Procesarea Imaginilor si Viziune Artificiala

Curs 3

Conf. dr. ing. Remus BRAD, Universitatea “Lucian Blaga” din Sibiu

Elemente de procesare de nivel inalt

Segmentarea (cont.)

Transformata Ahuja pentru segmentarea imaginilor

Considerații legate de segmentarea imaginilor

Transformata

$$\mathbf{F}(x,y;\sigma_g(x,y),\sigma_s(x,y)) = \iint_R d_g(\Delta I,\sigma_g(x,y)) \cdot d_s(\vec{r},\sigma_s(x,y)) \frac{\vec{r}}{\|\vec{r}\|} dwdv$$

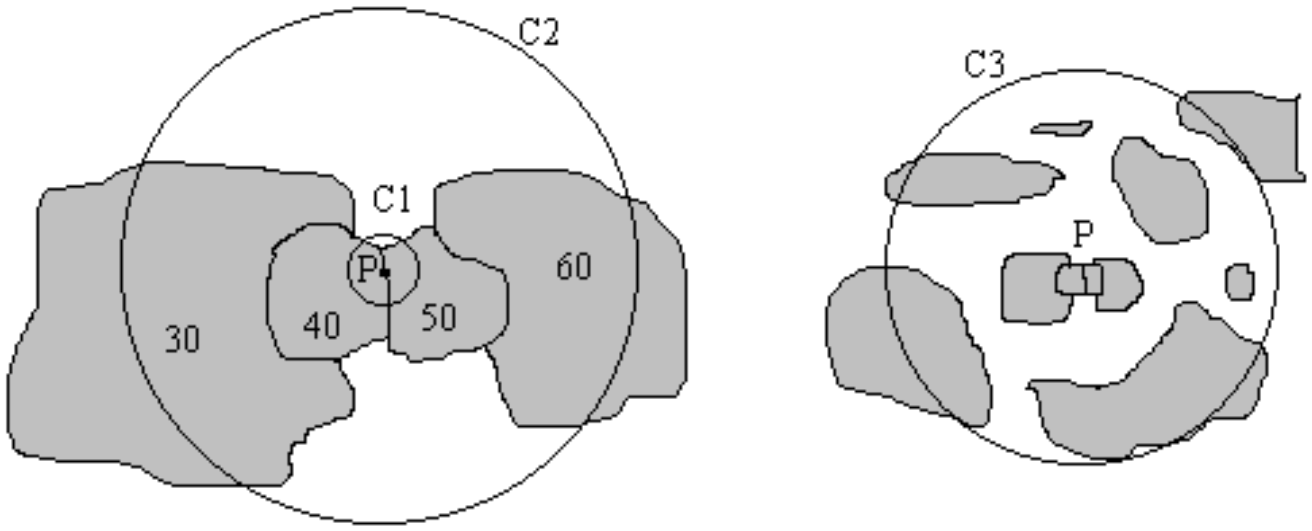
unde R = domeniul($I(u,v)$)\{(x,y)}, $\vec{r} = (v\text{-}x)\,\vec{i} + (w\text{-}y)\vec{j}$ $\Delta I = |I(x,y) - I(v,w)|$

$$d_g(\Delta I,\sigma_g) \approx B_{\Delta I}(\sigma_g)$$
$$d_s(\vec{r},\sigma_s) \approx B_{\|\vec{r}\|}(\sigma_s)$$

$$B_x(c) = \begin{cases} 1 & |\mathrm{x}| \le c \\ 0 & \text{altfel} \end{cases}$$

$$\mathbf{F}(x,y;\sigma_g(x,y),\sigma_s(x,y)) = \mathbf{F}_{\sigma_g}(x,y)$$

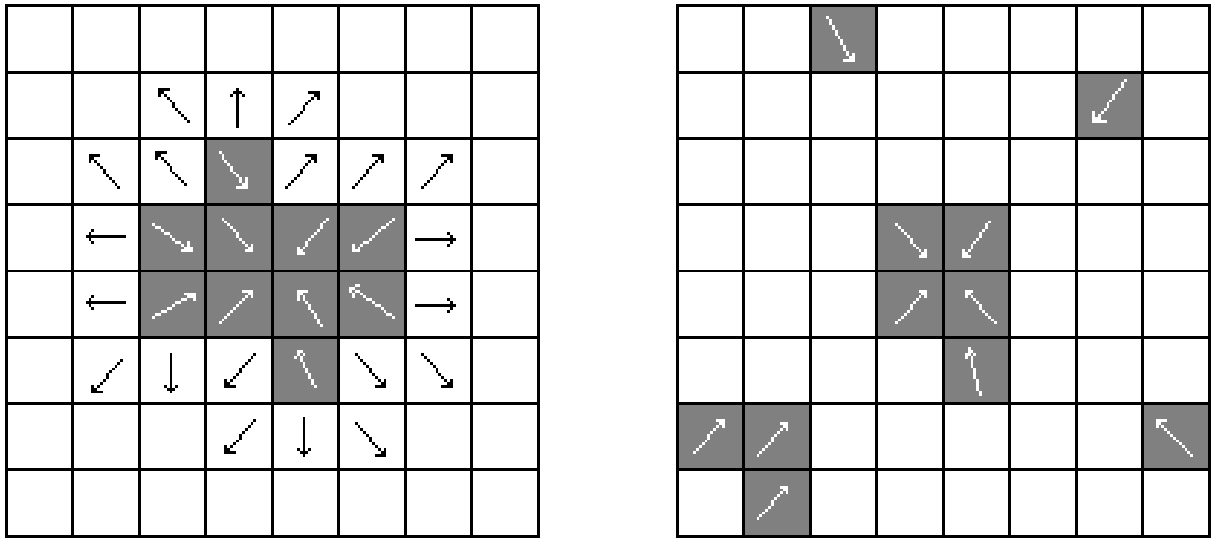
Transformata Ahuja pentru segmentarea imaginilor



a) b)

a) Pentru un σ_g <10, cercul C1 conține destulă informație pentru a identifica un contur în punctul P. Pentru o valoare mai mare a lui σ_g, decizia va fi posibilă folosind vecinătatea cercului C2. b) vecinătatea C3 contine prea multă informație pentru a decide asupra existenței conturului în punctul P.

Transformata Ahuja pentru segmentarea imaginilor



a) b)

a) Alegerea potrivită a lui σ_s duce la transformarea regiunii într-o zonă de atracție constând dintr-o singură regiune cu flux convergent. b) σ_s a fost ales prea mare şi zona de atracție conține mai multe regiuni de flux convergent.

Transformata Ahuja pentru segmentarea imaginilor

Determinarea valorilor lui σ_s

Se defineşte limita de jos a intervalului:

$$\sigma_s^- = \min(\sigma_s) \text{ astfel încât } \|\mathbf{F}_{\sigma_g}(x_0, y_0)\| \geq T(\sigma_g, \sigma_s)$$

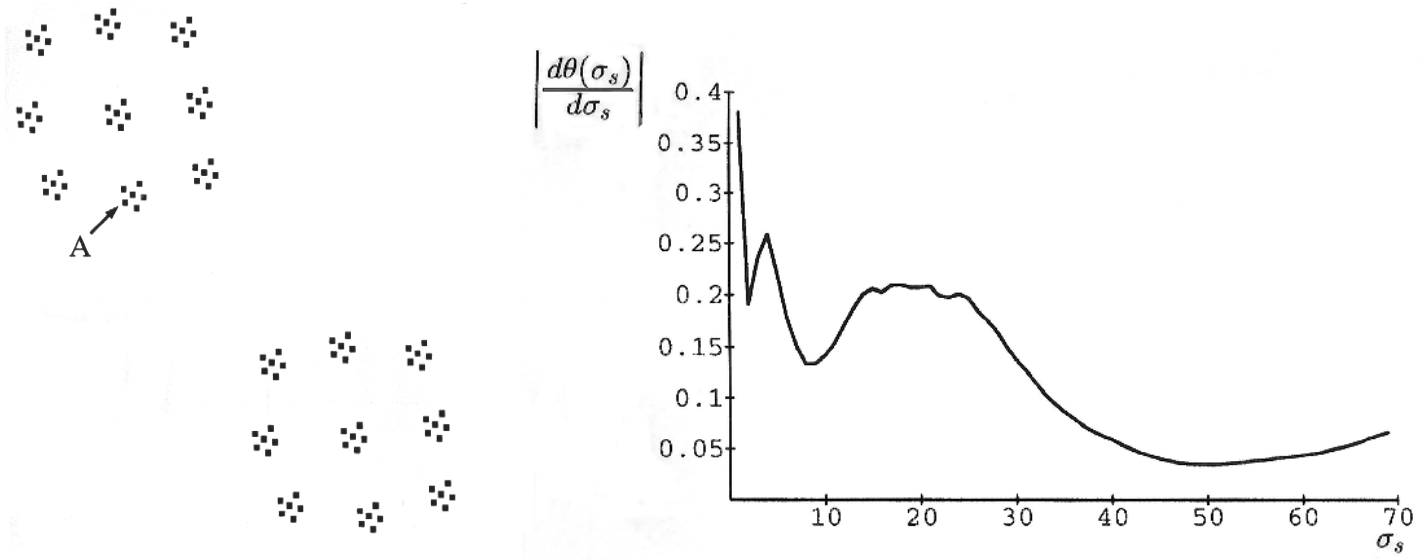
$$T = 4$$

marginea superioară a intervalului:

$$\sigma_s^+ = \min(\sigma_s) \text{ astfel încât } \frac{d}{d\sigma_s}\left|\frac{d}{d\sigma_s}\theta(\sigma_s)\right| \geq 0$$

pentru $\sigma_s \geq \sigma_s^-$. Alegând $\sigma_s(x_0, y_0) \in \left[\sigma_s^-, \sigma_s^+\right]$ pentru fiecare pixel, obținem un câmp **F** având fiecare vector de forța aparținând unei regiuni de atracție

Transformata Ahuja pentru segmentarea imaginilor

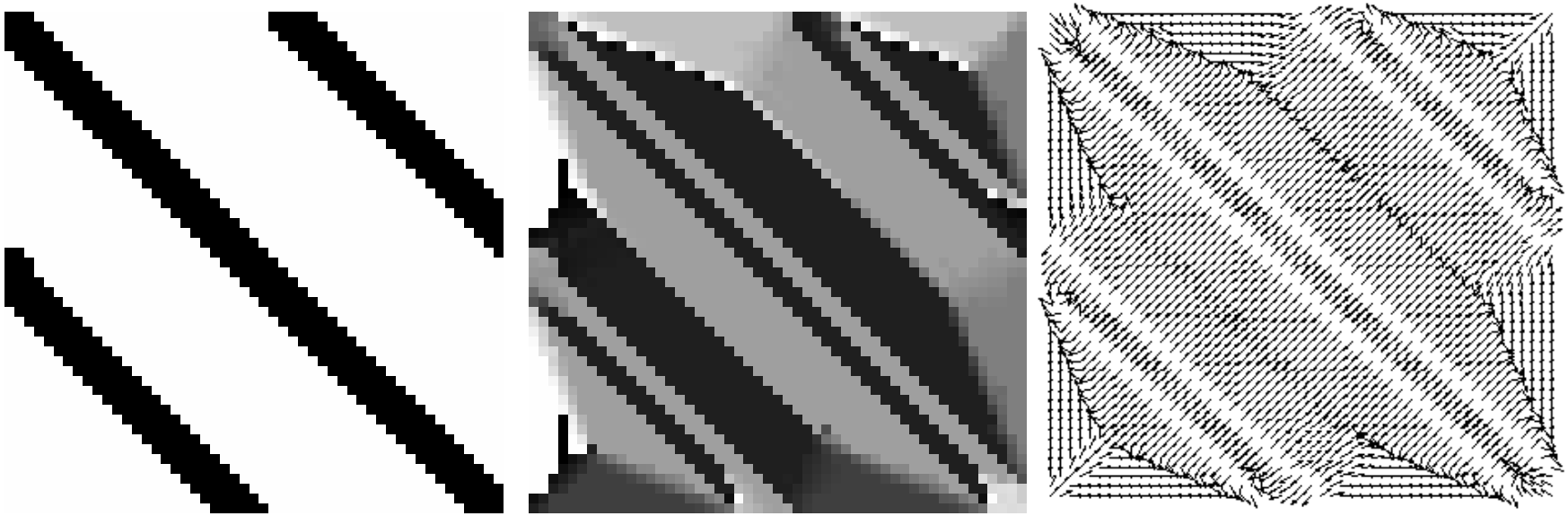


a) b)

a) O imagine conținând pătrate de culoare neagră. b) Graficul variației direcției forței corespunzător lui σ_s, pentru pixelul ce aparține pătratului A.

Transformata Ahuja pentru segmentarea imaginilor

Identificarea regiunilor de atracție din F



a) b) c)

a) Imaginea originală; b) unghiurile forțelor reprezentate în nuanțe de gri pentru σ_g=30; c) reprezentarea vectorială a câmpului de forțe **F**.

Transformata Ahuja pentru segmentarea imaginilor

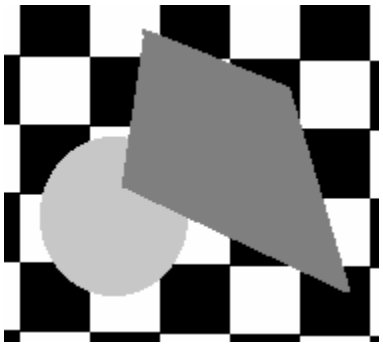
daca $\mathbf{F}_{\sigma_{g0}}(x_0, y_0) \le 0$ si $\mathbf{F}_{\sigma_{g0}}(x_0 + 1, y_0) \ge 0$

si $$\frac{\mathbf{F}_{\sigma_{g0}}(x_0, y_0) \cdot \mathbf{F}_{\sigma_{g0}}(x_0 + 1, y_0)}{\left\|\mathbf{F}_{\sigma_{g0}}(x_0, y_0)\right\| \left\|\mathbf{F}_{\sigma_{g0}}(x_0 + 1, y_0)\right\|} < \cos\varepsilon$$

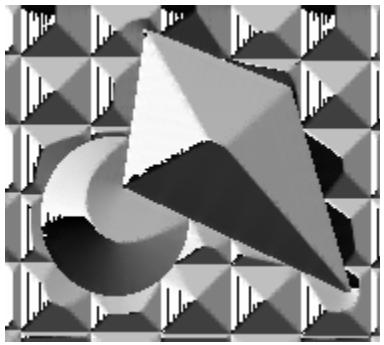
atunci (x_0, y_0) şi (x_0+1, y_0) sunt puncte de frontieră

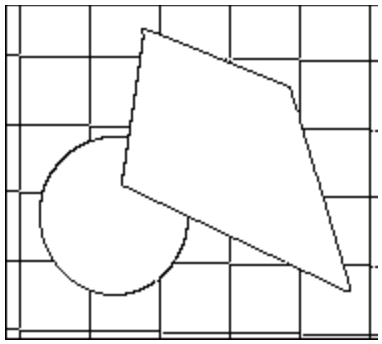
Transformata Ahuja pentru segmentarea imaginilor

influența factorului de omogenitate asupra segmentarii

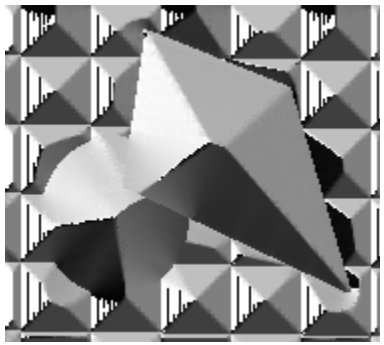


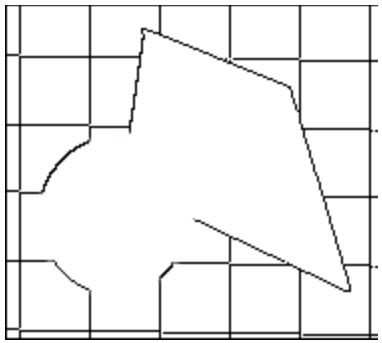
a) Imaginea originală





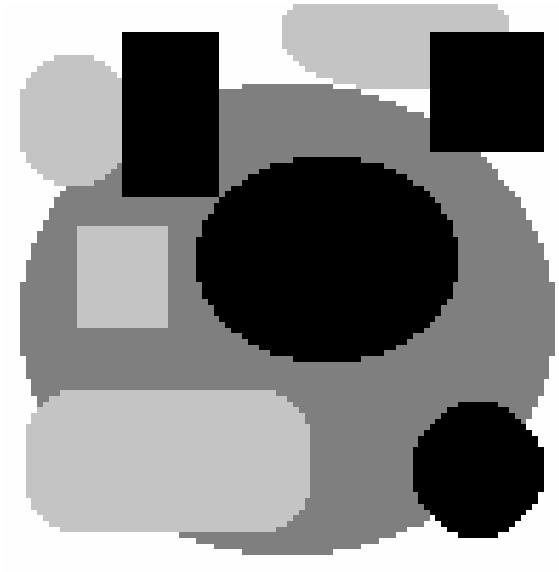
b) unghiurile forțelor şi contururile determinate pentru σ_g=30

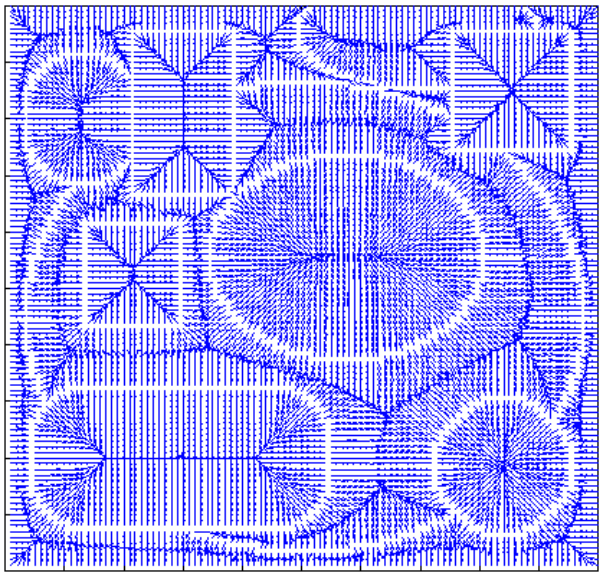


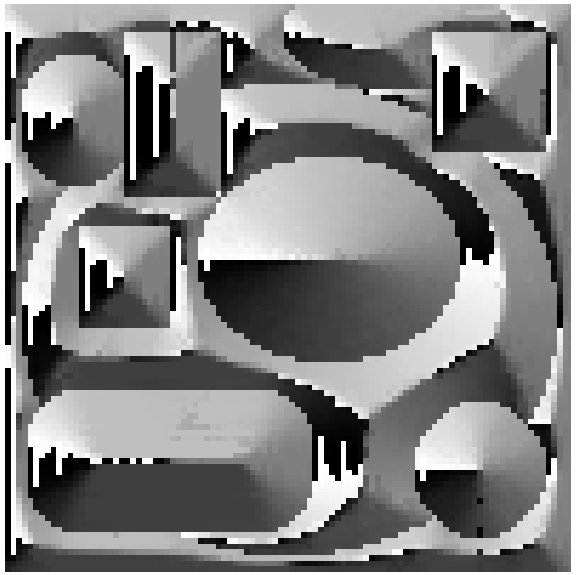


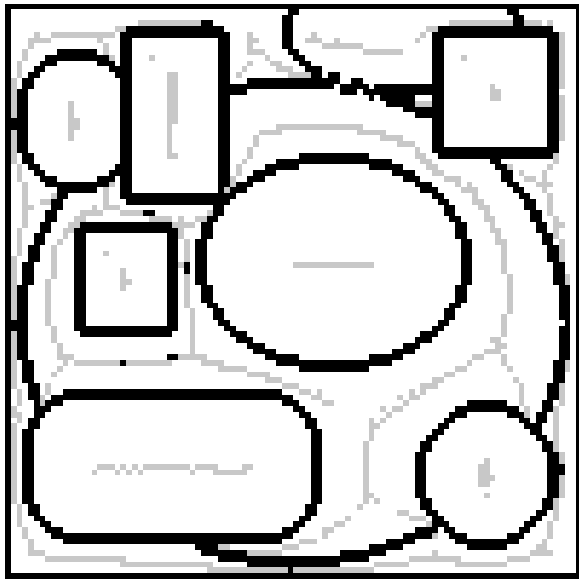
c) unghiurile forțelor şi contururile determinate pentru σ_g=60

Transformata Ahuja pentru segmentarea imaginilor





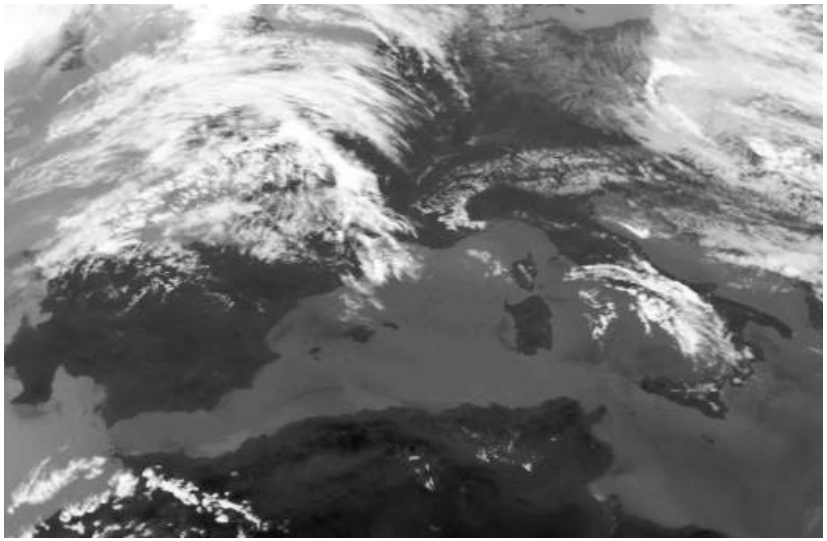




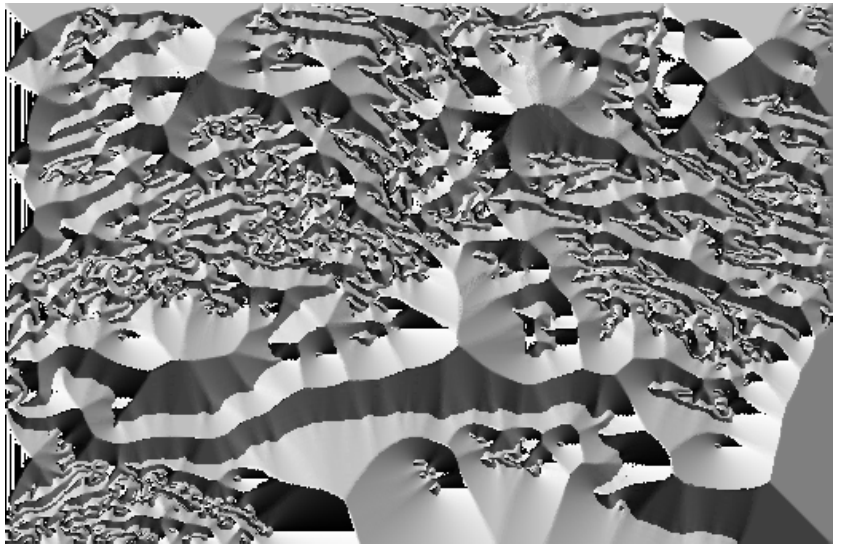
Segmentarea imaginilor satelit

```
// 1. preprocesarea imaginii în scopul netezirii
netezire cu un filtru gaussian având σ = 2

// 2. aplicarea transformatei
pentru o valoare prestabilită a omogenității σg
pentru fiecare pixel din imagine
        pentru valori crescânde ale vecinătății σs
                se determină valoarea forței
        se calculează unghiul forței pentru intervalul de stabilitate

// 3. determinarea apartenentei pixelilor la contur sau linii mediane
pentru fiecare pixel din imagine
        se compară cu vecinii săi
                divergență a forțelor  => pixel de contur
                convergență a forțelor => pixel al liniei mediane

// 4. determinarea regiunilor omogene prin "region growing"
pentru fiecare din pixelii aflați pe linii mediane din imagine
        se "creşte" regiunea, marcând toți pixelii vecini
        se opreşte "creşterea" atunci când
                se intâlnesc pixeli aparținând conturului sau
                se depăşeşte factorul de omogenitate σg
```

