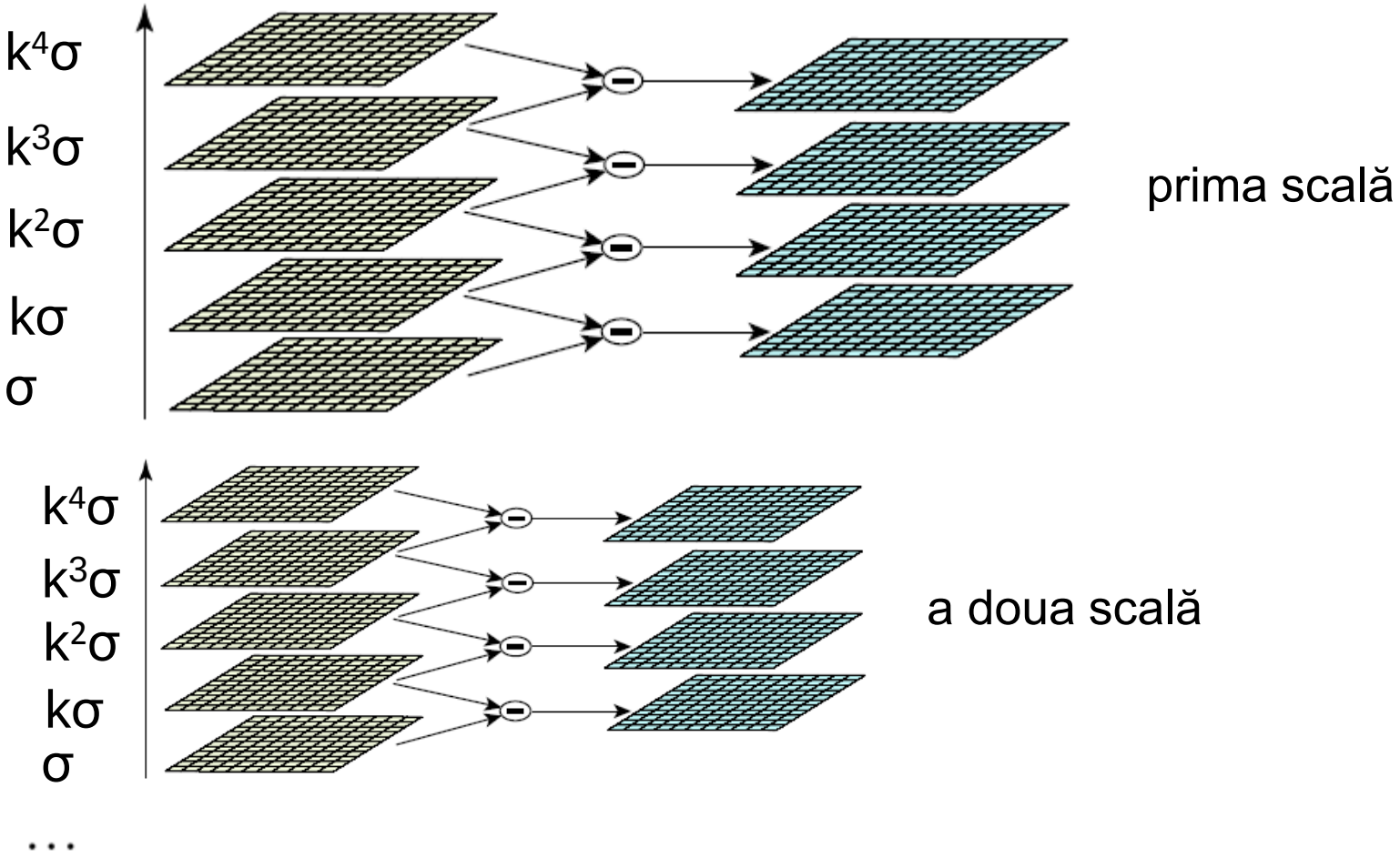
Implementare Python:

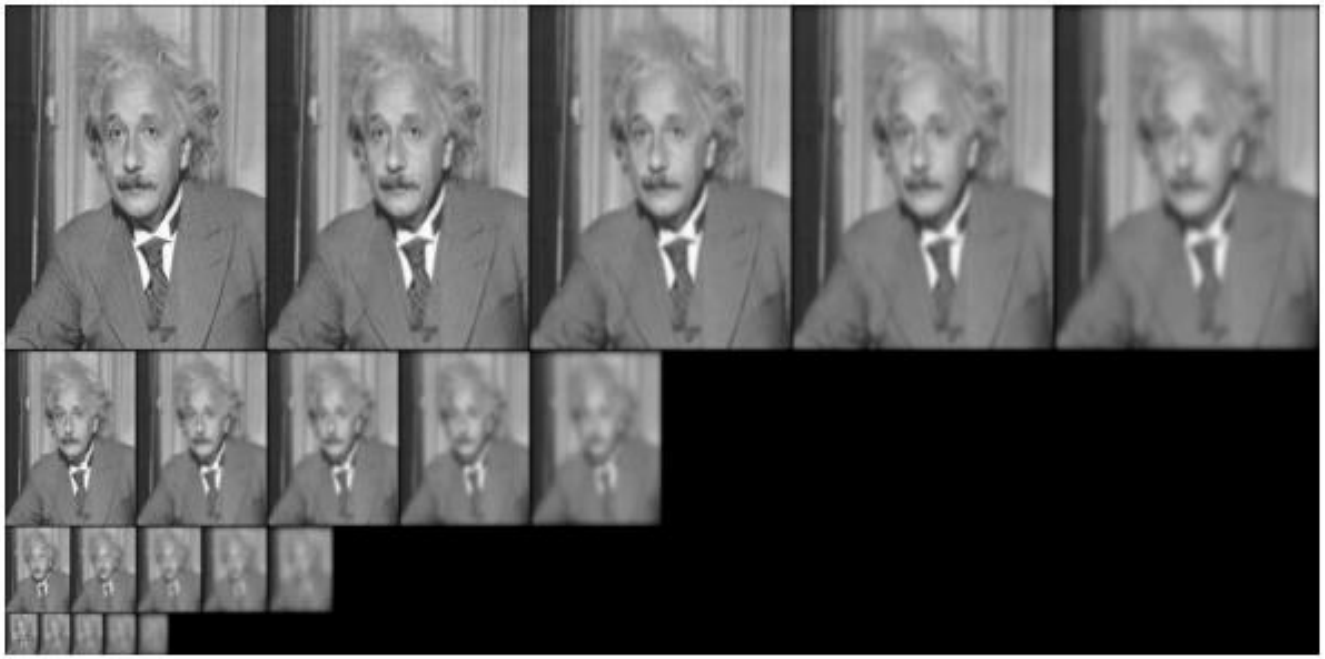
- scipy – ndimage.gaussian_filter
- OpenCV - cv2.GaussianBlur

III. Descriptori locali

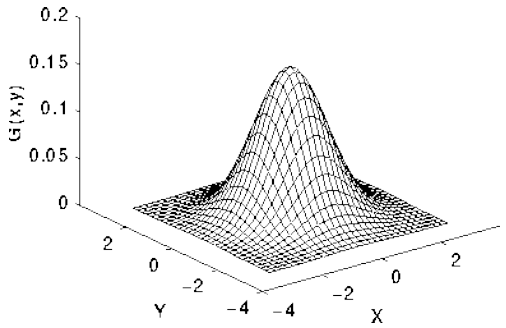
Utilizarea punctelor de interes

Algoritmul SIFT - Detecția punctelor de extrem

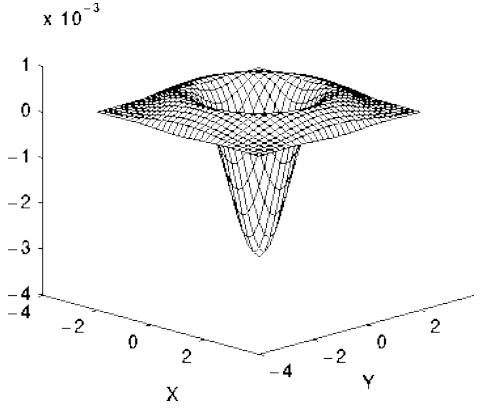




Imagini gausiene



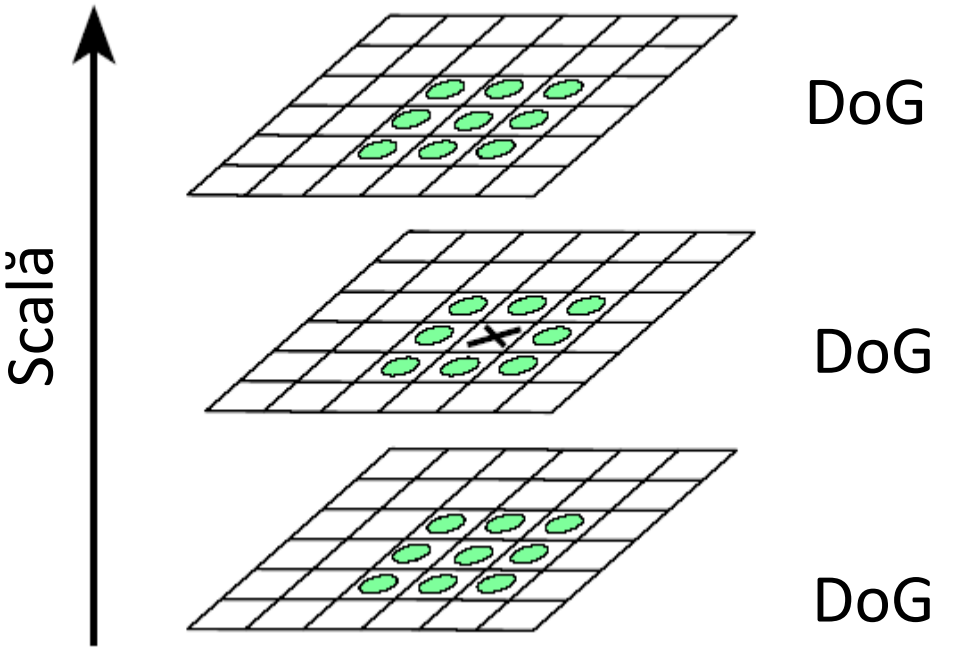
Imagini DoG



III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SIFT - Detecția punctelor de extrem



Dacă ***X*** este *cel mai mare* sau *cel mai mic* dintre toți vecinii, atunci ***X*** este denumit ***punct de interes (keypoint)***.

III. Descriptori locali

Utilizarea punctelor de interes

De ce utilizăm funcția DoG?

- Funcția DoG este utilizată pentru a înlocui funcția LoG (laplacian de gausiană - $\sigma^2\nabla^2 G$). Aceasta poate fi calculată într-un mod eficient, reprezentând o aproximare a funcției invariante la scală LoG:
  - Lindeberg a arătat că normalizarea funcției Laplacian cu un factor σ^2 este necesar pentru obținerea unei invarianțe la modificări de scală (1994);
  - Mikolajczyk a justificat că minimele și maximele din LoG creează cele mai stabile puncte de interes (2002);

DoG v.s. $\sigma^2\nabla^2 G$

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2\nabla^2 G$$

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SIFT - Detecția punctelor de extrem

- Pentru fiecare punct, se va calcula magnitudinea și orientarea gradientului utilizând formulele următoare:

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

$$\theta(x,y) = \tan^{-1}\left[\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right]$$

- De ce avem nevoie de orientare? – invarianță la rotație;
- Se va crea o histogramă de orientări și se vor reține acele valori maxime, împreună cu punctele care conțin minim 80% din valoarea maximă gasită (eliminandu-se astfel peste 95% din punctele extrase în procesul anterior);

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SIFT - Detecția punctelor de extrem

- După calculul extremelor, vor fi eliminate punctele cu contrast scăzut și muchii mai puțin ieșite în evidență. Punctele rămase vor reprezenta punctele de interes finale;

- Fiecare punct de interes va avea 4 dimensiuni: coordonatele (x,y), magnitudinea și orientarea.

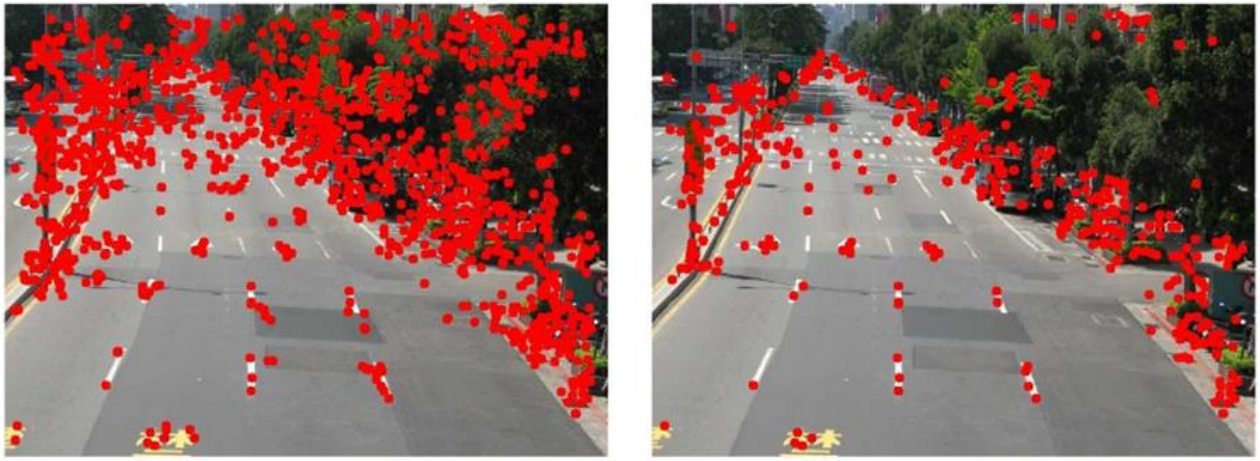
III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SIFT - Detecția punctelor de extrem

(a) puncte de interes înainte de aplicarea pragului;

(b) puncte de interes după aplicarea pragului;



(a) (b)

În funcție de selecția pragurilor anterioare vom avea un număr mai mare / mai mic de puncte de interes.

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SIFT - Detecția punctelor de extrem

(a) (b)

(a) Top 100 keypoints.
(b) Top 200 keypoints

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SIFT – Calculul descriptorului

1. Se preia o fereastră de 16 x16 în jurul punctului de interes.
2. Se împarte în 4x4 celule.
3. Se calculează o histogramă de gradienți pe 8 orientări pentru fiecare celulă.

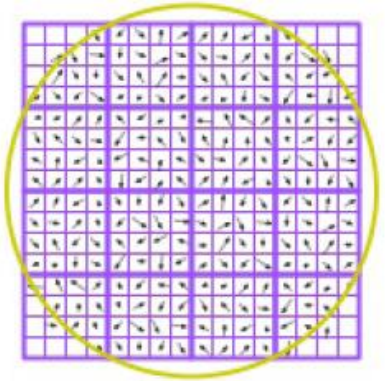
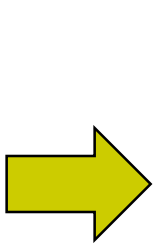
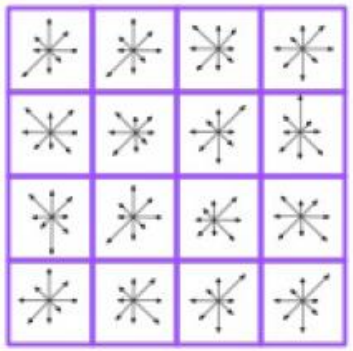


Image gradients





Keypoint descriptor

16 histograme x 8 orientări
= 128 trăsături

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SIFT – Proprietăți

- Grad ridicat de distinctivitate:
  - o singură trăsătură poate fi corect potrivită cu o mare probabilitate într-o bază de date de trăsături generate din multe imagini.
- Invariant la transformări de scală și rotație;
- Invariant parțial la schimbările de iluminare;
- Parțial invariant la vederea 3D a camerei:
  - poate tolera o rotație de până la 60 de grade;
- Este eficient;
- Totuși, este mai lent decât alte implementări.

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SIFT – software disponibil

- Articol complet: D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, **International Journal of Computer Vision**, 60(2):91-110, 2004 - probabil cel mai utilizat articol din CV

Download demo SIFT demo (autor):

- http://www.cs.ubc.ca/~lowe/keypoints/
- sau http://www.csie.ntnu.edu.tw/~myeh/courses/s10_ms/Assignments/siftDemoV4.zip

Alte surse:

- OpenCV,
- VLFeat.

III. Descriptori locali

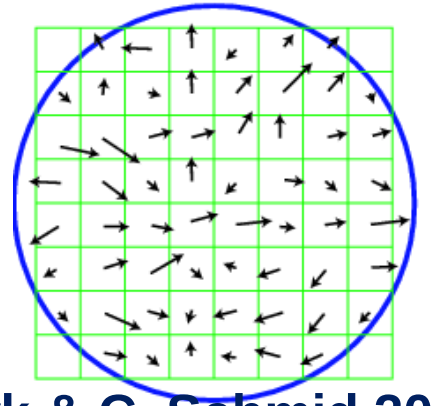
Utilizarea punctelor de interes

PCA-SIFT

- Algoritmul este identic cu SIFT, mai puțin pasul 4 (calculul descriptorului);

- În loc să se utilizeze histogramele ponderate din algoritmul SIFT clasic, se calculează gradienții (verticali / orizontali) locali pe o suprafață de 41x41 pixeli (2 pixeli reprezintă conturul);

- Se concatenează toate valorile într-un singur vector: 2 x 39 x 39 = 3042 elemente.



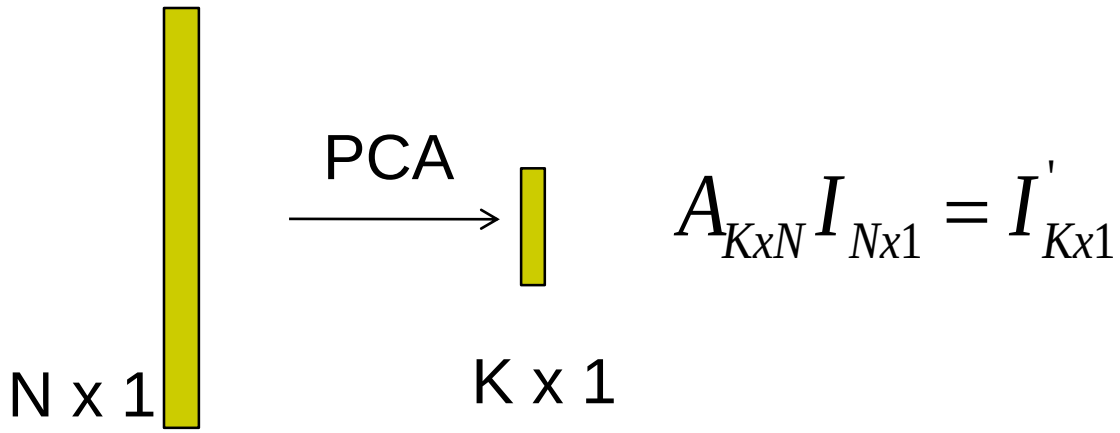
[K. Mikolajczyk & C. Schmid 2005]

III. Descriptori locali

Utilizarea punctelor de interes

PCA-SIFT

- Se reduce dimensionalitatea vectorului folosind Analiza Componentelor Principale (Principal Component Analysis - PCA) - de ex: de la 3042 la 64 / 36 / 20;



$$A_{KxN} I_{Nx1} = I'_{Kx1}$$

III. Descriptori locali

Utilizarea punctelor de interes

PCA-SIFT

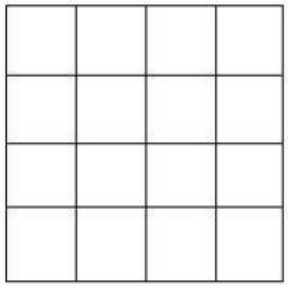
- PCA-SIFT reprezintă un descriptor mult mai compact iar în funcție de problemă poate fi mai mult sau mai puțin descriminatoriu.
- **Avantaje:**
  - Descriptorul este mult mai compact;
  - Mărește viteza algoritmului (keypoints matching);
  - Mai rezistent la zgomot.
- Mai multe detalii în: Yan Ke și Rahul Sukthankar: „PCA-SIFT: A More Distinctive Representation for Local Image Descriptors”, 2005;
- Download: http://www.cs.cmu.edu/~yke/pcasift/.

