



Technical University of Cluj - Napoca
Computer Science Department

Procesarea Imaginilor

Curs 12

Modele de culoare. Procesarea si segmentarea imaginilor color.



Achizitia imaginilor color

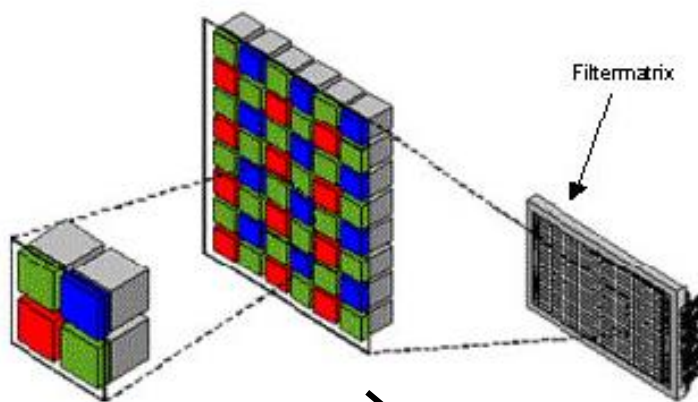
Senzori color

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

⇒ white balance, decodificare pattern Bayer

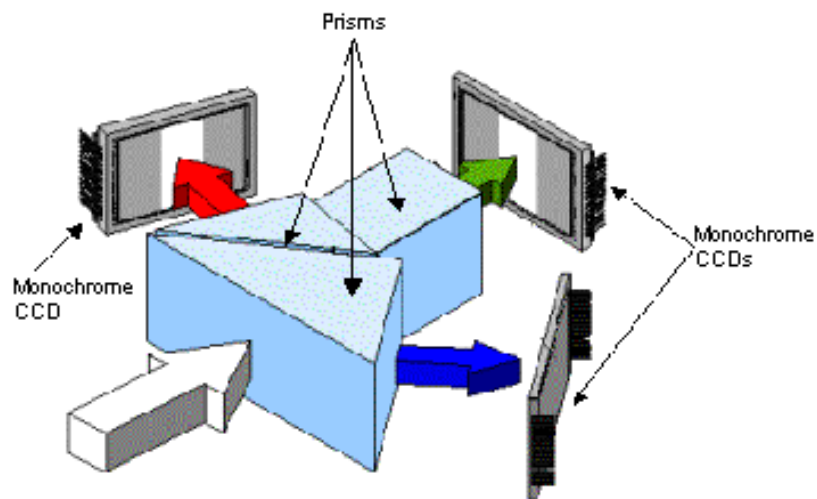
a) