



Technical University of Cluj - Napoca  
Computer Science Department

# Interactiune Om-Calculator

## Curs 2

**Procesarea imaginilor color:**

**Modele de culoare. Segmentare imagini color.**



# Achizitia imaginilor color

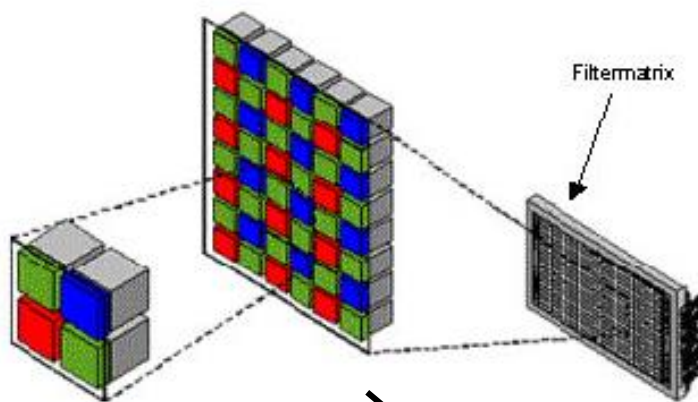
## Senzori color

<http://www.siliconimaging.com/RGB%20Bayer.htm>

<http://www.zeiss.de/c1256b5e0047ff3f/Contents-Frame/c89621c93e2600cac125706800463c66>

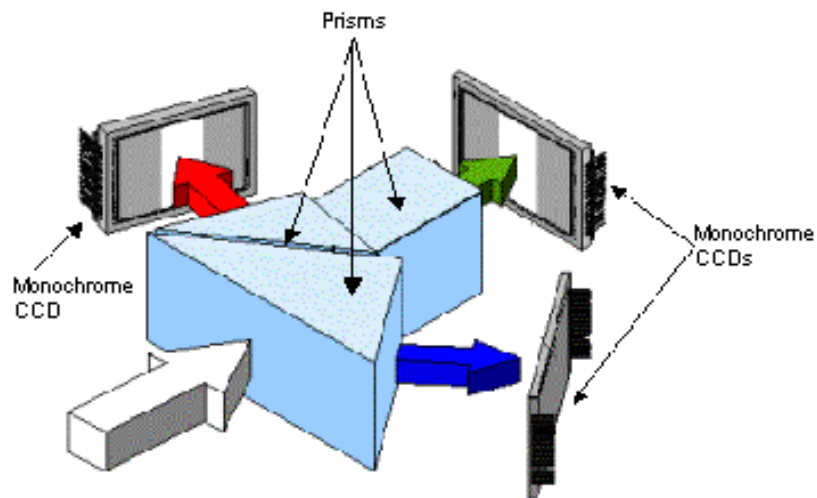
### a) Bayer mask

For color photos, the majority of commercial digital color cameras use pixels covered with special color filters in the three primary colors red, green and blue.



decodificarea  
pattern-ului Bayer

### c) 3-CCD camera



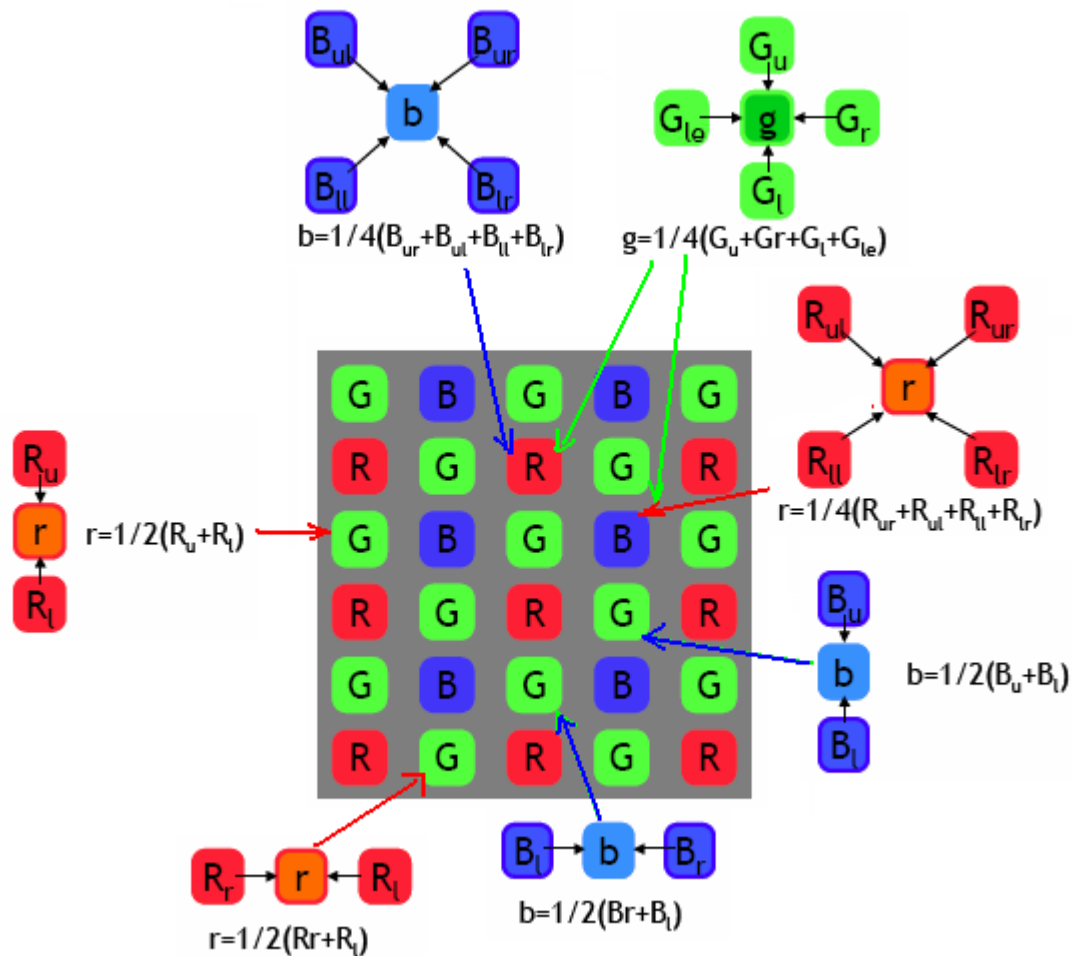
direct

Imagine RGB



# Achizitia imaginilor color

## Decodificarea pattern-ului Bayer



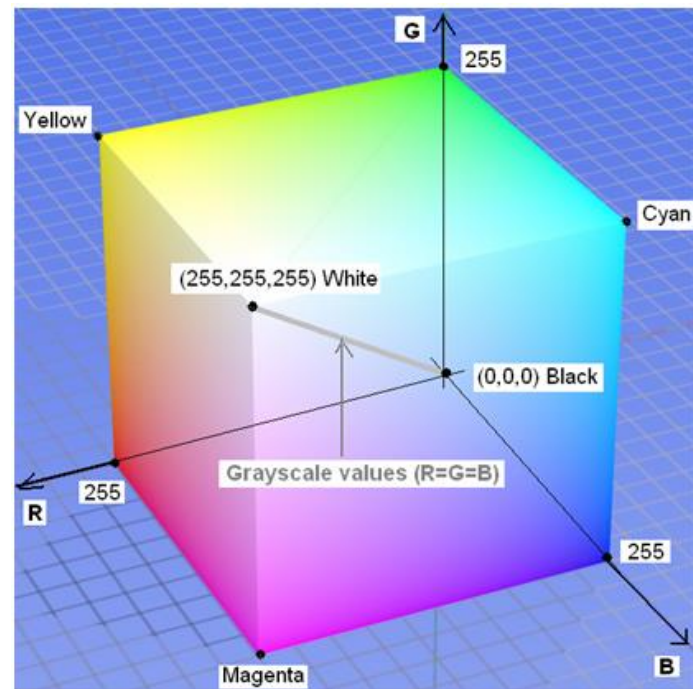
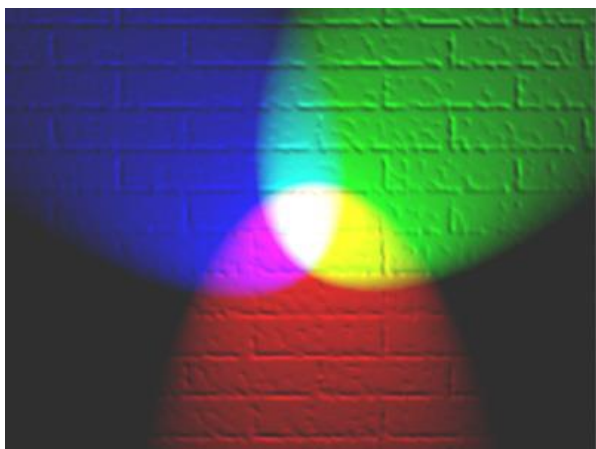
Calitatea imaginii (Bayer pattern vs. 3CCD) ???



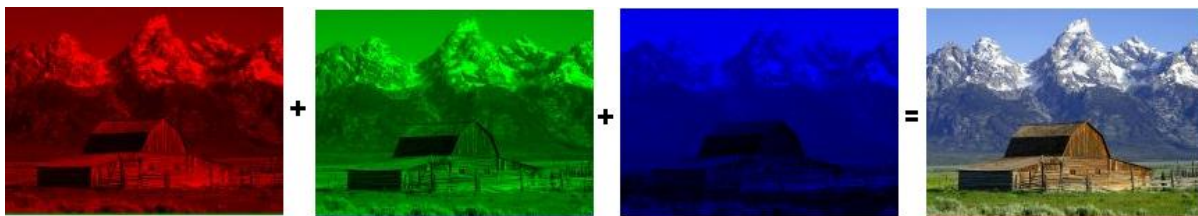
# Spatiul de culoare: RGB

**RGB**  $\Rightarrow$  Culoarea fiecărui pixel (atât pentru echipamentele de achiziție – camere) cât și pentru afișare (TV, CRT, LCD) se obține prin combinația a trei culori primare: roșu, verde și albastru. (Red, Green și Blue)

$\Rightarrow$  spațiu de culoare aditiv ( $R+G+B \Rightarrow$  Alb)



Modelul de culoare RGB mapat pe un cub. În acest exemplu fiecare culoare este reprezentată pe câte 8 biți (256 de nivele) (imagini bitmap RGB24). Numărul total de culori este  $2^8 \times 2^8 \times 2^8 = 2^{24} = 16.777.216$ .





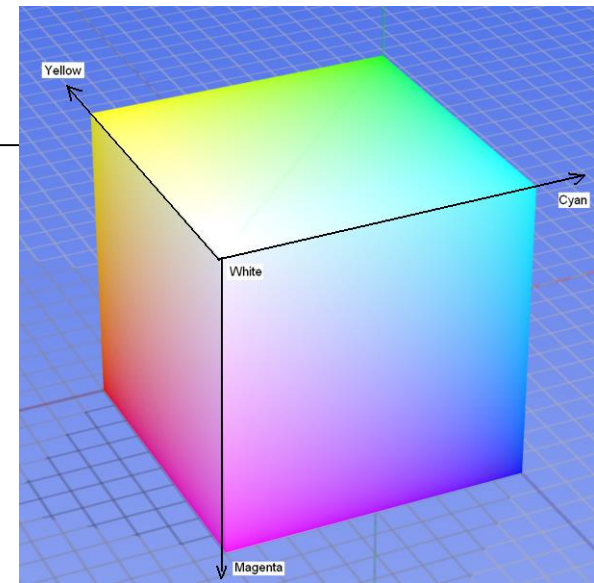
# Spatiul de culoare: CMY

**CMY: spațiu de culoare complementar fata de RGB folosit la dispozitive de imprimare color.**

**CMY model diferential (“substractive”):**

**Alb = absenta componentelor de culoare**

**Negru = C + M + Y**



## CMYK





# Spatiul de culoare: RGB normalizat

---

Reduce dependenta de iluminare a culorii obiectului

Se poate aplica doar daca variatiile de intensitate sunt uniforme de-a lungul spectrului RGB

$$r = \frac{R}{R + G + B}, \quad g = \frac{G}{R + G + B}, \quad b = \frac{B}{R + G + B}$$

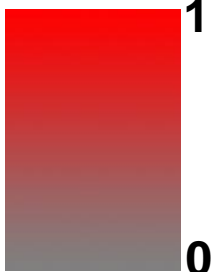
$$r + g + b = 1$$



# Modele de culoare: HSI (HSV, HSB, HSL)

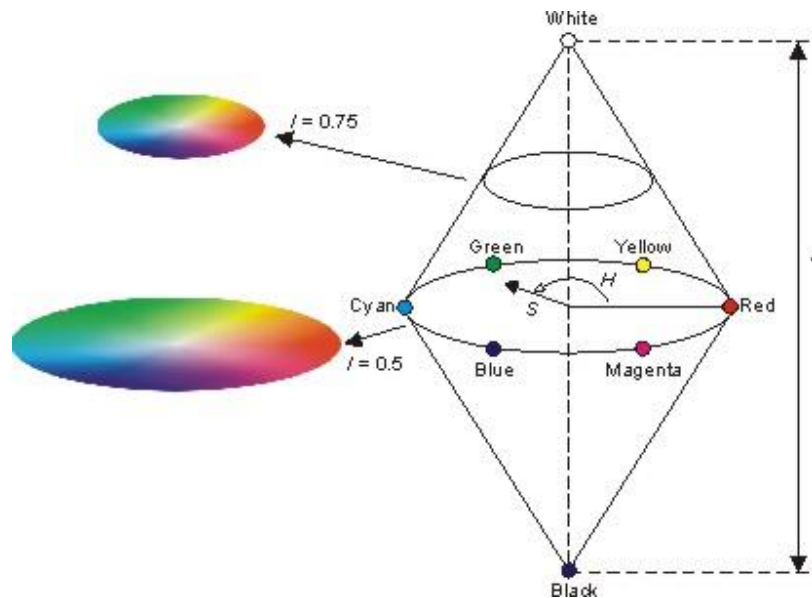
HSI: (H, S, I),  $H=0 \dots 2\pi$ ,  $S=0 \dots 1$ ,  $I=0 \dots 255$

$$I = \frac{1}{3}(R + G + B)$$

$$S = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)]$$


$$H = \begin{cases} \Phi & \text{if } G \geq B \\ 2\pi - \Phi & \text{if } G < B \end{cases}$$

$$\Phi = \cos^{-1} \left( \frac{\frac{1}{2} [(R - G) + (R - B)]}{\left[ (R - G)^2 + (R - B)(G - B) \right]^{\frac{1}{2}}} \right)$$





# Proprietati ale trasaturilor cromatice

## Invarianta la translatie si scalare

**Hue** – invarianta la scalarea uniforma a RGB:  $H(\alpha R, \alpha G, \alpha B) = H(R, G, B)$

**RGB-norm** – invarianta la scalarea uniforma RGB:

$$\begin{aligned}r(\alpha R, \alpha G, \alpha B) &= r(R, G, B) \\g(\alpha R, \alpha G, \alpha B) &= g(R, G, B) \\b(\alpha R, \alpha G, \alpha B) &= b(R, G, B)\end{aligned}$$

**Hue** – invarianta la translatia uniforma RGB:  $H(R + \beta, G + \beta, B + \beta) = H(R, G, B)$

**RGB-norm** – **nu prezinta invarianta** la translatia uniforma in spatial

RGB:

$$\begin{aligned}r(R + \beta, G + \beta, B + \beta) &\neq r(R, G, B) \\g(R + \beta, G + \beta, B + \beta) &\neq g(R, G, B) \\b(R + \beta, G + \beta, B + \beta) &\neq b(R, G, B)\end{aligned}$$





# Proprietati ale trasaturilor cromatice

---

## Singularitate Hue pt. RGB $\approx 0$

$R=G=B=0 \Rightarrow H$  nedefinit

Exemple:

$$H(1,0,0) = 0, H(0,1,0) = 2\pi/3$$

$$H(1,1,0) \Rightarrow H(2,1,0) : \delta H = \pi/6$$

Concluzie: calcularea H in zone cu intensitatea mica  $\Rightarrow$  erori numerice



# Alte modele (liniare)

**XYZ tristimulus** - transformare liniara asupra RGB:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\text{sat} = \sqrt{a^2 + b^2},$$

$$\text{hue} = \arctan'(b, a)$$

## CIE(Lab) space

$$L = 25(100Y/Y_0)^{1/3} - 16,$$

$$a = 500 \left[ (X/X_0)^{1/3} - (Y/Y_0)^{1/3} \right]$$

$$b = 200 \left[ (Y/Y_0)^{1/3} - (Z/Z_0)^{1/3} \right]$$

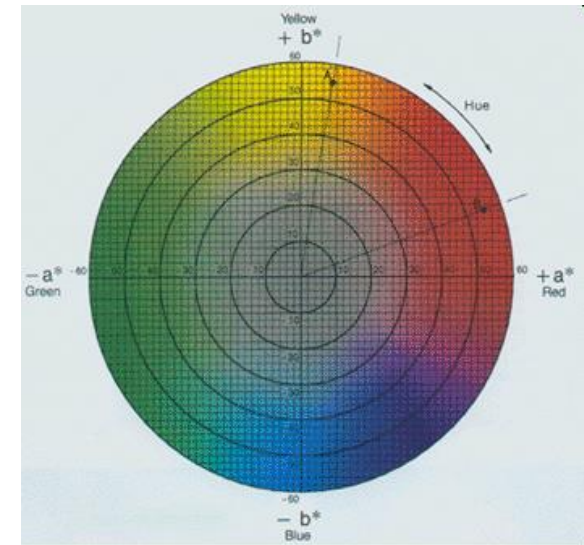
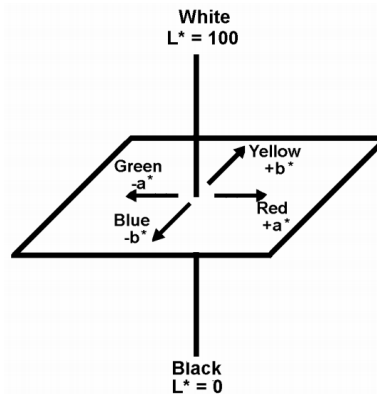
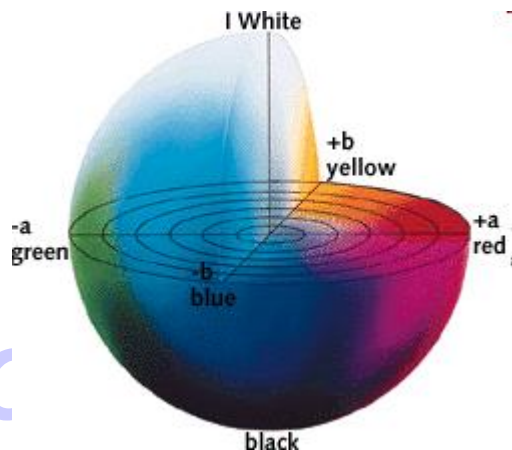
L – componenta de intensitate

a, b - componentele de culoare cu variatie liniara

## CIE(Luv) space

$$u = 13W(4X/(X + 15Y + 3Z) - 0.199)$$

$$v = 13W(6Y/(X + 15Y + 3Z) - 0.308)$$

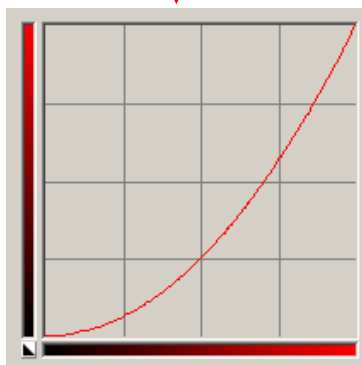
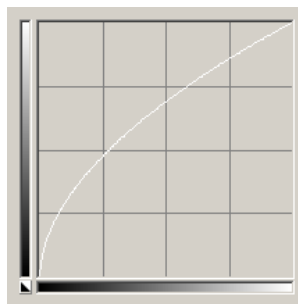




# Procesari pe imagini color

Similare cu cele pe imagini grayscale

Procesarile se aplica pe fiecare componenta de culoare in parte.





# Metoda Canny (imagini color)

---

[2] A. Koschan, M. Abidi, Digital Color Image Processing, Wiley & Sons, 2008.

## Algoritm

1. Filtrare zgomot cu un filtru trece jos ([2], cap. 5.3, pp102-117)

**2. Calculul modulului si directiei gradientului** ([2], cap 6.1.1, pag 126-128)

3. Supresia non-maximelor

4. Trhresholding cu histereza



# Metoda Canny (imagini color)

Pas2 :

Pixel (x,y)  $\Rightarrow$  culoarea :  $C(x,y)=(R,G,B)$

Gradient  $\Rightarrow$  **Jacobian** (matricea derivatelor partiale ale vectorului C):

$$\mathbf{J} = \begin{pmatrix} R_x & R_y \\ G_x & G_y \\ B_x & B_y \end{pmatrix} = (\mathbf{C}_x, \mathbf{C}_y).$$

$$R_x = \frac{\partial R}{\partial x} \quad \text{and} \quad R_y = \frac{\partial R}{\partial y}$$



# Metoda Canny (imagini color)

**Directia** - vectorul propriu al  $J^T J$  corespunzator celei mai mici valori proprii:

$$\tan(2\theta) = \frac{2 \cdot \mathbf{C}_x \cdot \mathbf{C}_y}{\|\mathbf{C}_x\|^2 - \|\mathbf{C}_y\|^2}$$

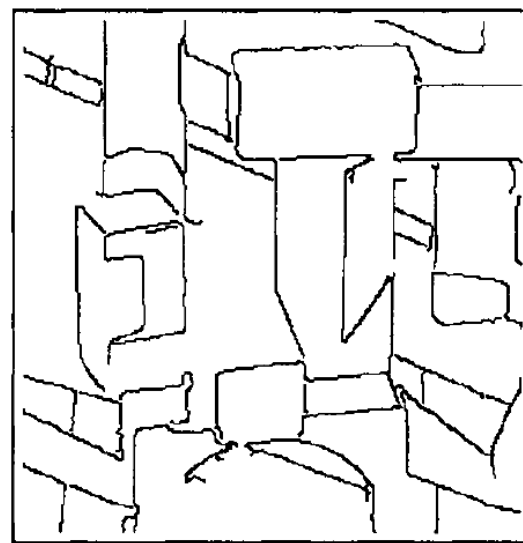
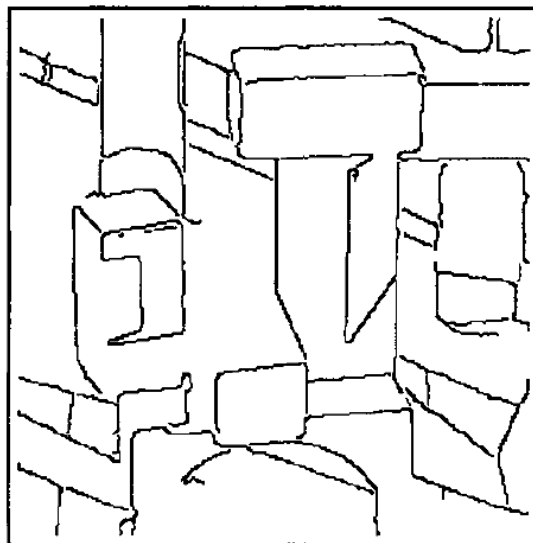
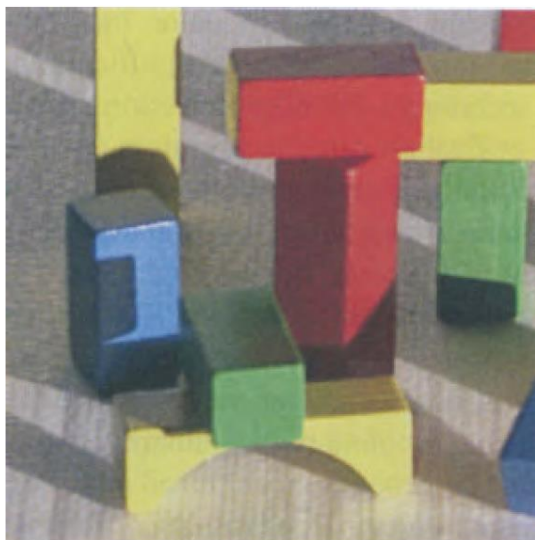
$$\mathbf{C}_x = (R_x, G_x, B_x)$$

**Magnitudinea:**

$$m^2 = \|\mathbf{C}_x\|^2 \cos^2(\theta) + 2 \cdot \mathbf{C}_x \cdot \mathbf{C}_y \cdot \sin(\theta) \cos(\theta) + \|\mathbf{C}_y\|^2 \sin^2(\theta).$$



# Rezultate Canny [2]



Color

Technical University of Cluj Napoca  
Computer Science Department  
**Grayscale**



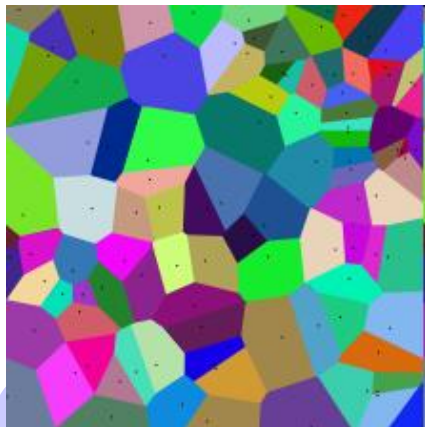
# Segmentarea imaginilor color

**Segmentare** := identificare zone omogene din imagine

**Segmentare imaginii color** := identificare regiuni (componente conexe) care satisfac anumite criterii de omogenitate, bazate pe trasaturi derivate din componentele spectrale. Aceste componente sunt definite in spatiul de culoare considerat.

**(1) Regiune (definitie bazata culoarea pixelului)** := componenta conexa a unui set de pixeli specificata printr-o functie de aparteneta la o clasa definita in spatiul de culoare considerat:

- (a) Culoare pixelului este intr-un semispatiu definit de un plan;
- (b) Culoare pixelului se incadreaza intr-un poliedru;
- (c) Culoare pixelului se incadreaza intr-o celula Voronoi data de niste puncte reprezentative;



**Decompozitie (spatiu) Voronoi** := In cazul cel mai simplu (2D) se dau un set de puncte  $S$  in plan (centre Voronoi). Fiecare centru  $s$  are asociata o celula Voronoi  $V(s)$  continand toate punctele mai apropiate de  $s$  decat de toate celelalte centre.

Segmentele diagramei Voronoi sunt multimi de puncte (segmente de dreapta) care sunt egal departate de 2 centre (cele mai apropiate)

Nodurile Voronoi sunt puncte echidistante fata de 3 sau mai multe centre Voronoi





# Segmentarea imaginilor color

---

**(2) Regiune** (definitie bazata pe pozitia pixelului) := setul maximal de pixeli pentru care este satisfacuta o conditie de uniformitate (predicat de omogenitate):

(a) Regiuni uniforme obtinute prin cersterea unui bloc/seed prin unirea altor pixeli sau blocuri de pixeli

(b) Regiuni uniforme obtinute prin impartirea unor regiuni mai mari care nu sunt omogene

**(3) Regiune** (definitie bazata pe notiunea de muchie) := set de pixeli delimitati de pixeli de muchie (countur) - (predicat de ne-omogenitate):

**(4) Regiune** := corespunde unei suprafete a unui obiect din material omogen (physics based vision methods – modele de reflexie bazate pe proprietatile materialelor)



# 1. Segmentare la nivel de pixel

---

**Segmentare se face in spatiul trasaturilor (culorilor) – prin clasificarea culorii pixelilor la un set de clase !!!**

**Abordari:**

1. Bazate pe histograma: detectia maximelor si gruparea culorilor (clustering) in jurul maximelor + clasificarea pixelilor in aceste clustere
2. Clusering in spatiul de culoare – punctele din spatiul de culoare sunt grupate in jurul unor centre reprezentative + clasificarea pixelilor in aceste clustere
3. Clusterring fuzzy – functi de aparteneta fuzzy evaluate pentru totii pixelii si clusterele fuzzy + obtinerea clusterilor de pixeli prin defuzzificare si diviziune in seturi connexe maximale



# 1.1. Segmentare bazata pe impartirea histogramei

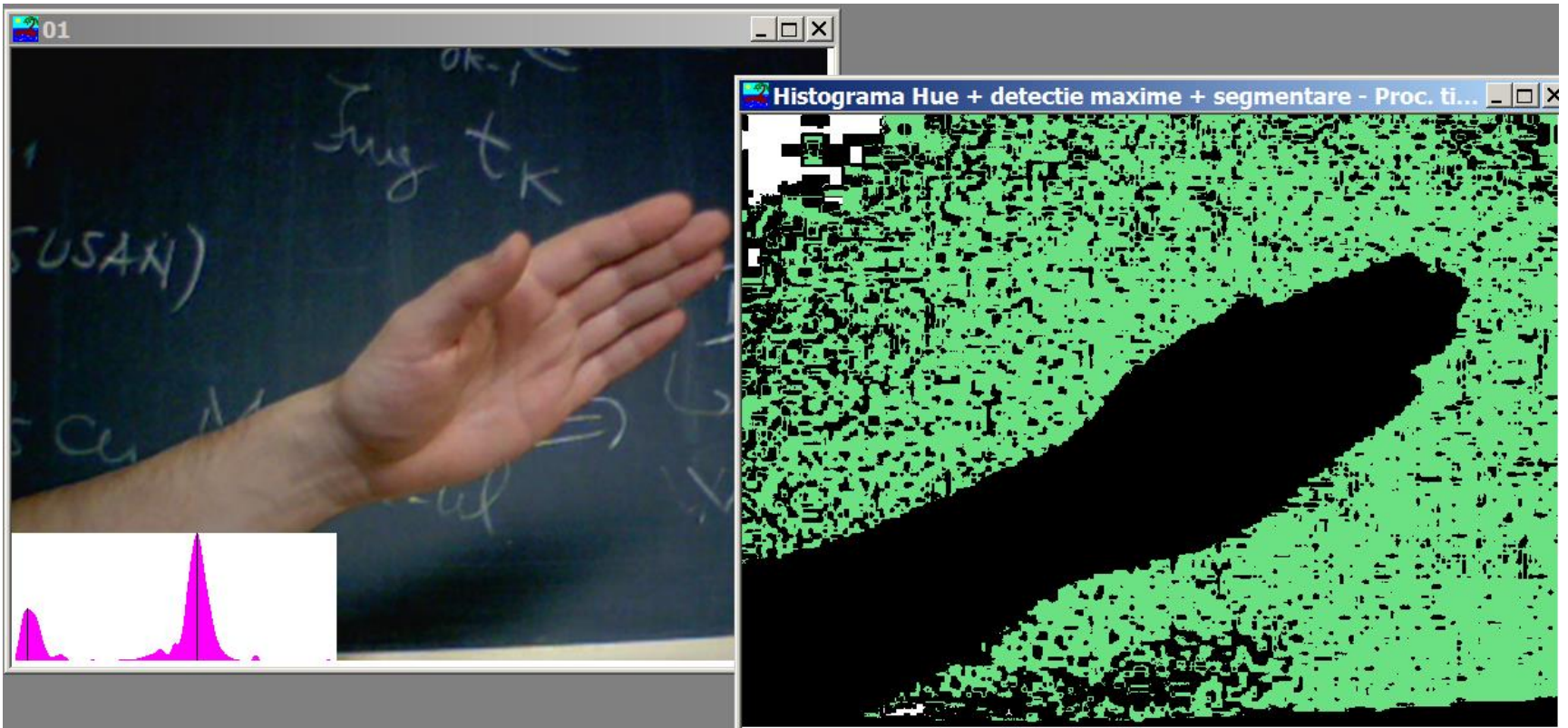
Detectia maximelor histogramei Hue (filtrate) si impartirea spatiului Hue in Clusteri avand ca centre aceste varfuri + clasificare pixeli

The image displays a medical ultrasound software interface with a color-coded elasticity map. The map is overlaid on a grayscale B-mode image. A color scale on the right indicates stiffness, ranging from 'Soft' (red) to 'Hard' (blue). The map is divided into four clusters, labeled 1, 2, 3, and 4. A 'Filtered Hue Histogram' window is open, showing a multi-modal distribution of hue values. The histogram has four distinct peaks corresponding to the four clusters. The software interface includes a menu bar (File, View, Scale, Processing, ROI, Measurements, Convert, Image, Geometry, Filters, Movie, Window, Help) and a toolbar with various tools. The status bar at the bottom shows technical parameters: BG:15, L54M, 70I-/2/5/1/A/6, 26%, 2I-/1-/2/3/4 T-Elasto, 50mm, BG:15, L54M, 10M, Superficial, 50mm, and a patient ID 'No.14/84'. The coordinates at the bottom are x:759, y:458; R:13, G:13, B:25, A:17, H:240.



## 1.1. Segmentare bazata pe impartirea histogramei

Detectia maximelor histogramei Hue (filtrate) si impartirea spatiului Hue in Clusteri avand ca centre aceste varfuri  $\cong$  clasificare pixeli (fiecare pixel se clasifica la culoarea vf. celui mai apropiat) – vezi L3 / PI





## 1.1. Segmentare bazata pe impartirea histogramei

Detectia maximelor histogramei Hue (filtrate) si impartirea spatiului Hue in Clusteri avand ca centre aceste varfuri  $\cong$  clasificare pixeli (fiecare pixel se clasifica la culoarea vf. celui mai apropiat) – vezi L3 / PI





## 1.2 Clustering in spatiul de culoare

Gruparea culorilor (din spatul culorilor) si asignarea ficarui grup a unei culori echivalente (medie)

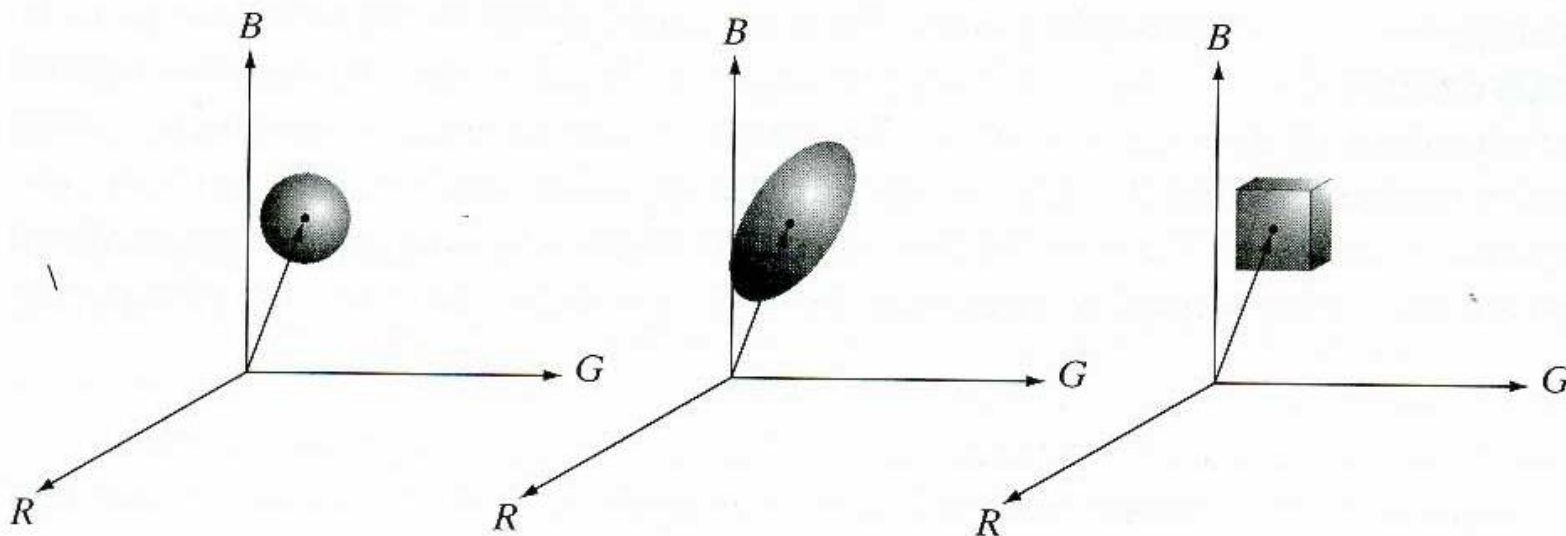
- **cuantizare/posterizare** (impartirea spatiului trasaturilor in subspatii de dimensiuni fixe);

- **clustering**:

- supervizat - info. apriori: se stiu nr. clusterilor / pozitiile centrelor (de exemplu farfurile histogramelor)

- nesupervizat – fara informatii apriori

Nu se aplica de obicei pe RGB ci pe **HS** sau pe de preferat pe **ab** sau **uv** (*liniar*)