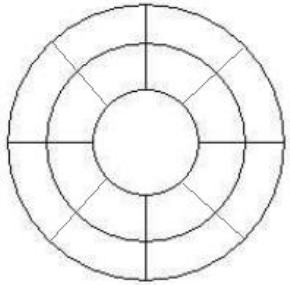
III. Descriptori locali

Utilizarea punctelor de interes

Gradient location-orientation histogram (GLOH)

- GLOH reprezintă un descriptor similar cu PCA-SIFT, singura diferență fiind calculul descriptorului final:
- Se împarte spațiul din jurul punctului de interes în coordonate log-polare;
- Lungime descriptor: (2 x 8 + 1) * 16 = 272;
- Se aplică PCA și se reține 128 elemente.





- **Proprietăți:**
  - Acurateţe mai mare de clasificare;
  - Viteză mai mică;
  - Mai rezistent la zgomot.

[Mikolajczyk & Schmid ‘05]

III. Descriptori locali

Utilizarea punctelor de interes

SURF

Algoritmul **SURF (**Speeded Up Robust Features) îmbunătățește viteza de calcul prin:

(1) utilizarea matricei de aproximare Hessiană

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(\sigma) & L_{xy}(\sigma) \\ L_{yx}(\sigma) & L_{yy}(\sigma) \end{bmatrix}$$

(2) a imaginii integrale în calculul descriptorului.

Ce este o imagine integrală?

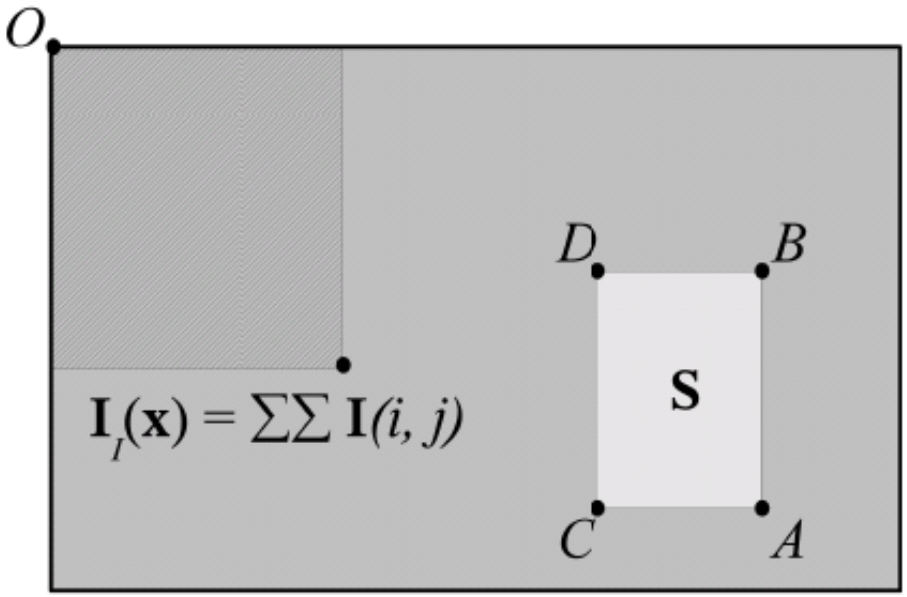
[H. Bay '08]

III. Descriptori locali

Utilizarea punctelor de interes

SURF – Imaginea integrală

Imaginea integrală $I_\Sigma(x,y)$ a unei imagini I(x, y) reprezintă suma tuturor pixelilor din I(x,y) dintr-o regiune dreptunghiulară.



$$I_\Sigma(x,y) = \sum_{i=0}^{i\le x}\sum_{j=0}^{j\le y} I(i,j)$$

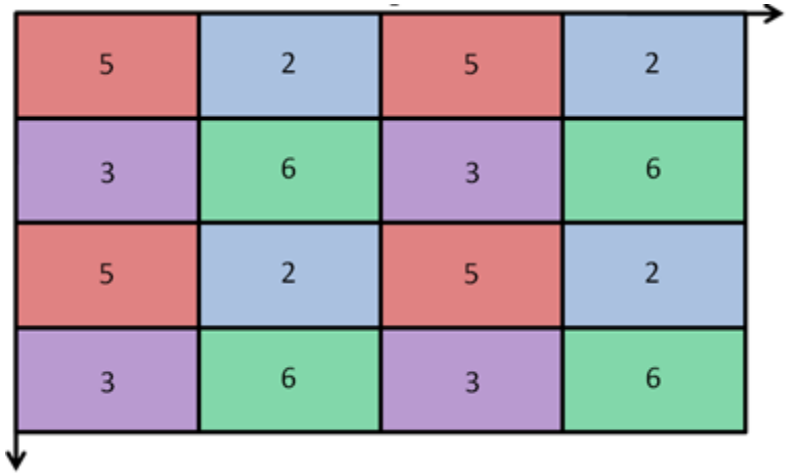
Prin utilizarea imaginii integrale este nevoie de doar patru valori pentru calculul sumei pixelilor dintr-o suprafată dreptunghiulară

$$\mathbf{S} = A - B - C + D$$

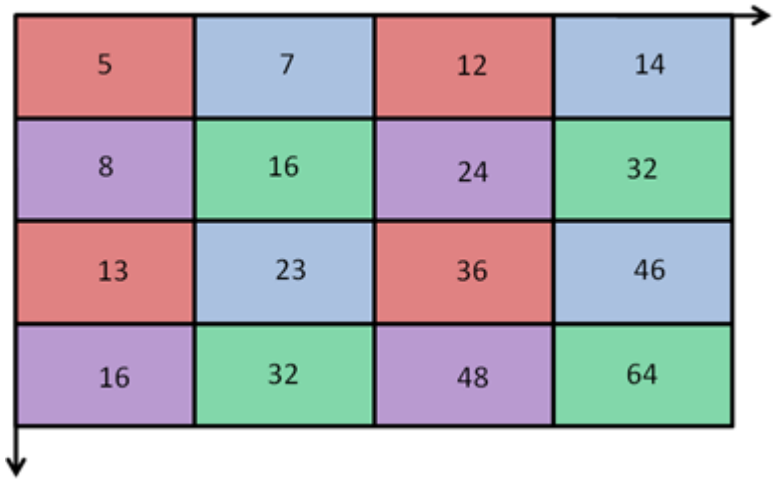
III. Descriptori locali

Utilizarea punctelor de interes

SURF – Imaginea integrală



Imagine



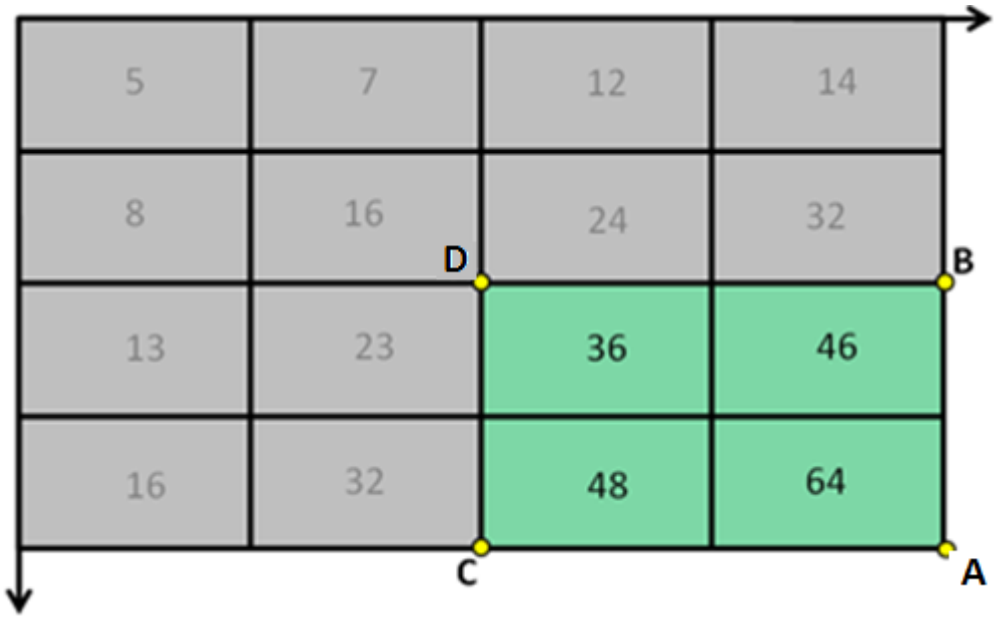
Imagine integrală

$$\mathbf{S}=A-B-C+D$$

III. Descriptori locali

Utilizarea punctelor de interes

SURF – Imaginea integrală



$$\mathbf{S}=A-B-C+D$$

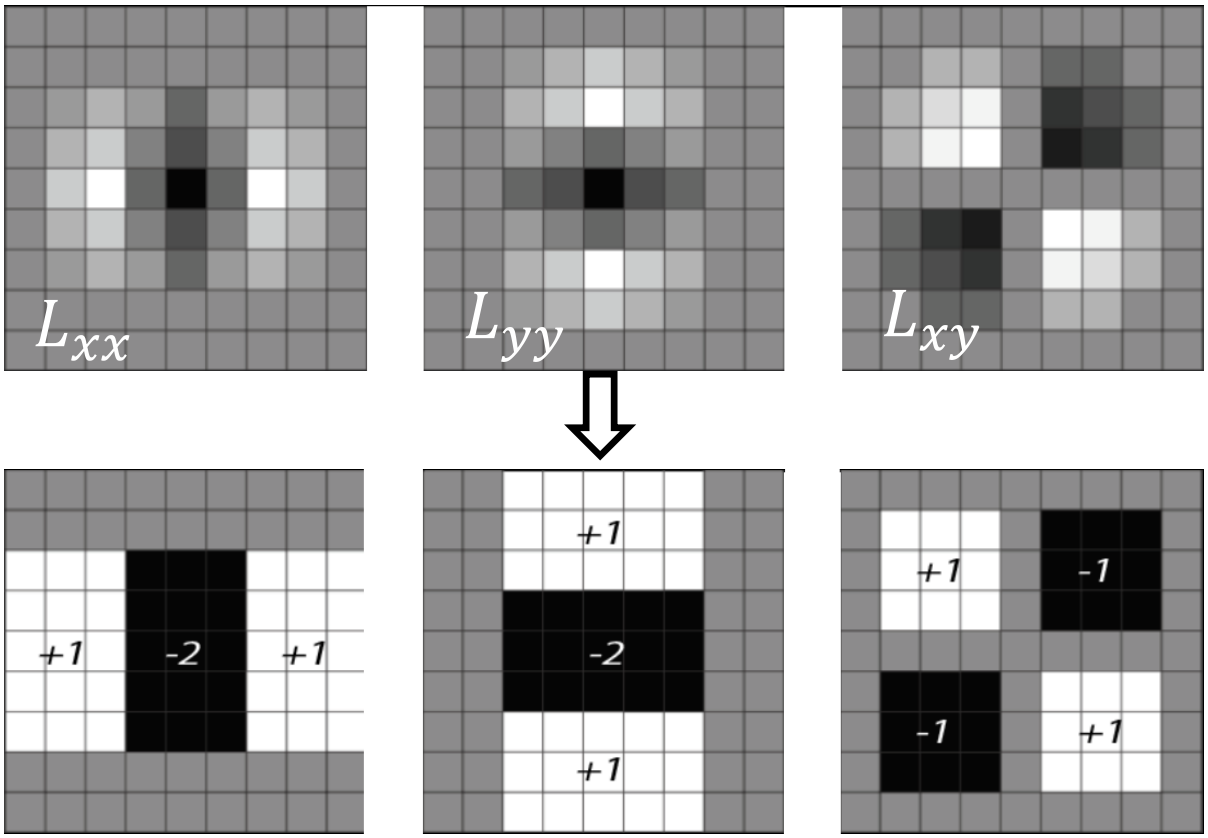
S = 64-32-32+16 = 16

III. Descriptori locali

Utilizarea punctelor de interes

SURF – Aproximarea matricei Hessiene

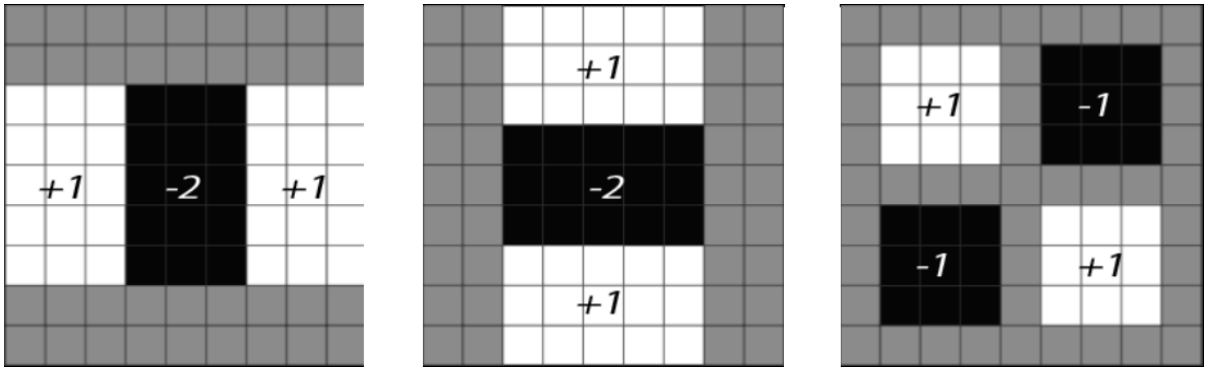
Pentru σ=1.2 (9x9), funcția LoG poate fi aproximată din:



III. Descriptori locali

Utilizarea punctelor de interes

SURF – Aproximarea matricei Hessiene



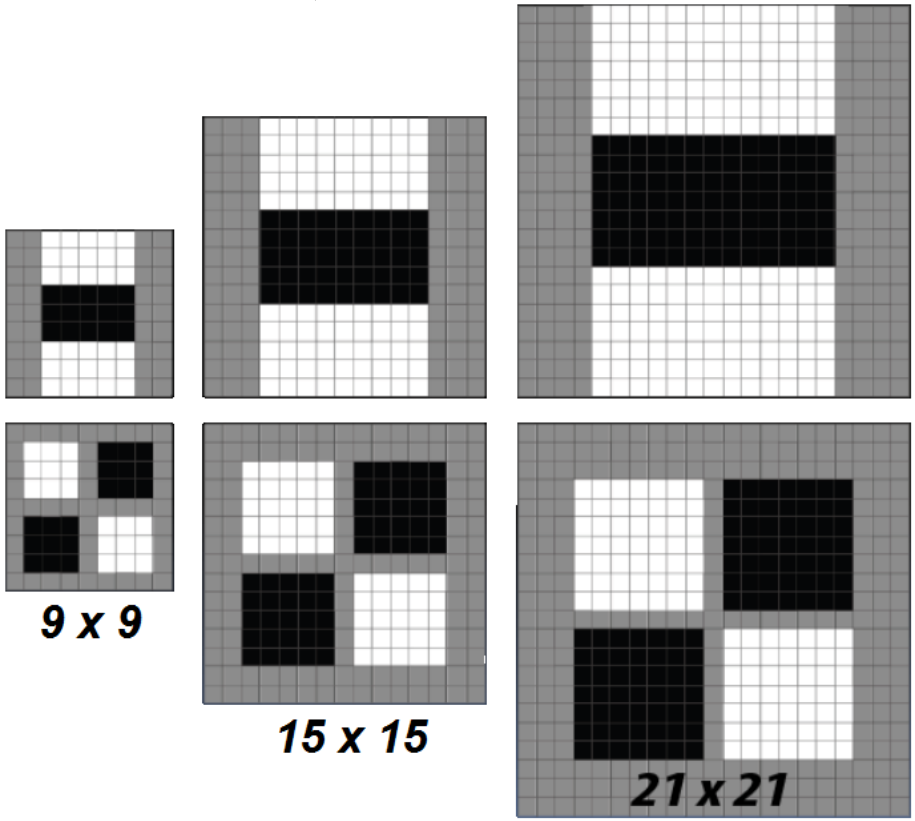
Acestea pot fi foarte ușor calculate utilizând principiul de imagine integrală.

III. Descriptori locali

Utilizarea punctelor de interes

SURF – Aproximarea matricei Hessiene

Similar cu algoritmul SIFT, se aplică filtrul se calculează la mai multe octave (3 sau 4 în funcție de implementare):

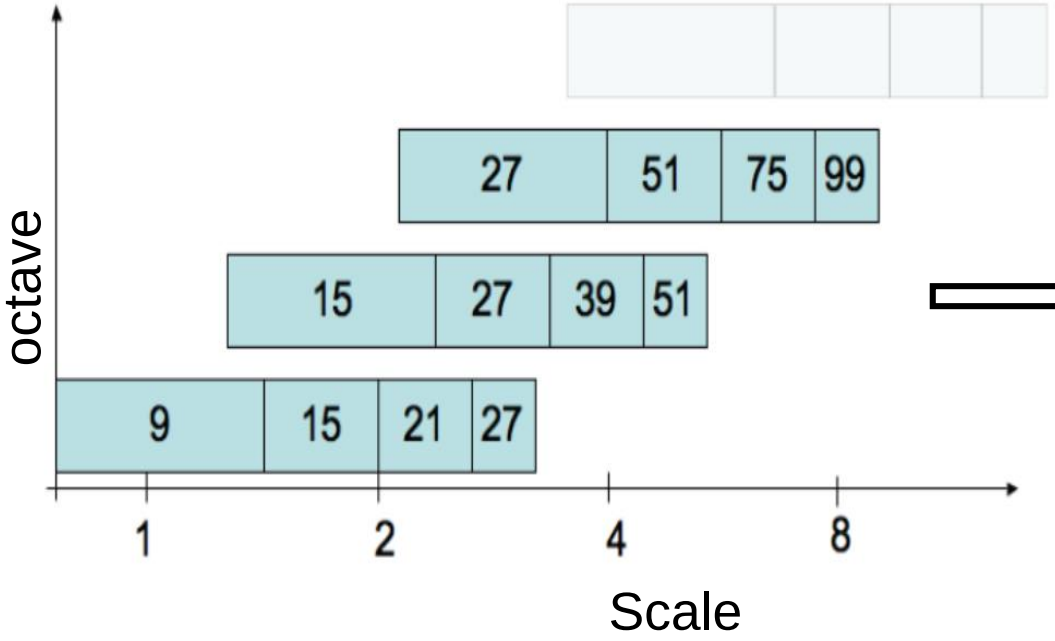


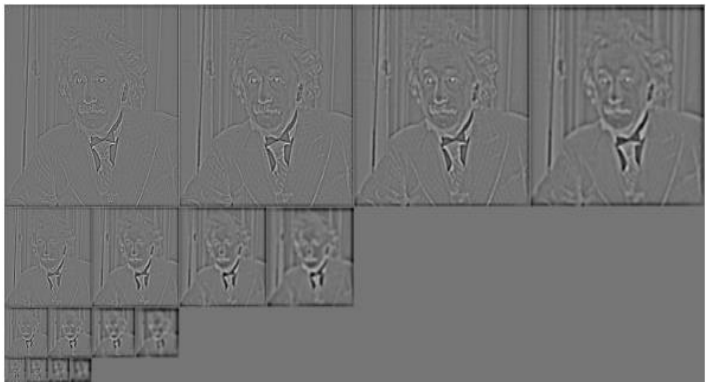
III. Descriptori locali

Utilizarea punctelor de interes

SURF – Aproximarea matricei Hessiene

Similar cu algoritmul SIFT, se aplică filtrul se calculează la mai multe scale (3 sau 4 în funcție de implementare):



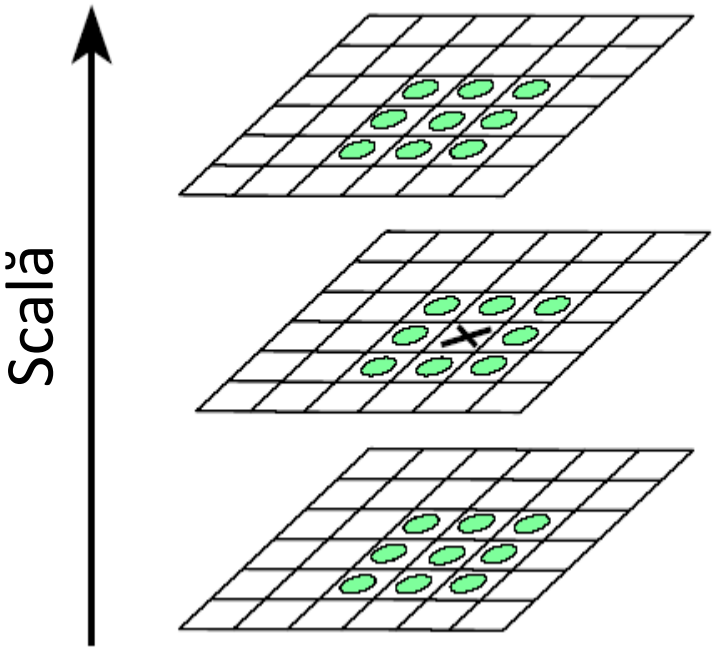


Datorită imaginilor integrale, filtrele de orice dimensiune pot fi aplicate cu aceeași viteză!

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SURF- Detecția punctelor de extrem



Dacă ***X*** este *cel mai mare* sau *cel mai mic* dintre toți vecinii, atunci ***X*** este denumit ***punct de interes (keypoint)***.

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SURF

Odată ce punctele au fost localizate, care sunt următorii pași?

(1) alocarea orientării,

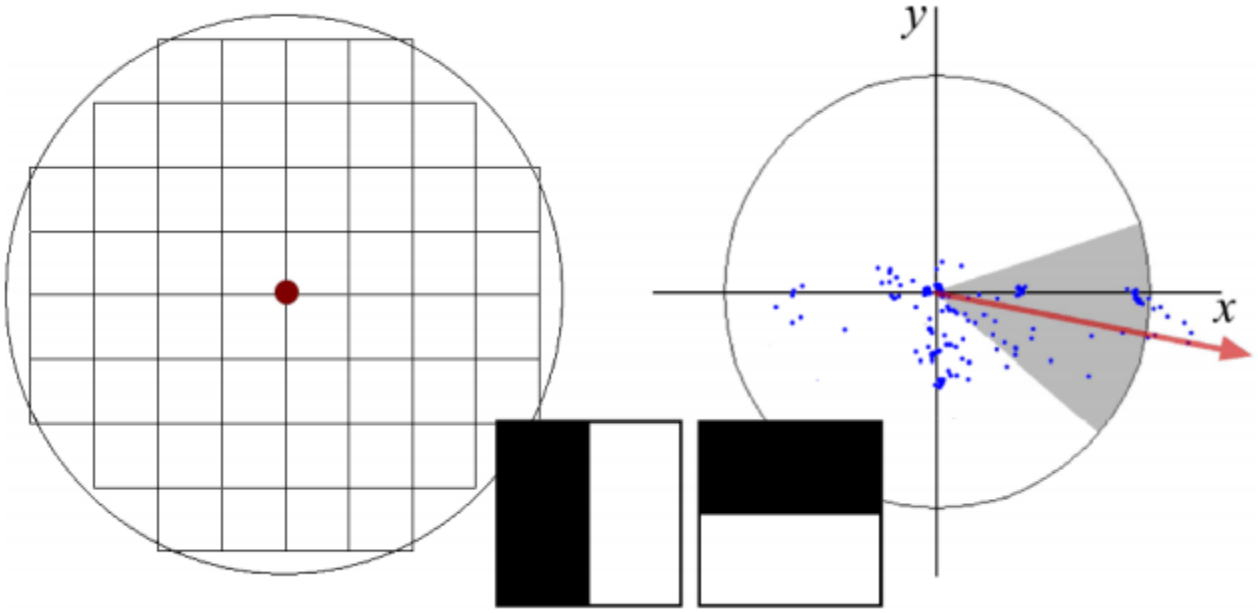
(2) extragerea trăsăturii punctului de interes.

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SURF - alocarea orientării

Se realizează o interpolare a orientării pe diferite scale:

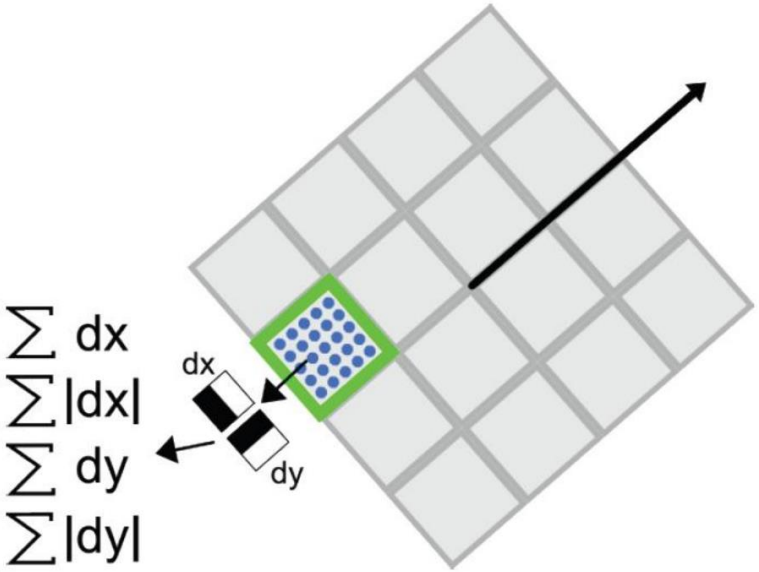


III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SURF – calculul descriptorului

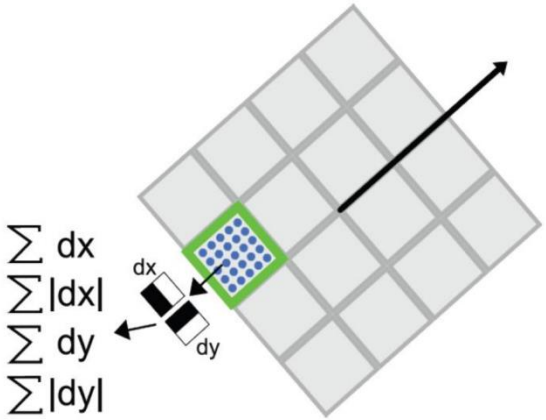
- Se împarte regiunea în 4x4 celule;
- Pentru fiecare celulă se calculează 4 valori:
  - suma răspunsului (intensitate) pe dx și dy;
  - suma modulelor răspunsului pe dx și dy;



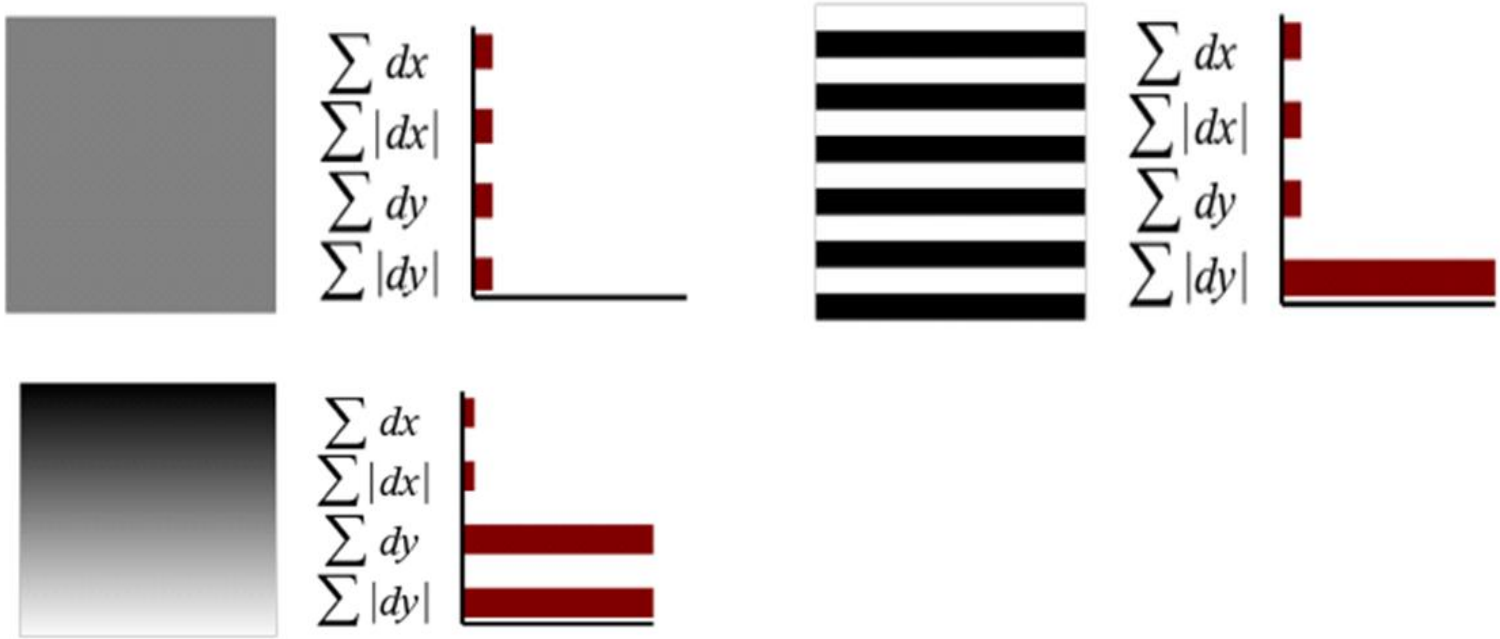
III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SURF – calculul descriptorului



- Lungimea finală a descriptorului va fi:
  - **4 x 16 = 64** elemente.
- **Exemple:**



III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SURF – calculul descriptorului

- Mai există o variantă de descriptor de 128 de elemente;
- Suma componentelor dx și |dx| sunt calculate separat pentru punctele *dy < 0* și *dy >0;*
- Similar se calculează suma pentru dy și |dy|;
- În final, lungimea descriptorului va fi:
  2 x 4 x 16 = 128 elemente
- Descriptorul final este mai descriminativ.

III. Descriptori locali

Utilizarea punctelor de interes

Algoritmul SURF – alte informații

- Mai multe informații pot fi citite în articolul original:
- Herbert Bay, Tinne Tuytelaars, și Luc Van Gool, “SURF: Speeded Up Robust Features”, European Computer Vision Conference (ECCV), 2006;
- Surse:
  - OpenSURF:

    http://www.chrisevansdev.com/computer-vision-opensurf.html,
  - OpenCV,
  - Implementări pe FPGA.

III. Descriptori locali

Utilizarea punctelor de interes

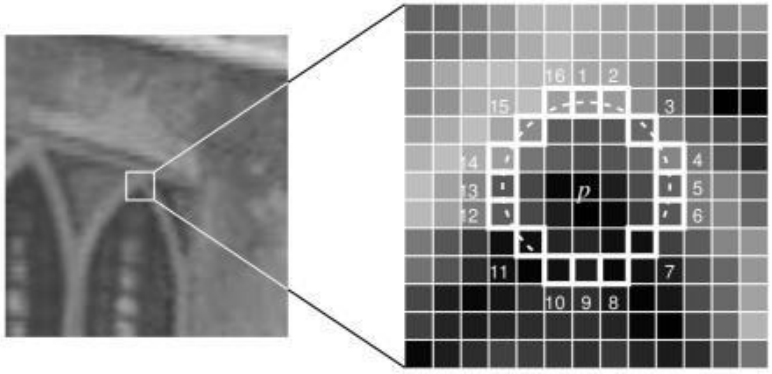
SIFT vs SURF

- Calculul algoritmului SURF este de 3-4 ori mai rapid;
- Poate fi adaptat la calcul paralel deoarece fiecare matrice hesiană poate fi estimată în mod paralel;
- SURF are o acuratelețe ușor mai scăzută;
- SURF este mai puțin invariant la schimbările de iluminare și de schimbare a punctului de observație.

III. Descriptori locali

Utilizarea punctelor de interes

FAST (Features from Accelerated Segment Test**)**



- Selectează un pixel care ar putea fi punct de interes Ip;
- Selectează o valoare de prag *t*;
- Se consideră un cerc de rază 16;
- Un pixel *p* este considerat muchie dacă există un set de *n* pixeli care sunt mai deschiși / închiși decât valoarea centrală a punctului de interes;
- Pentru a mări viteza se examinează doar 4 pixeli (eliminare pixelilor nerelevanți): pixelii 1, 9, 5 și 13 (mai întâi 1 și 9, iar apoi 5 și 13). Dacă *p* este muchie atunci minim 3 puncte trebuie să fie mai închise sau deschise decît *Ip + t / Ip − t.*

[PAMI Rosten ‘10]

III. Descriptori locali

Utilizarea punctelor de interes

FAST (Features from Accelerated Segment Test**)**

- Algoritmul este foarte rapid, dar are câteva puncte slabe:
  - Nu elimină suficienți candidați dacă n < 12;
  - Alegerea pixelilor nu este optimală deoarece eficiența este dependentă de ordinea întrebărilor și de distribuția punctelor;
  - Multe trăsături adiacente sunt găsite unul lângă altul.

[PAMI Rosten '10]

III. Descriptori locali

Utilizarea punctelor de interes

FAST – selecție optimală

- Utilizează arbori de decizie pentru detecția punctelor de interes:

- Se selectează un set de imagini pentru antrenare
- Se rulează algoritmul FAST pe fiecare imagine

- Pentru fiecare trăsătură se reține o vecinătate de 16 pixeli.

$$S_{p\to x} = \begin{cases} d, & I_{p\to x} \le I_p - t & \text{(darker)} \\ s, & I_p - t < I_{p\to x} < I_p + t & \text{(similar)} \\ b, & I_p + t \le I_{p\to x} & \text{(brighter)} \end{cases}$$

- Fiecare pixel va avea unul din următoarele stări: (deschis / similar, închis)

- Se utilizează algoritumul ID3. De fiecare dată se selectează coloana cu cea mai importantă entropie – până când entropia maximă devine zero.

- Arborele creat se utilizează pentru a detecta în mod rapid punctele de interes.

[PAMI Rosten '10]

III. Descriptori locali

Utilizarea punctelor de interes

FAST – obținerea punctelor de extrem

- Eliminarea valorilor non-maximale din zonele alăturate cu puncte de interes reprezintă o altă problemă;

- Se calculează o funcție de scor, *V* pentru toate trăsăturile extrase. *V* reprezintă suma absolută a diferențelor dintre punctul central și celălalte 16 puncte alăturate;

- Dintre două puncte alăturate se va elimina punctul cu valoare *V* mai mică.

[PAMI Rosten '10]

III. Descriptori locali

Utilizarea punctelor de interes

Alți algoritmi

- MSER („Maximally Stable Extremal Region Detector”) - [M. Donoser & H. Bischof 2006],
- STAR - [V. Agrawal et al. 2008],
- FAST – [E. Rosten & T. Drummond 2006],
- GOOD („Good Features to Track”) [J. Shi & C. Tomasi 1998],
- SUSAN [S. M. Smith & J. M. Brady 97],
- Detectorul de margini Harris [C. Stephens & M. Harris 1988].

III. Descriptori locali

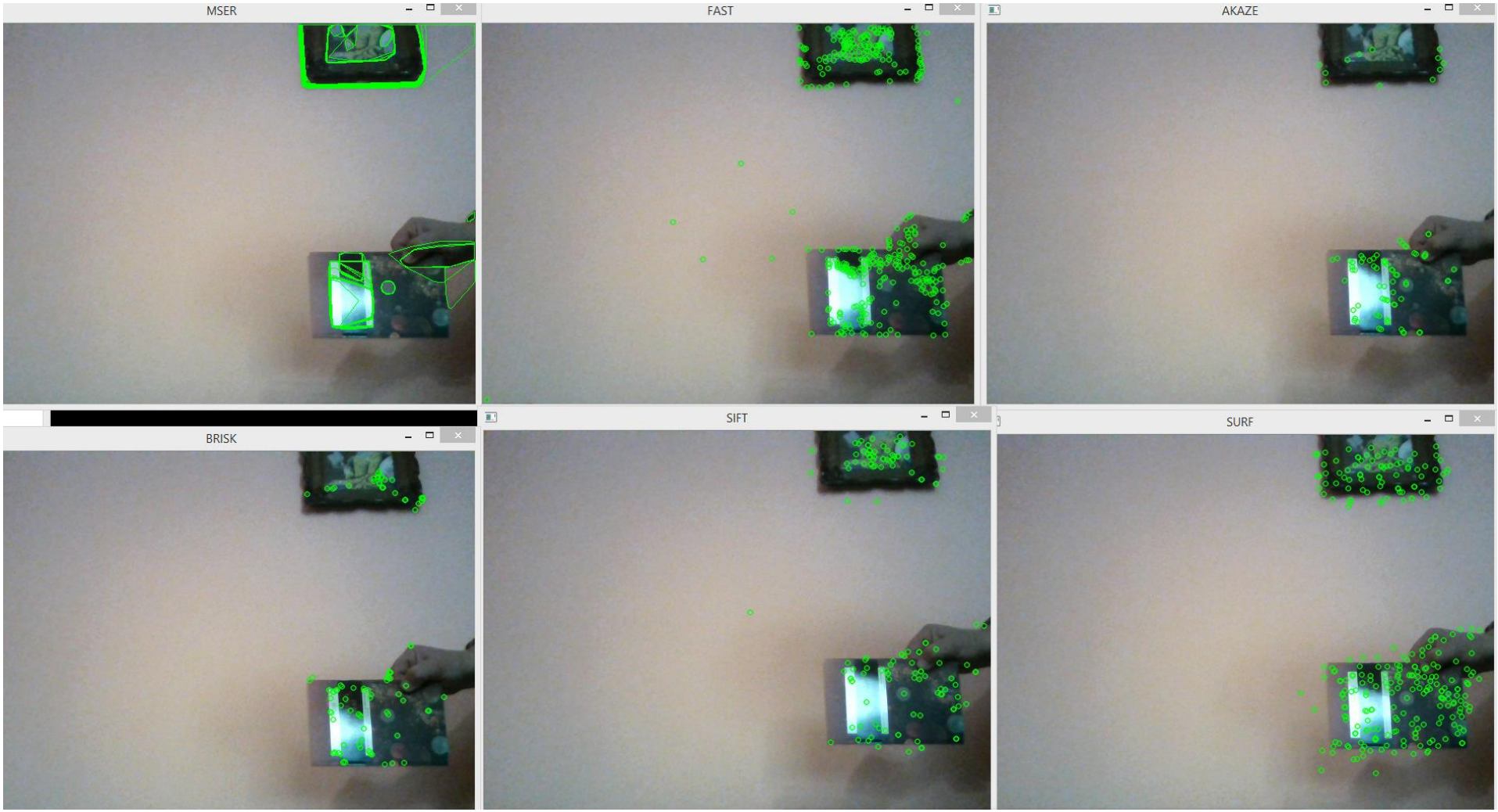
Utilizarea punctelor de interes

Demo

[https://github.com/imironica/IVOM-Demo/tree/master/IVOM_Demo/KeypointsDetector]

III. Descriptori locali

Utilizarea punctelor de interes



III. Descriptori locali

Utilizarea punctelor de interes

Căutarea cu puncte de interes

Algoritm general

(1) se detectează regiunile în care se extrag punctele de interes;

(2) pentru fiecare punct de interes se definește o regiune în jurul acestuia. Pentru fiecare regiune se calculează un descriptor;

(3) se calculează o distanță între lista de puncte de interes din șablonul de antrenare și celelalte cadre.

III. Descriptori locali

Utilizarea punctelor de interes

Potrivirea punctelor de interes

Se utilizează algoritmul Nearest Neighbor (cei mai apropiați vecini) cu distanță euclidiană:

- Se verifică care din punctele din prima imagine se potrivesc cu imaginile din cea de-a doua imagine;
- Totuși este puțin probabil să găsim trăsături identice dintre două imagini;
- Pentru a rezolva problema trăsăturilor care nu se potrivesc perfect (ex: zgomot de fundal, mișcare) se va calcula o distanță dintre trăsături:
  - cel mai apropiat vecin;
  - în cazul în care distanța este mai mică de un prag global va exista o potrivire (nu funcționează intotdeauna bine);
  - un raport în funcție de primii 2 vecini.

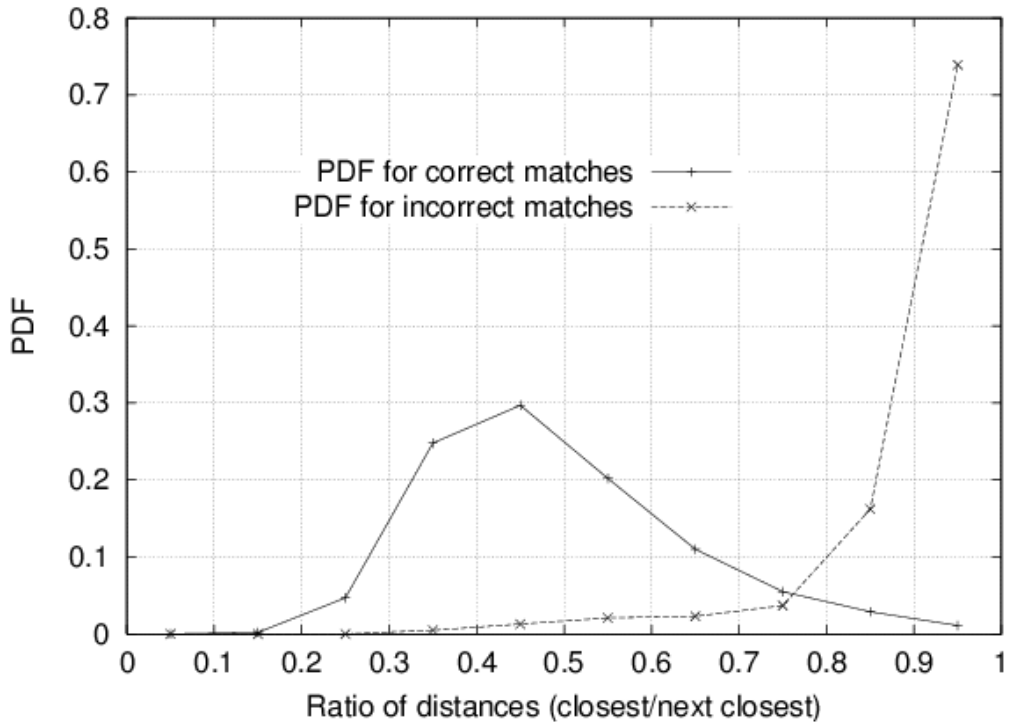
III. Descriptori locali

Utilizarea punctelor de interes

Potrivirea punctelor de interes

$$\frac{cel\ mai\ apropiat\ vecin}{al\ doilea\ cel\ mai\ apropiat\ vecin} < 0.8$$

Elimină 90% din potrivirile false în timp ce eroarea de potrivire este mai mică de 5%.



III. Descriptori locali

Utilizarea punctelor de interes

Potrivirea punctelor de interes

- Nu există algoritmi foarte eficienți pentru spații de dimensiuni mari (mai mari de 9-10 dimensiuni);

- Un algoritm utilizat pentru calculul eficient este Best-Bin-First (BBF) [Beis and Lowe, 1997];

- Returnează cei mai apropiați vecini cu o probabilitate mare:

  Preț computațional scăzut – căutarea se termină după ce un număr suficient de potriviri este găsit.

  Suficient de bun – avem nevoie doar de primele 2 elemente pentru a calcula raportul.

III. Descriptori locali

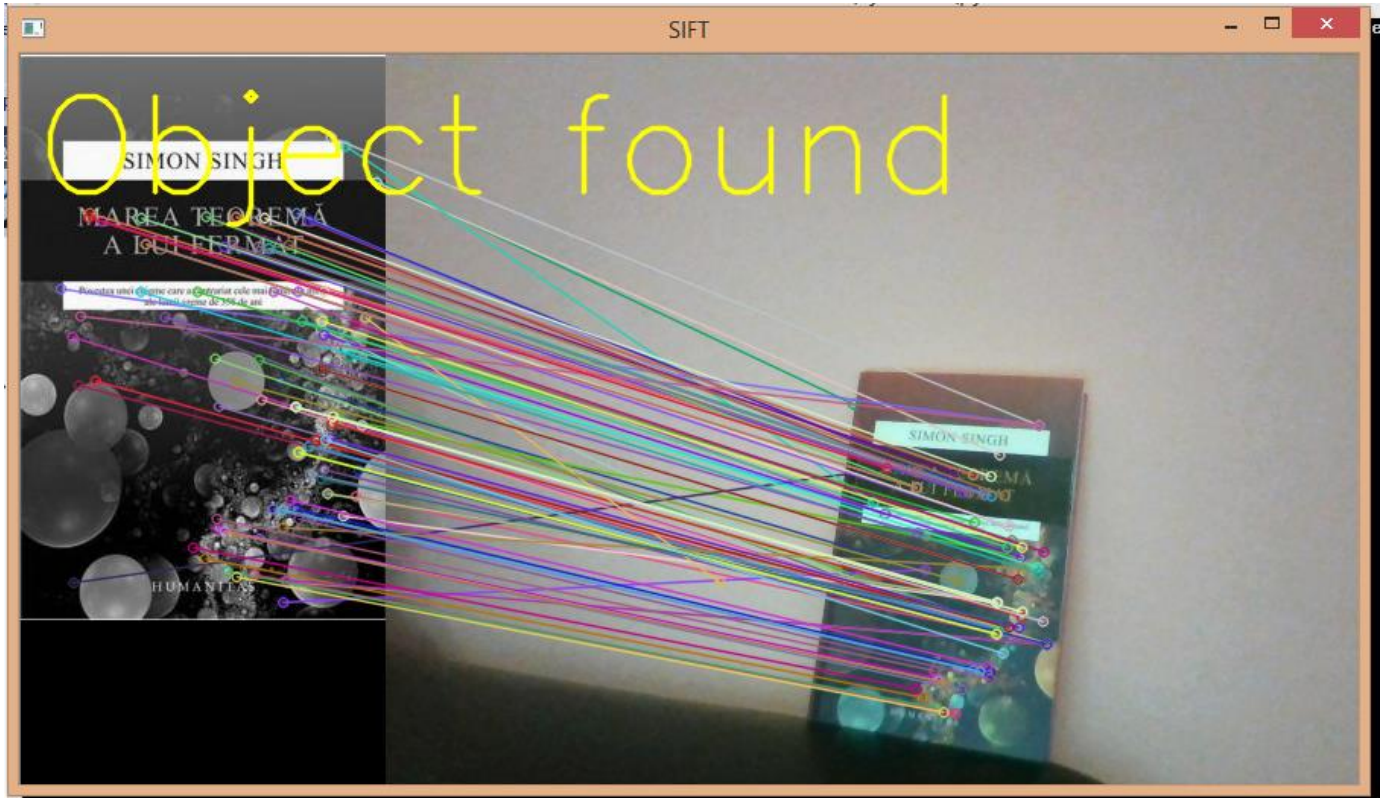
Utilizarea punctelor de interes

Demo

[https://github.com/imironica/IVOM-Demo/tree/master/IVOM_Demo/KeypointsDetector]

III. Descriptori locali

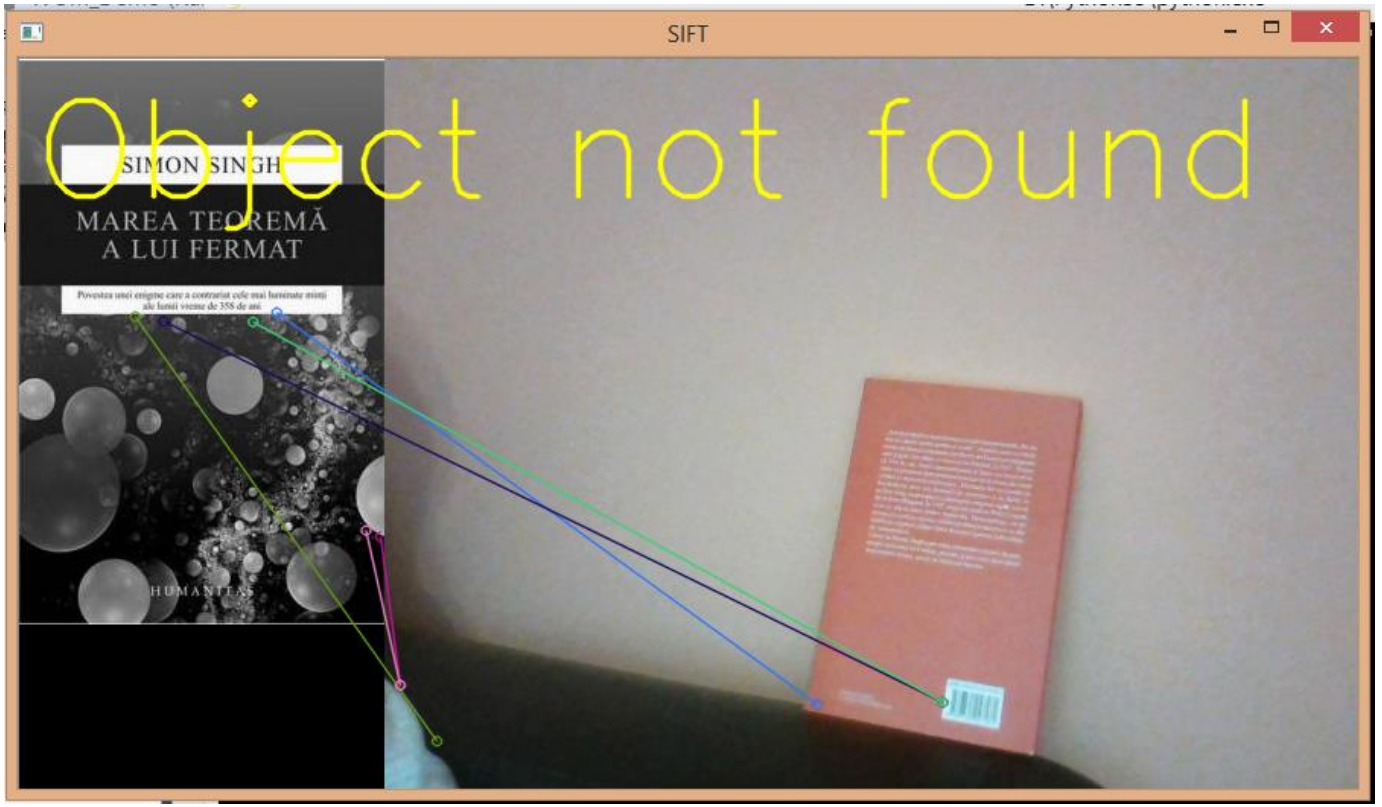
Utilizarea punctelor de interes



[https://github.com/imironica/IVOM-Demo/tree/master/IVOM_Demo/KeypointsDetector]

III. Descriptori locali

Utilizarea punctelor de interes



[https://github.com/imironica/IVOM-Demo/tree/master/IVOM_Demo/KeypointsDetector]

III. Descriptori locali

Agregarea punctelor de interes

Căutarea cu puncte de interes

Algoritmul de căutare poate fi unul destul de consumator de resurse de calcul.

Posibilă soluție:

- Agregarea punctelor de interes într-un descriptor global.

III. Descriptori locali

Modelul Bag-of-Words

Obiect → **Bag of 'words'**

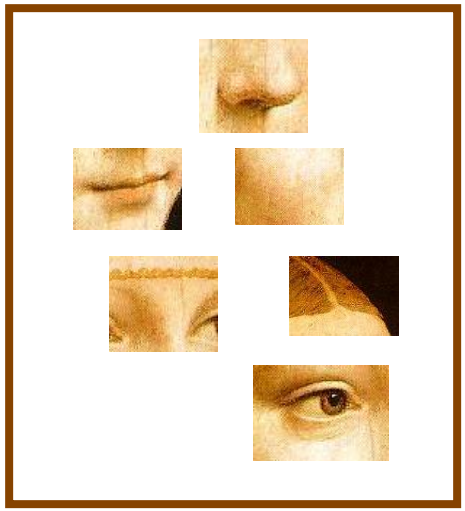
- Extragere trăsături (puncte de interes / descriptori de mișcare)
- Învățarea unui dicționar (vizual / de mișcare)
- Cuantizaea trăsăturilor prin utilizarea vocabularului
- Reprezentarea cadrelor prin utilizarea frecvenței de apariție cuvintelor

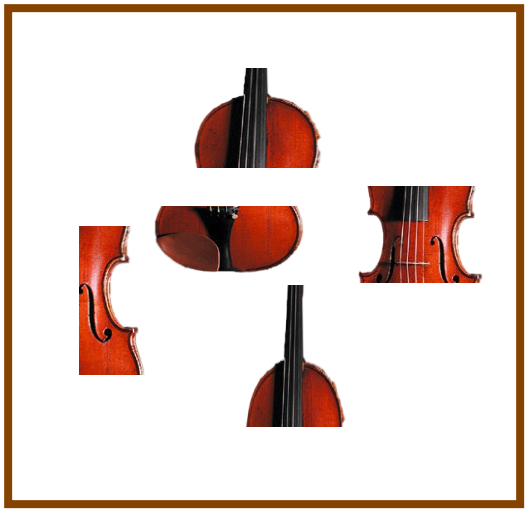
[Slide-uri adaptate din prezentările lui Rob Fergus, Svetlana Lazebnik și Noah Snavely]

III. Descriptori locali

Modelul Bag-of-Words

- Extragere trăsături (puncte de interes / descriptori de mișcare)





III. Descriptori locali

Modelul Bag-of-Words

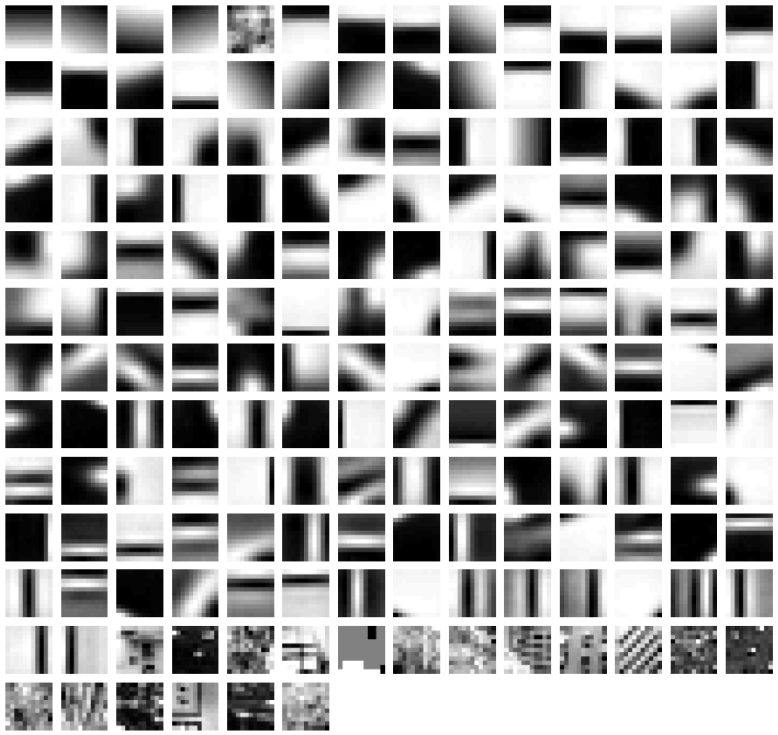
- Învățarea unui dicționar (vizual / de mișcare)

- K-means
- Clusterizare ierarhică
- Gaussian Mixture Model
- Arbori aleatori

III. Descriptori locali

Modelul Bag-of-Words

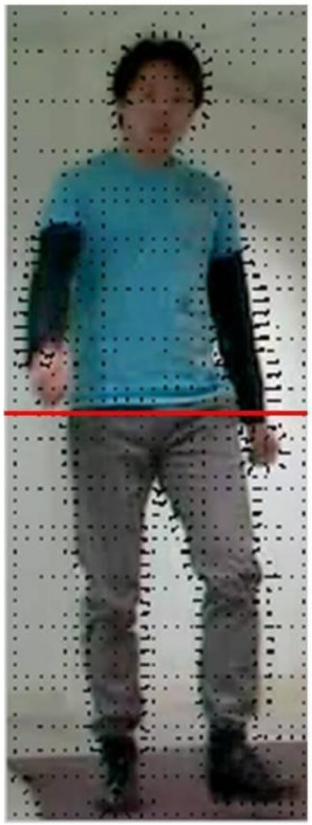
- Învățarea unui dicționar (vizual / de mișcare)



III. Descriptori locali

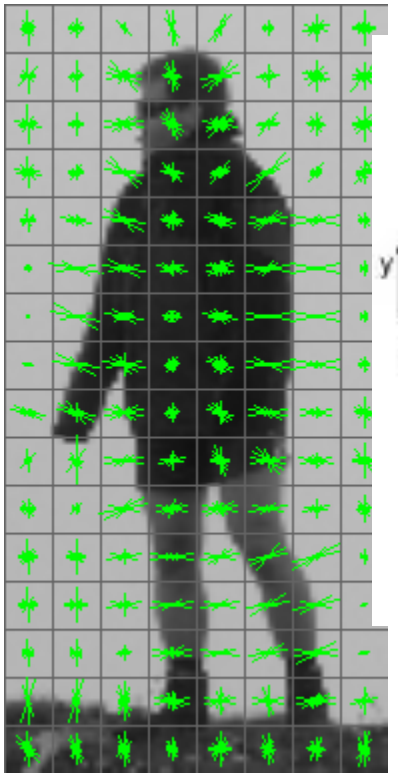
Modelul Bag-of-Words

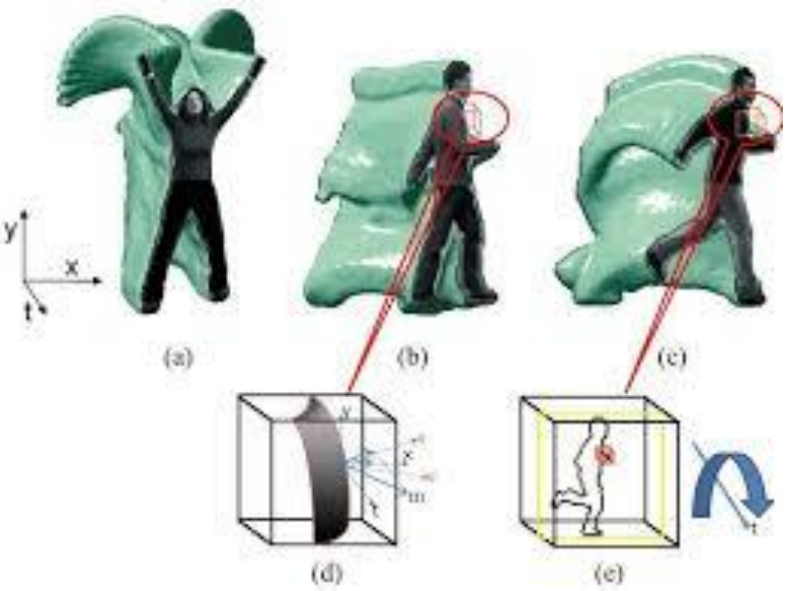
- Învățarea unui dicționar (vizual / de mișcare)



(a)

(b)

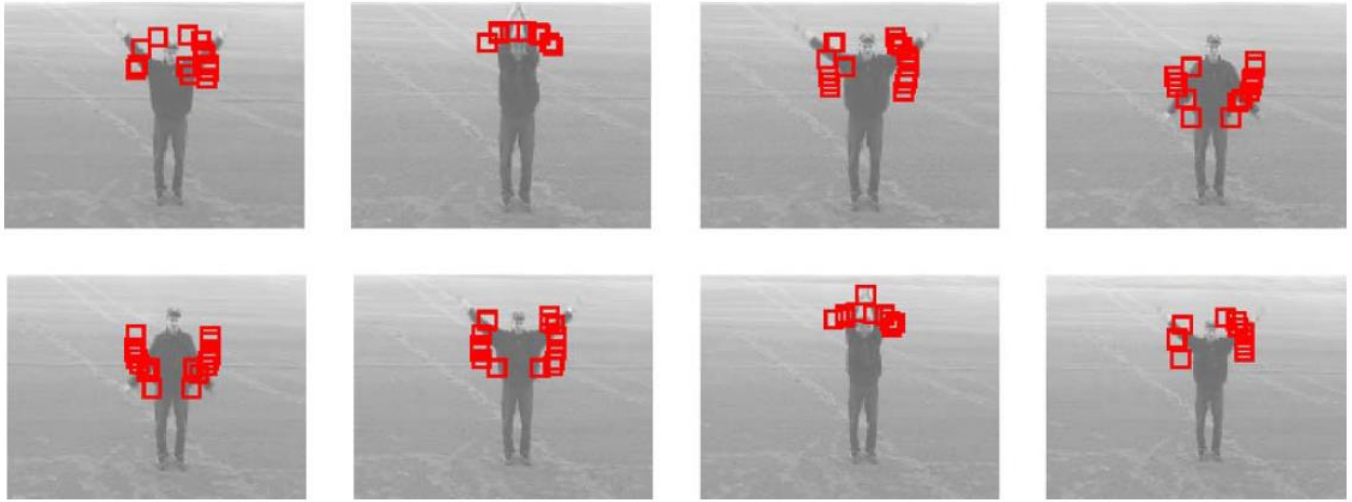


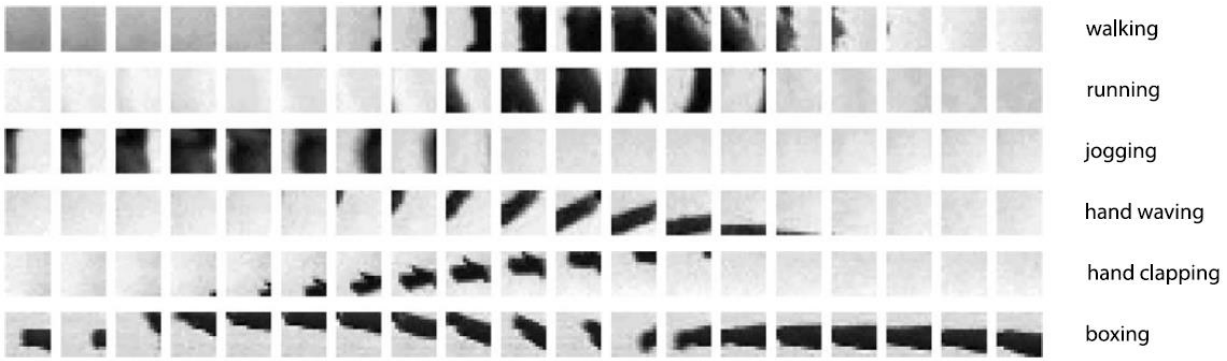


III. Descriptori locali

Modelul Bag-of-Words

- Învățarea unui dictionar (vizual / de miscare)





J.C. Niebles, H. Wang, L. Fei-Fei: Unsupervised learning of human action categories using spatial-tempotal words. IJCV, 79(3): 299-318, 2008.