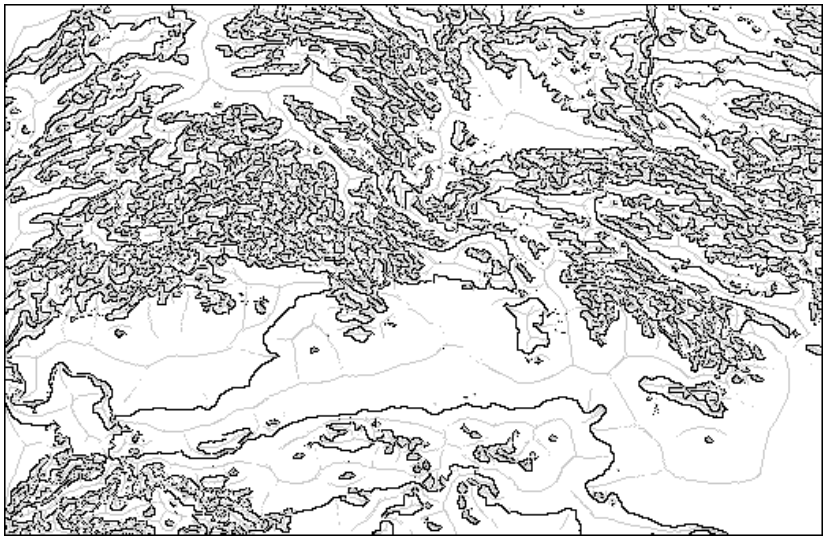
Rezultate



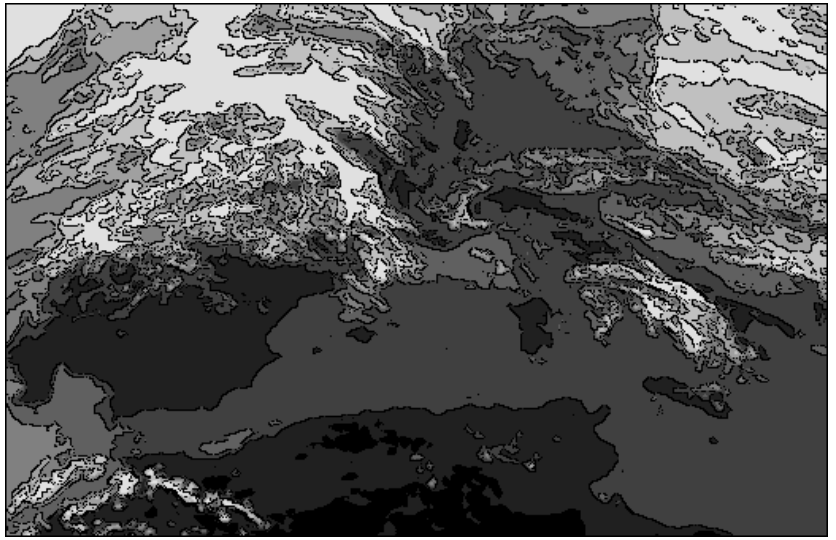
a) după etapa 1



b) după etapa 2



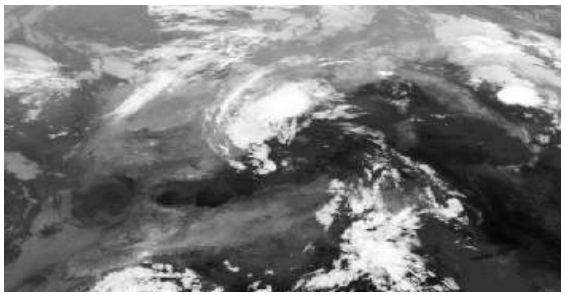
c) după etapa 3



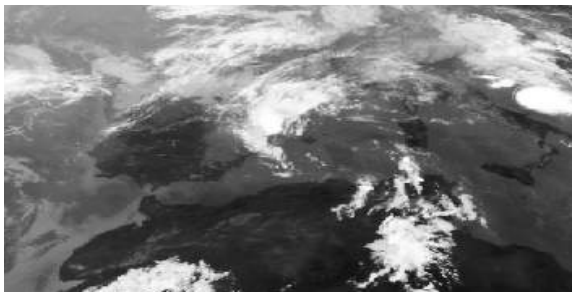
d) după etapa 4

Rezultate

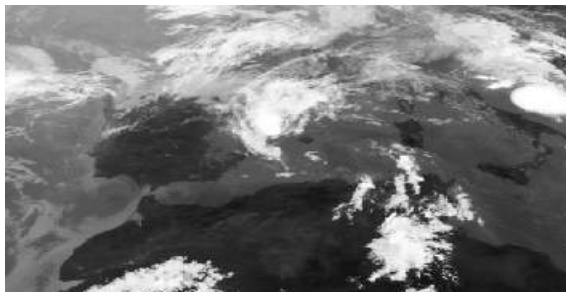
secvență de **imagini METEOSAT** infraroşu



a)

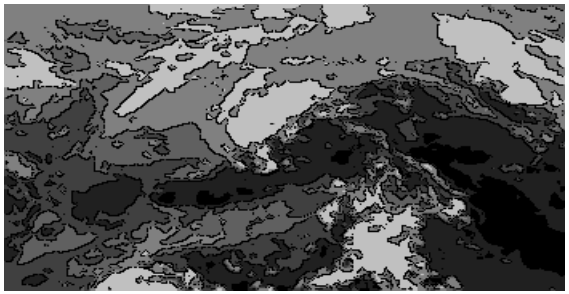


b)

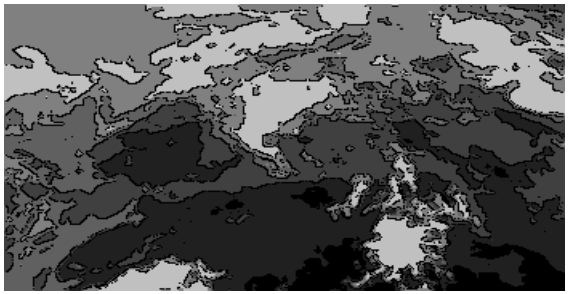


c)

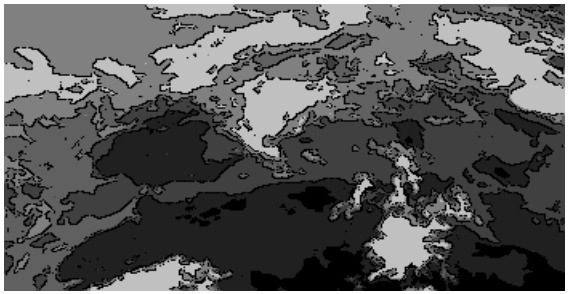
secvență de **imagini segmentate** (σ_g ales în concordanță cu specificația EUMETSAT)



d)



e)



f)

Elemente de procesare de nivel inalt

Recunoasterea de forme

Transformata Hough

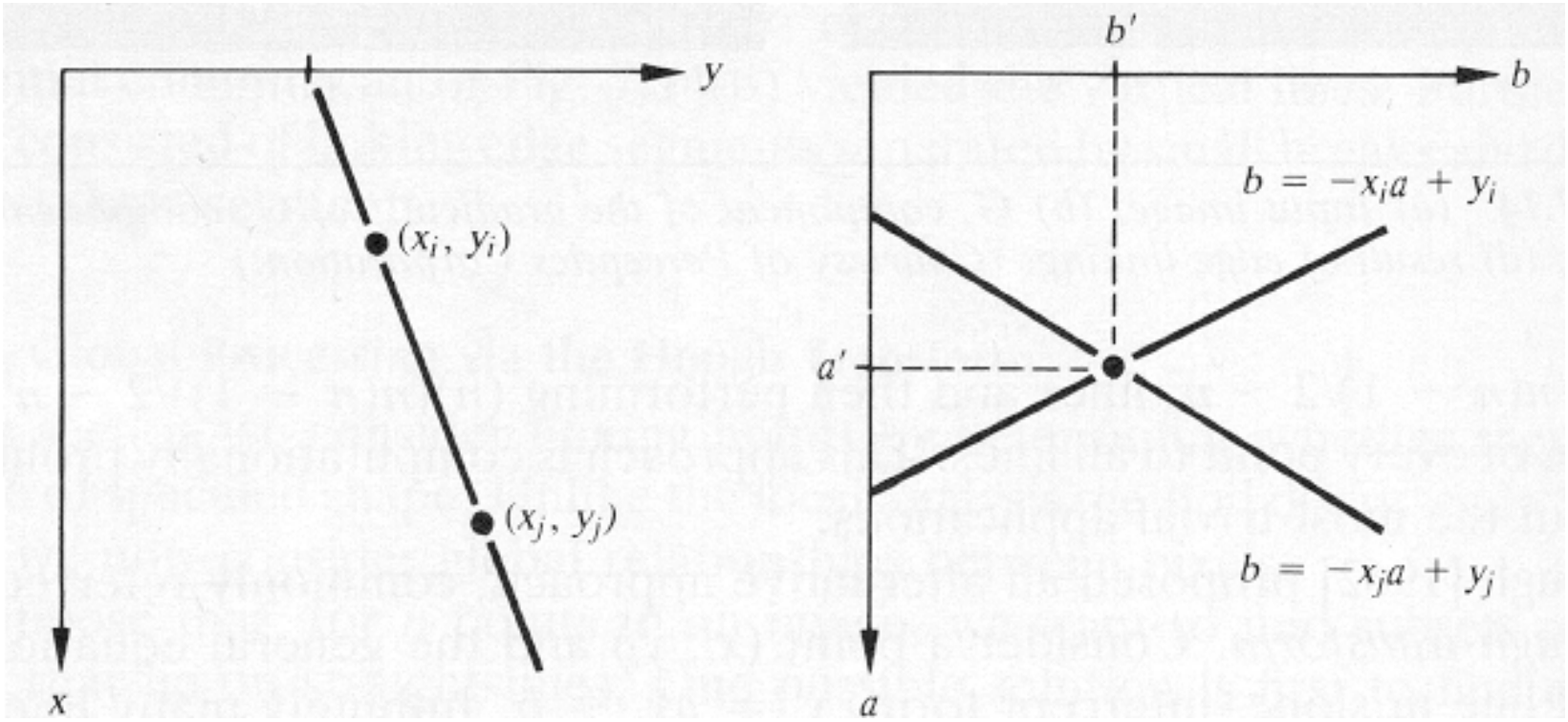
- Ecuatia dreptei ce trece prin punctul de coordonate (x_i,y_i):

$$y_i = ax_i + b$$

- Dar daca rescriem: $b = -x_i a + y_i$ **à**

ecuatia unei drepte in spatiul a b

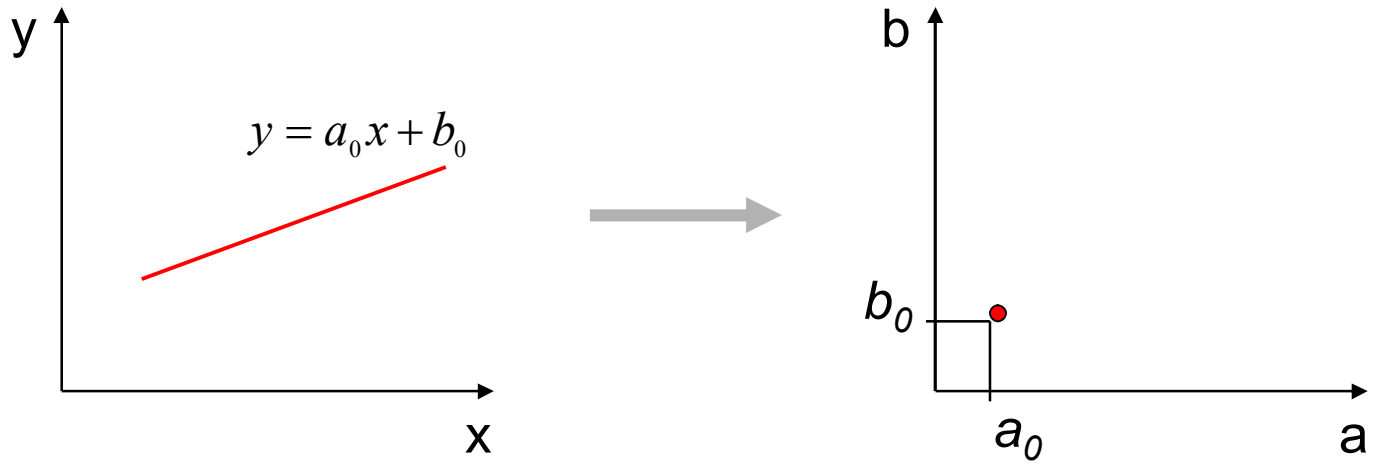
Transformata Hough



Transformata Hough

- O dreapta in spatiul (x,y) ce trece prin mai multe puncte, va avea o valoare precisa a parametrilor (a,b)

- O dreapta in spatiul parametric (a,b) semnifica toate dreptele ce trec printr-un anumit punct (x_i,y_i) (pot fi o infinitate)



spatiul imagine

spatiul Hough

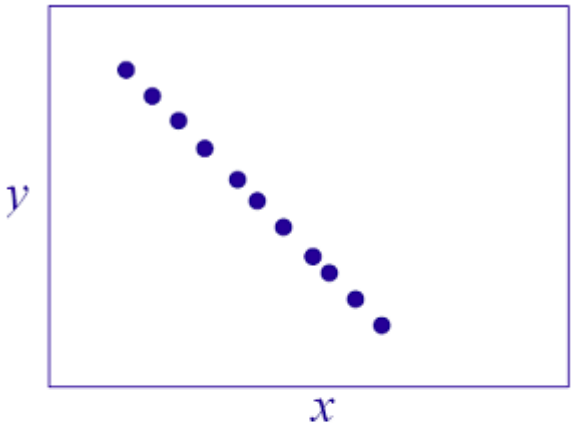
Transformata Hough

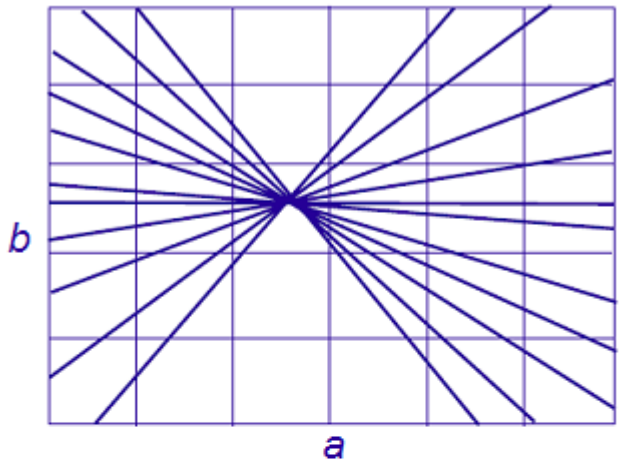
In concluzie:

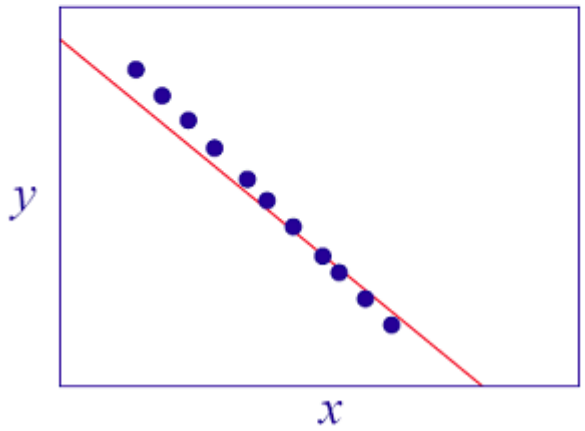
- O dreapta se reprezinta printr-un punct in spatiul (a,b)
- Doua drepte in spatiul (a,b) parametric ce se intersecteaza intr-un punct, denota ca exista puncte ce apartin aceleiasi drepte in spatiul (x,y)

Transformata Hough

Se creaza un accumulator







b

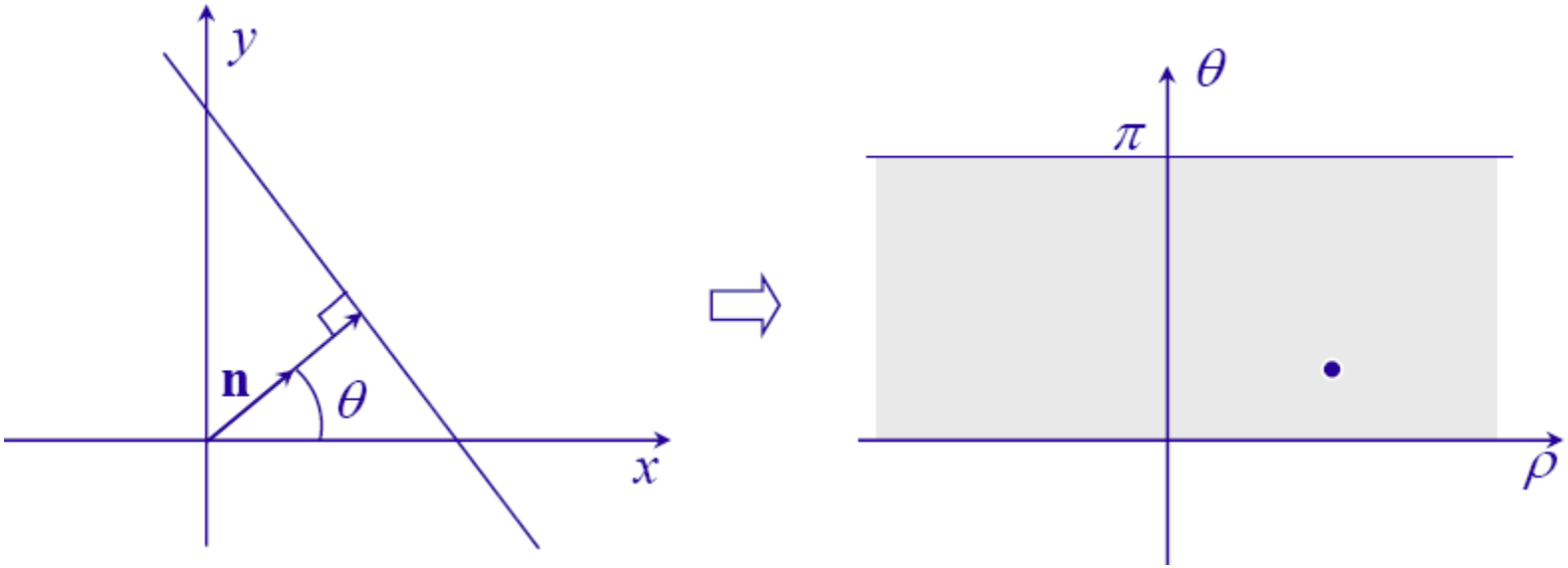
2	2	0	1	2	2
3	5	3	3	2	2
3	7	11	10	4	3
2	3	1	4	5	2
2	1	0	1	3	3

a

Transformata Hough

- Deoarece a,b devin infinit atunci cind dreapta devine verticala, se foloseste de obicei reprezentarea:

$$d = xcos\theta + ysin\theta$$

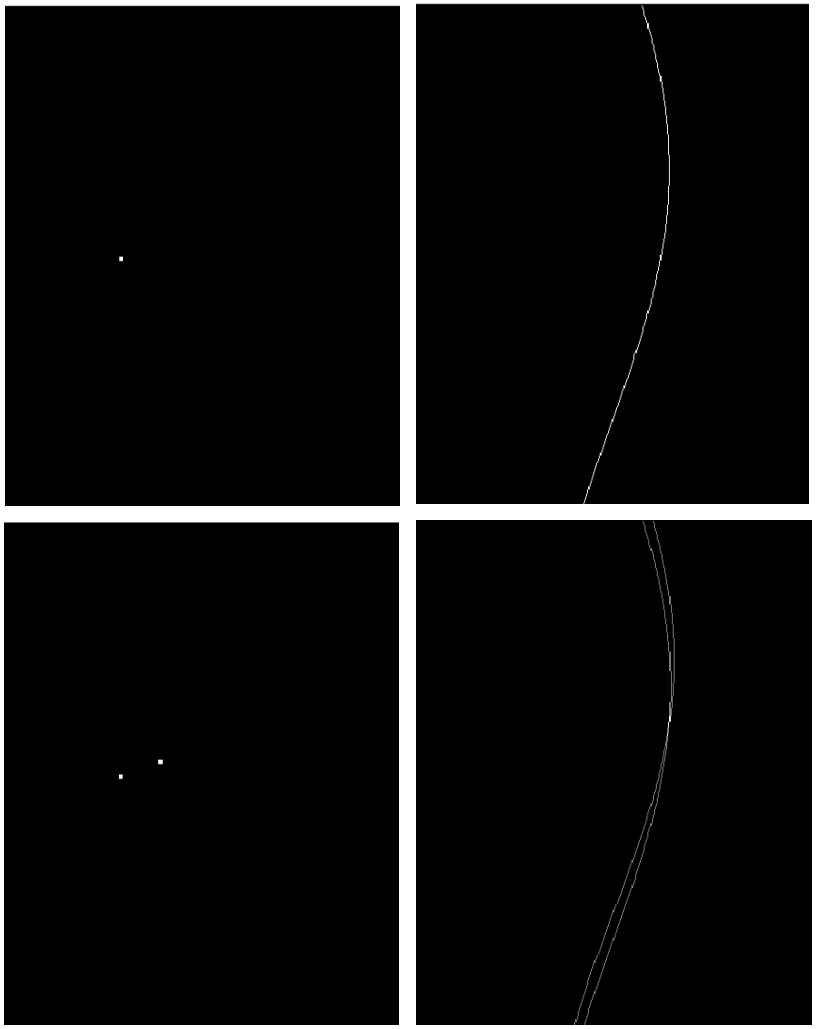


Transformata Hough

- 1 punct => o sinusoida

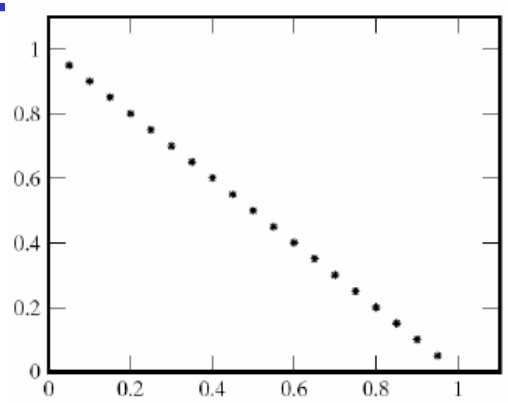
- 2 puncte => 2 sinusoide

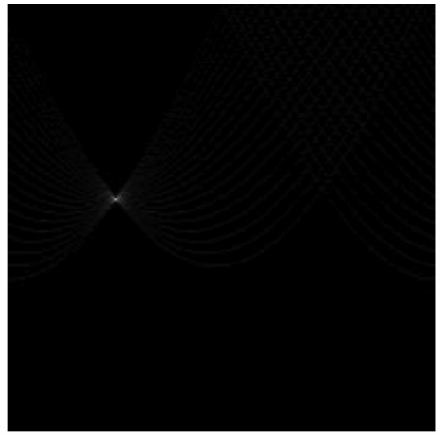
Punctul de intersectie a lor reprezinta dreapta ce trece prin cele doua puncte