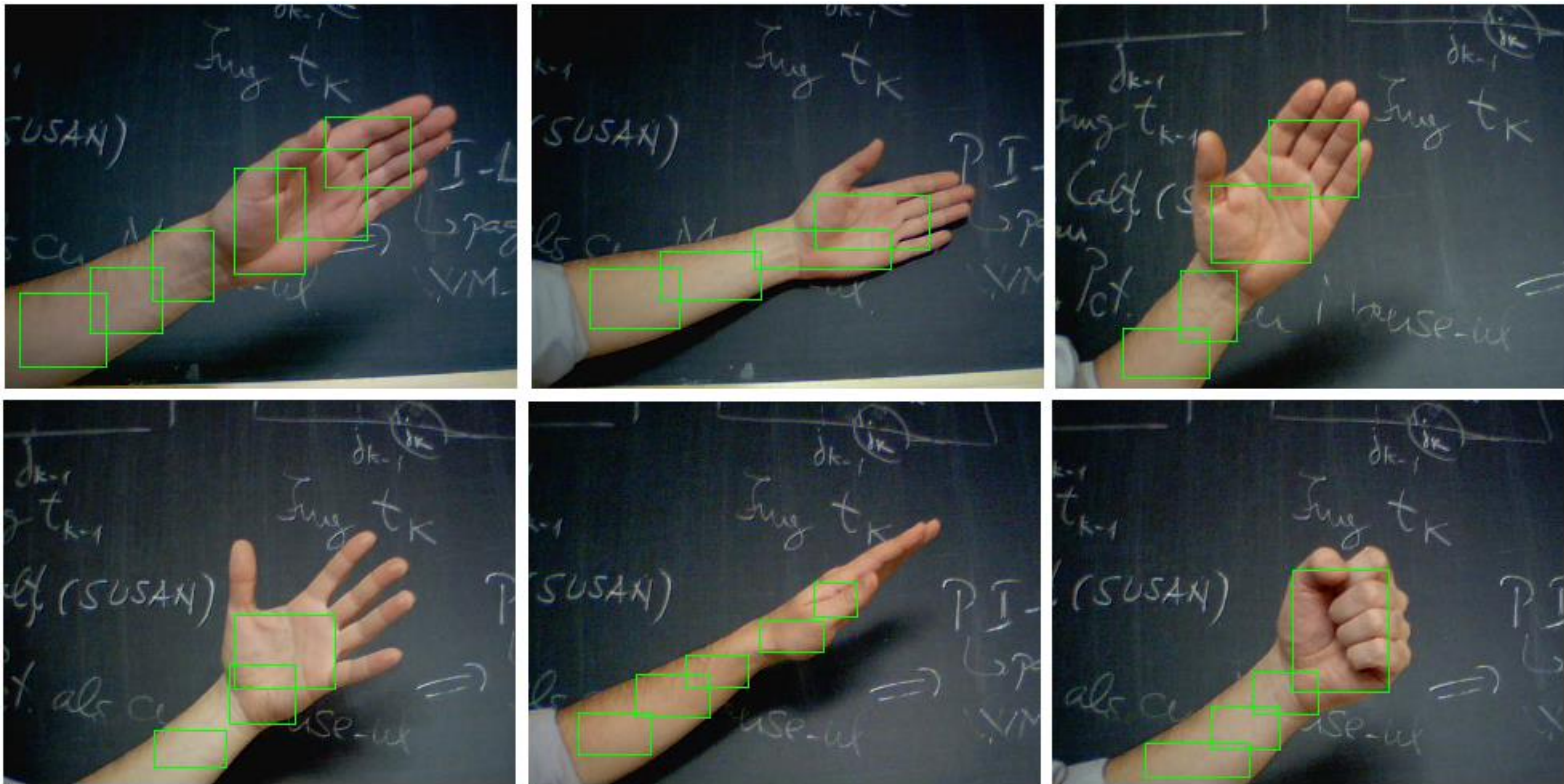




# Clustering folosind un model de culoare (1D)

**Construirea modelului de culoare al mainii** ( $\Rightarrow$  metoda spurvizata)

$\Rightarrow$  histograma globala (o distributie de probabilitate de forma cvasi-gaussiana) obtinuta prin acumularea histogramelor locale ale culorii (ex. Hue) de pe un set de esantioane (samples)

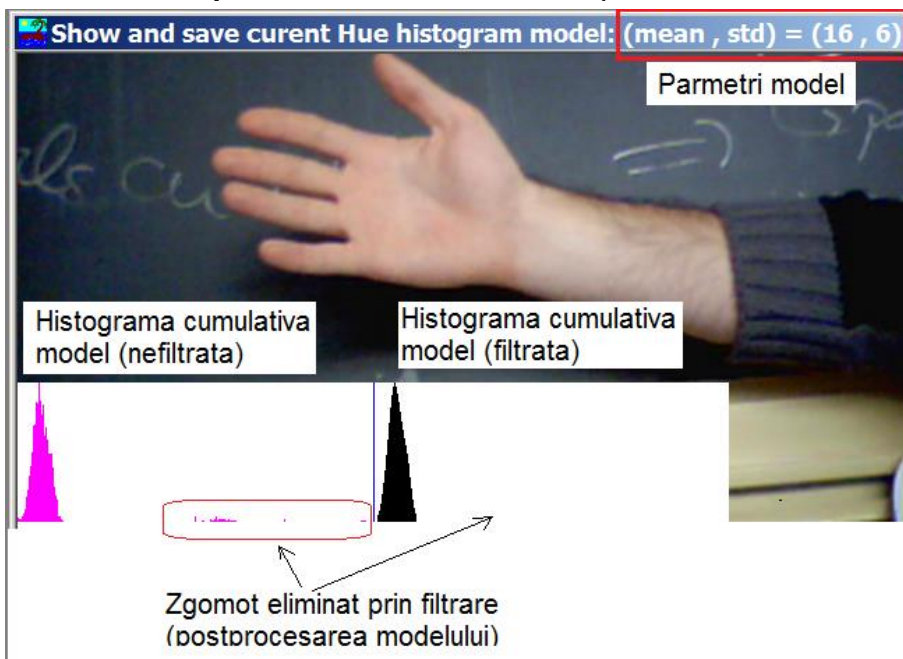




# Clustering folosind un model de culoare (1D)

## Postprocesarea modelului de culoare al mainii (optional )

- se pot filtra elementele histogramei cumulative (modelului) cu un filtru trece jos (FTJ) de tip medie aritmetica sau gaussian.
- se pot filtra valorile din histograma model care sunt mai mici decat x% (x% = 1 ..10 %) din valoarea maxima a histogramei (pt. eliminarea eventualelor „zgomote” datorate selectiei imprecise – daca se selecteaza si zone de fond din afara perimetrului mainii).



## Calcularea parametrilor modelului de culoare al mainii

$$\bar{g} = \mu = \int_{-\infty}^{+\infty} g \cdot p(g) dg = \sum_{g=0}^L g \cdot p(g) = \frac{1}{M} \sum_{g=0}^L g \cdot h(g)$$

$$\sigma = \sqrt{\sum_{g=0}^L (g - \mu)^2 \cdot p(g)}$$

Pt. setul de antrenare folosit:

hue\_mean = 16

hue\_std = 6





# Clustering folosind un model de culoare (1D)

## Segmentarea imaginii

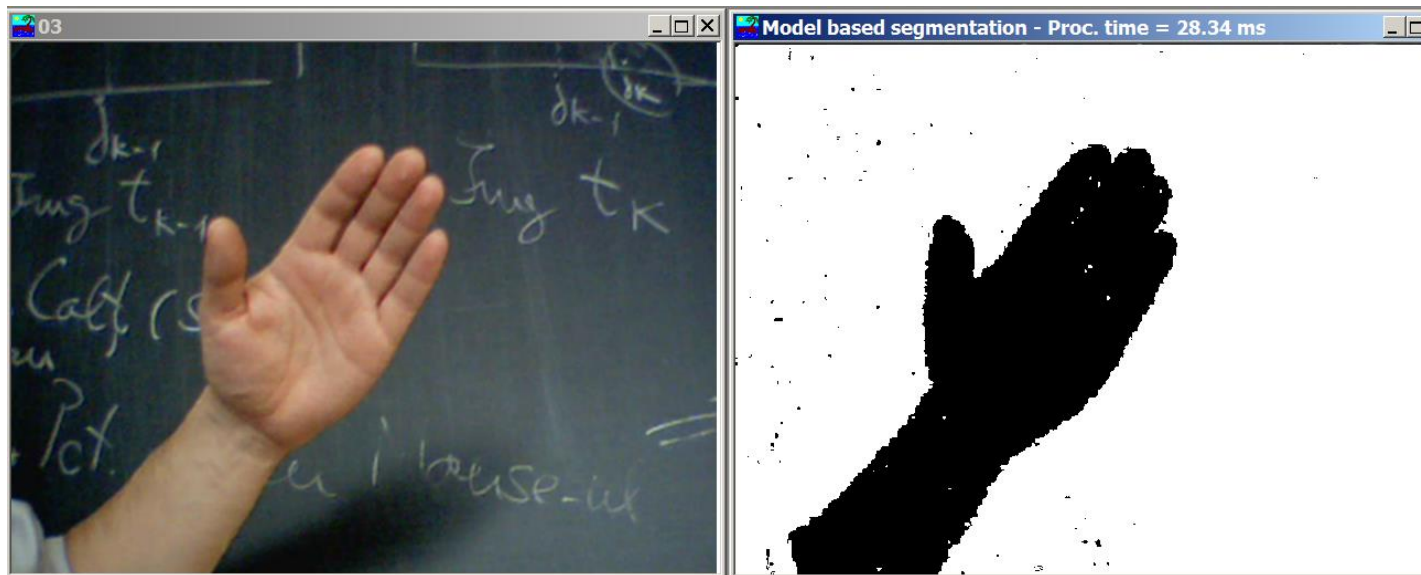
Aceasta etapa consta in clasificarea pixelilor din imagine pe baza valorii curente a componentei de culoare (ex. Hue):

```
if hue(I,j)  $\in$  [hue_mean-k*hue_std .. hue_mean+k*hue_std]
```

```
    label(I,j) = OBIECT
```

```
else
```

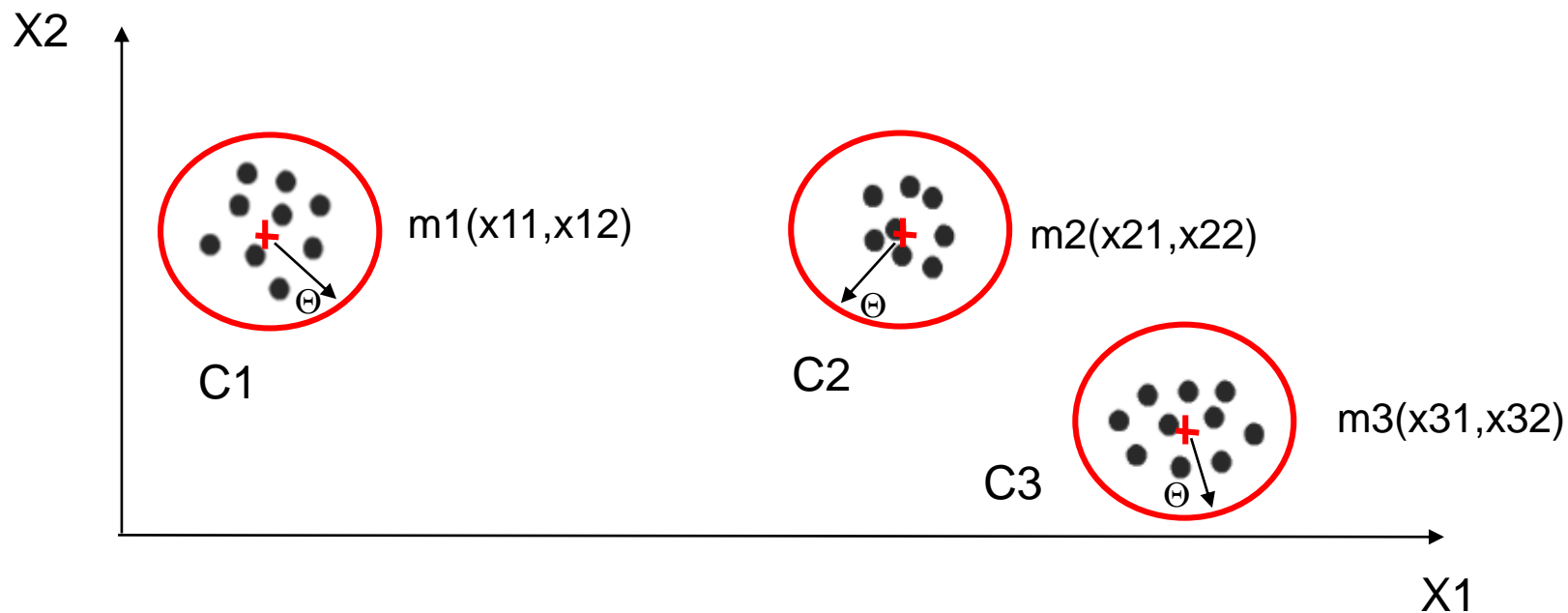
```
    label(I,j) = FOND
```



**Postprocesarea imaginii segmentate – op. morfologice etc.**



# Clustering in spatiu de culoare 2D



$X1$ ,  $X2$  – pot fi cele doua axe de coordonate corespunzatoare componentelor de culoare: (H,S) sau (a,b) sau (u,v)



# Clustering supervizat

## K-means (se poate aplica intr-un spatiu nD)

Partitioneaza un set de  $n$  observatii (ex. culorile pixelilor in spatiul considerat) in  $k$  cluster, in care fiecare observatie va apartine de clusterul cu cel mai apropiat centru (medie).

Setul de observatii:  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ ,  $\mathbf{x}_i$  – vector  $d$ -dimensional (ex:  $\mathbf{x}_i = (a_i, b_i)$ )

Scop  $\Rightarrow k$  multimi ( $k \leq n$ )  $\mathbf{S} = \{S_1, S_2, \dots, S_k\}$  astfel incat sa minimizam suma patratelor distantelor de la fiecare punct din cluster la centrul (media) clusterului

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

$\boldsymbol{\mu}_i$  – centroidul (media clasei  $S_i$ )



# K-means clustering

## Algoritmul standard (Lloyd)

**Pas 1: Initializare:** se da un set initial de centroide (medii/means):

$$m_1(1), \dots, m_k(1)$$

**Pas 2: “atribuire”** – se atribuie fiecare observatie la clusterul cu centroidul (media) cea mai apropiata (se partitioneaza observatiile dupa diagrama Voronoi data de medii).

$$S_i^{(t)} = \left\{ \mathbf{x}_j : \|\mathbf{x}_j - \mathbf{m}_i^{(t)}\| \leq \|\mathbf{x}_j - \mathbf{m}_{i^*}^{(t)}\| \text{ for all } i^* = 1, \dots, k \right\}$$

**Pas 3: “actualizare”** – se re-actualizeaza mediile fiecarui cluster pe baza observatiilor incorporate

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j$$

Repeta pasi 2 si 3. Oprete cand se atinge convergenta (nu mai sunt schimbari sau se atinge un anumit numar de pasi).