III. Descriptori locali

Modelul Bag-of-Words

- Cat de mare trebuie să fie un dicționar?

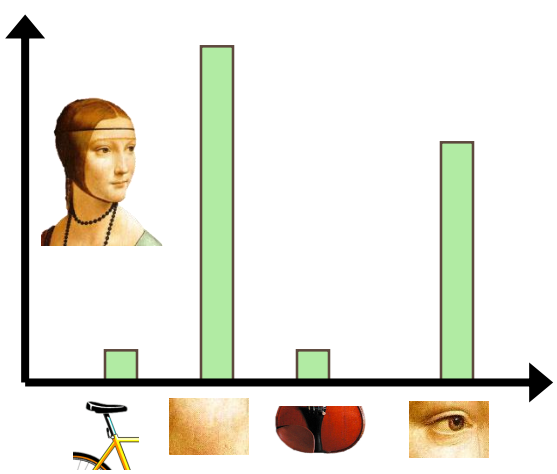
  - Prea mic: numărul de cuvinte vizuale nu vor fi reprezentative pentru toate conceptele
  - Prea mare: supra-învățare (overfitting)

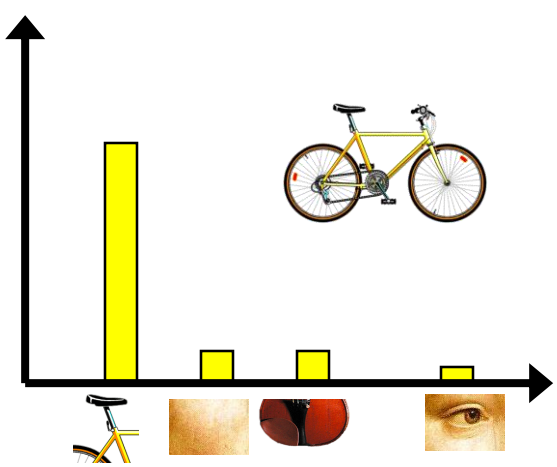
  Există metode care propun de la câteva sute de cuvinte la sute de mii de cuvinte.

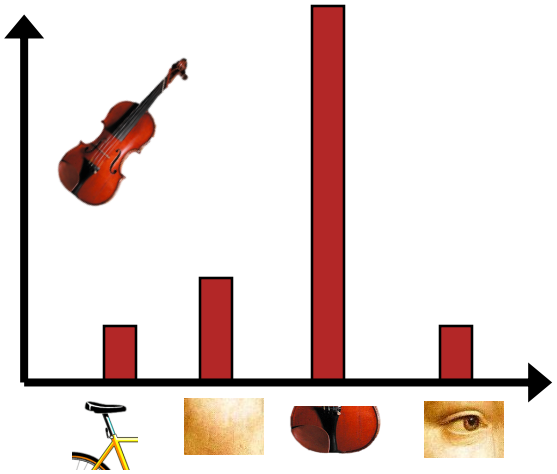
III. Descriptori locali

Modelul Bag-of-Words

- Reprezentarea cadrelor prin utilizarea frecvenței de apariție cuvintelor

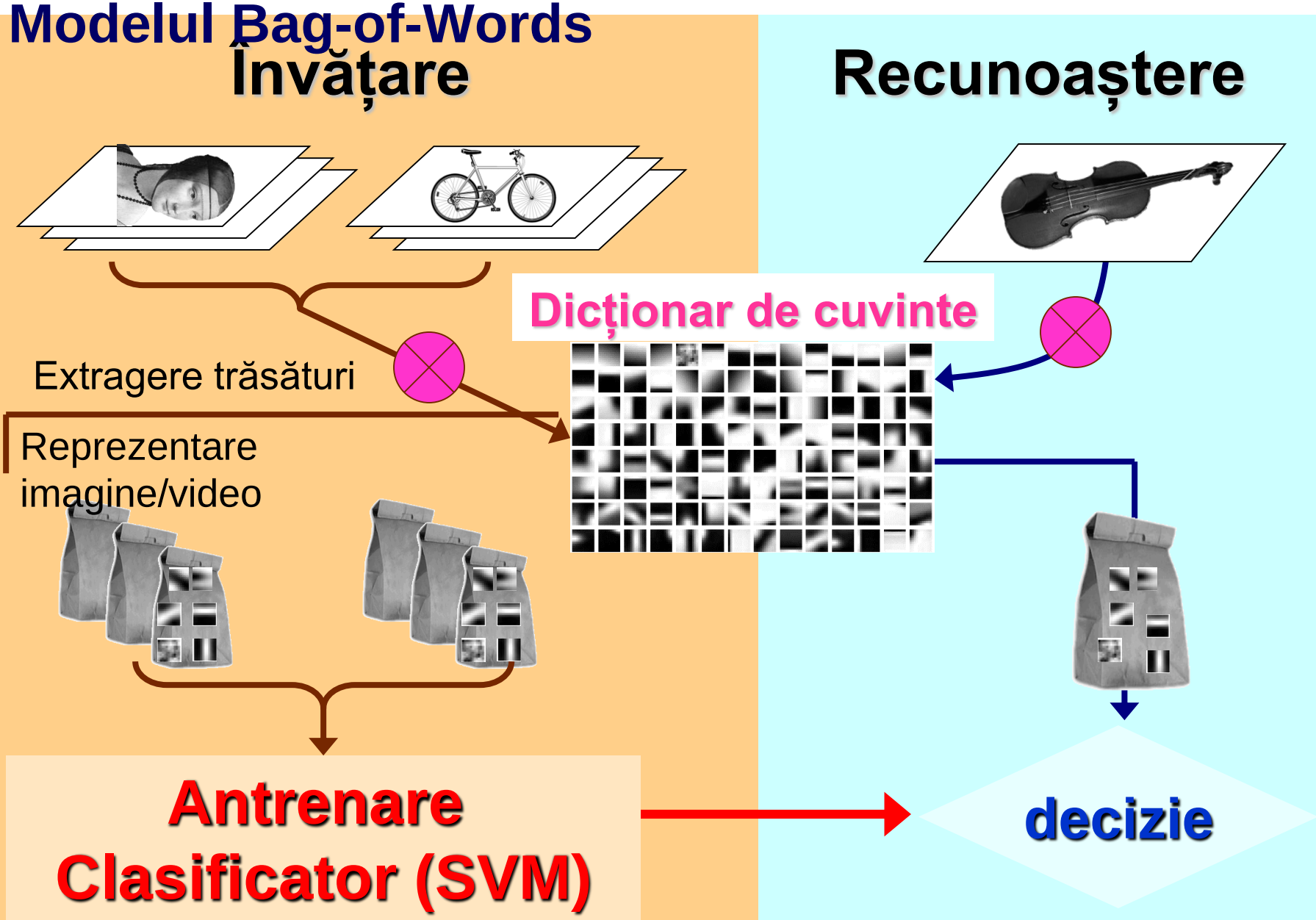






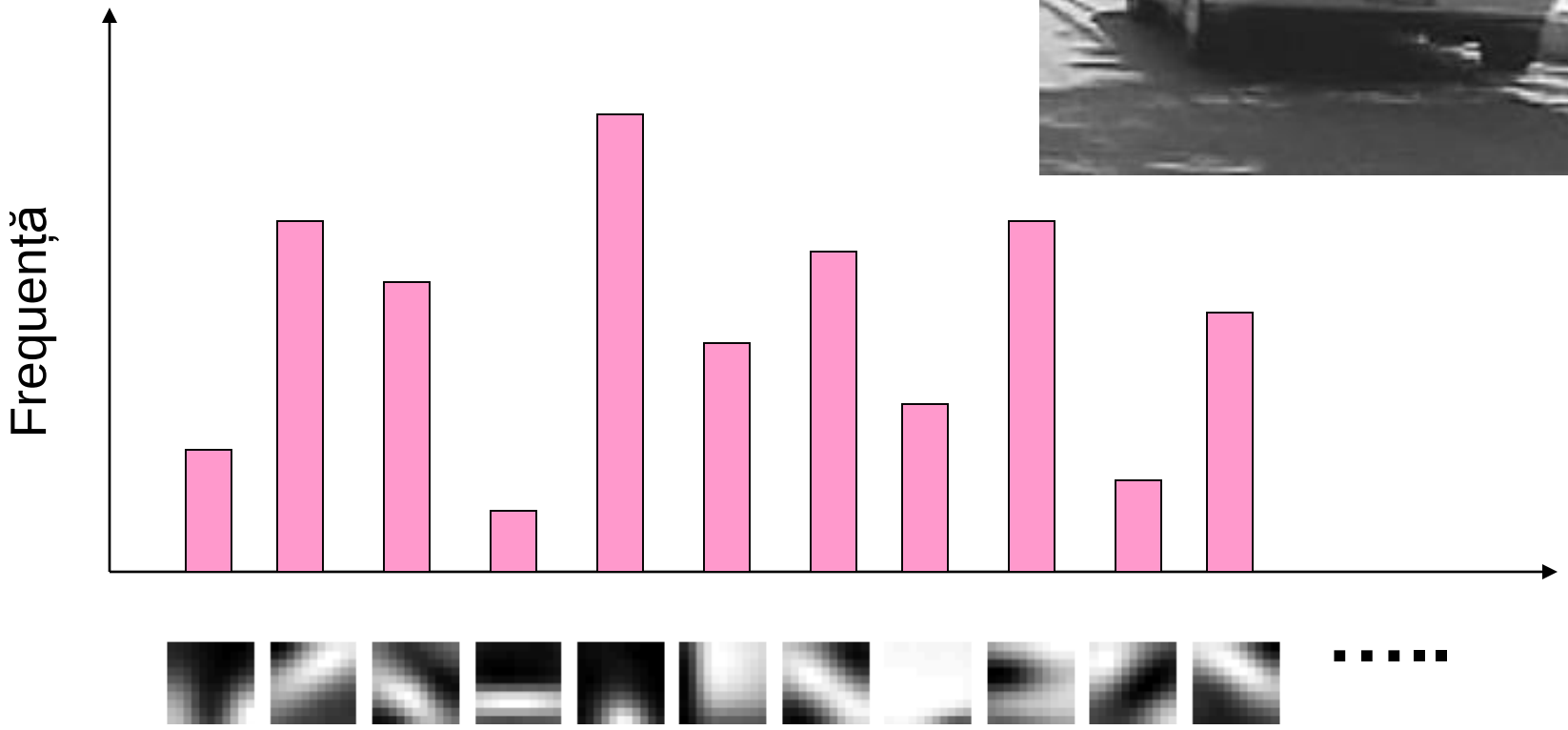
III. Descriptori locali

Modelul Bag-of-Words



III. Descriptori locali

Modelul Bag-of-Words



III. Descriptori locali

Modelul Bag-of-Words

Descriptori - Bag of Words

- Au rezultate bune atunci când obiectele sunt asemănătoare

- Dar ce facem cu scaunele?

III. Descriptori locali

Modelul Bag-of-Words

Dezavantaje model „Bag of Words”

nu există nici o metodă riguroasă de reprezentare a distribuției spațiale dintre anumite perechi de cuvinte.

există multe cuvinte care nu sunt relevante

procesul de cuantizare a cuvintelor generează zgomot de cuantizare.

costul computațional crește foarte mult odată cu dimensiunea vocabularului de cuvinte.

III. Descriptori locali

Modelul Fisher kernel

FK vs BoW

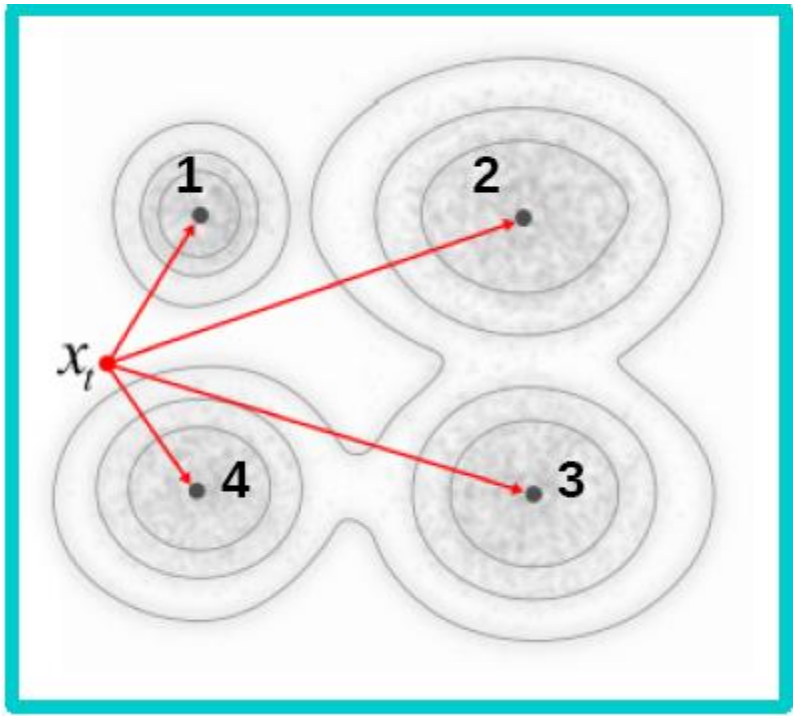
Bag of Words

conține apartenența

fiecărui punct proeminent către

un element al unui dicționar

(histogramă de cuvinte)



Rezultat: D = [0;0;0;1];

Dimensiune: K

III. Descriptori locali

Modelul Fisher kernel

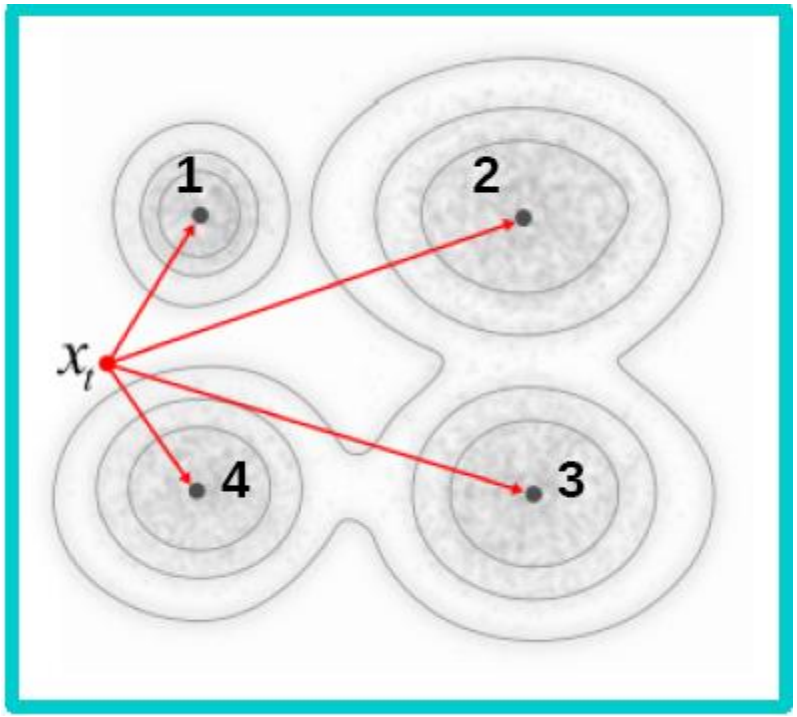
FK vs BoW

Fisher Kernels

Calculează probabilitățile de apartenență la un cuvânt din dicționar

Rezultat: D = [0.3;0.1;0.1;0.5];

- calculează gradientul mediei și a variației probabilităților de apartenență la un cuvânt din dicționar.



Dimensiune: 2*D*K

III. Descriptori locali

Modelul Fisher kernel

Îmbunătățiri FK

Normalizare L2

- elimină erorile ce apar din diferența de scală a obiectelor

Normalizare de putere (Power Normalization)

- eliminarea efectului de matrice rară (majoritatea elementelor din FK au valori foarte mici)

$$f(z) = \mathrm{sign}(z)|z|^{\alpha}$$

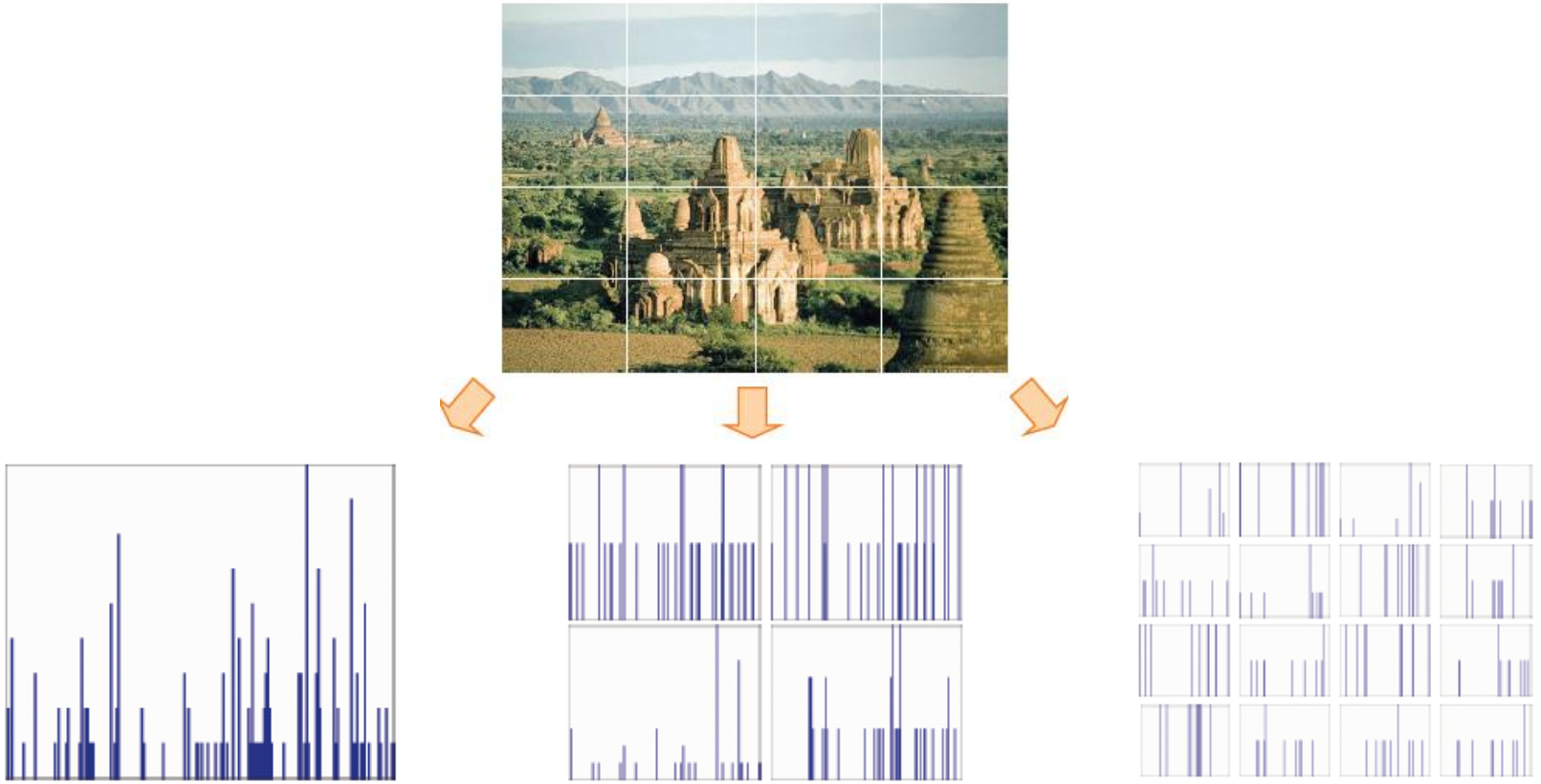
Aplicare Piramide Spațiale

- utilizează informația geometrică a obiectelor

III. Descriptori locali

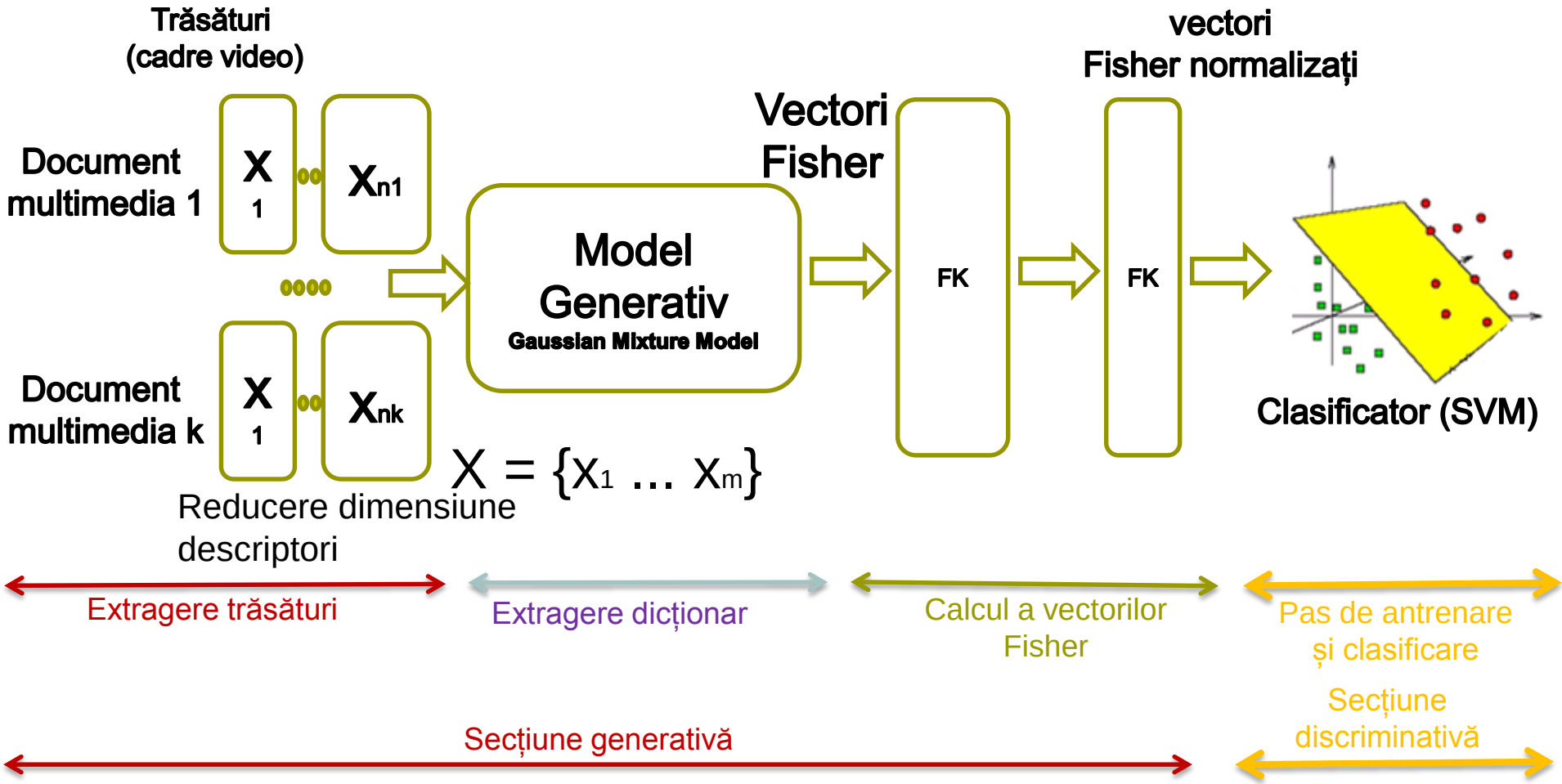
Modelul Fisher kernel

Piramide spațiale



Modelul Fisher kernel

Arhitectura reprezentării „Fisher kernel”

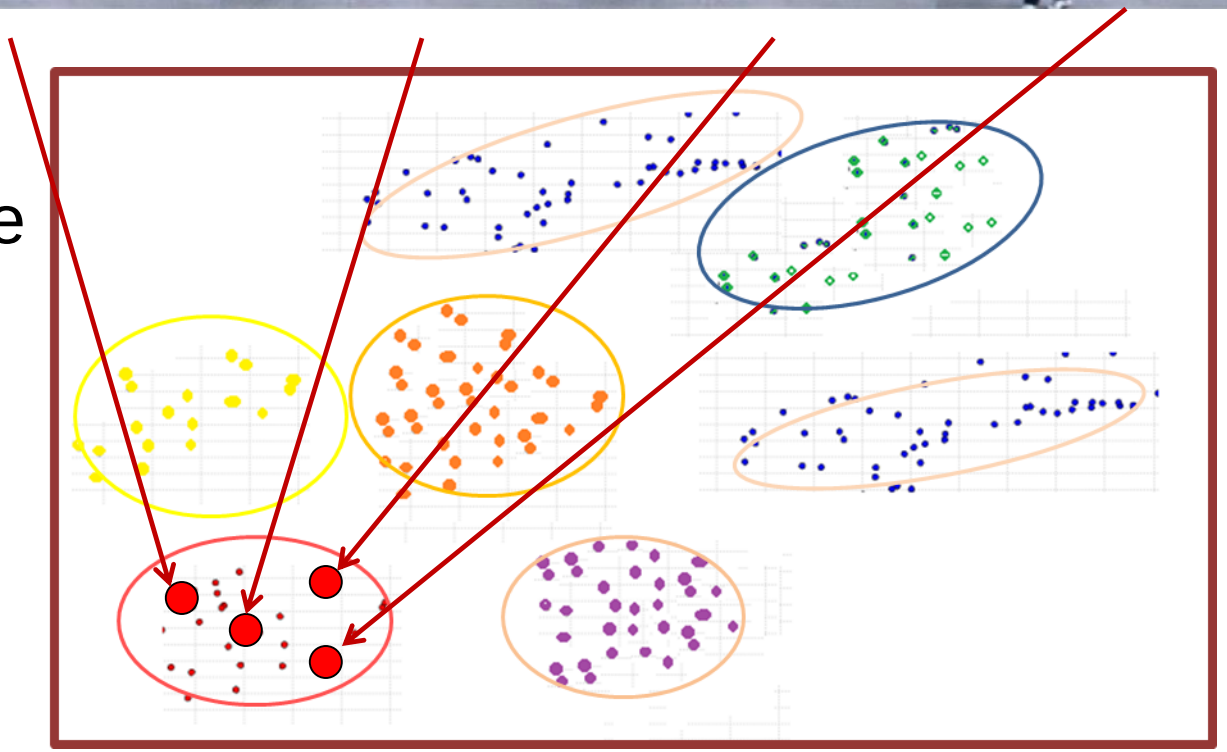


[Mironică et al., ICMR'13 ACM]

Modelul Fisher kernel

Agregarea cadrelor cu reprezentarea „Fisher kernel”

Cadrele similare vor face parte din aceeași componentă, modelând variațiile subtile de timp.



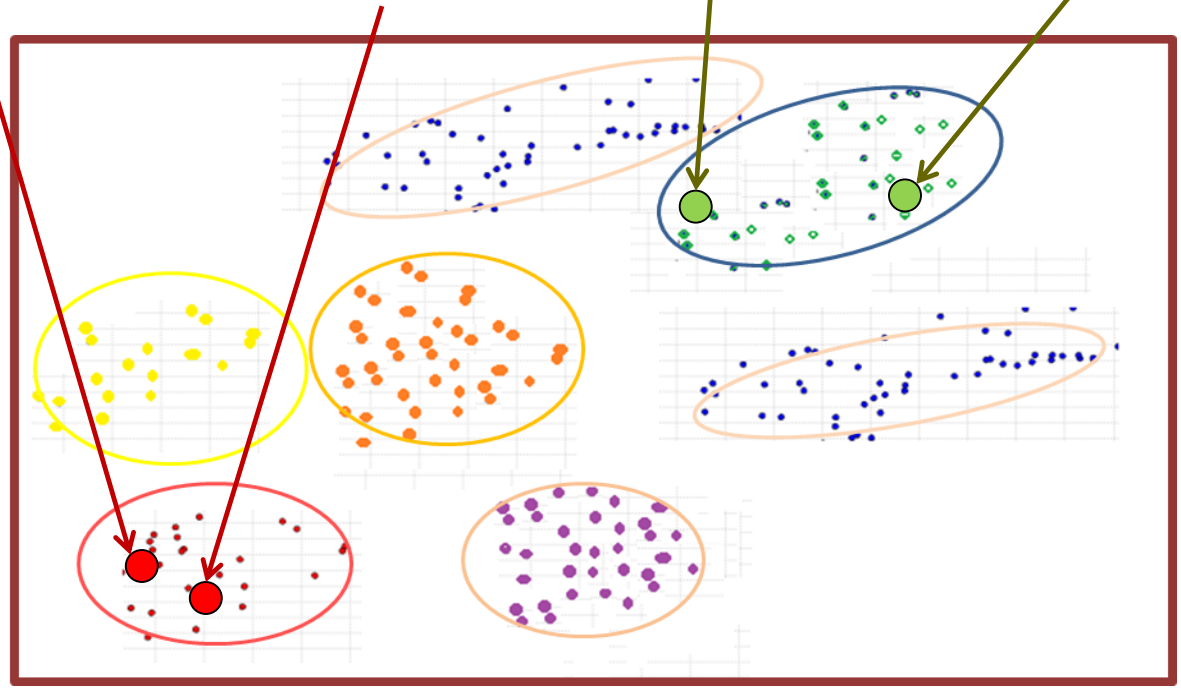
Reprezentare „Fisher kernel”

[Mironică et al., Multimedia’13 ACM]

Modelul Fisher kernel

Agregarea cadrelor cu reprezentarea „Fisher kernel”

Cadrele nesimilare vor face parte din componente separate, prevenind amestecarea conceptelor nesimilare.

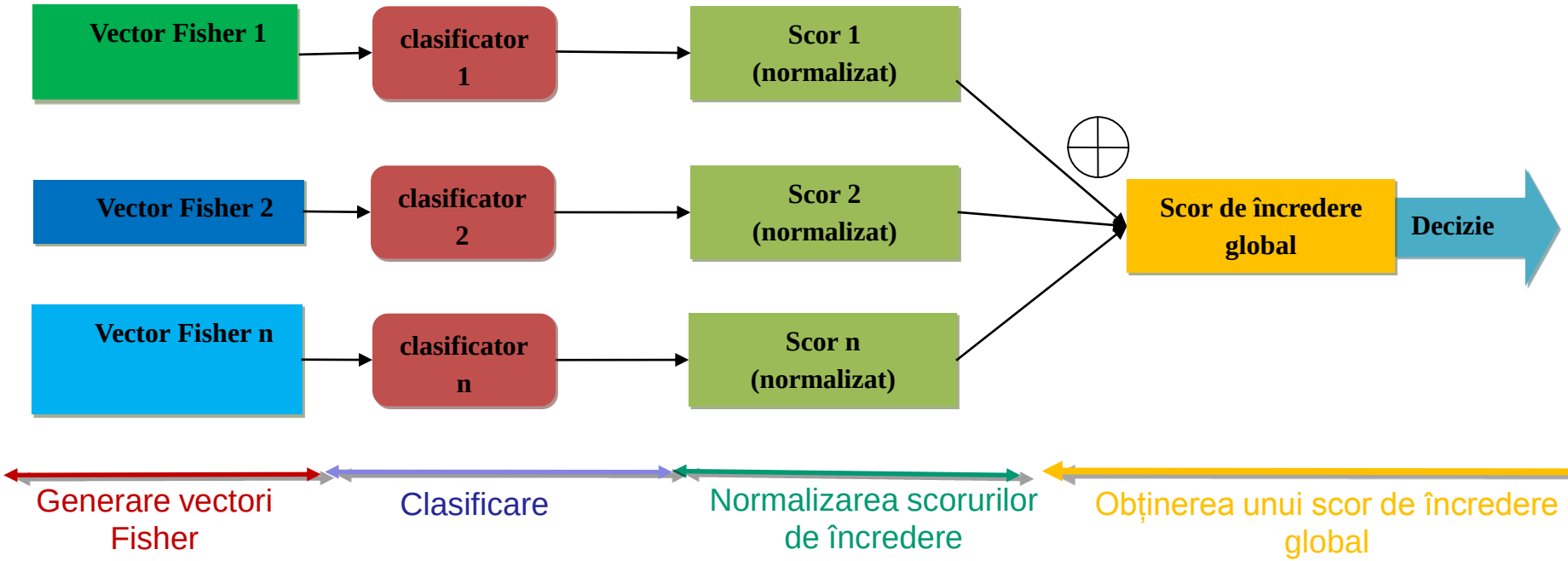


Reprezentare „Fisher kernel”

[Mironică et al., Multimedia’13 ACM]
[Mironică et al., ICMR’13 ACM]

Modelul Fisher kernel

Fuziunea trăsăturilor – „Late Fusion”



[Mironică et al., CBMI 2013, IEEE/ACM]

Urmărirea traiectoriei

III. Urmărirea traiectoriei

Urmărirea traiectoriei

Trăsături utilizate

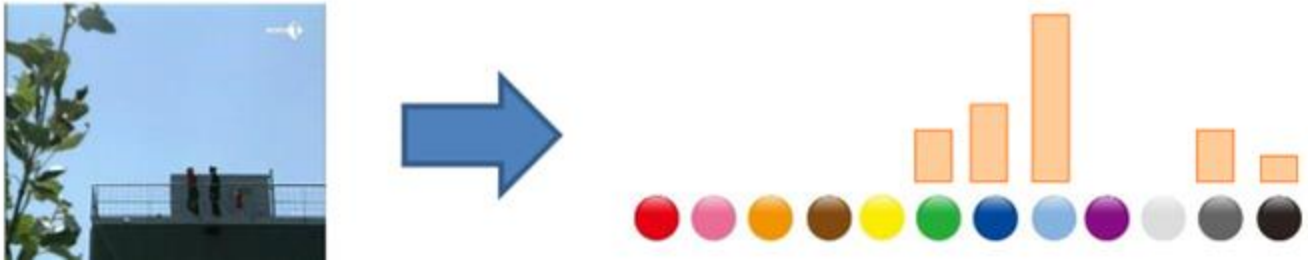
- Culoare
- Textură
- Unghiuri

III. Urmărirea traiectoriei

Urmărirea traiectoriei

Trăsături utilizate

- Culoarea: spațiile de culoare: RGB, L∗u∗v∗, L∗a∗b∗, HSV, etc. Nu există o „rețetă” pentru care spațiu de culoare este mai bun de utilizat, există o varietate mare de spații de culoare utilizate.

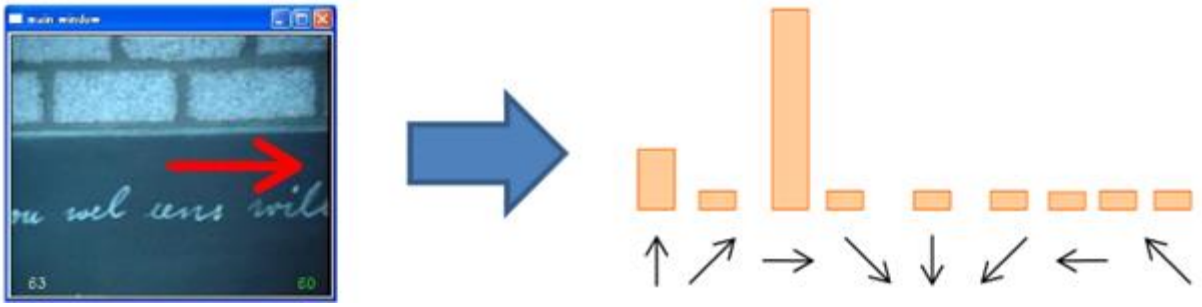


III. Urmărirea traiectoriei

Urmărirea traiectoriei

Trăsături utilizate

- Muchii: mai puțin sensibile la schimbările de iluminare în comparație cu trăsăturile de culoare. Datorită simplității, cel mai popular detector de muchii este detectorul Canny. După calculul muchiilor se va calcula o histogramă de muchii.



III. Urmărirea traiectoriei

Urmărirea traiectoriei

Trăsături utilizate

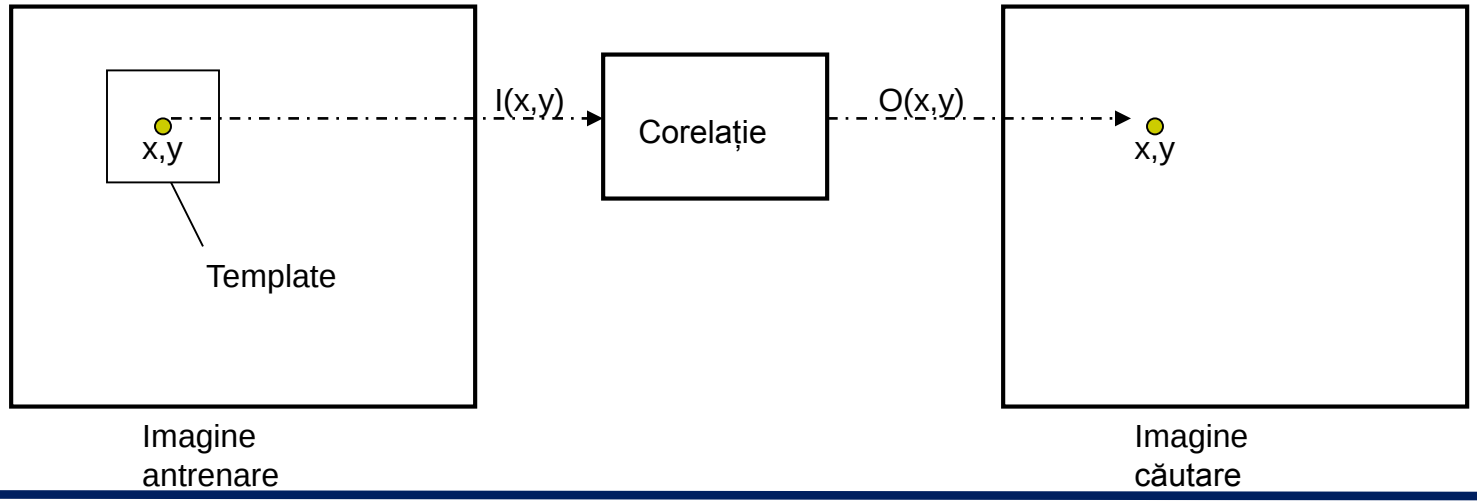
- Textura: reprezintă un concept foarte vast, atribuit oricărei suprafeţe naturale;
- În general, textura reprezintă o structură de suprafaţă spaţial repetitivă, formată prin repetiţia de elemente în diverse poziţii relative;
- Repetiţia poate implica variaţii locale de scală, orientare şi rotaţie;
- O textură este definită de următoarele trăsături: asprime, fineţe, granularitate, liniaritate, direcţionalitate, rugozitate, regularitate, nivel haotic.

III. Urmărirea traiectoriei

Urmărirea traiectoriei

Căutarea prin șablon

- Căutarea noii poziții se face utilizând metoda brute-force (se caută în toate pozițiile posibile);
- Pentru fiecare poziție se calculează un scor al similarității;
- Daca scorul este mai pare decât un prag acesta va fi eliminat ca și posibilă locație;
- Se poate utiliza atunci când deviația standard a modelului folosit este minimă.



III. Urmărirea traiectoriei

Urmărirea traiectoriei

Căutarea prin șablon

Algoritm:

- se definește o zonă de căutare;
- se așează template-ul definit în cadrul anterior pe fiecare poziție și se calculează distanța dintre candidat și șablon;
- se selectează cel mai bun candidat, care va fi și noua poziție.