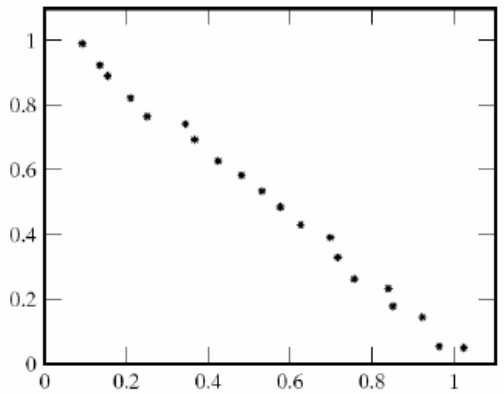


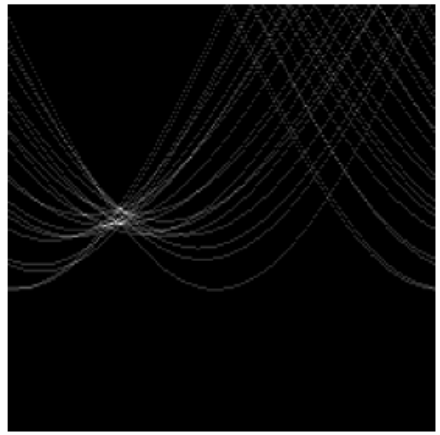
Transformata Hough

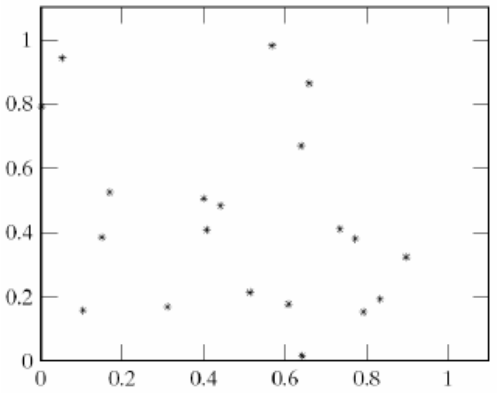
Efectul zgomotului asupra detectiei

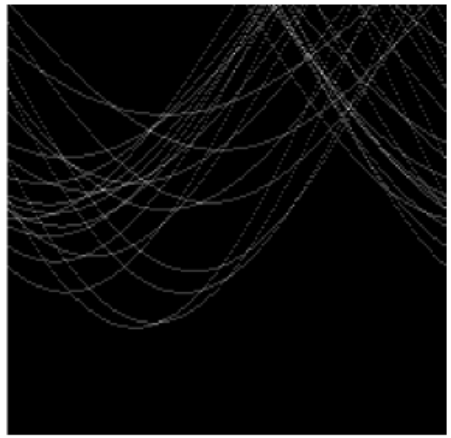






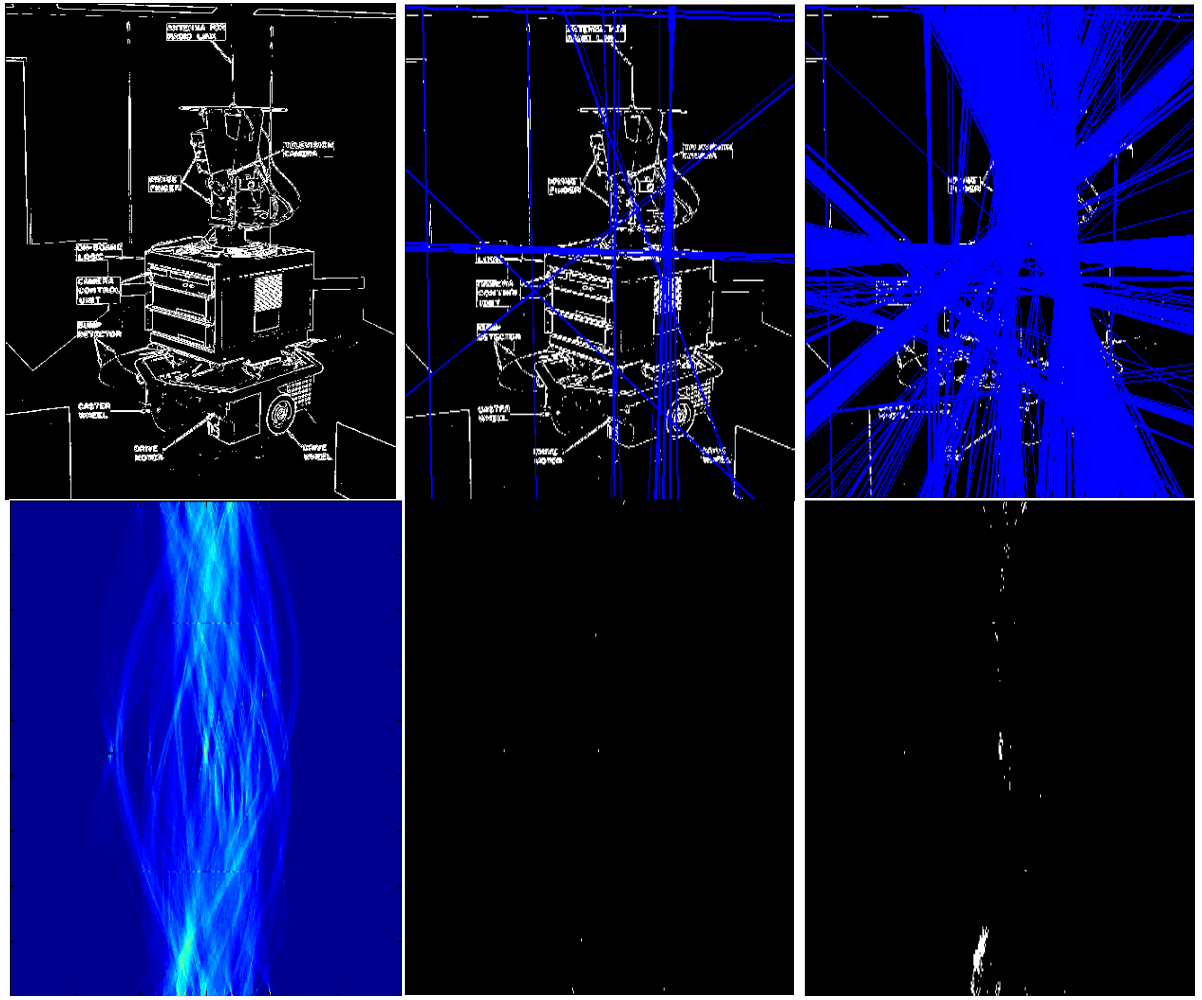






Transformata Hough

Exemplu



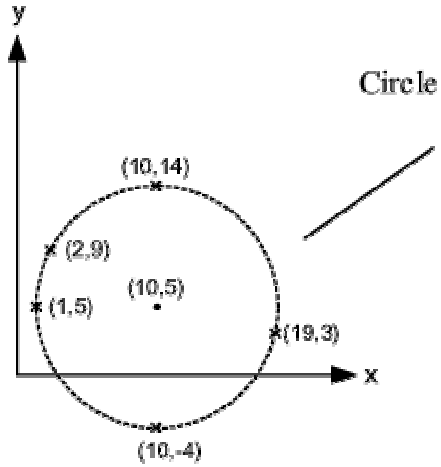
Transformata Hough

- Transformata se poate aplica oricarei functii de forma g(v,c) = 0 unde v este vectorul coordonatelor iar c reprezinta coeficientii

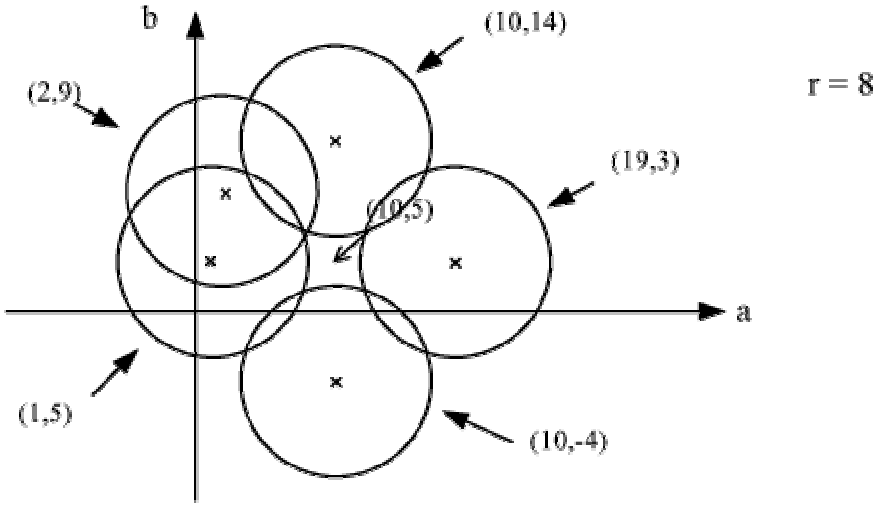
- Spre exemplu, punctele de pe un cerc:

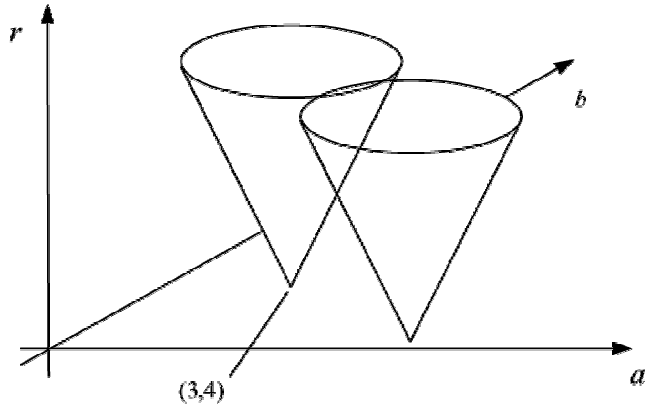
$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

Transformata Hough

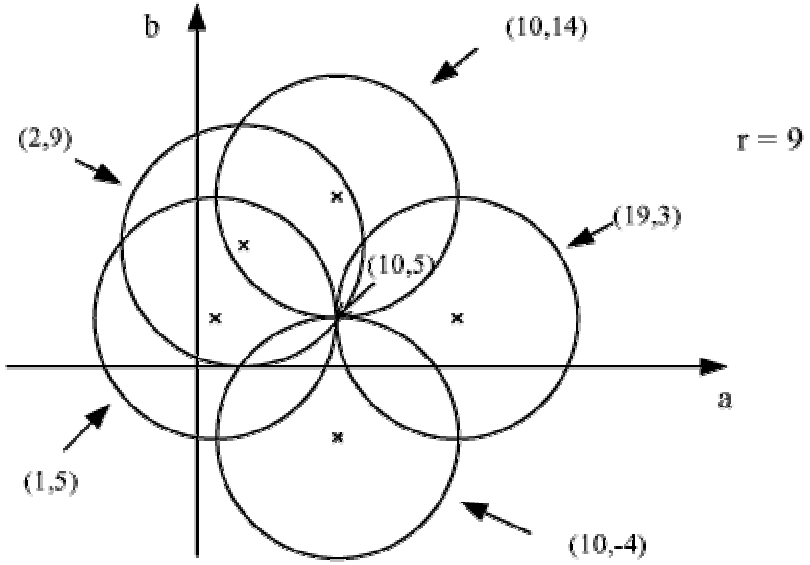


Spatiul coordonatelor



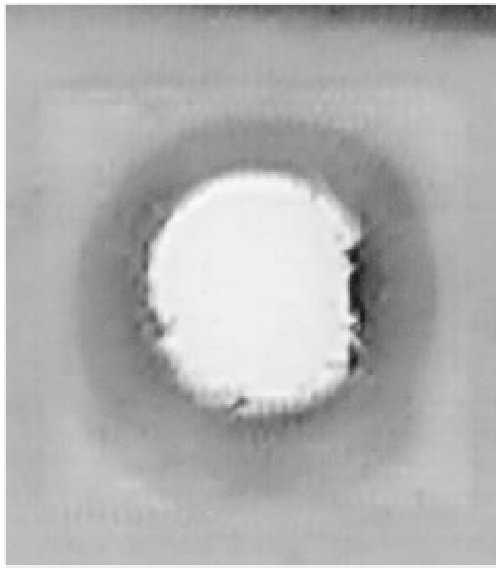


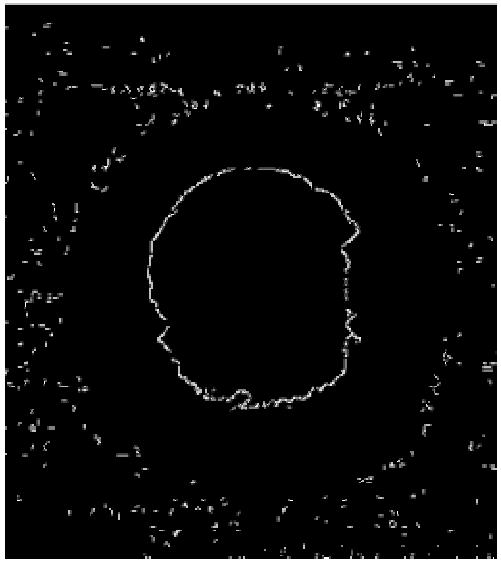
Spatiul parametric

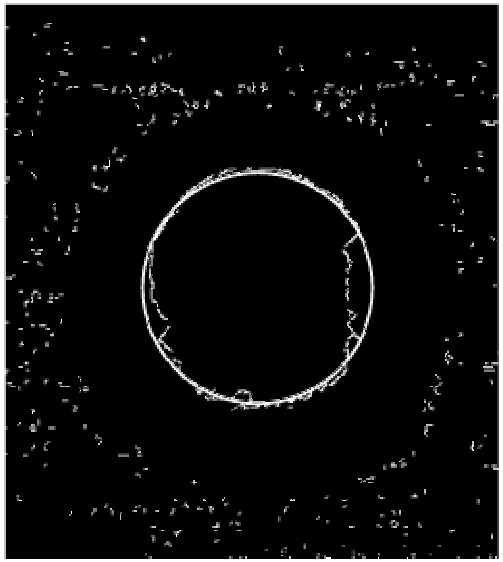


Transformata Hough

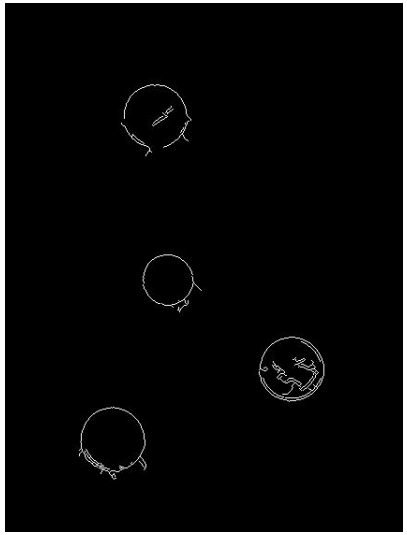
Exemplu

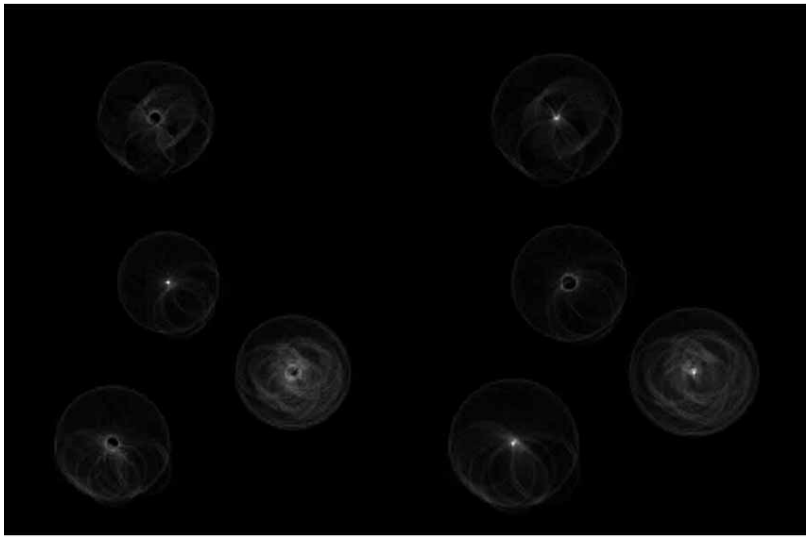


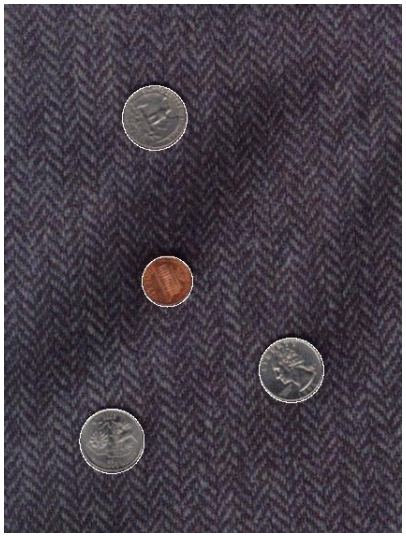




Transformata Hough







Transformata Hough

Linii:

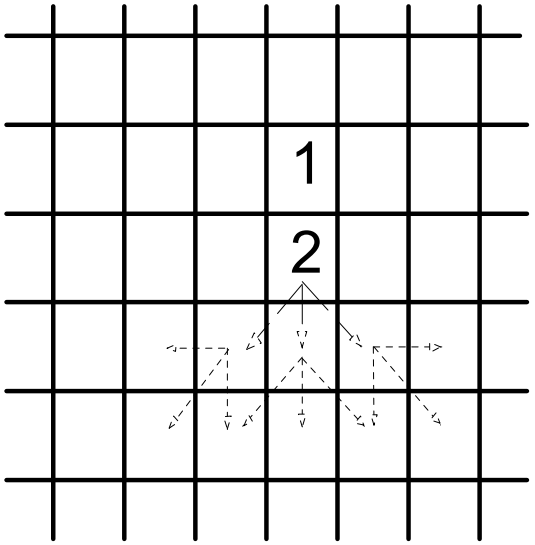
- http://www.dis.uniroma1.it/~iocchi/slides/icra2001/java/hough.html

Cercuri:

- http://www.markschulze.net/java/hough/

Urmarirea contururilor

- Poate fi aplicata imaginilor ce contin informatii legate de contur (spre exemplu detectie de contur/binarizare)
- Dupa ce a fost detectat un punct de contur, operatorul cauta toti ceilalti pixeli ce apartin conturului
- Metoda consta in:



• Se gaseste pixelului initial de contur (1);

• Se cauta cei 8 vecini pentru a se gasi urmatorul pixel de contur (2);

• Se cauta in aceeasi directie (se admite o deviatie de 1 pixel);

• Se repeta pasul 3 pina cind nu se mai gasesc alte puncte de contur.

Algoritmul de urmarire al unui contur

Se verifica pixelii vecini ai punctului de start, astfel:

- Se aleg 3 pixeli cu valoarea cea mai mare a gradientului
- Se sorteaza cei 3 pixeli in ordine descrescatoare, iar tripleta de pixeli va fi stocata intr-o lista
- Se alege pixelul cu valoarea cea mai mare a gradientului
- Setam directia de deplasare conform conventiei
- Se inainteaza cu un pas in directia aleasa (dir)
- Se seteaza pixelul curent ca vizitat
- Se verifica cei trei pixeli vecini din directiile: dir – 1, dir , dir + 1
- Se sorteaza pixelii descrescator si se stocheaza intr-o lista ca un triplet
- Se alege primul pixel (cu gradientul cel mai mare)

Algoritmul de urmarire al unui contur

Se repeta pana cand se gaseste un pixel candidat valid

- IF pixelul candidat este nevizitat si gradientul este > threshold, THEN pixelul este parte a conturului si se scoate pixelul din lista

  ELSE
- IF mai sunt pixeli candidati in triplet THEN

  Se trece la urmatorul pixel din triplet

  ELSE se sterge tripletul gol si se trece cu o pozitie inapoi in lista
- Se face un pas inainte in directia respectiva

0	1	2
7	*	3
6	5	4

Exemplu

Imagine Originala

	1	2	3	4	5	6	7	8	9	10
1	8	9	10	11	12	13	14	15	16	17
2	9	10	11	12	13	14	15	16	17	18
3	10	11	12	13	14	15	16	17	18	19
4	11	12	13	**24**	**25**	**26**	**27**	18	19	20
5	12	13	14	**25**	**26**	**27**	**28**	19	20	21
6	13	14	15	**26**	**27**	**28**	19	20	21	22
7	14	15	16	**27**	**28**	19	20	21	22	23
8	15	16	17	18	19	20	21	22	23	24
9	16	17	18	19	20	21	22	23	24	25
10	17	18	19	20	21	22	23	24	25	26

0	1	2
7	*	3
6	5	4

Sa alege punctul cu coordonatele [4, 4] = 76 ca punct de start. Setam pragul de gradient la 20.
Pentru primul punct alegem primii 3 pixeli cu gradientul cel mai mare: [4, 3], [5, 3] si [3, 4] in aceasi ordine.
Se trece la pixelul [4, 3], urmatorii pixeli: [3, 2], [4, 2] si [5, 2] nu respecta testul pragului, deci punctul [4, 3] este sters.
Urmatorul pixel este [5, 3] si produce tripeltul: [6, 3], [5, 2] si [6, 2] din care se alege primul, etc.

Gradienti Sobel

	1	2	3	4	5	6	7	8	9	10
1	-	-	-	-	-	-	-	-	-	-
2	-	16	16	16	16	16	16	16	16	-
3	-	16	36	56	**56**	**56**	40	20	16	-
4	-	16	56	**76**	56	56	**60**	40	16	-
5	-	16	**56**	56	16	4	**44**	24	16	-
6	-	16	**56**	56	4	**44**	44	4	16	-
7	-	16	40	**60**	**44**	44	4	16	16	-
8	-	16	20	40	24	4	16	16	16	-
9	-	16	16	16	16	16	16	16	16	-
10	-	-	-	-	-	-	-	-	-	-

Step	1st candidate	2nd candidate	3rd candidate
1	~~**(4,3)=56**~~	**(5,3)=56**	(3,4)=56
2	~~**(3,2)=16**~~	~~**(4,2)=16**~~	~~**(5,2)=16**~~
3	**(6,3)=56**	(5,2)=16	(6,2)=16
4	**(7,4)=60**	(7,3)=40	(7,2)=16
5	**(7,5)=44**	(8,4)=40	(8,5)=24
6	**(6,6)=44**	(7,6)=44	(8,6)=4
7	**(5,7)=44**	(6,7)=44	(5,6)=4
8	**(4,7)=60**	(4,8)=40	(5,8)=24
9	**(3,6)=56**	(3,7)=40	(3,8)=20
10	**(3,5)=56**	(2,5)=16	(2,6)=16
11	**(4,4)=76**	(3,4)=56	(2,4)=16

Reprezentarea contururilor

- **Trebuie stabilita anumite reguli pentru determinarea carui obiect ii apartine un pixel**
- **Sint utilizate doua scheme:**
  - **conectivitate cu 4 vecini**
    - un pixel este considerat avind 4 vecini
  - **conectivitate cu 8 vecini**
    - un pixel este considerat avind cei mai apropiati 8 vecini

	1	
2	X	0
	3	

Pixeli cu 4 vecini

3	2	1
4	X	0
5	6	7

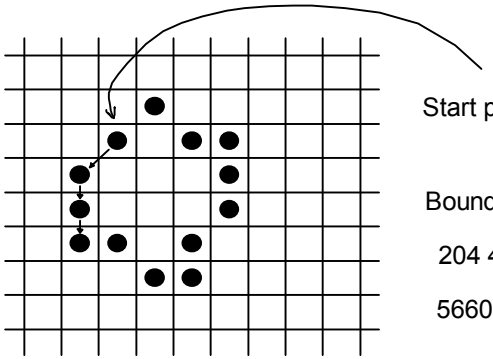
Pixeli cu 8 vecini

Reprezentarea contururilor

- Utilizarea unui format compact pentru stocarea informatiilor referitoare la un obiect
- Se poate coda directia contururilor unui obiect
- Foloseste proprietatea de vecinatate a pixelilor de contur (vecinatate de 4 sau de 8)
- Reprezentare pe 3 biti pentru fiecare punct

- Se alege un punct se start arbitrar pe contur
- Se cauta pixelul vecin de pe contur folosind o regula (sensul acelor de ceasornic)
- Se memoreaza directiile contururilor ca si o lista de coduri

3	2	1
4	**n**	0
5	6	7



Detecția mișcării din secvențe de imagini

Iluzii optice

- Bazate pe miscare

http://www.sandlotscience.com/Guided_Tours/Tour2/Tour_2_Main.htm

Introducere

Există trei grupe principale de probleme legate de mişcare:

1. **Detecția mişcării**
2. **Detectia mişcării obiectelor şi localizarea lor**
3. **Reconstituirii caracteristicilor obiectelor 3D**

Analiza mişcării

Fazele analizei mişcării

1. Faza detecţiei mişcării: Acest proces are drept scop extragerea dintr-o secvenţă de imagini a suprafeţelor aflate în mişcare şi a suprafeţelor imobile, printr-o serie de observaţii bazate pe diferenţele între cadre. Rezultatul acestui proces este o informaţie binară a prezenţei sau a absenţei mişcării şi o mapare spaţială a contururilor obiectelor mobile. Un set de două imagini poate fi insuficient pentru a reduce ambiguităţile de detecţie cum ar fi suprapunerile de obiecte, acoperirile, modificarea suprafeţei obiectului în mişcare.

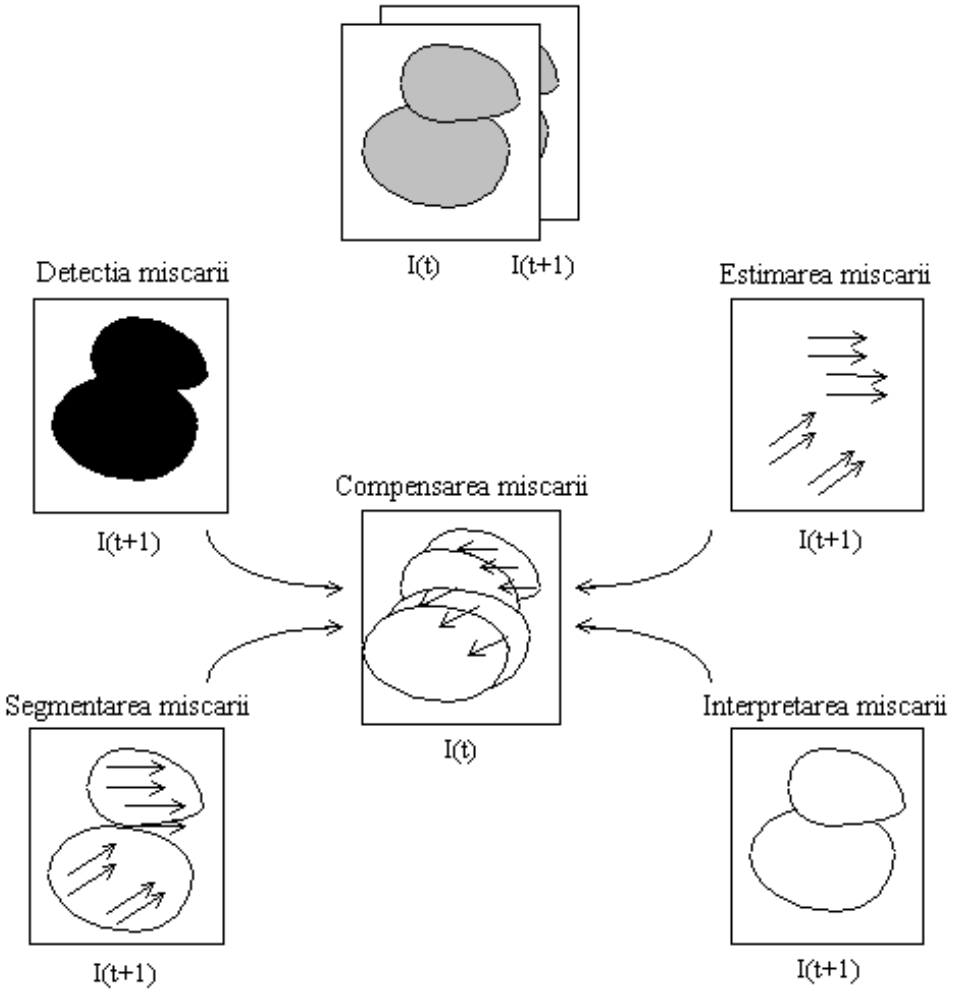
2. Faza estimării mişcării: Scopul acestei etape este de a cuantiza informaţia percepută relativ la mişcare (viteza aparentă), pe baza observării imaginilor. Studiul acestei faze a produs un numar mare de algoritmi, în sensul determinării câmpului vectorilor de mişcare. Câmpurile astfel obţinute sunt caracterizate prin:

- densitatea lor: câmpul dens, dacă măsura a fost aplicată fiecărui pixel, sau câmpul dispers, dacă faza de estimare este făcută la nivelul trasăturilor din imagini.
- tipul de primitive implicat: mişcări locale bazate pe pixel (cazul 1-D), mişcări ale contururilor, mişcări ale regiunilor (cazul 2-D), mişcări de obiecte (cazul 3-D).

3. Faza segmentării bazate pe mişcare: La fel ca şi în faza detecţiei, se poate analiza o scenă prin intermediul regiunilor omogene, ca un criteriu de mişcare. Spre deosebire de detecţia mişcării, un algoritm de segmentare bazat pe mişcare realizează împărţirea imaginii în regiuni având mişcări distincte şi identificabile.

4. Faza interpretării mişcării: Pentru multe aplicaţii de analiză a scenei, cât şi pentru cele de codare semantică cu modele de mişcare prestabilite, este de interes analiza unei secvenţe de imagini numai prin intermediul definiţiei calitative a mişcării: obiectul 1 - rotaţie, obiectul 2 - scalare etc.. Aceste interpretări de nivel înalt se obţin din faze anterioare, de nivel scăzut, ale detecţiei sau segmentării. Acest tip de procesare este esenţial atunci când, într-un sistem de comunicaţie, este necesară realizarea unei compresii şi diagnoze cu aceeaşi procesare algoritmică. Acesta este cazul sistemelelor de monitorizare activă ce utilizează reţelele de comunicaţie video.

5. Faza compensării mişcării: Această fază de procesare este specifică etapei de compresie a datelor. Compensarea mişcării unei imagini implică o analiză a priori a mişcării şi exploatarea datelor astfel obţinute pentru a crea o imagine virtuală $I_{comp}(x,y;t)$.

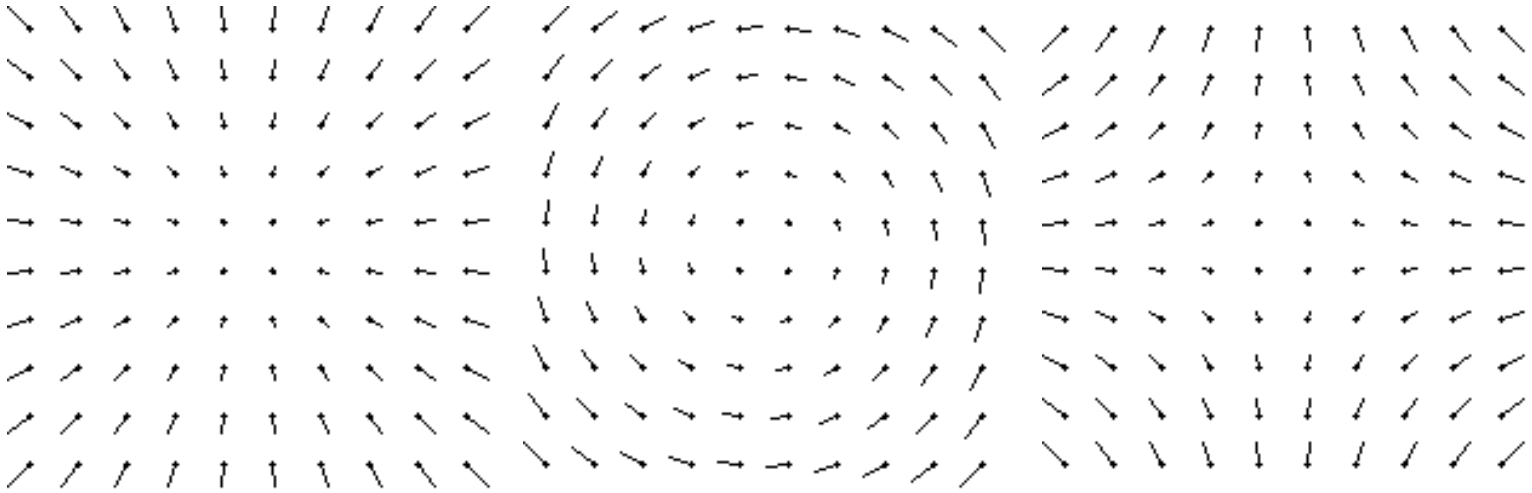


Fazele analizei mişcării

Elemente ale analizei mişcării

Reprezentarea în două dimensiuni a unei mişcări în trei dimensiuni se numeşte **câmp de mişcare.** Fiecărui punct din cadrul acestui câmp îi este asociat un **vector de viteză** în concordanţă cu direcţia mişcării, a vitezei şi a distanţei faţă de observator.

O abordare diferită analizează mişcarea prin calculul **fluxului optic**. În acest caz este necesară o diferenţă de timp foarte mică între două imagini consecutive, fără să existe schimbări importante de la o imagine la alta. Calculul fluxul optic rezultă din determinarea direcţiei mişcării şi a vitezei de deplasare a tuturor punctelor imaginii.



Exemple de fluxuri optice

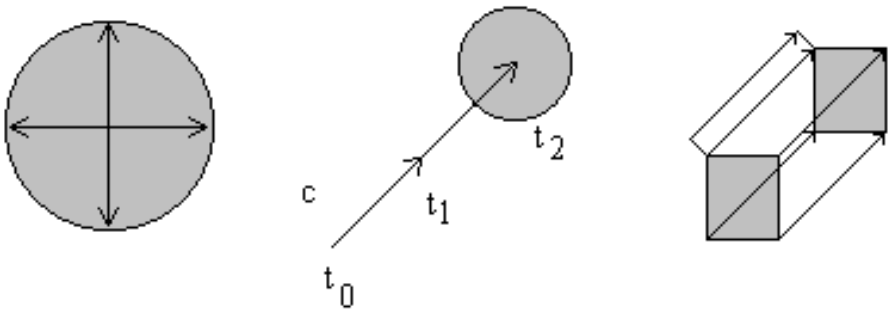
Dacă analiza mişcării se bazează pe detecţia obiectelor în mişcare, atunci următoarele ipoteze pot uşura etapa de localizare a obiectelor:

- **Viteza maximă**. Să presupunem că sunt preluate imagini ale unui obiect în mişcare la intervale de timp *dt*. Poziţia unui punct ce face parte din obiect, în imaginile viitoare, va fi în interiorul unui cerc ce are centrul în poziţia iniţială a punctului din obiect şi de raza $c_{max}dt$, unde c_{max} este viteza maximă de mişcare a obiectului.

- **Acceleraţii mici**. Modificarea în timp a vitezei este limitată la valoarea unei constante.

- **Similaritate în mişcare**. Toate punctele obiectului se mişcă în mod similar.

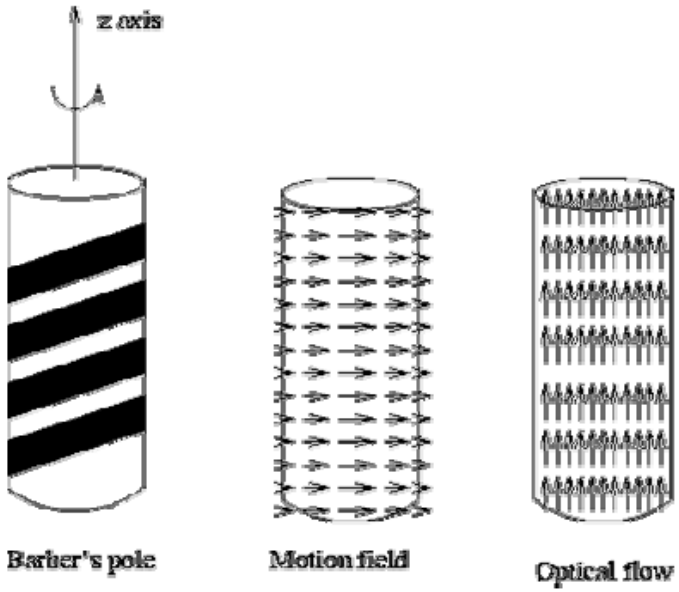
- **Corespondenţa**. Obiectele rigide au aceeaşi conformaţie în timp. Fiecare punct al unui obiect corespunde exact unui punct din următoarea imagine şi vice versa. Există şi excepţii datorate rotaţiilor sau ocluziilor.



a) b) c)

Ipoteze asupra mişcării obiectelor: a) viteza maximă;
b) acceleraţie mică; c) obiecte rigide

Barber pole illusion



Metode de analiză a mişcării

Metoda diferențială

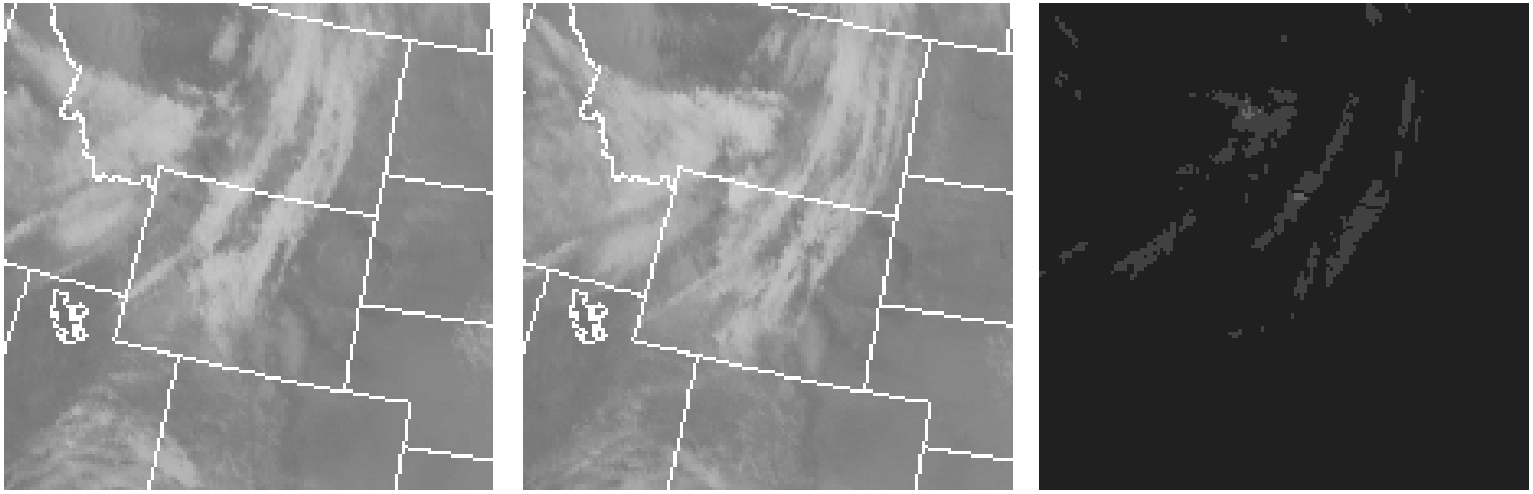
Simpla "scădere" a imaginilor, surprinse la un anumit interval de timp poate să facă posibilă detecția mişcării, presupunând o cameră video fixă şi o iluminare constantă. Diferența dintre imaginile consecutive $I(i,j,t\text{-}1)$ şi $I(i,j,t)$, $d(i,j)$, este o imagine binară, în care valorile diferite de zero aparțin zonelor aflate în mişcare, adică zone în care există o diferență semnificativă între nivelele de gri.

$$d(i,j) = \begin{cases} 0 & \text{daca} \left| I(i,j,t) - I(i,j,t-1) \right| \le \varepsilon \\ 1 & \text{altfel} \end{cases} \qquad (2.2)$$

unde ε este un număr pozitiv mic.

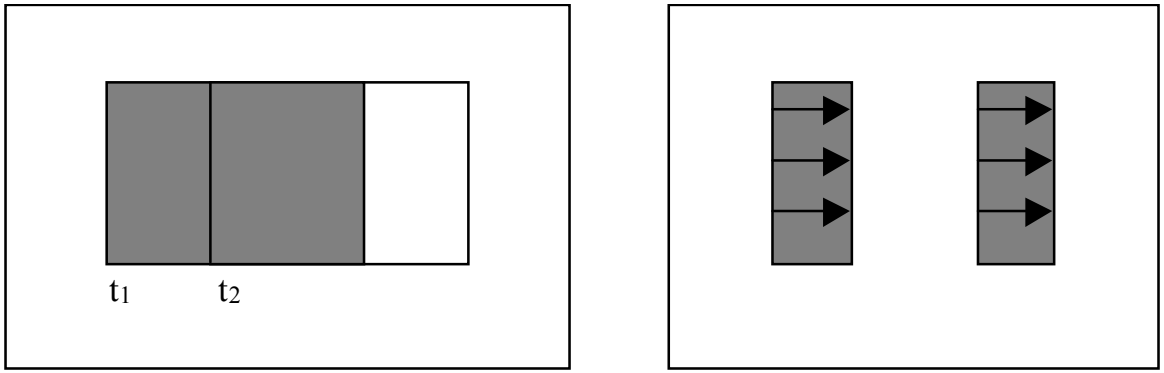
Fie $I(i,j,t\text{-}1)$ şi $I(i,j,t)$ două imagini consecutive preluate la un interval de timp. Un element $d(i,j)$ al imaginii diferență dintre $I(i,j,t\text{-}1)$ şi $I(i,j,t)$ poate avea valoarea 1 în oricare din cazurile următoare:

a) $I(i,j,t)$ este un pixel ce aparține unui obiect aflat în mişcare
   $I(i,j,t\text{-}1)$ este un pixel ce aparține fundalului static (sau invers)
b) $I(i,j,t\text{-}1)$ este un pixel ce aparține unui obiect aflat în mişcare
   $I(i,j,t)$ este un pixel ce aparține altui obiect aflat şi el în mişcare
c) $I(i,j,t\text{-}1)$ este un pixel ce aparține unui obiect aflat în mişcare
   $I(i,j,t)$ este un pixel ce altei părți a aceluiaşi obiect în mişcare
d) zgomot, imprecizia în poziționarea camerei video statice.



a) b) c)

Metoda diferențială de detecție a mișcării: a) și b) secvența de imagini ale satelitului NOAA GOES 8; c) imaginea diferența



a) b)

Erori de detecție a mișcării; a) poziția obiectului la momentele de timp t_1 (gri) și t_2; b) câmpul de mișcare rezultat

Fluxul optic

Fluxul optic reflectă modificările din imagini datorate mişcării dintr-un intervalul de timp *dt*. Câmpul de flux optic este câmpul de viteze ce reprezintă mişcarea tridimensională a punctelor obiectului transpusă într-o imagine bidimensională. Fluxul optic este o abstracție ce ar trebui să reprezinte numai acele modificări de intensitate din imagine datorate mişcării, iar toate celelalte schimbări din imagini reflectate în fluxul optic să fie considerate erori ale detecției fluxului optic. De exemplu fluxul optic nu ar trebui să fie sensibil la schimbări ale iluminării sau la mişcări ale umbrelor. Totusi, un flux optic diferit de zero este detectat în cazul unei sfere statice iluminate de o sursă în mişcare. În mod asemănător, o sferă de suprafață netedă în rotație, iluminată de o sursă statică produce un flux optic nul, în ciuda unui câmp de viteze diferit de zero.

Calculul fluxului optic se bazează pe două premize:
1. strălucirea fiecărui punct al obiectului este constantă în timp;
2. puncte vecine din planul imaginii au mişcări similare.

Fluxul optic poate oferi o descriere a mişcării şi contribuie la buna interpretare a imaginii, chiar dacă nu se obțin parametrii cantitativi din analiza mişcării. Fluxul optic se foloseşte pentru a studia o varietate largă de mişcări:
- observator ce se mişcă şi obiecte statice;
- observator static şi obiecte ce se mişcă;
- observator şi obiecte aflate în mişcare.

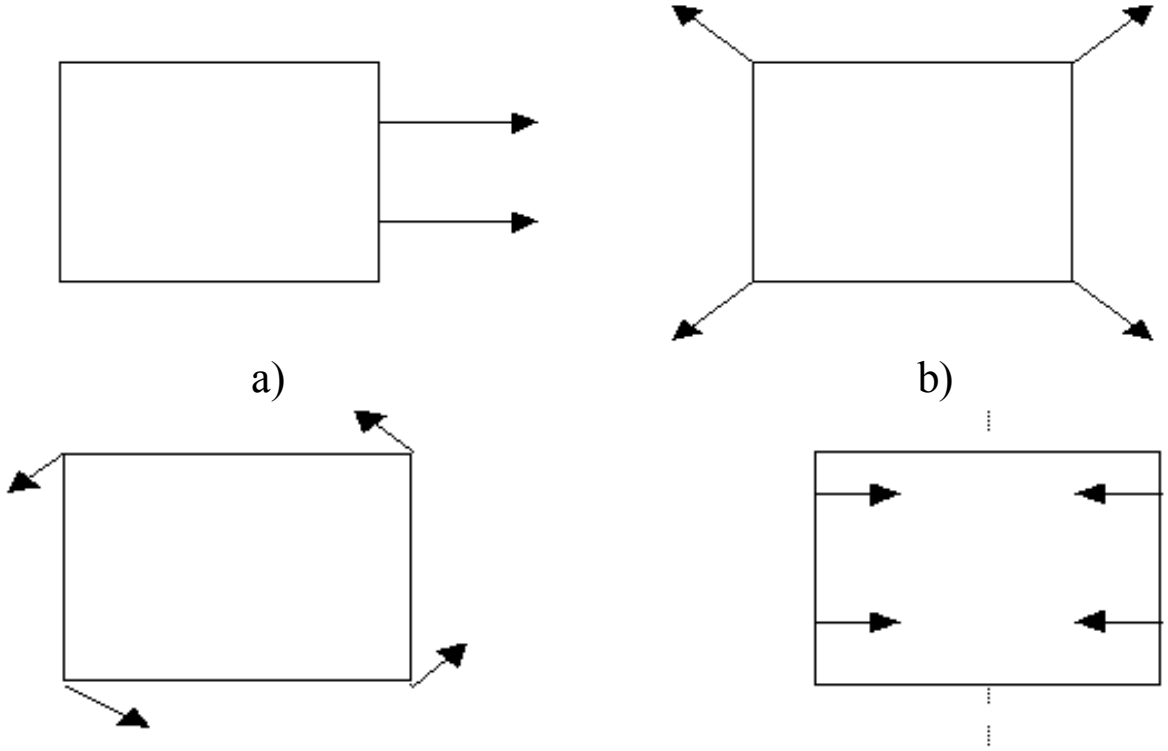
Mişcarea, aşa cum apare în imaginile dinamice, este în general o combinaţie de patru elemente de bază:

1. translaţie la o distanţă constantă de observator:
2. translaţie în adâncime relativ la observator;
3. rotaţie la distanţă constantă pe axa de rotaţie;
4. rotaţie a unui obiect perpendicular pe axa vederii.

Analiza mişcării bazată pe fluxul optic poate recunoaşte aceste elemente de bază, prin aplicarea unor operatori relativ simplii. Recunoaşterea mişcării se bazează pe următoarele date, aşa cum sunt ilustrate şi în figura 2.6:

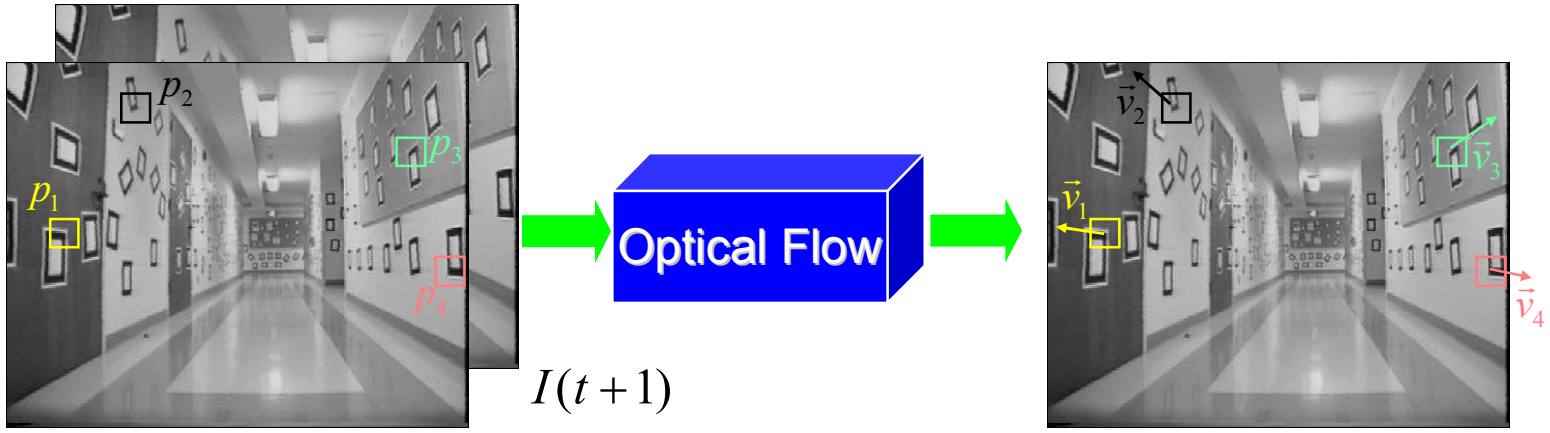
- translaţia la distanţă constantă este reprezentată ca un set de vectori paraleli;
- translaţia în adâncime formează un set de vectori ce au un punct comun de expansiune;
- rotaţia la distanţă constantă rezultă dintr-un set vectori de mişcare concentrici;
- rotaţia perpendiculară pe axa de vedere formează unul sau mai multe seturi de vectori pornind direct de la liniile de segment drepte.

Determinarea exactă a axei de rotație sau traiectoriei de translație poate fi realizată, dar cu o creştere considerabilă a efortului de analiză.



Formele de mişcare: a) translație la distanță constantă; b) translație în adâncime; c) rotație la distanță constantă; d) rotație perpendiculară pe axa de vedere.

Flux optic



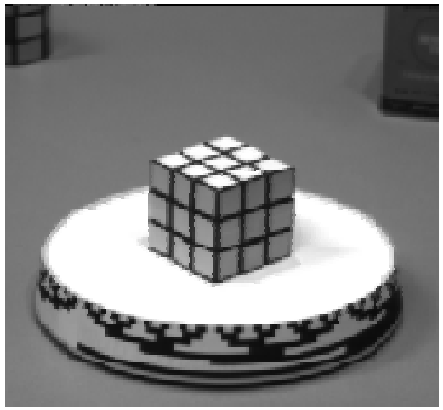
$I(t), \{p_i\}$

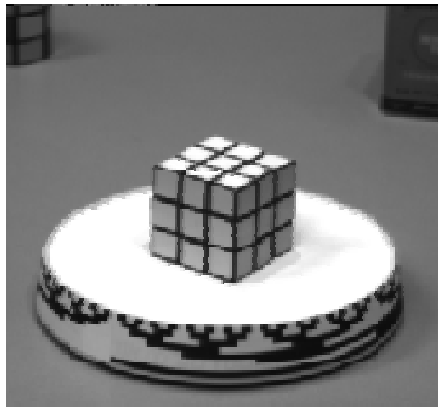
Secventa de imagini

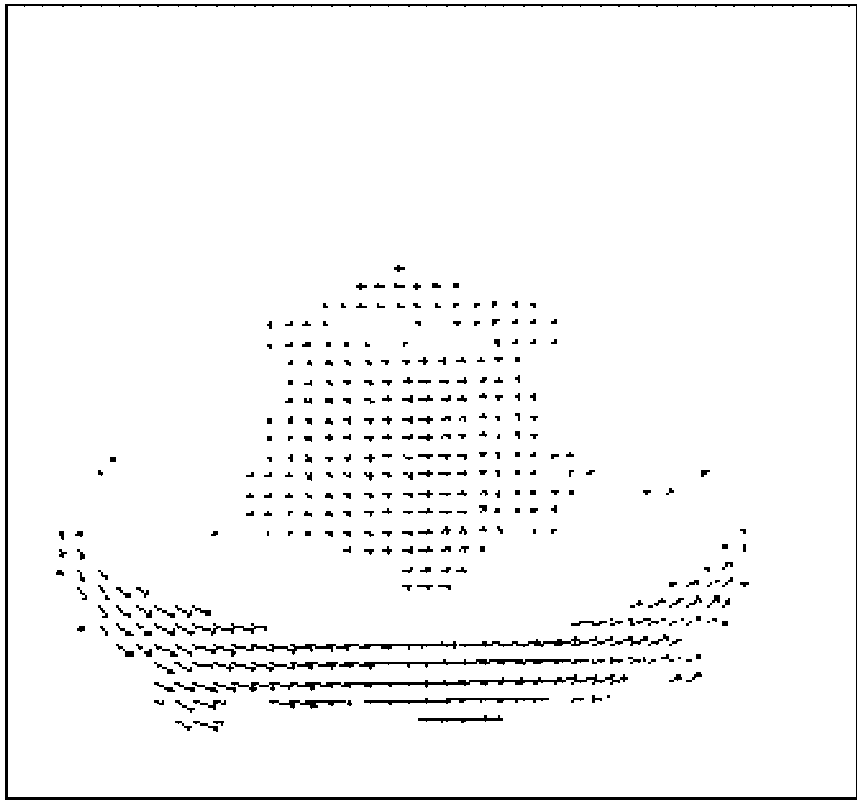
Vectori de miscare $\{\vec{v}_i\}$

Flux optic

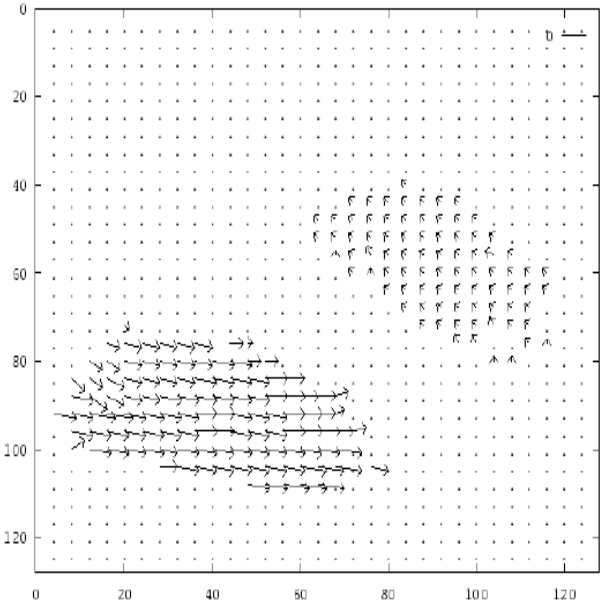
exemplu







Flux optic



Tehnici de estimare locală a mişcării

Aproape toate tehnicile de estimare a miscarii pot fi grupate într-una din urmatoarele categorii:

- bazate pe region matching
- bazate pe gradient
- bazate pe o analiză a frecvenței
- bazate pe urmărirea anumitor trăsături

Toate metodele au la bază estimarea locală a mişcării de tranzlație din imagine.

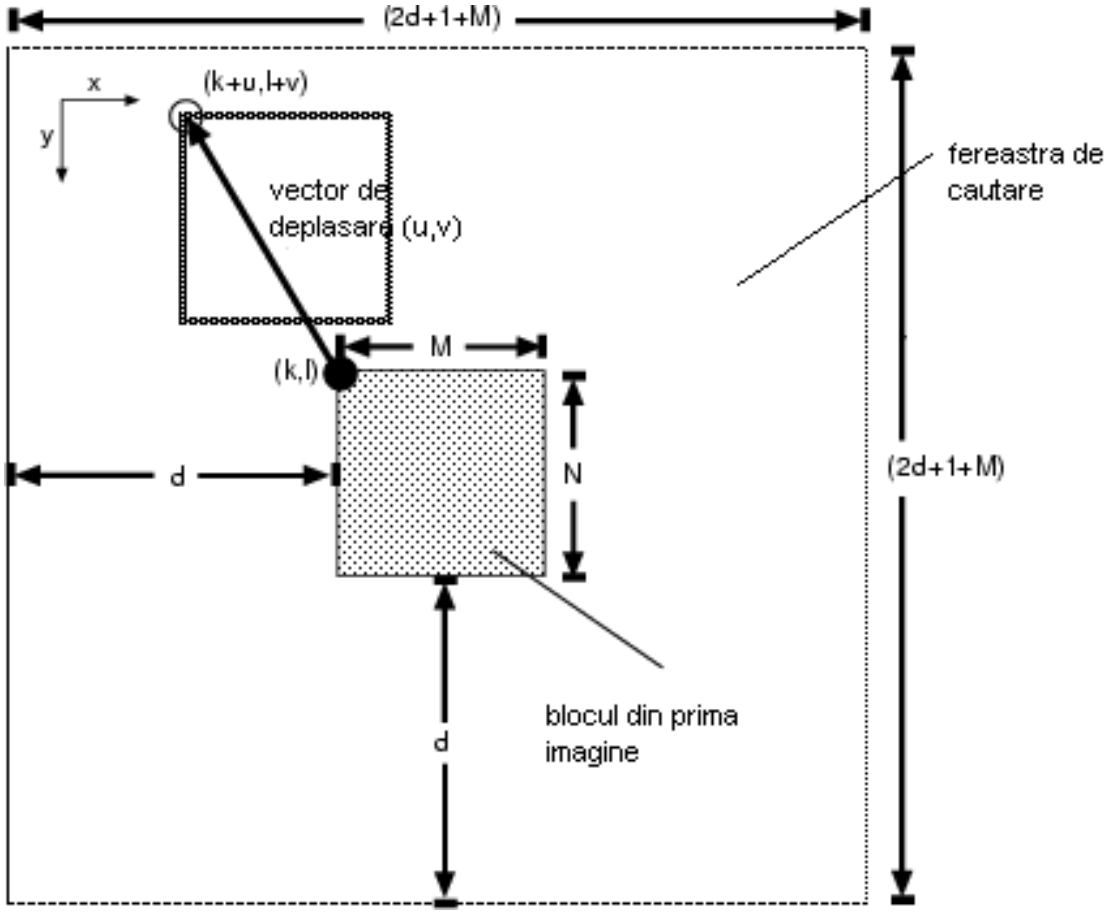
Block Matching

Metoda block matching este general aplicabilă unei regiuni rectangulare a cărei dimensiune a fost dinainte fixată. Se defineşte o suprafață de căutare pentru vectorul de deplasare. Fie ea:

$$|u| \le \Delta_h \text{ şi } |v| \le \Delta_v \tag{4.42}$$

Ca o regulă generală, vectorii de deplasare acceptabili sunt multipli întregi ai perioadei de eşantionare spațiale sau a unei jumătăți de perioadă. Dacă dimensiunea blocului este de K linii şi L coloane, mărimea suprafeței de căutare este $(K + 2\Delta_h)(L + 2\Delta_v)$. Numărul de deplasări posibile în acelaşi caz este de $(2\Delta_h + 1)(2\Delta_v + 1)$. De exemplu, pentru un deplasament maxim pe orizontală şi verticală de 7 pixeli, numărul de deplasamente posibile este 225. După definirea suprafeței de căutare trebuie stabilit criteriul ce trebuie optimizat.

Block Matching



Modelul block matching

Criteriul cel mai frecvent utilizat este media valorii absolute a diferenței dintre cadrele deplasate:

$$(\hat{u},\hat{v}) = \underset{\substack{(u,v)\in Z^2 \\ |u|\le\Delta_h,|v|\le\Delta_v}}{\arg\min} \sum_{i=0}^{K-1}\sum_{j=0}^{L-1}\left|I_{DFD}(i,j;u,v)\right| \tag{4.43}$$

sau media pătratelor diferențelor cadrelor deplasate:

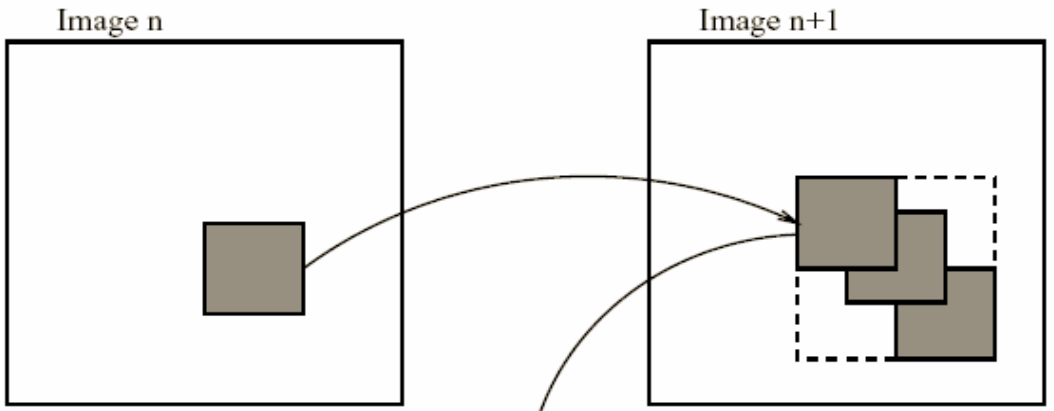
$$(\hat{u},\hat{v}) = \underset{\substack{(u,v)\in Z^2 \\ |u|\le\Delta_h,|v|\le\Delta_v}}{\arg\min} \sum_{i=0}^{K-1}\sum_{j=0}^{L-1}\left(I_{DFD}(i,j;u,v)\right)^2 \tag{4.44}$$

sau valoarea normalizată a mediei pătratelor diferențelor cadrelor deplasate:

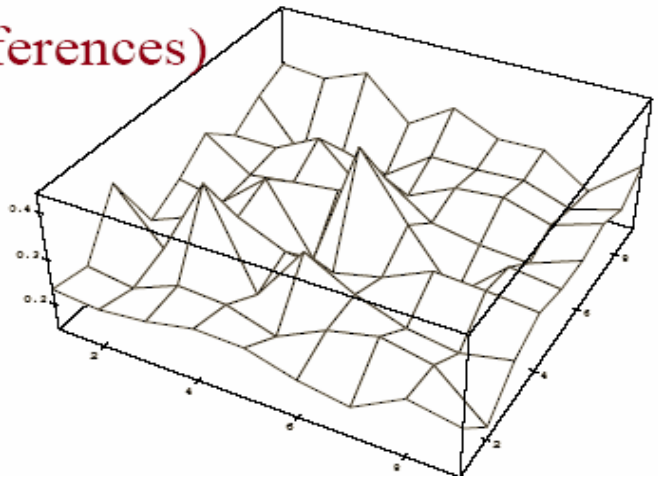
$$(\hat{u},\hat{v}) = \underset{\substack{(u,v)\in Z^2 \\ |u|\le\Delta_h,|v|\le\Delta_v}}{\arg\min} \frac{\sum_{i=0}^{K-1}\sum_{j=0}^{L-1}\left(I_{DFD}(i,j;u,v)\right)^2}{\sum_{i=0}^{K-1}\sum_{j=0}^{L-1}I^2(i,j;k)} \tag{4.45}$$

unde $I_{DFD}(i,j;u,v) = I(i,j;k) - I(i-u,j-v;k-1)$ reprezintă diferența cadrului deplasat. Deoarece primul criteriu este cel mai puțin costisitor, el este cel mai adesea utilizat.

Block Matching



SSD surface
(sum of squared differences)



Block Matching

Să presupunem că blocul de referinţă este de dimensiune MxN pixeli, iar deplasarea maximă este $\pm d$, în ambele direcţii, orizontală şi verticală. Vectorul de mişcare (u,v) se obţine determinând blocul din cea de-a doua imagine, ce se potriveşte cel mai bine cu cel de referinţă, realizând o căutare în interiorul ferestrei de dimensiune $(2d+1)$x$(2d+1)$. Fereastra de căutare este centrată pe blocul de referinţă din prima imagine.

Cel mai direct algoritm BMA este algoritmul de căutare totală (FS). Acesta găseşte vectorul de mişcare optimal dintre toţi vectorii de mişcare din interiorul ferestrei. Dar el necesită un număr de $(2d+1)^2$ calcule pentru determinarea distanţei dintre cele două blocuri şi determinarea distanţei minime. Aceste calcule complexe fac ca acest algoritm să nu poată fi folosit în aplicaţii de timp real. In acest sens, s-au conceput mai mulţi algoritmi pentru a reduce complexitatea calculelor. Dintre aceştia amintim: căutare în trei paşi, căutare 2D logaritmică, căutare ortogonală, căutare în cruce, căutare în patru paşi, căutarea de tip gradient descend.

Block Matching

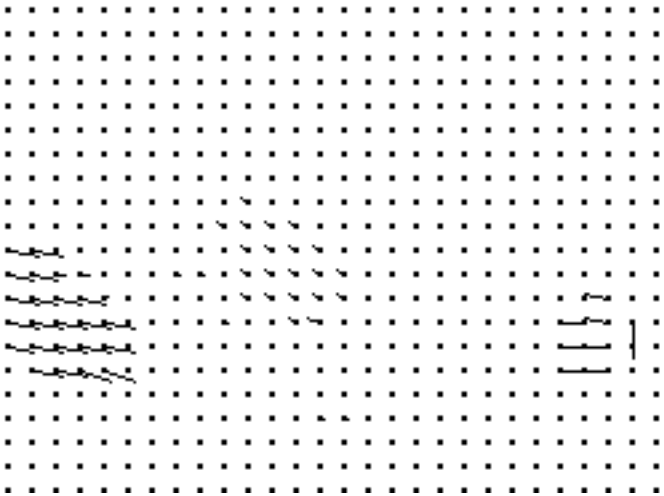
a) b) c)

Secvenţa “translating tree”; a) şi b) doua cadre succesive;
c) fluxul optic determinat cu NFS pentru un bloc de 13

a)

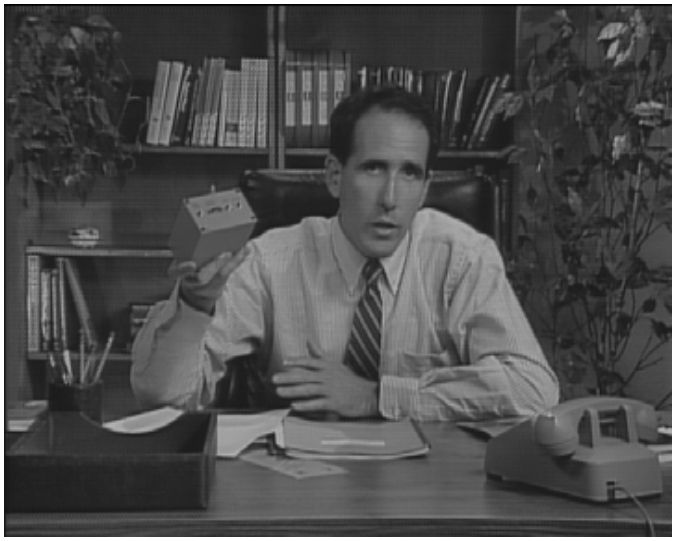
b)

c)

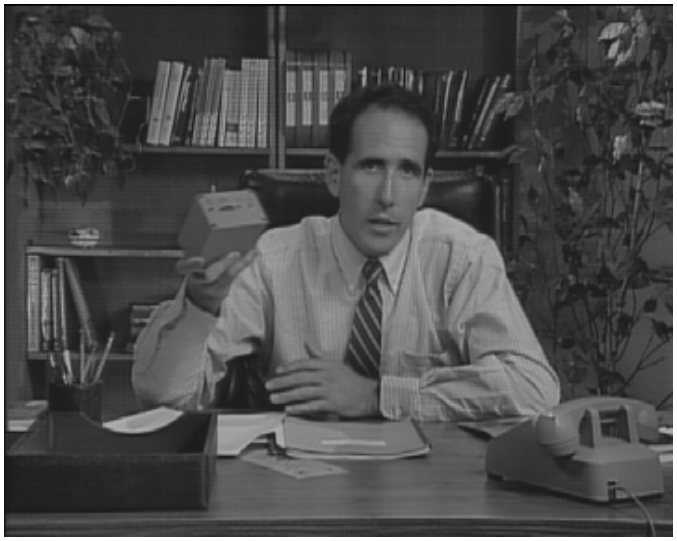


d)

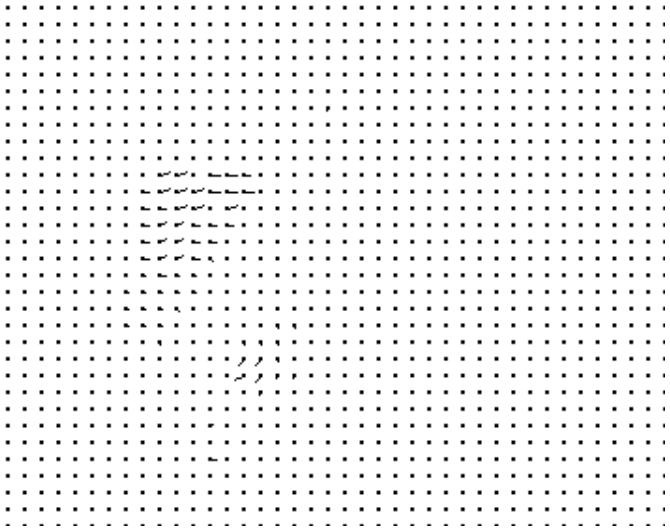
Secvența “Hamburg Taxi”; a) şi b) cadrele 1 si 2;
fluxul optic NFS pentru c) bloc de 5 şi d) bloc de 9



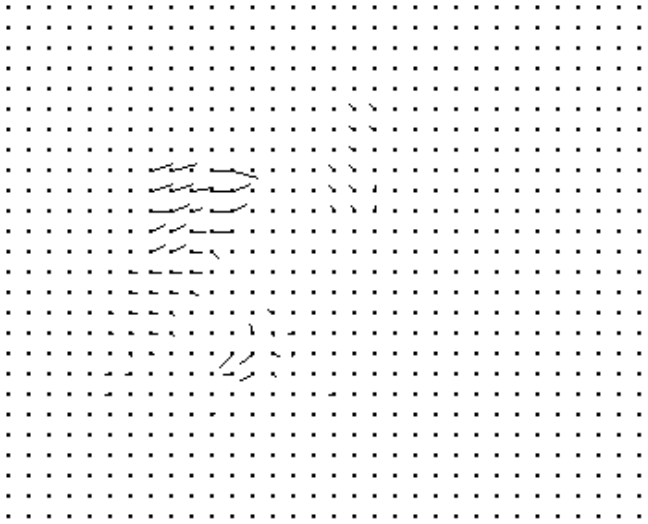
a)



b)



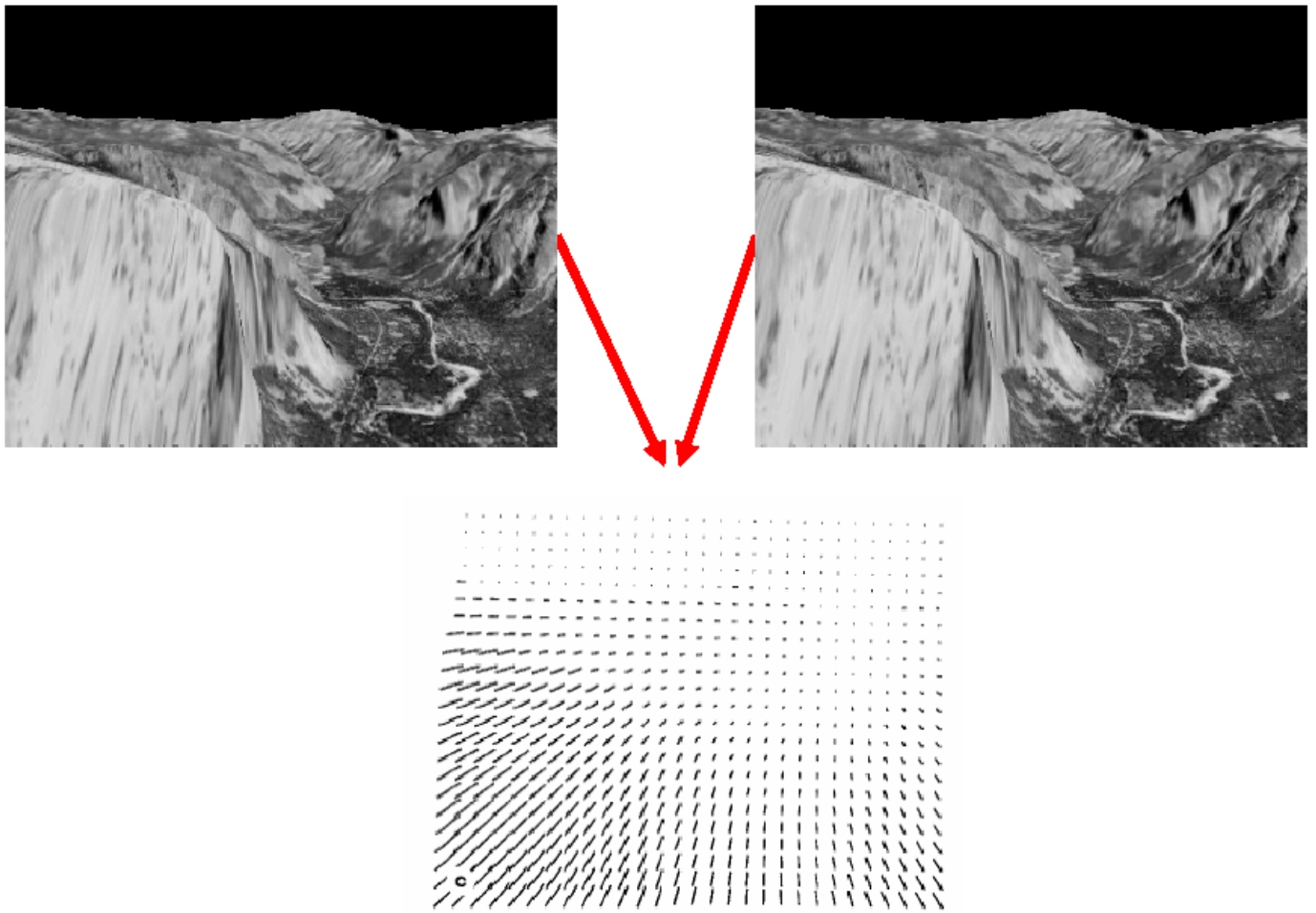
c)



d)

Secvența “salesman”; a) cadrul 1 şi b) 5; fluxul optic pentru c) d=7 şi d) d=13

Block Matching



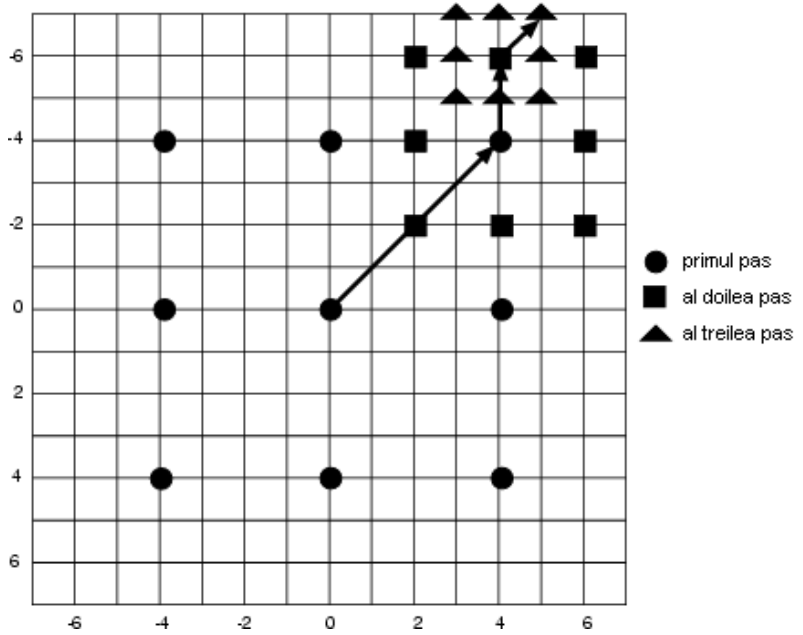
Algoritmi de reducere a complexității căutării

Căutarea în trei paşi

Metoda “three step search” (3SS) se bazează pe o abordare a căutării de la grosier la fin, ceea ce duce la o scădere logaritmică la fiecare pas.

Dimensiunea pasului inițial este jumătate din mărimea maximă a deplasării d.

Pentru fiecare pas se verifică nouă puncte, iar punctul de minim al unei măsuri devine centrul de plecare al pasului următor. Numărul de puncte care se verifică este: $[1+8[\log_2(d+1)]]$



Algoritmi de reducere a complexității căutării

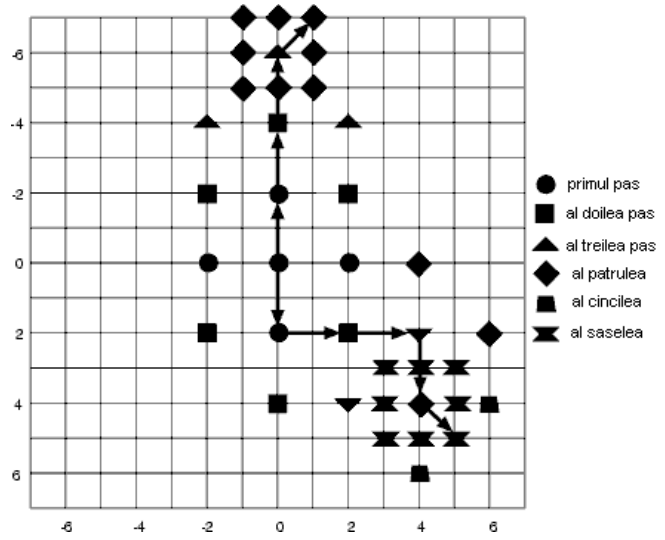
Căutarea logaritmică 2-D

Pasul inițial este de mărime d/4.

La fiecare pas, căutarea este făcută doar în cinci poziții care includ un punct de mijloc şi patru puncte din două direcții principale, orizontală şi verticală.

În primul pas, cele patru puncte utilizate se află la mijloc, între centrul suprafeței şi marginile ei. Dacă optimul este în punctul de mijloc, suprafața de căutare este decrementată cu jumătate.

Procedura continuă până când suprafața de căutare ajunge la dimensiunile de 3x3. În acest ultim pas, toate cele nouă poziții sunt testate pentru a determina poziția minimului .



Algoritmi de reducere a complexității căutării

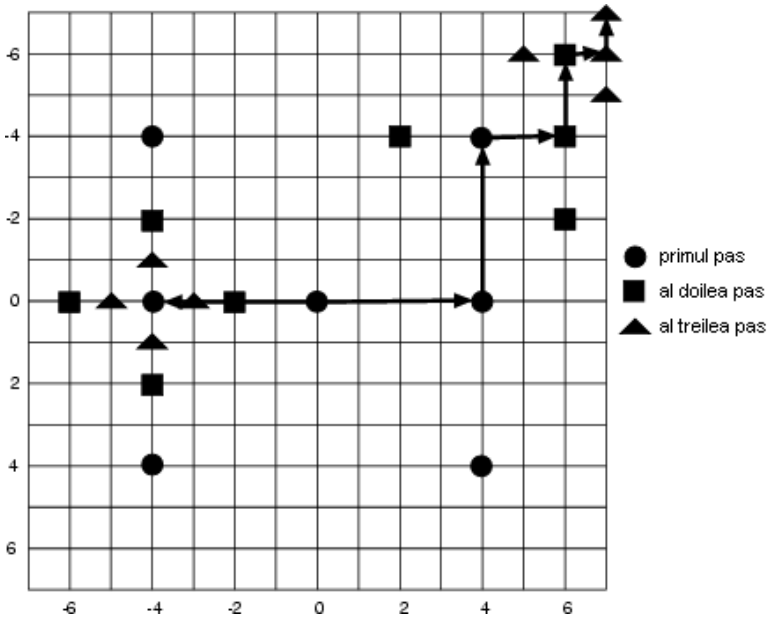
Căutarea ortogonală

Aceasta tehnică (orthogonal search) constă într-o căutare prin perechi de paşi orizontali şi verticali, având o scădere logaritmică la fiecare pas.

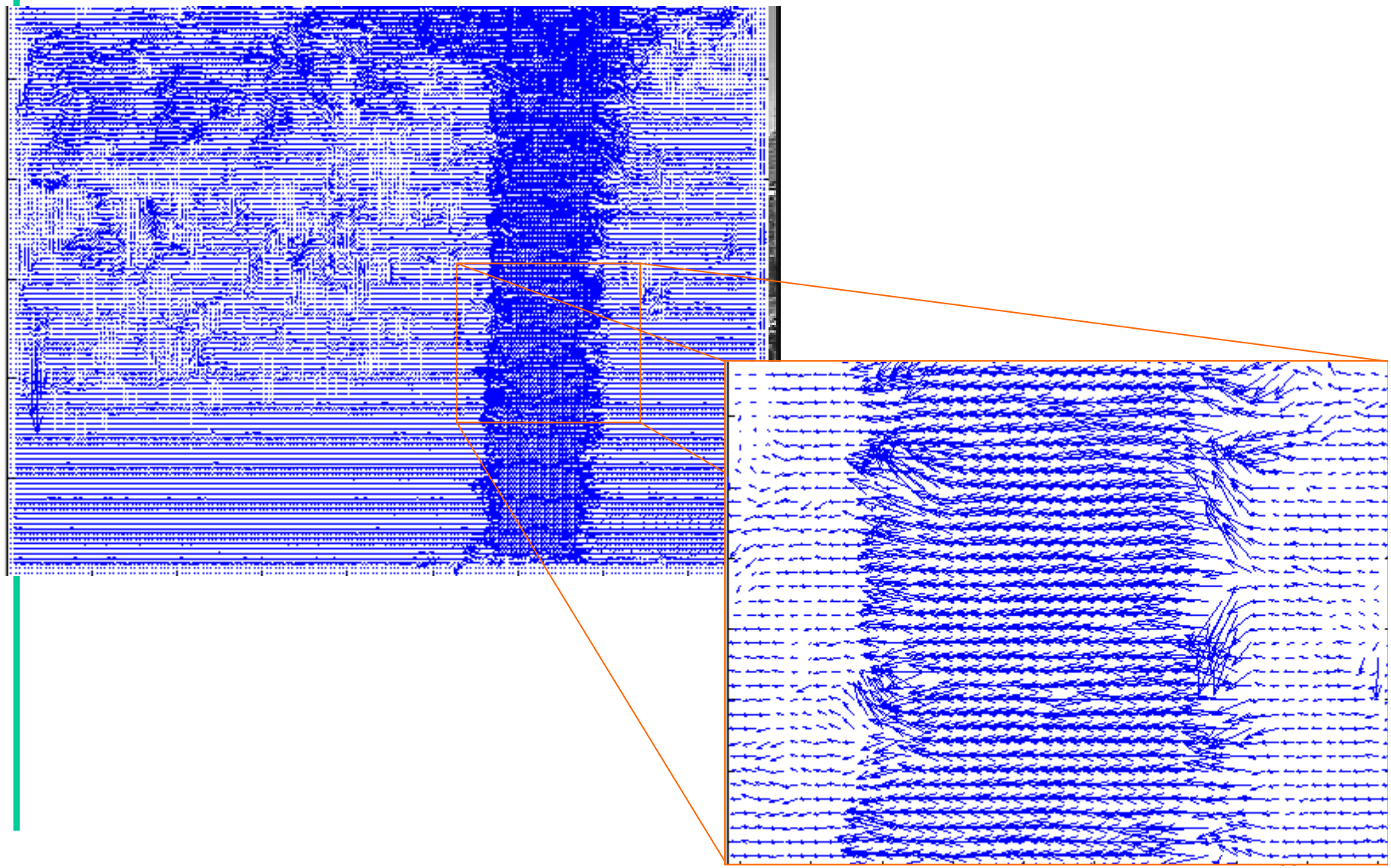
Pasul inițial este d/2. Se începe cu o căutare pe orizontală şi sunt verificați câte trei pixeli pe această direcție.

Minimul devine centru de căutare pe verticală, care de asemenea constă în verificarea a trei puncte. Apoi pasul descreşte la jumătate şi se reia algoritmul. Algoritmul se termină când pasul devine 1. În cazul general, algoritmul necesită

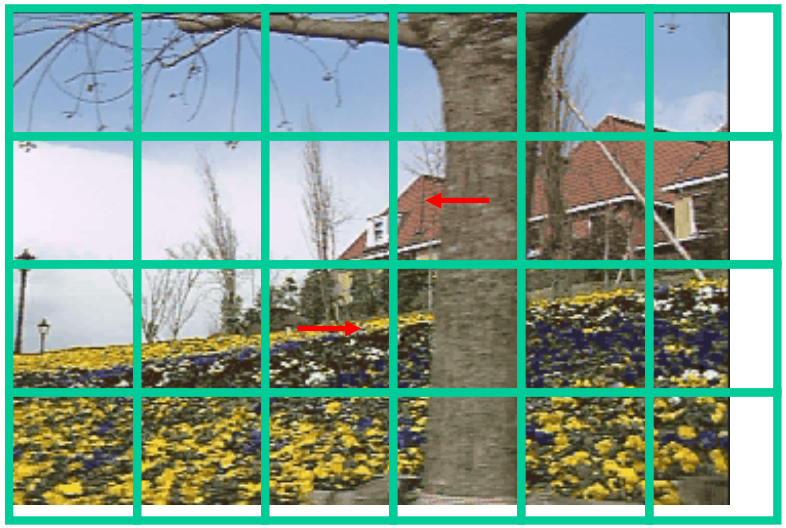
$[1+4(\log_2(d+1))]$ puncte de căutare.

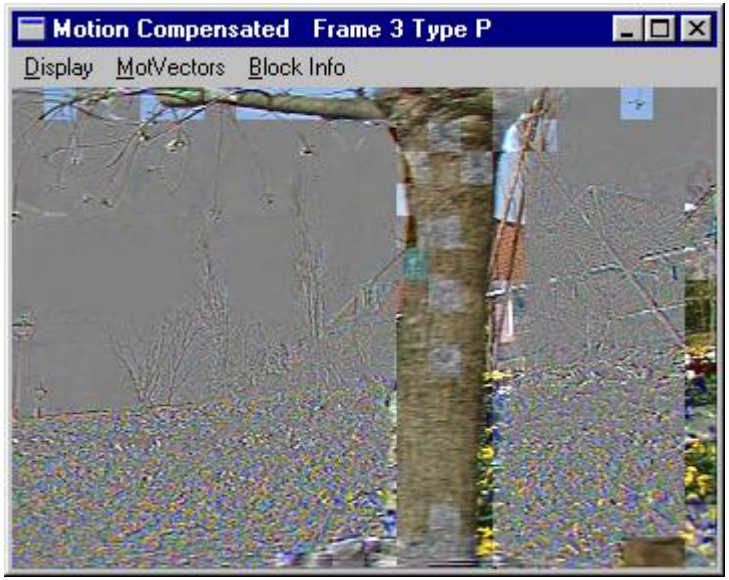


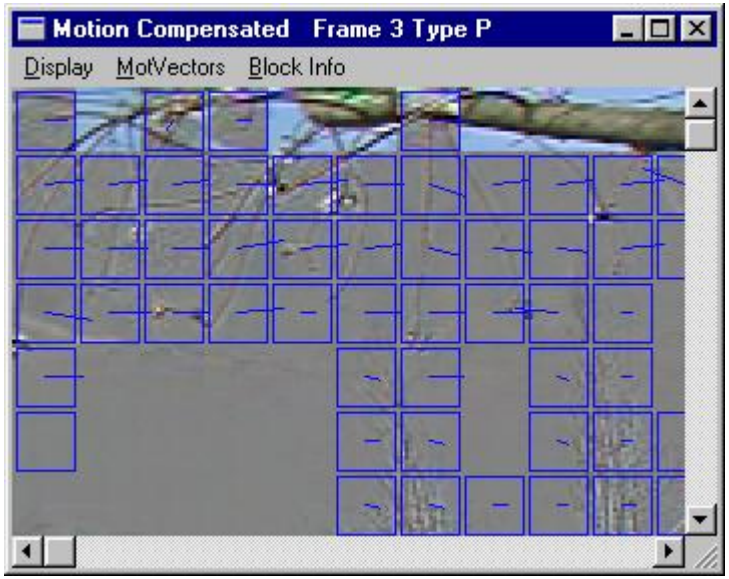
Aplicatii ale fluxului optic



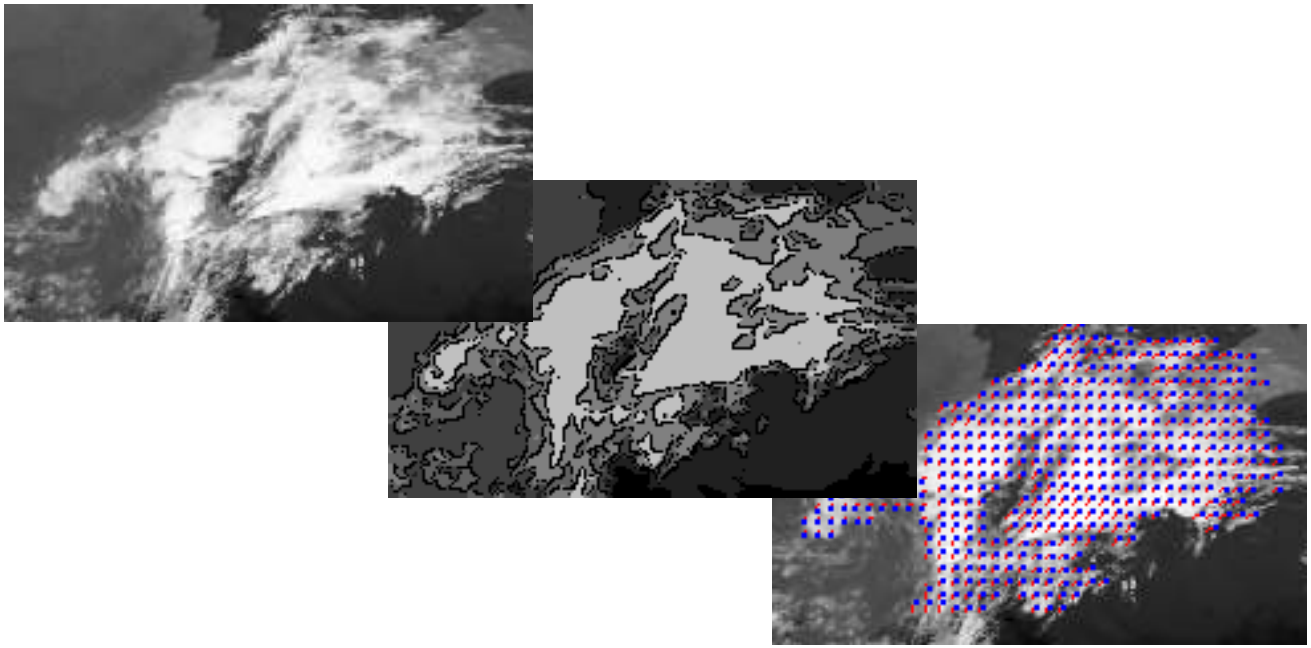
Aplicatii ale fluxului optic







Tehnici de procesare a mişcării norilor din secvențe de imagini satelit

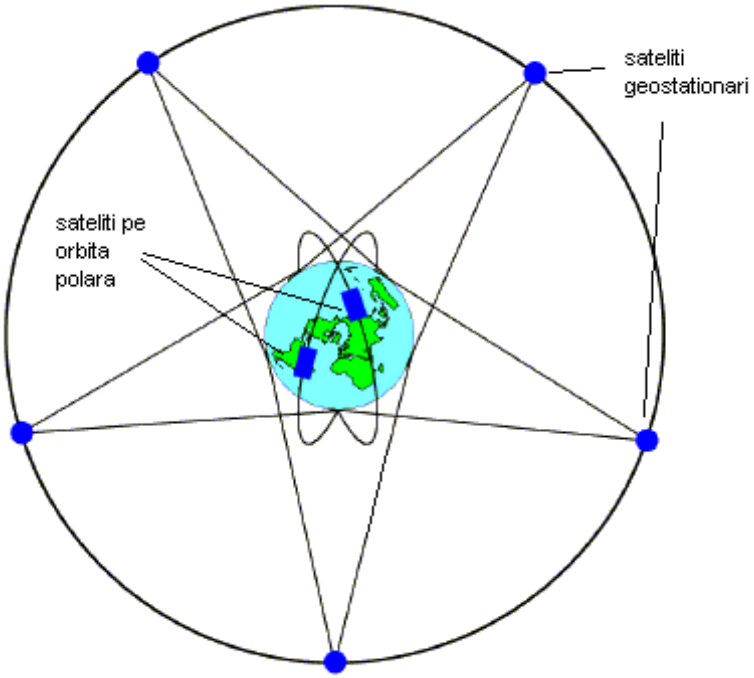


Introducere

- a observa, a înțelege şi a prevedea fenomenele meteorologice
- meteorologia s-a dezvoltat datorită progresului ştiințific şi tehnologic
- sateliții meteorologici - imagini cu conținut ce variază în funcție de mărimile fizice măsurate
- tehnicile pot oferi asistența meteorologilor în vizualizarea şi interpretarea secvențelor de imagini satelit

Problematica imaginilor meteorologice satelit

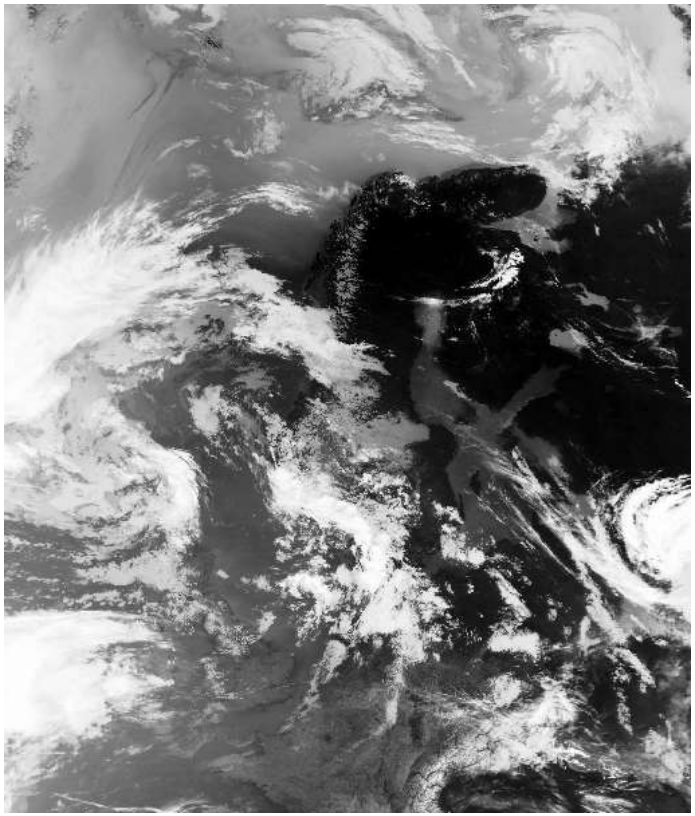
Tipuri de sateliți meteorologici



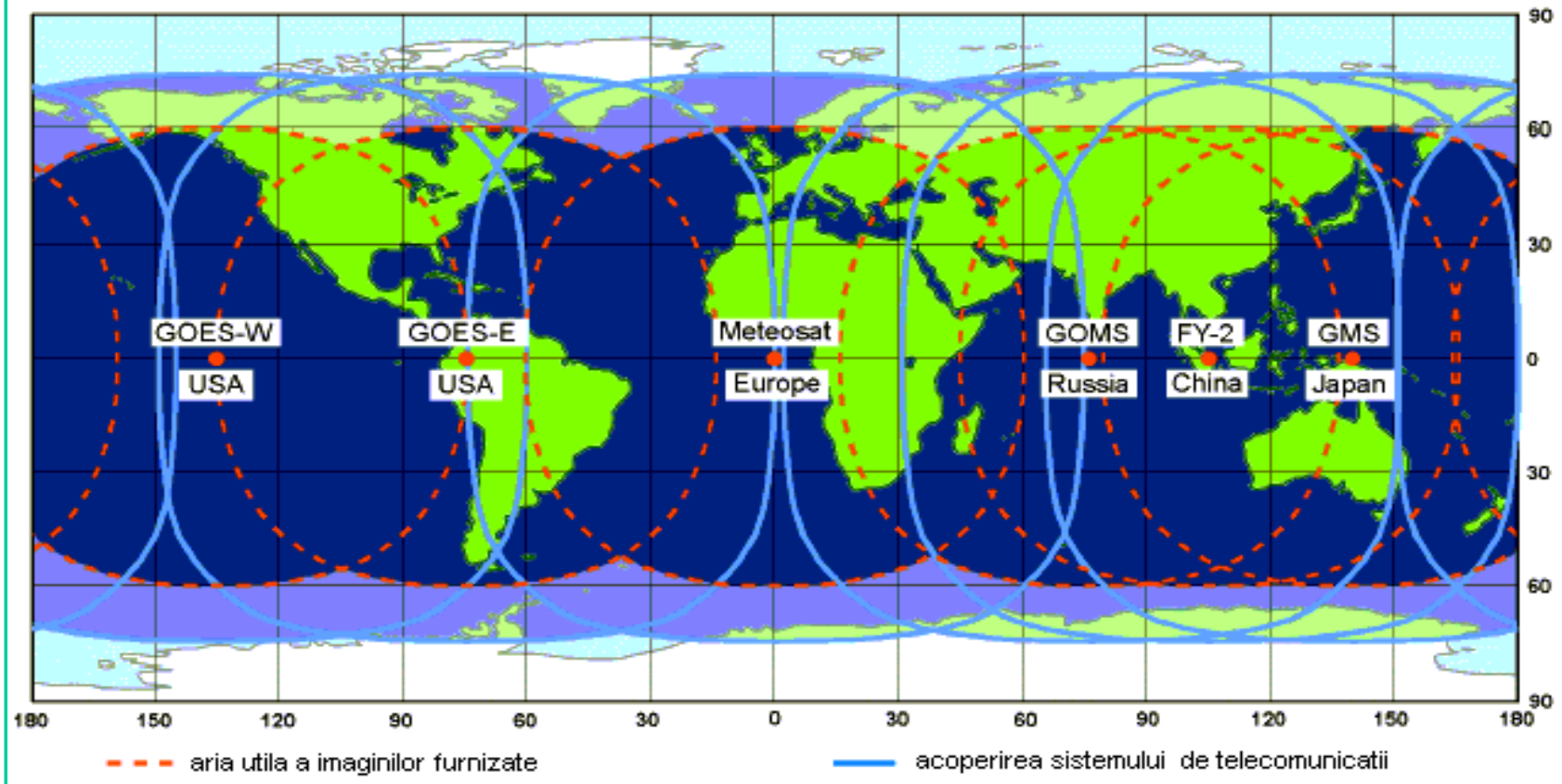
Sateliți pe orbite polare

NOAA 6-9

Altitudine	Înclinare	perioada de revoluție	decalaj	rezoluție spațială
850 km	99°	102 min.	259°5	1.1×1.1 km^2



Sateliți geostaționari

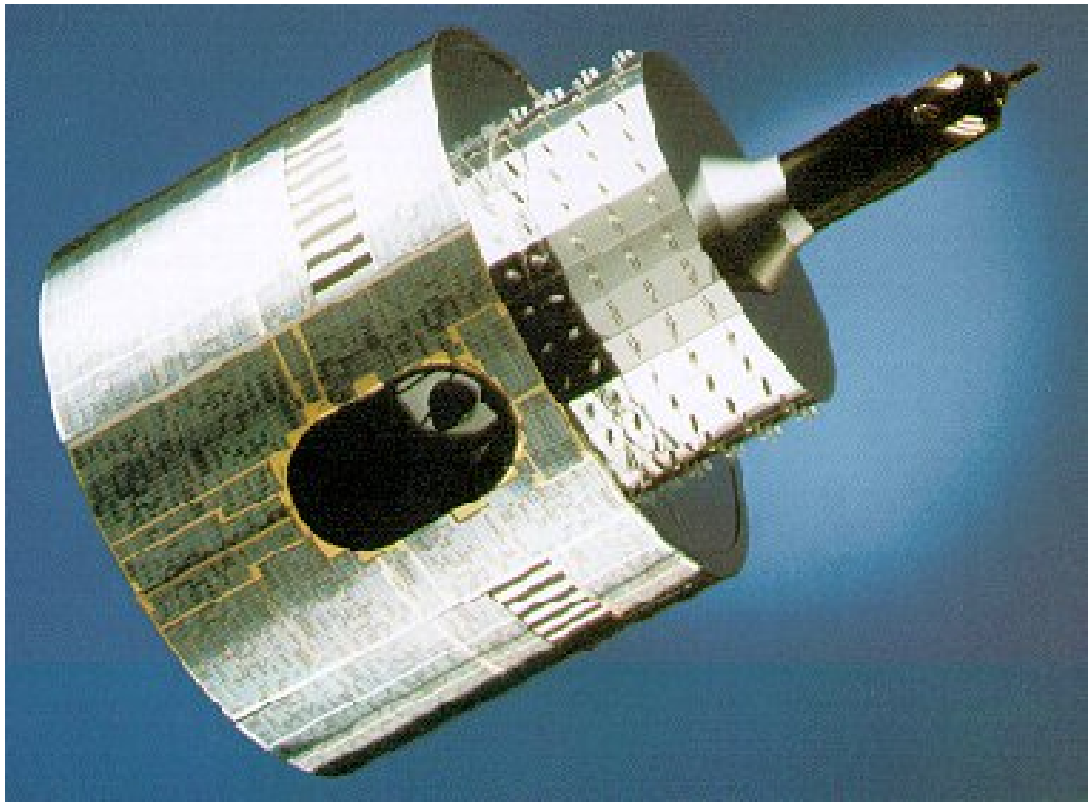


Achiziția imaginilor METEOSAT

METEOSAT 4-7

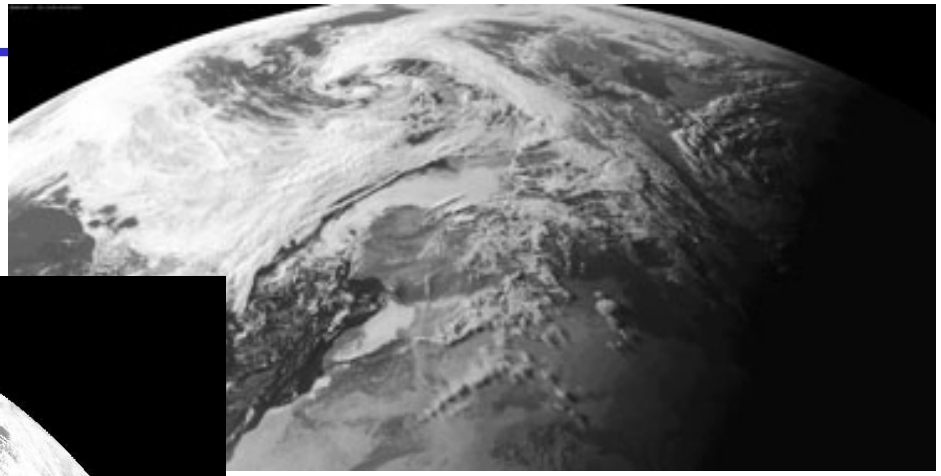
Denumirea canalului	vizibil	vapori de apă	infraroşu
banda spectrală [μm]	**0.45 - 1.0**	**5.7 - 7.1**	**10.5 - 12.5**
detectori (+redundanți)	**2(+2)**	**1(+1)**	**1(+1)**
linii × pixeli pe linie	**5000×5000**	**2500×2500**	**2500×2500**
rezoluție la verticală [km^2]	**2.5×2.5**	**5×5**	**5×5**
frecvența de achiziție [min]	**30**	**30**	**30**
disponibilitatea imaginilor [TU]	**7 – 17**	**0 – 24**	**0 – 24**
biți per pixel	**8**		
viteza de transmisie la sol	**333 kbps (normal)**		**2.7 Mbps (burst)**

Achiziția imaginilor METEOSAT

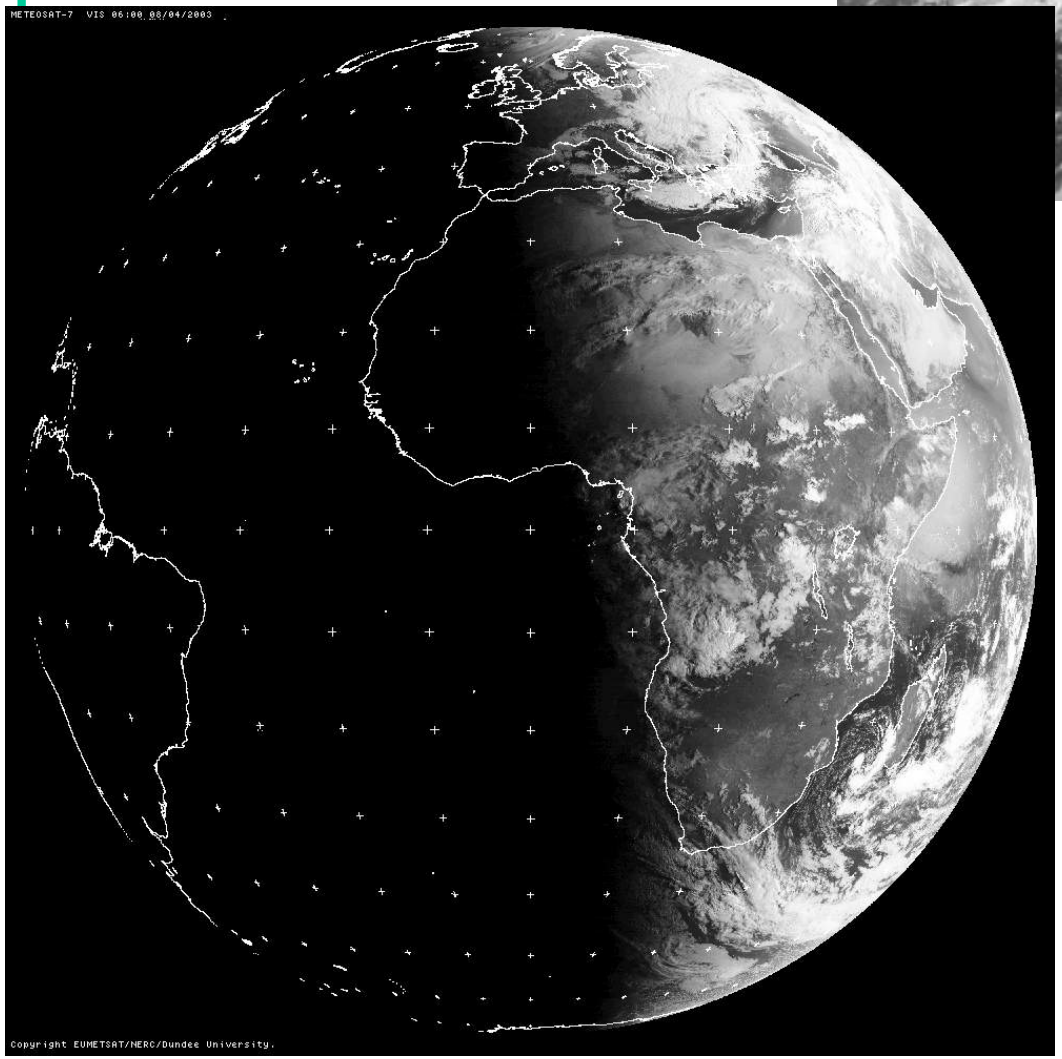


Satelitul METEOSAT 7

Canalul vizibil



Sectorul C2

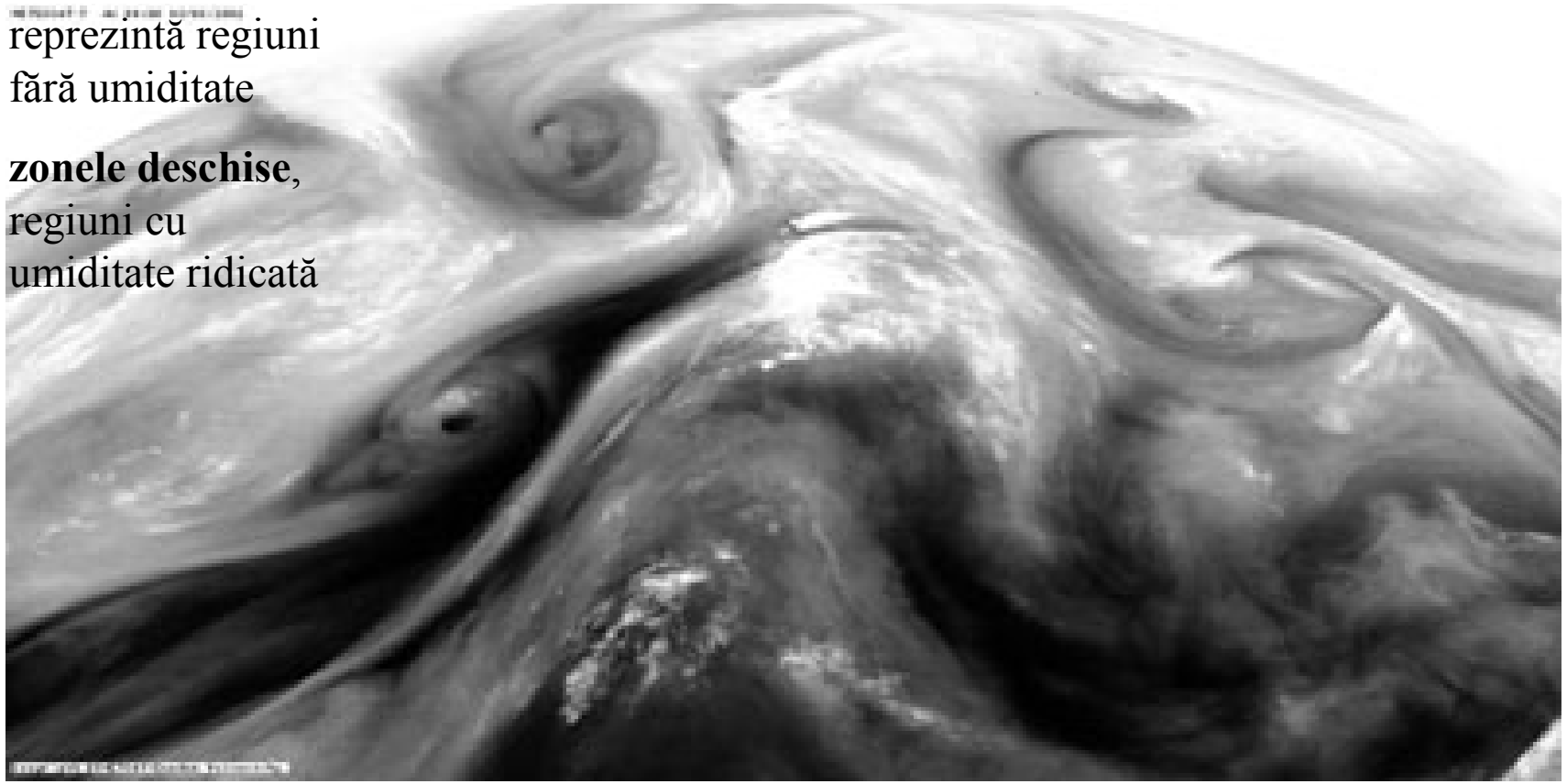


Full earth disk

Canalul vapori de apă

zonele închise
reprezintă regiuni
fără umiditate

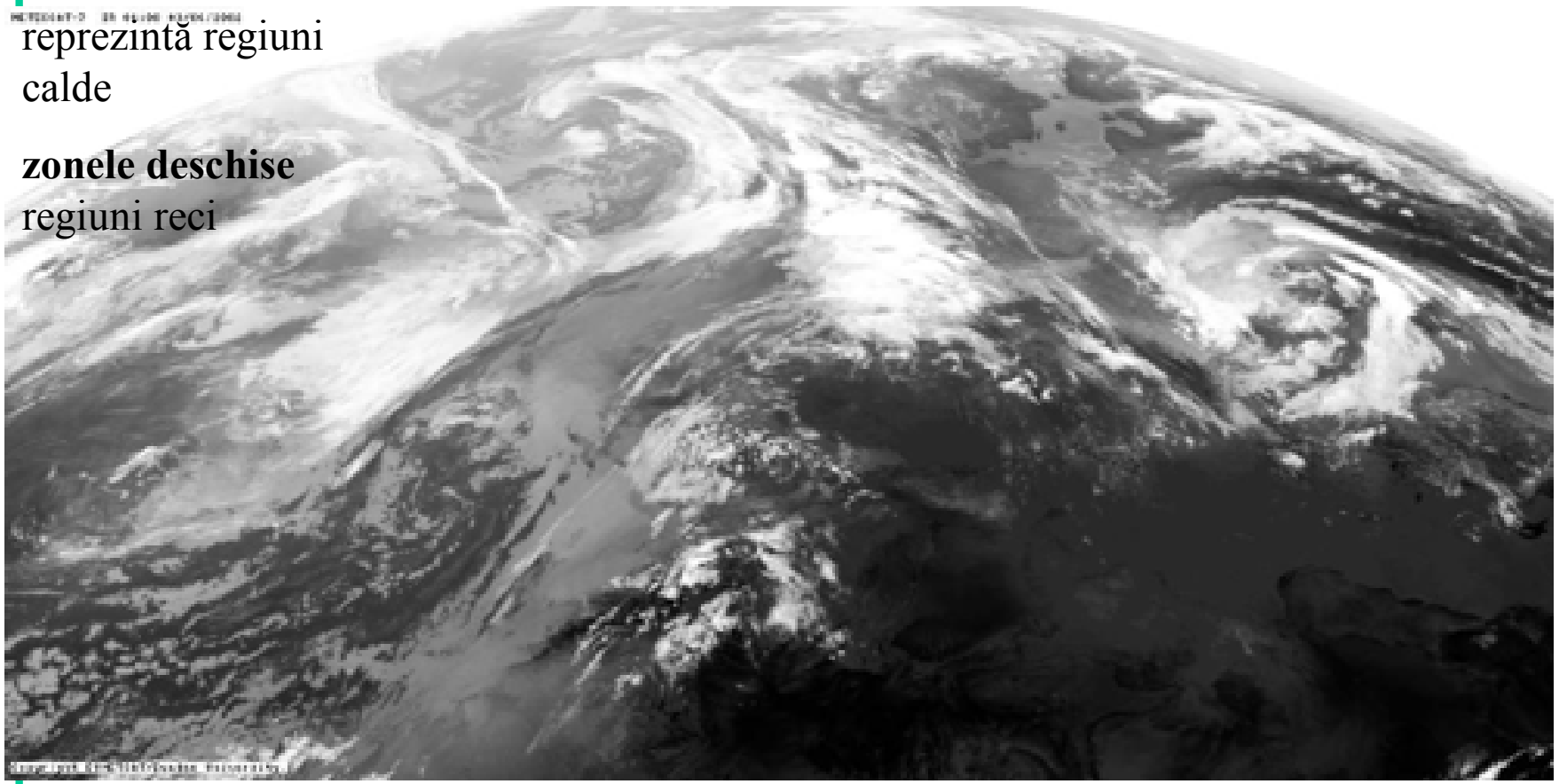
zonele deschise,
regiuni cu
umiditate ridicată



Canalul infra-roşu

zonele închise
reprezintă regiuni calde

zonele deschise
regiuni reci



Clasificarea norilor (de interes în segmentare)

In funcție de altitudine:

- înalți **(6-12 km): cirrus, cirrostratus, cirrocumulus, cumulonimbus**
- medii **(2-6 km): altostratus, altocumulus**
- joşi **(0-2 km): stratus, stratocumulus, cumulus**

a)

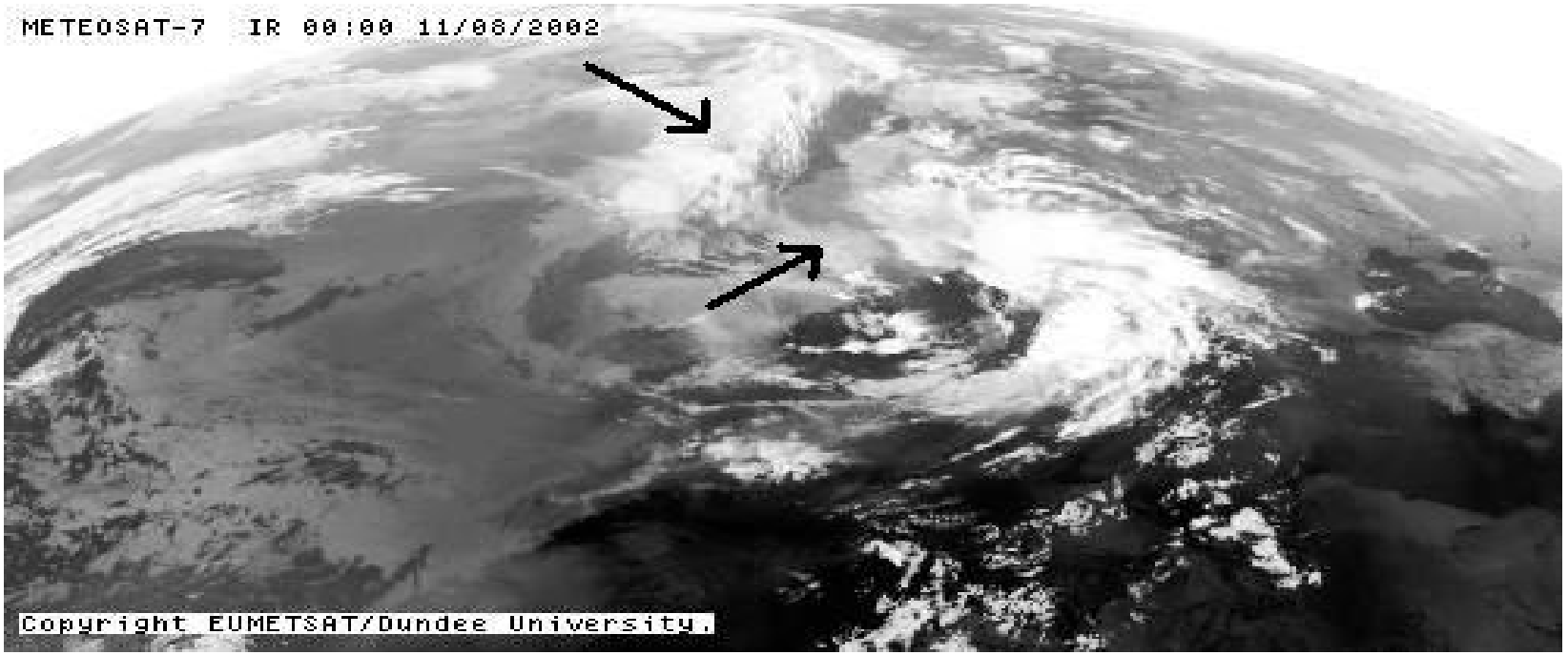
b)

Tipuri de formațiuni noroase: a) nori înalți (cirrus); b) nori medii (altocumulus)

Nori de joasă altitudine (stratus)