# K-means clustering (2D)

## Metode de initializare

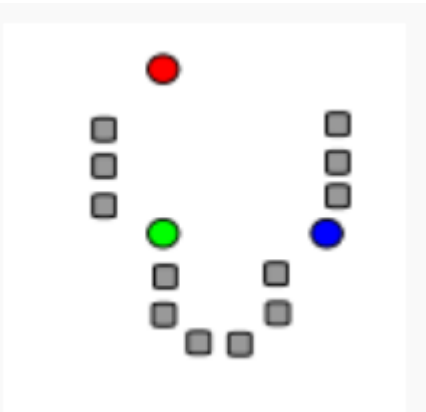
-**Forgy** – aleg aleator k observatii  $\Rightarrow$  medii initiale

-**Random partition** - initial asigneaza aleator un cluster la fiecare observatie

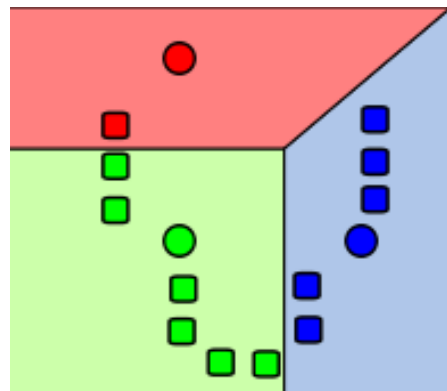
- Mediile initiale: centroidele unor puncte alese aleator pt. fiecare cluster

**Obs:** pt. clustering in spatiul de culoare (HS) sau (ab) sau (uv) se pot lua cele mai pronuntate K varfuri ale histogramei bi-dimensionale

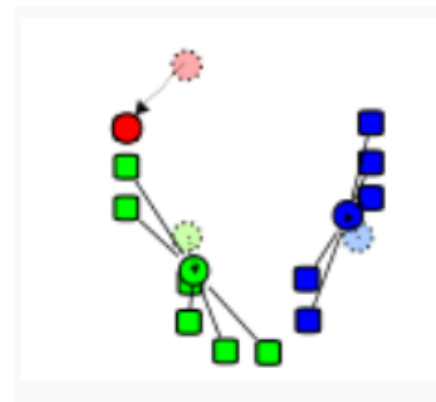
## Exemplu:



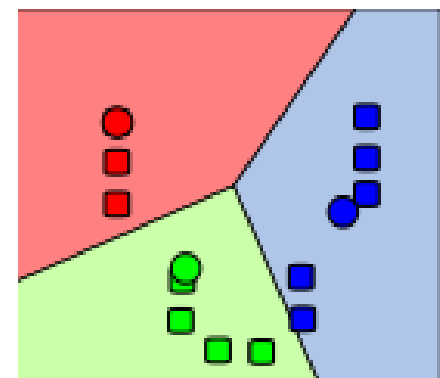
Initializare:  
k=3, centre  
alese aleator



Pas 2:  
Partitionare  
Voronoi



Pas 3:  
Recalculare  
centroide



Repeta pasii 2 si 3  
pana la  
convergenta



# Metrice de distanta

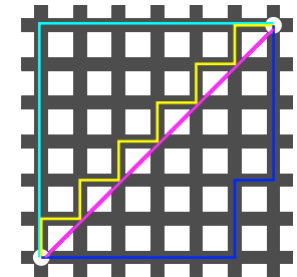
## Exemple pt. cazul 2D:

- doua puncte  $P_1 = (x_1, y_1)$  si  $P_2 = (x_2, y_2)$

**Distanta Euclidiană** - distanta geometrica intre doua puncte in spatiul bidimensional definita ca linia dreapta care le uneste:

$$d_{Euclidian}(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = \|P_1 - P_2\|$$

**City block sau Manhattan** - distanta este definita ca si calea de parcurgere prin unul din cei 4 vecini (sus, jos, stanga, dreapta - fara a parcurge ca directie diagonala intre pixeli):



$$d_{CityBlock}(P_1, P_2) = |x_2 - x_1| + |y_2 - y_1|$$

**Chessboard sau Chebyshev** (miscarea regelui pe tabla de sah) - parcurgerea se poate realiza in cele opt directii spatiale:

$$d_{Chessboard}(P_1, P_2) = \max(|x_2 - x_1|, |y_2 - y_1|)$$



## 2. Segmentare la nivel de regiune

---

- In spatiul imagine
- Criterii de uniformitate
- 1. Region Growing
- 2. Region Splitting
- 3. Split & merge

### **Segmentarea imaginii prin Region Splitting (impartire)**

- (1) Imparte imaginea in blocuri B de dimensiune  $N \times N$ ;  $N = 2^n$ , n - rangul blocului.
  - (2) Pentru fiecare bloc:
    - (3) Dc. NEUNIFORMITATEA (B) > T si  $k = \text{rang}(B) > 0$ ; atunci
      - divide blocul B into 4 blocuri egale B1;B2;B3;B4;
      - repeta pasul (3) pt. B1;B2;B3;B4;
- Altfel raporteaza B ca un bloc aparte.



# Region Growing

---

Metoda region growing are la baza un proces iterativ prin care regiuni ale imaginii sunt fuzionate incepand de la regiuni primare (care pot fi pixeli sau alte regiuni mici – celule de baza). Iteratiile de crestere se opresc atunci cand nu mai sunt pixeli de procesat !

## Algorithm:

1. Se segmenteaza imaginea in celule de baza (dimensiune  $\geq 1$  pixel).
2. Fiecare celula este comparata cu vecinii ei folosind o masura de similaritate. In caz afirmativ (valoarea metricii de similaritate  $<$  prag) celulele sunt fuzionate intr-un fragment mai mare si se actualizeaza trasaturile regiunii folosite la masura similaritatii (de obicei prin mediere ponderata).
3. Se continua procesul de crestere al fragmentului prin examinarea tuturor vecinilor pana cand nu se mai pot realiza fuziuni.
4. Se trece la urmatoarea celula ramasa nemarcata si se repeta pasii 2-3. Algoritmul se opreste atunci cand nu au mai ramas celule nemarcate.

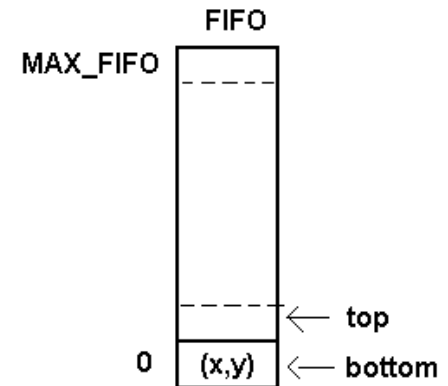




# Exemplu de implementare

O implementare eficientă folosește o coadă (similar cu pasul 4 de la Canny). Algoritmul rulează astfel:

1. Parcurge imaginea de la stânga la dreapta și de sus în jos și găsește primul seed point (celula de baza) și pune coordonatele sale în coadă și se stabilește o eticheta unică pentru aceea regiune
2. Cât timp coada nu este vidă, repeta:
  - Extrage primul punct din coada (sterge)
  - Găsește toți vecinii acestui punct care satisfac condiția de similaritate
  - Marchează în imagine vecinii acestui punct cu eticheta seed pointului inițial
  - Pune coordonatele acestor puncte în coadă
  - Continuă cu următorul punct din coadă
3. Continuă de la pasul 1 cu următorul seed point (neparcurs încă).





# Region Growing - examples

RG from a user defined seed point in (H,S) space



RG in (R,G,B) space





- [1] W. Skarbek, A. Koschan, Colour Image Segmentation: A Survey, Technical report 94-32, Technische Universität Berlin, Fachbereich 13 Informatik Franklinstrasse 28/29, 10587 Berlin, Germany
- [2] A. Koschan, M. Abidi, Digital Color Image Processing, Wiley & Sons, 2008.
- [3] [http://en.wikipedia.org/wiki/K-means\\_clustering](http://en.wikipedia.org/wiki/K-means_clustering)