Limitări ale căutării cu șablon:

- Cost computațional foarte ridicat datorită metodei brute-force;
- Se limitează căutarea doar în vecinătatea lui (datorită volumului de ridicat calcul)

III. Urmărirea traiectoriei

Urmărirea traiectoriei

Căutarea prin șablon

Corelația este o măsură a gradului de asemănare dintre două variabile. În cazul de față, cele două variabile pot fi reprezentate și de pixelii celor două regiuni.

$$cor = \frac{\sum_{i=0}^{N-1}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1}(x_i - \bar{x})^2 \cdot \sum_{i=0}^{N-1}(y_i - \bar{y})^2}}$$

X reprezintă valorile pixelilor din template

$\bar{x}$ reprezintă media pixelilor din template

y reprezintă pixelii din imaginea de căutare

$\bar{y}$ reprezintă media pixelilor din imaginea de căutare

N reprezintă numărul de pixeli a șablonului (N = nr coloane * nr rânduri)

Valoarea funcției cor este între –1 și +1, unde valori apropiate de 1 reprezintă un grad de corelare ridicat.

[R. Miezianko '08]

III. Urmărirea traiectoriei

Detecția traiectoriei

Căutarea prin șablon

Comparație directă: între templateul t(i,j) și regiunea candidat g(i,j)

$$d_1(t,g) = \sum_{i=1}^{m}\sum_{j=1}^{n}|g(i,j)-t(i,j)|$$

$$d_2(t,g) = \sum_{i=1}^{m}\sum_{j=1}^{n}(g(i,j)-t(i,j))^2$$

$$d_3(t,g) = \sum_{i=1}^{m}\sum_{j=1}^{n}g(i,j)\cdot t(i,j)$$

Comparație între distribuții: utilizând distanța Bhattacharyya

$$bd(H_1,H_2) = \sqrt{1-bc(H_1,H_2)}$$

$$bc(H_1,H_2) = \sum_{i=1}^{n}\sqrt{H_1(i)\cdot H_2(i)}$$

III. Urmărirea traiectoriei

Detecția traiectoriei

Căutarea prin șablon – exemple de aplicații

Urmărirea traiectoriei capului (se calculează o diferenţă a nivelelor de gri ale pixelilor)

(C/C++ cod sursă: http://robotics.stanford.edu/~birch/headtracker/)

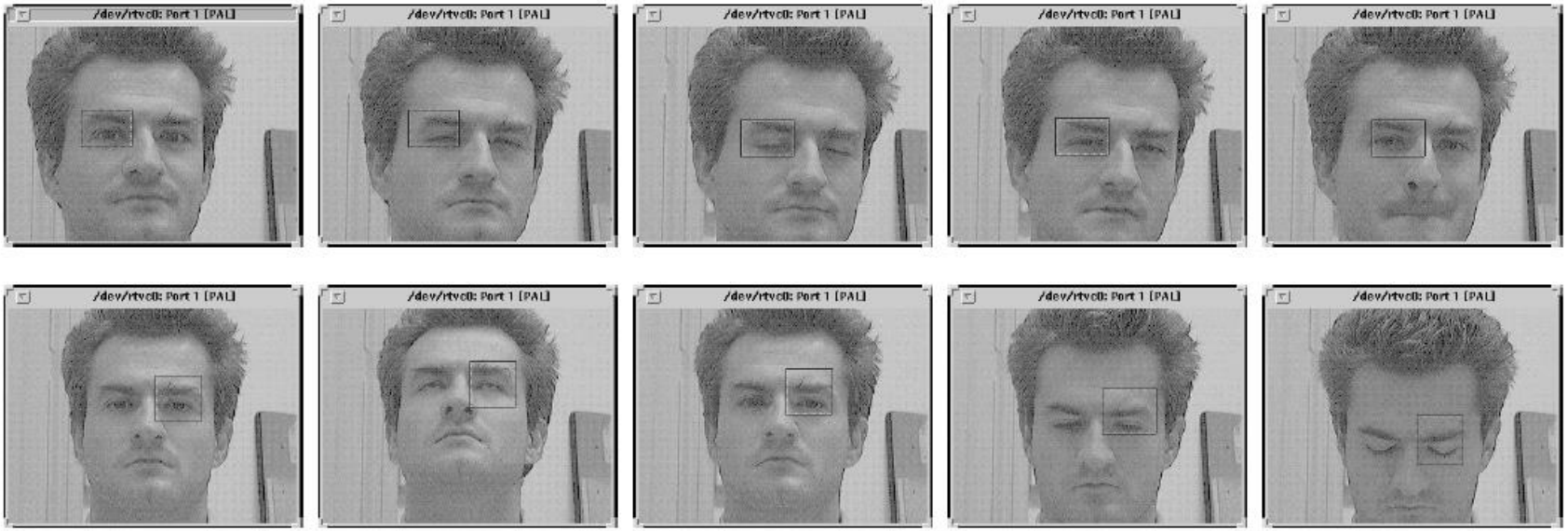
[S. Birchfield '98]

III. Urmărirea traiectoriei

Detecția traiectoriei

Căutarea prin șablon – exemple aplicații

Urmărirea traiectoriei ochiului (diferențe de nivele de gri ale pixelilor)



[R. Miezianko '08]

III. Urmărirea traiectoriei

Detecția traiectoriei

Căutarea prin șablon

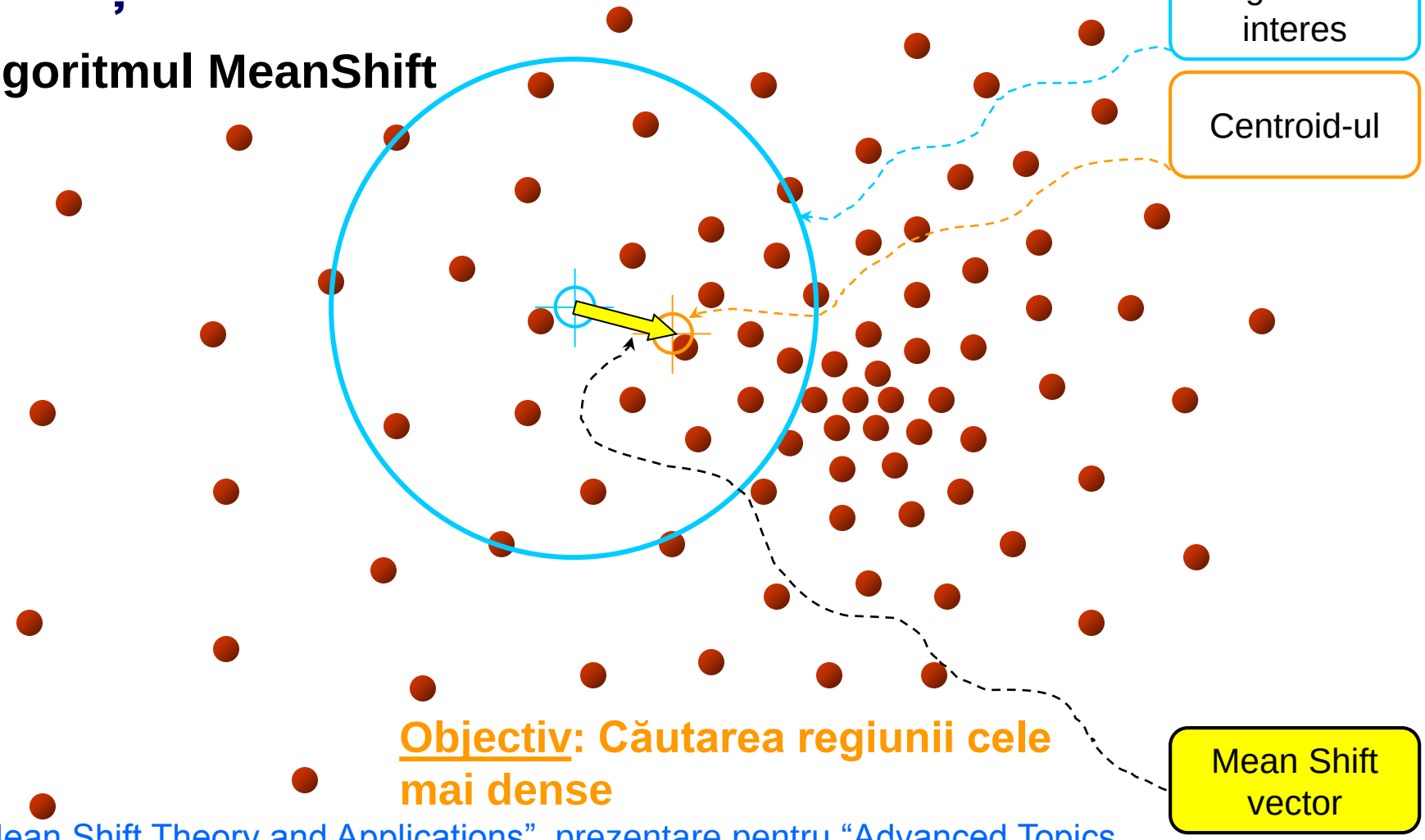
Alte surse de bibliografie:

- H. Schweitzer, J.W. Bell, F. Wu: *Very fast template matching*. Proc. of ECCV (4): 358-372, 2002.
- H. Schweitzer, R.A. Deng, R.F. Anderson: *A dual-bound algorithm for very fast and exact template matching*. IEEE-TPAMI, 33(3): 459-470, 2011.

III. Urmărirea traiectoriei

Detecția traiectoriei

Algoritmul MeanShift



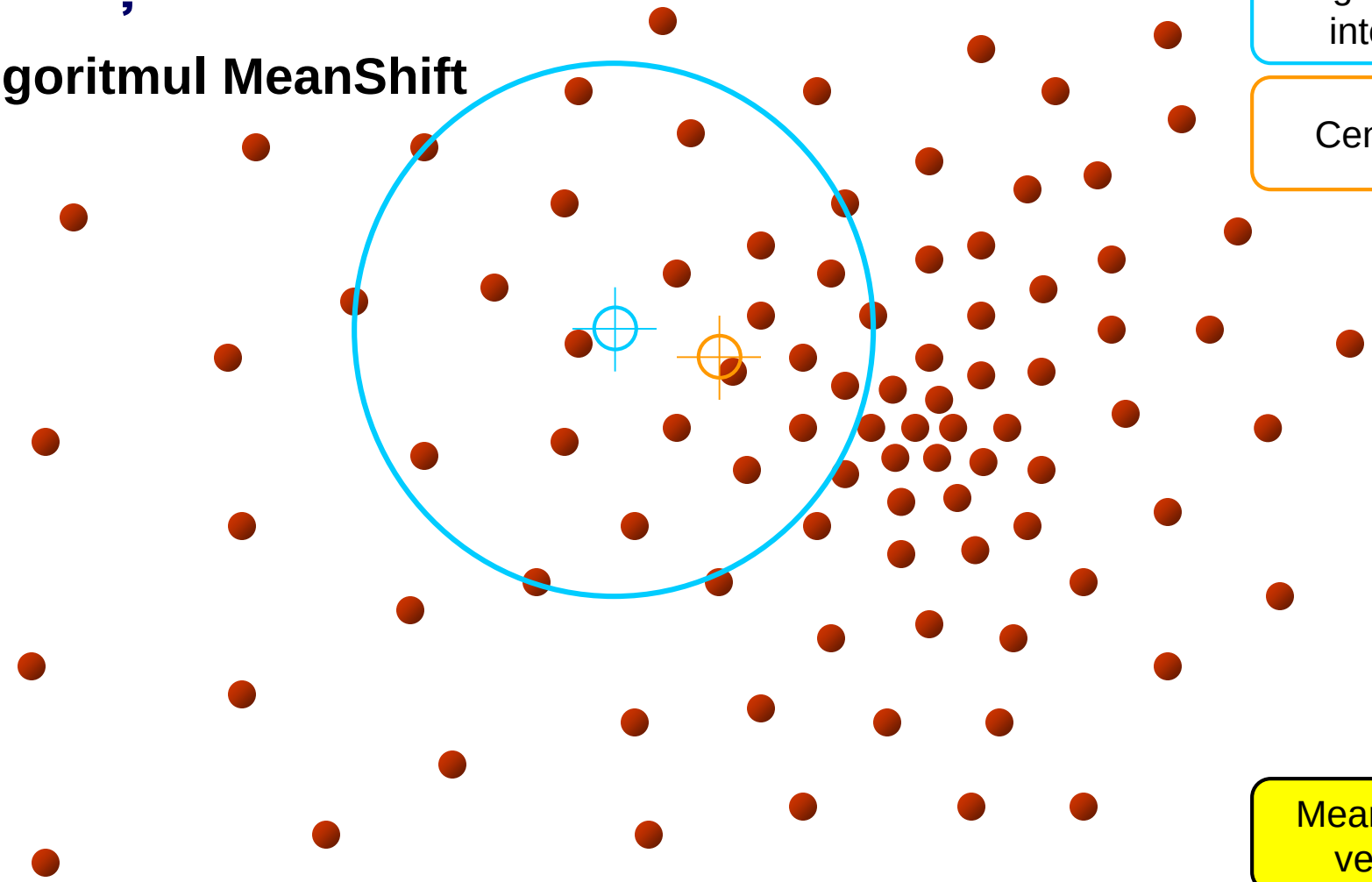
Objectiv: Căutarea regiunii cele mai dense

Din “Mean Shift Theory and Applications”, prezentare pentru “Advanced Topics in Computer Vision”, Institututul Weizmann.