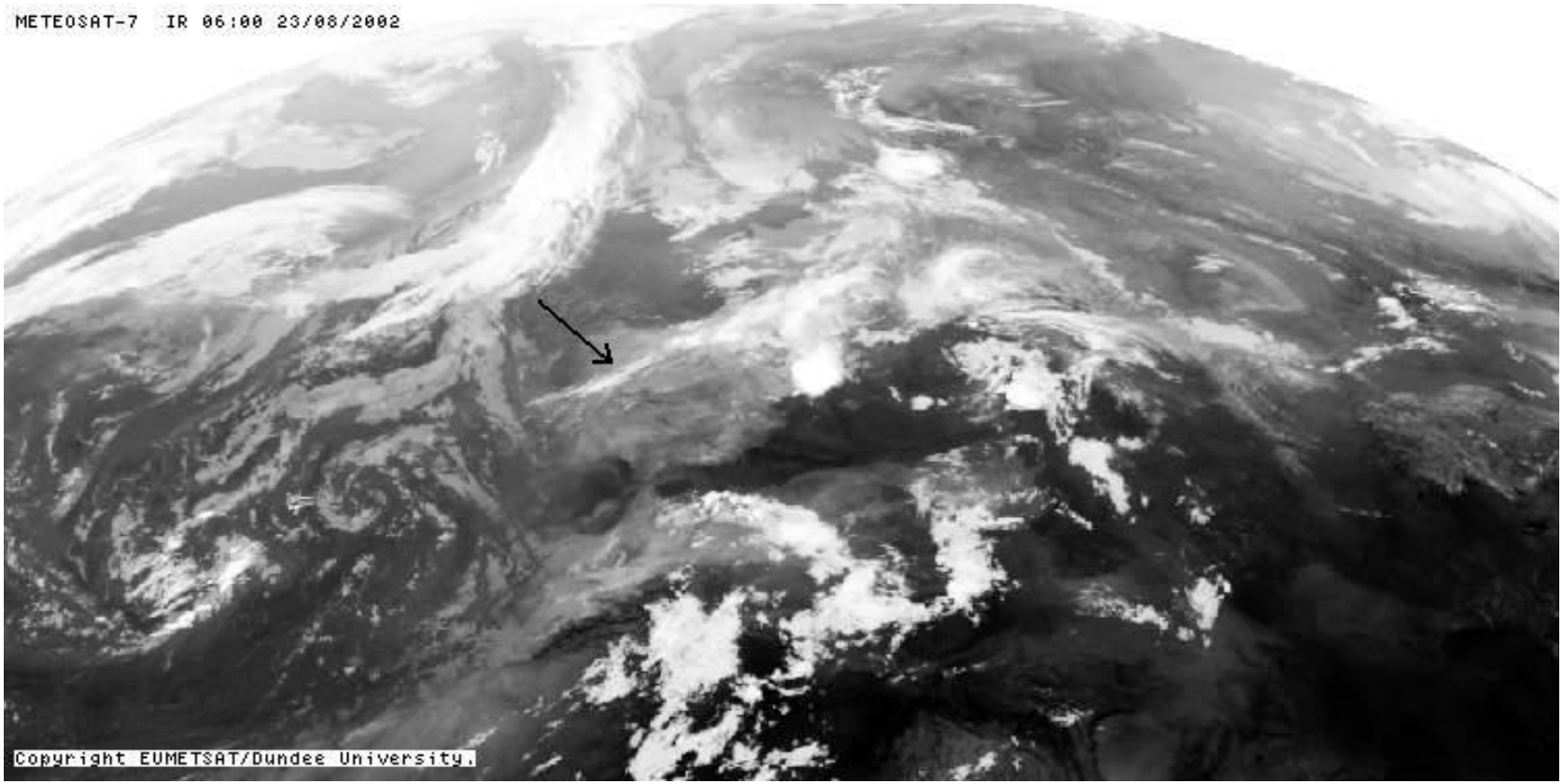
Dificultăți legate de interpretarea imaginilor satelit

1. Multistratificare



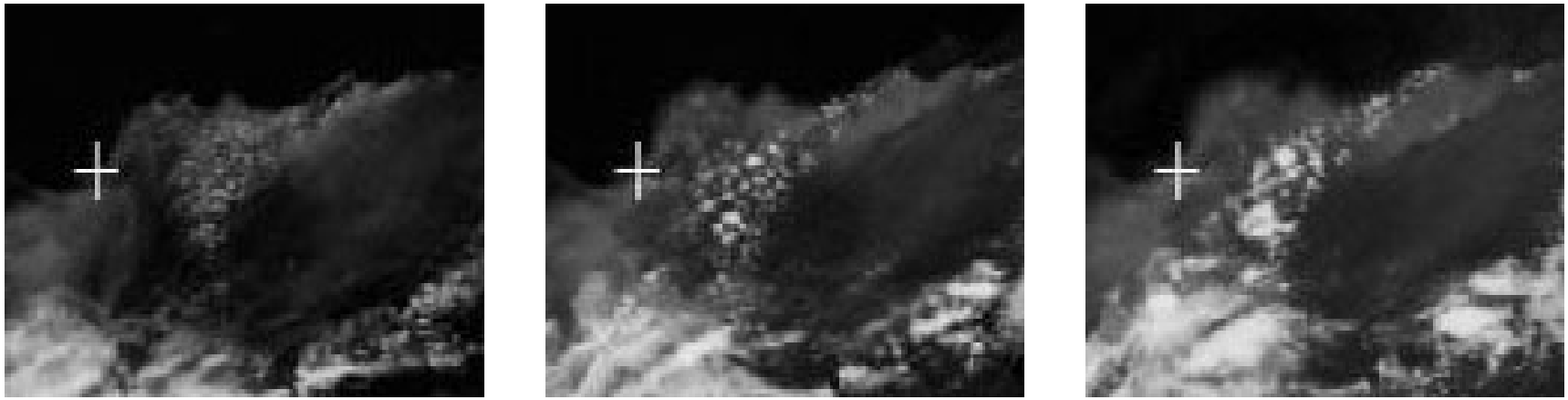
Dificultăți legate de interpretarea imaginilor satelit (cont.)

2. Lipsa de contrast termic



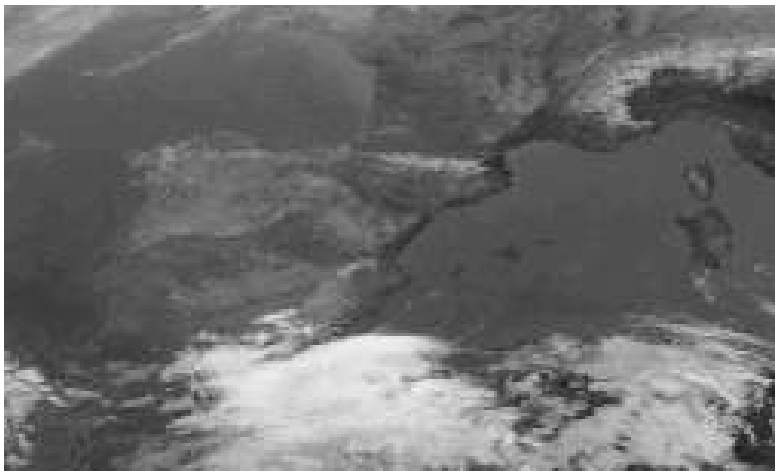
Dificultăți legate de interpretarea imaginilor satelit (cont.)

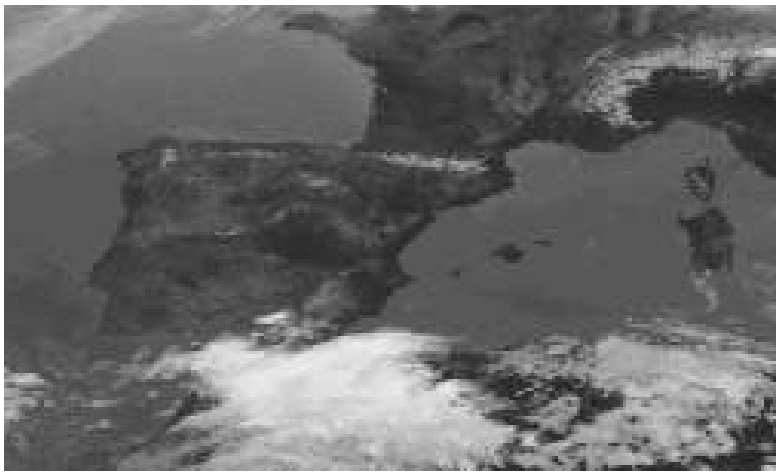
3. Evoluția rapidă a fenomenelor meteo



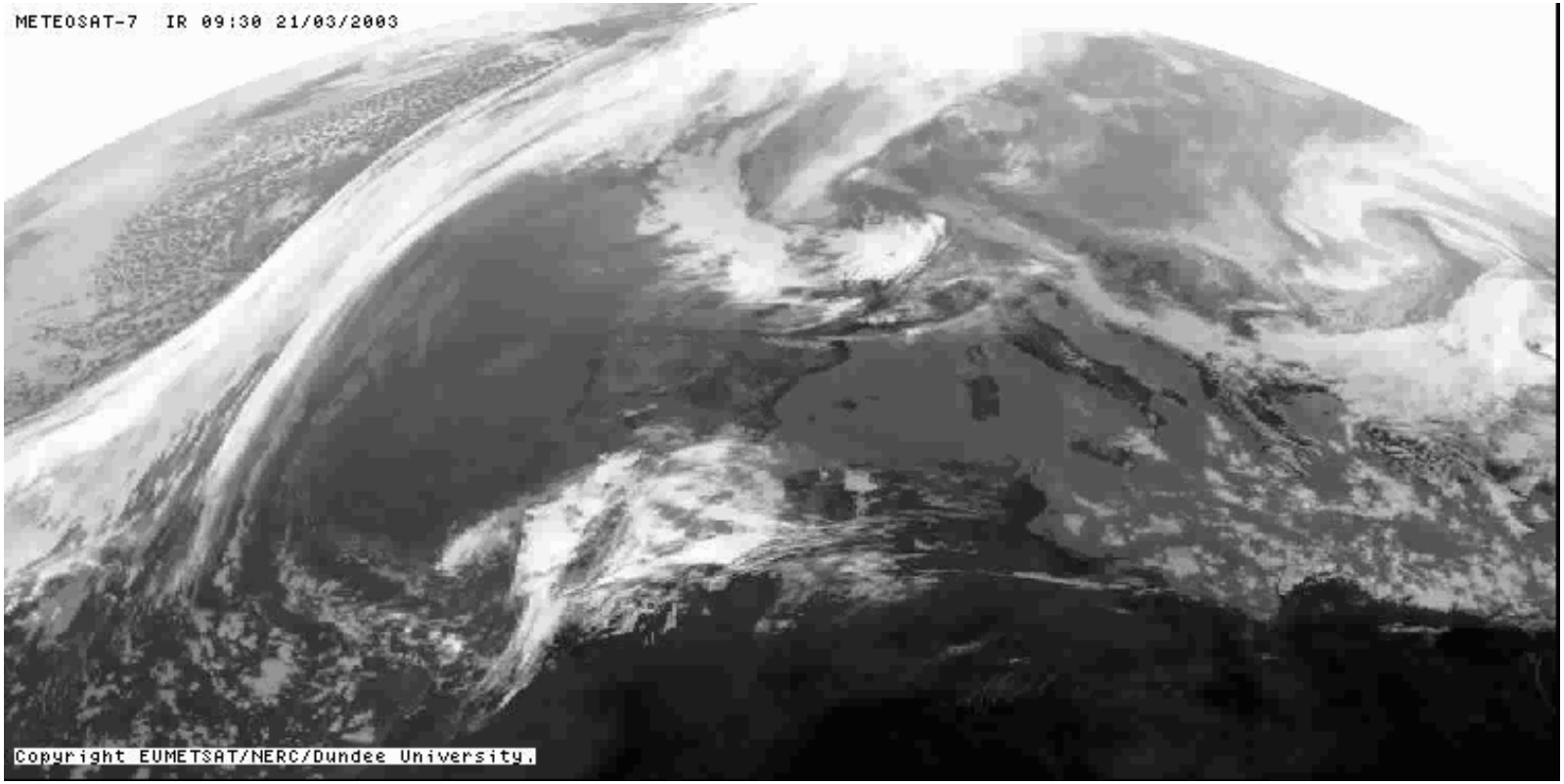
Dificultăți legate de interpretarea imaginilor satelit (cont.)

4. Inversiune a contrastului termic





Secvență de test



Secvență METEOSAT infraroşu

din 21.04.2003 9:30-14:00 GMT

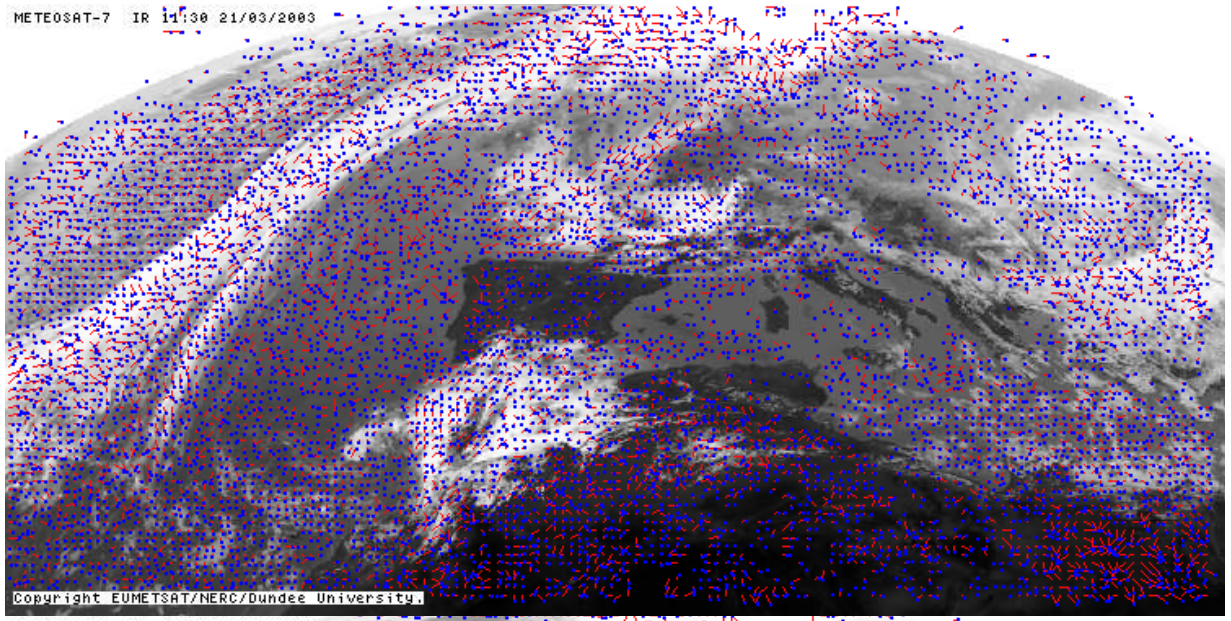
Adaptarea metodei Block Matching la specificul imaginilor satelit (cont.)

Rezultate

Vectorii de mişcare

algoritmul

block-matching FS

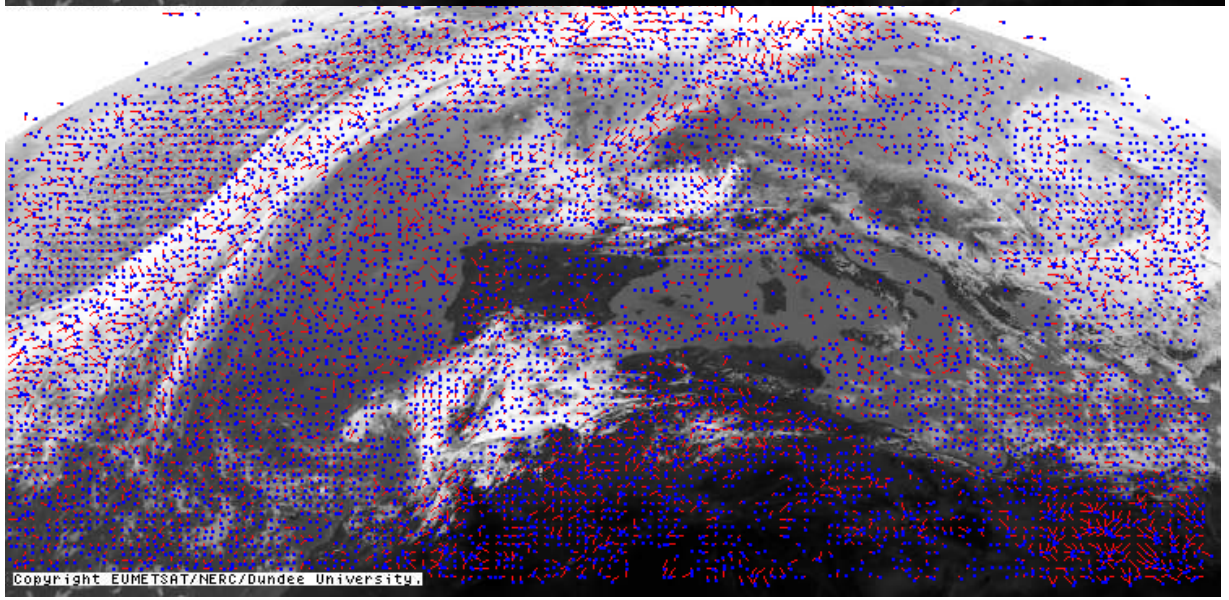


Vectorii de mişcare

metoda celui mai

bun candidat

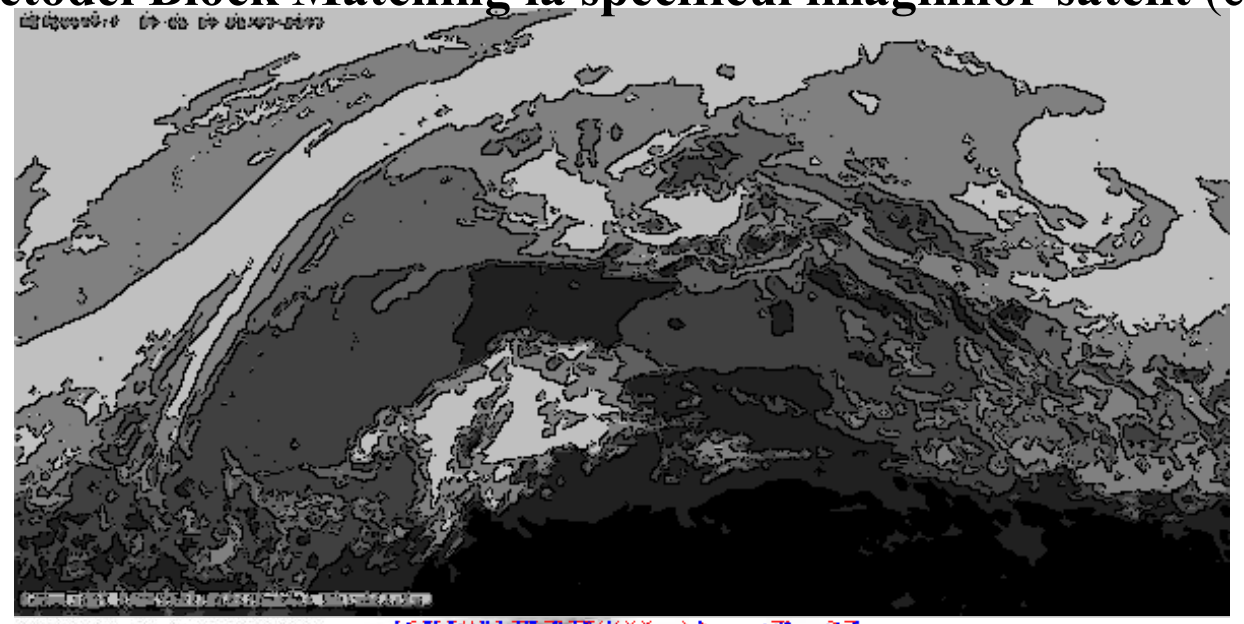
neadaptat



Adaptarea metodei Block Matching la specificul imaginilor satelit (cont.)

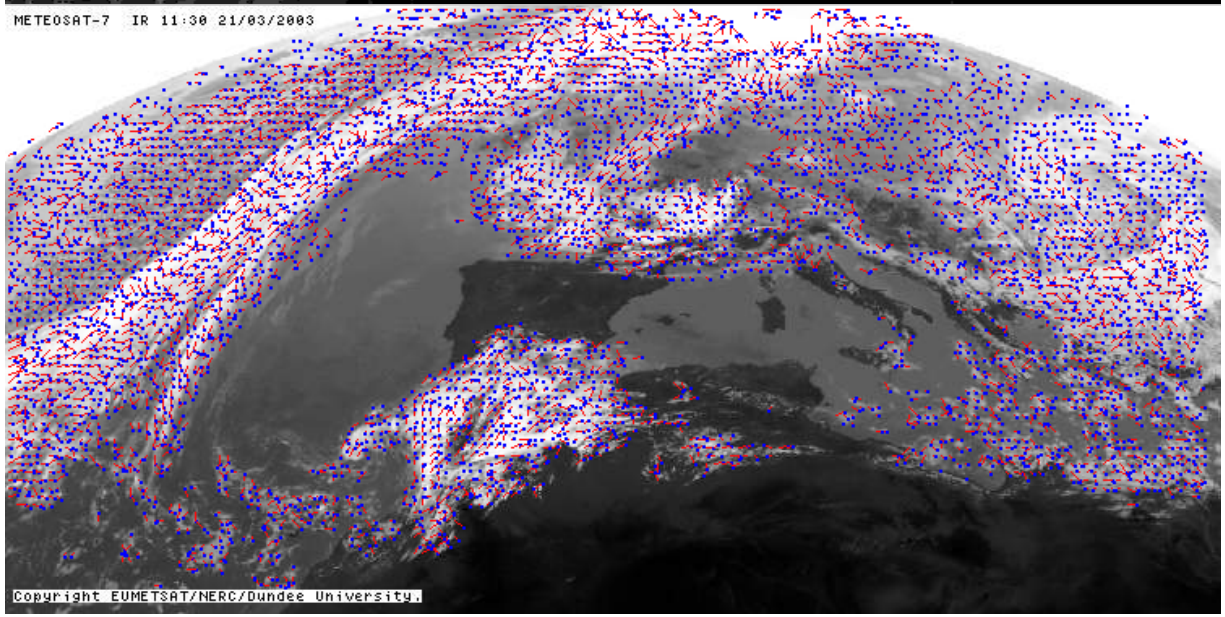
Rezultate

Segmentarea imaginii de la ora 11:00 GMT



Vectorii de mişcare

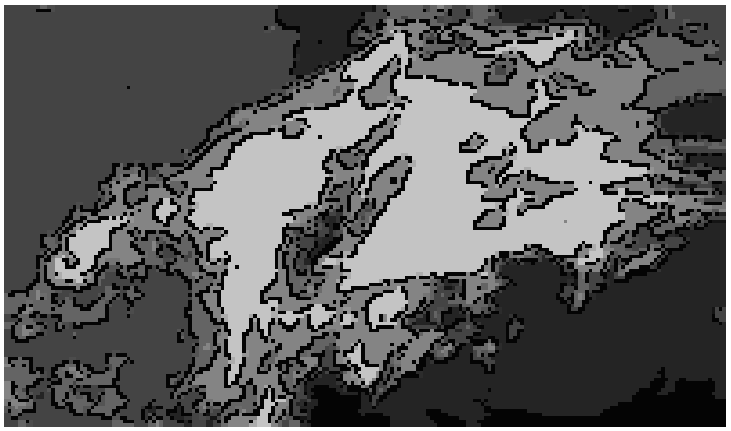
metoda celui mai bun candidat adaptat



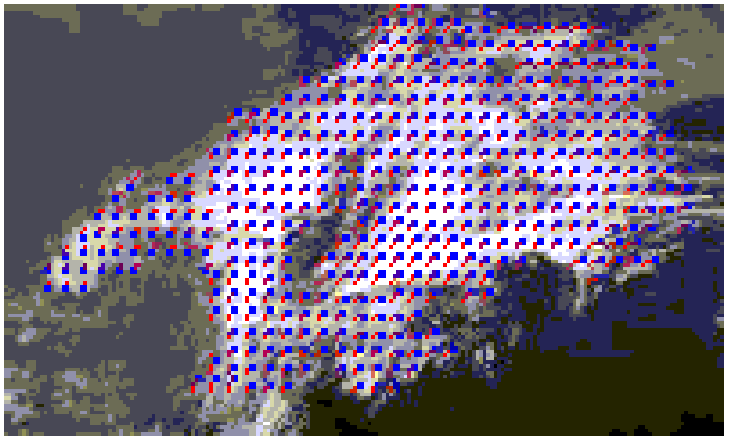
Rezultate (cont.)

detalierea unei porțiuni din secvența anterioară

Secvența METEOSAT



Secvența segmentată



Secvența vectorilor de mişcare

Fluxul optic pentru sectorul C2