III. Urmărirea traiectoriei

Detecția traiectoriei

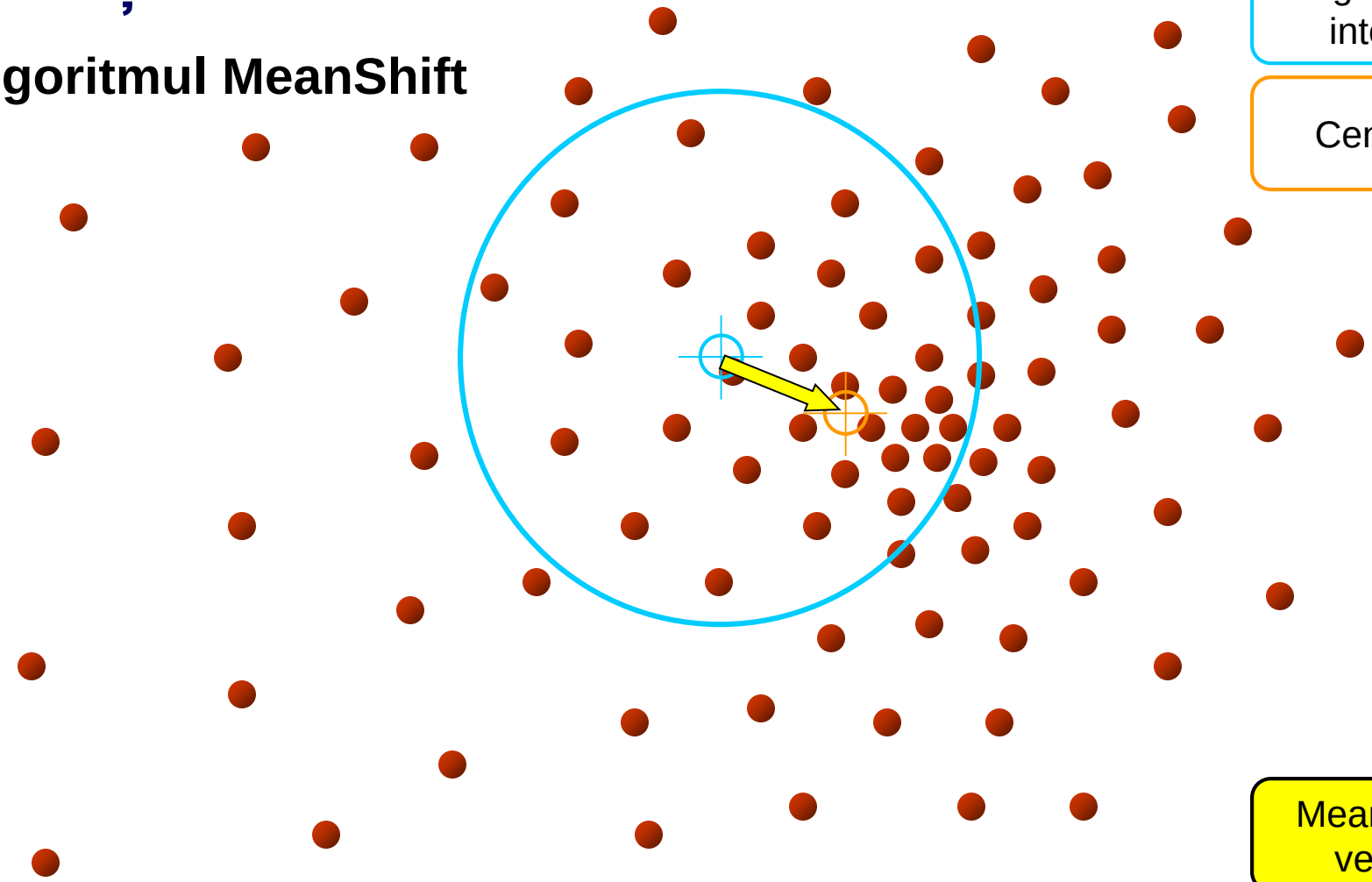
Algoritmul MeanShift



III. Urmărirea traiectoriei

Detecția traiectoriei

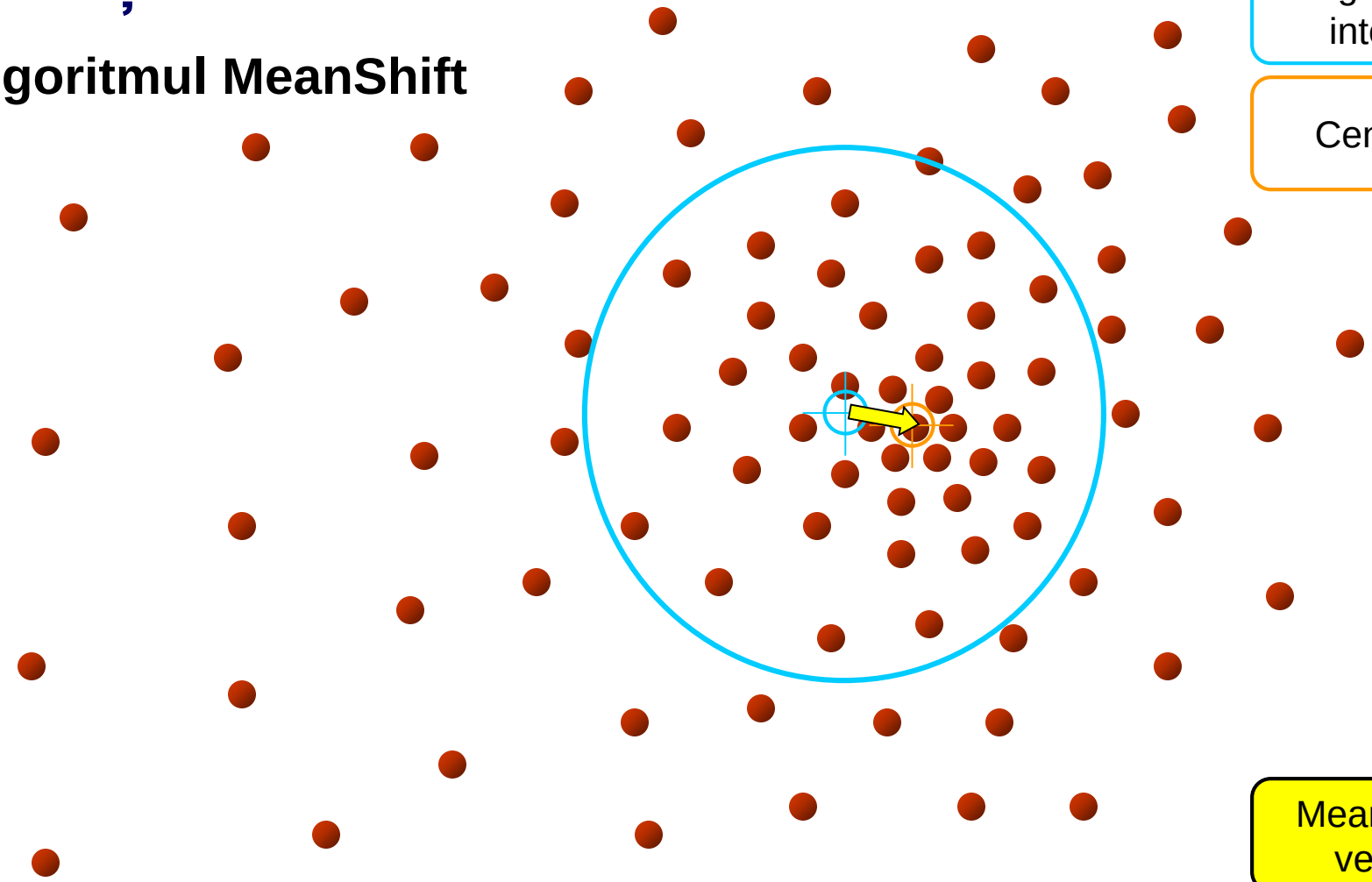
Algoritmul MeanShift



III. Urmărirea traiectoriei

Detecția traiectoriei

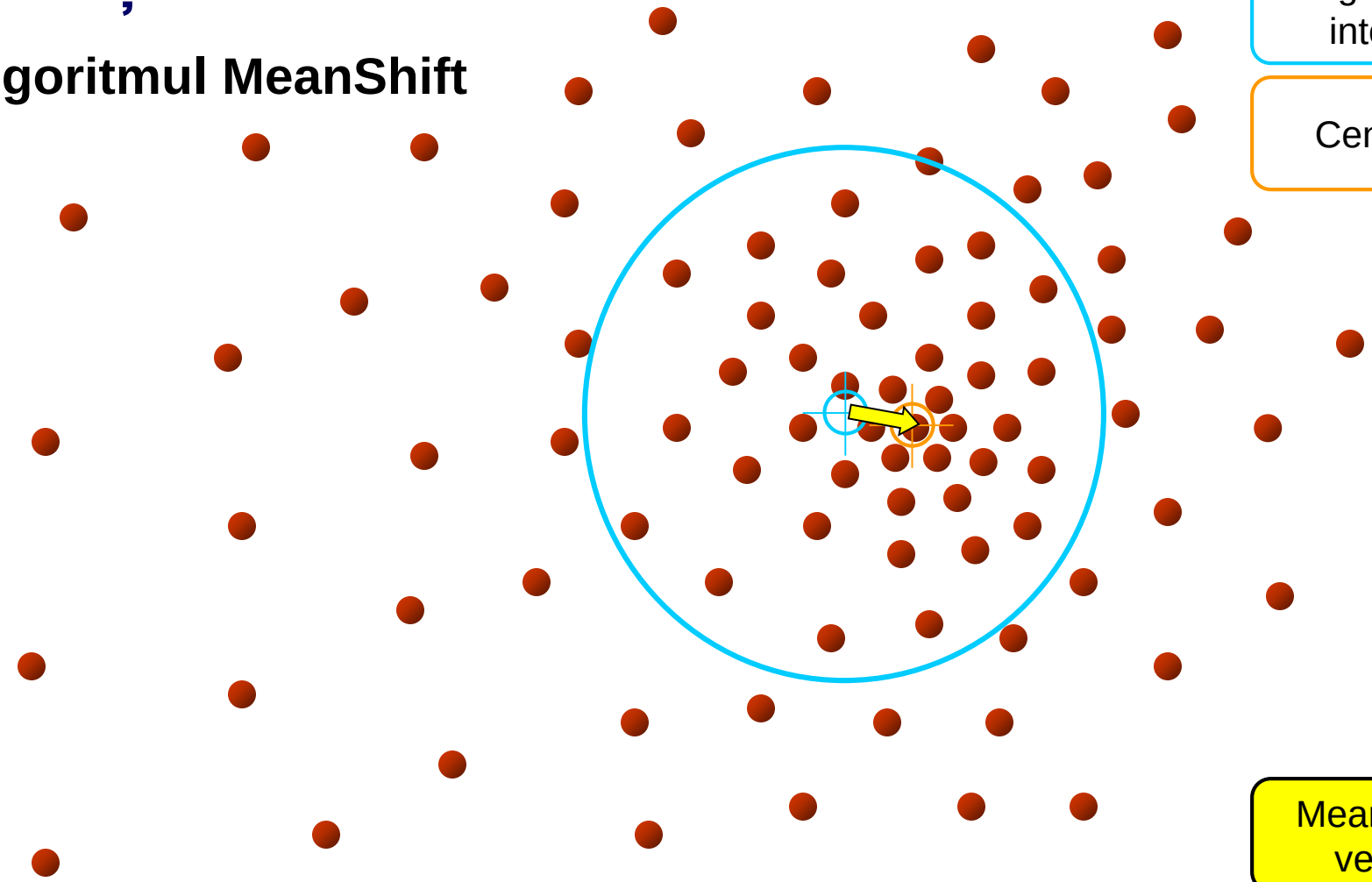
Algoritmul MeanShift



III. Urmărirea traiectoriei

Detecția traiectoriei

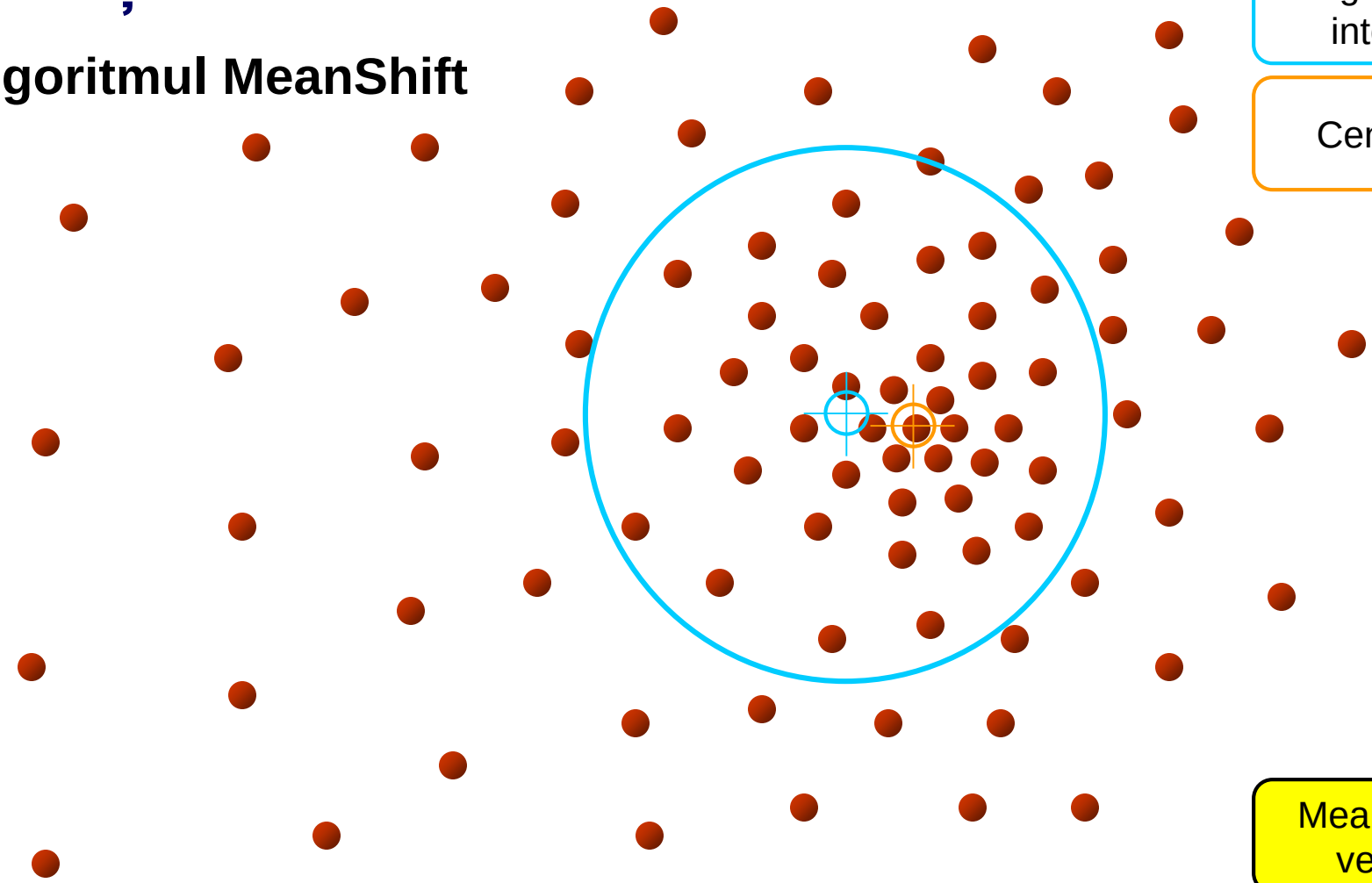
Algoritmul MeanShift



III. Urmărirea traiectoriei

Detecția traiectoriei

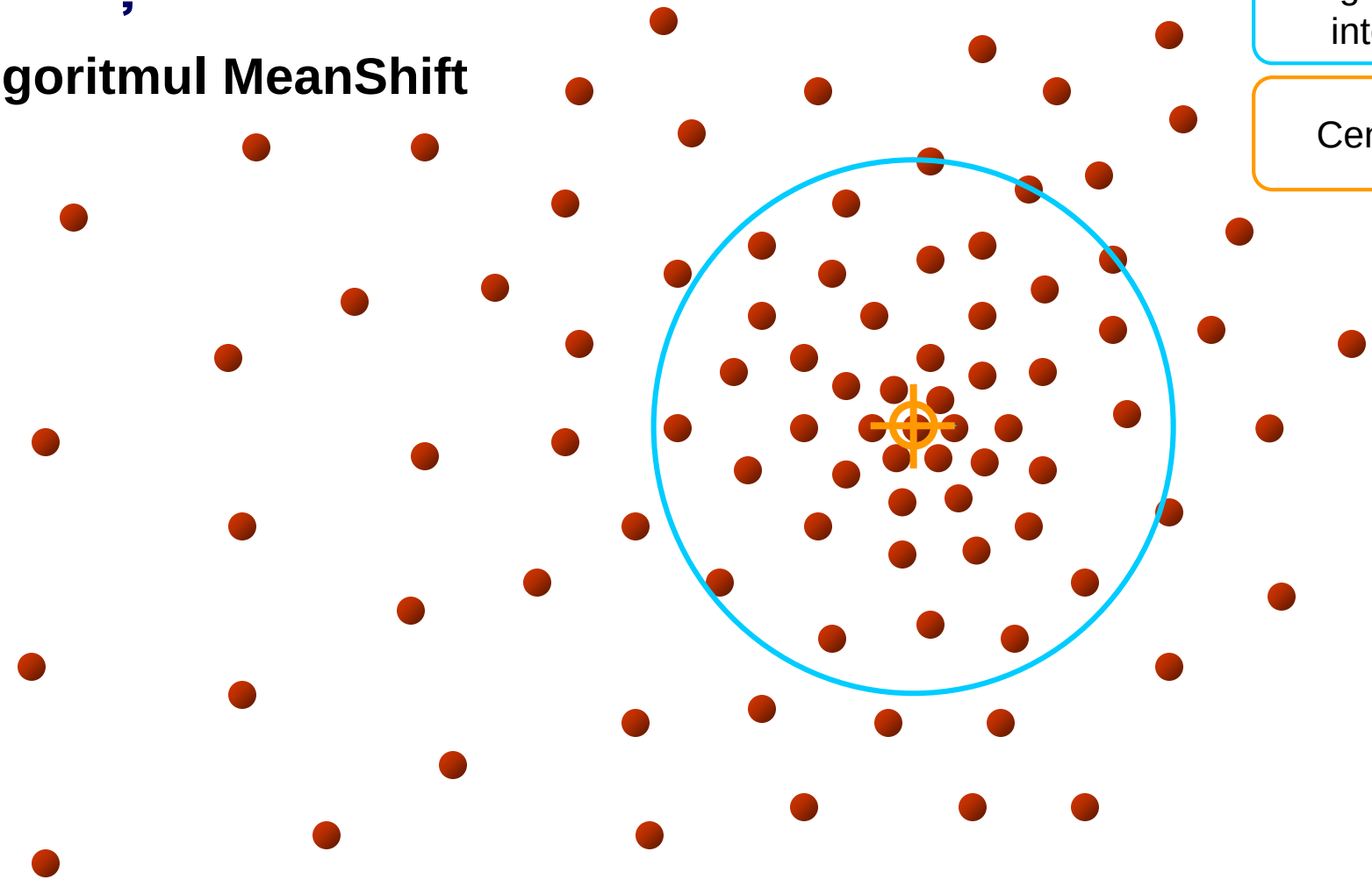
Algoritmul MeanShift



III. Urmărirea traiectoriei

Detecția traiectoriei

Algoritmul MeanShift



III. Urmărirea traiectoriei

Detecția traiectoriei

Algoritmul MeanShift

- Algoritmul MeanShift este o procedură iterativă care localizează maximele locale ale unei funcţii de probabilitate;

- Mijlocul dreptunghiului se mută de la o locaţie la alta până când centrul converge către un punct stabil;

- Două criterii pentru oprire sunt utilizate:
  - un număr maxim de iteraţii;
  - o valoare de plasare a centrului dreptunghiului sub care poziţia este considerată ca a fi conves deja catre un punct stabil.

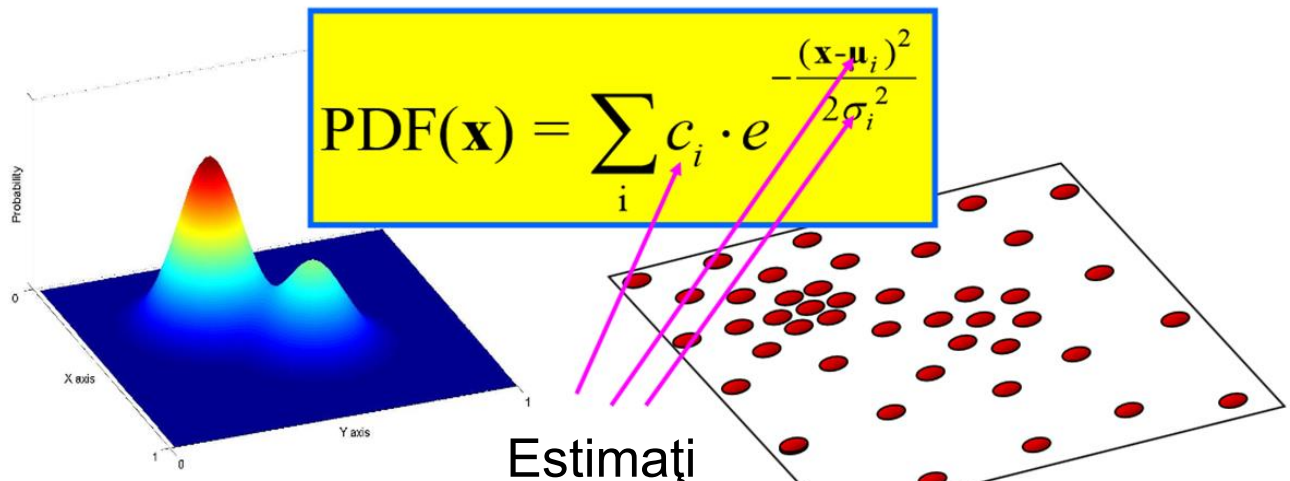
III. Urmărirea traiectoriei

Detecția traiectoriei

Algoritmul MeanShift

Presupunere

- Punctele pot fi mapate la o funcţie de densitate de probabilitate



Funcţia de densitate de probabilitate asumată

Eşantioane preluate

III. Urmărirea traiectoriei

Detecția traiectoriei

Algoritmul MeanShift

Funcţii kernel utilizate pentru modelarea *pdf*

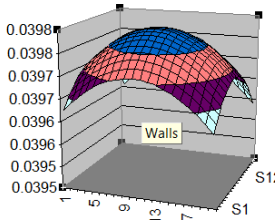
$$\int_{-\infty}^{+\infty} K(w)dw = 1$$

$$K(-w) = K(w) \text{ for all values of w.}$$

$$w = \left\| \frac{x - x_i}{h} \right\|^2$$

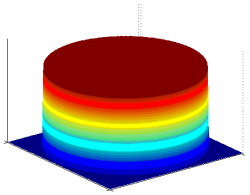
Epanechnikov

$$K_E(w) = \begin{cases} c\left(1 - \| w \|^2\right) & \| w \| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



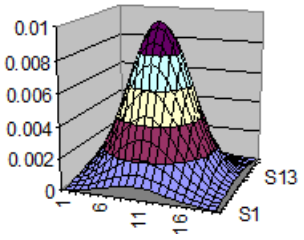
Uniformă

$$K_U(w) = \begin{cases} c & \| w \| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



Normală (gausiană)

$$K_N(w) = c \cdot e^{-\frac{1}{2}\|w\|^2}$$



III. Urmărirea traiectoriei

Detecția traiectoriei

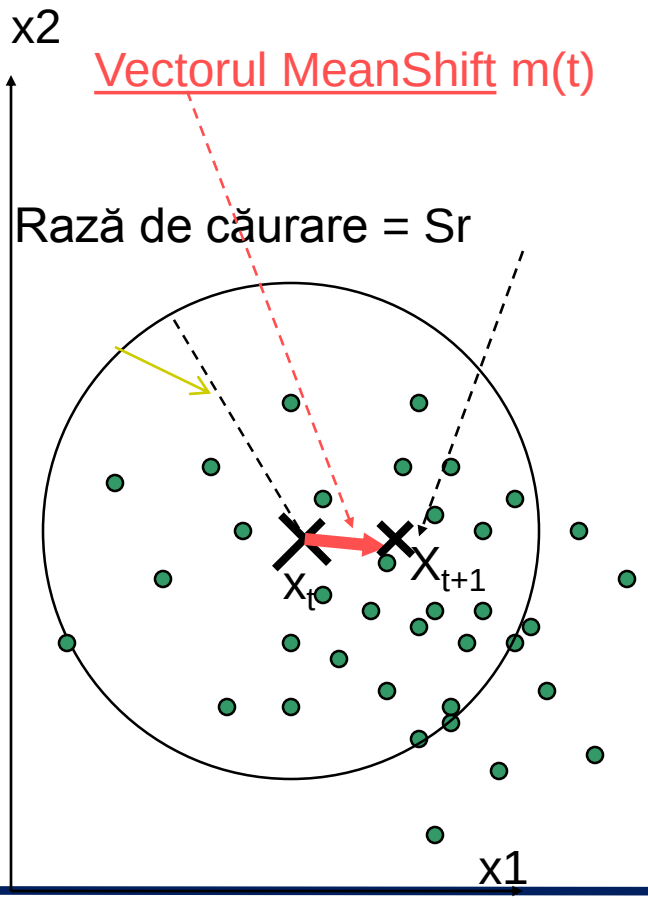
Algoritmul MeanShift

Fie punctul iniţial $x_{t=0}$

1) Desenează un cerc de rază Sr
2) Găseşte media elementelor din cadrul razei (fereastra de căutare) mean_loc(x_{t+1})
3) Mută centrul cercului în punctul x_{t+1}
4) Calculează m(t)=mean_loc(x_t)-x_t=mean-shift vector
5) t=t+1, $x_t = x_{t+1}$

Repetă paşii 1-5, până când m(t) este suficient de mic

Poziţia finală este x_{t+1}



III. Urmărirea traiectoriei

Detecția traiectoriei

Algoritmul MeanShift

Algoritmul clasic MeanShift pleacă de la premiza că scala şi orientarea obiectelor rămân constante în timpul deplasării.

Totuşi, obiectele pot avea forme diverse, iar scala şi orientarea se pot modifica în timp.

III. Urmărirea traiectoriei

Detecția traiectoriei

Algoritmul CamShift

- Reprezintă o extensie a algoritmului MeanShift;
- CamShift este capabil să lucreze cu distribuţii dinamice prin ajustarea ferestrei de căutare de la un cadru la altul. Astfel modelul evoluează de-a lungul timpului;
- Algoritmul are rezultate foarte bine şi atunci când au loc mişcări bruşte care schimbă radical modelul;
- Demo (implementare OpenCV)
  https://www.youtube.com/watch?v=iBOlbs8i7Og

[D. Comaniciu '02]

III. Urmărirea traiectoriei

Detecția traiectoriei

Alte referinţe bibliografice

K. Fukunaga, L.D. Hostetler, “The estimation of the gradient of a density function, with applications in pattern recognition,” IEEE Trans. Information Theory, vol. 21, pp. 32-40, 1975.

Y. Cheng, “Mean Shift, Mode Seeking, and Clustering”, IEEE Trans. PAMI, vol. 17, no. 8, pp. 790-799, 1995.

Dorin Comaniciu, Peter Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Trans. PAMI, Vol. 24, No. 5, 2002.

D. Ramanan, D.A. Forsyth, A. Zisserman: *Tracking People by Learning their Appearance*. IEEE-TPAMI, 29(1): 65-81, 2007.

http://www.ics.uci.edu/~dramanan/papers/pose/index.html

III. Urmărirea traiectoriei

Concluzii

- No free lunch: algoritmii de recunoaștere a obiectelor și urmărire a traiectoriei au fiecare avantaje și dezavantaje.
- Algoritmii pe bază de puncte de interes sunt utilizate pentru detecția inițială a obiectelor
- Metodele pe bază de șablon / mean-shift sunt ulterior utilizate pentru urmărirea obiectelor.

Întrebări?


V
A


M
U
L
T
U
M
E
S
C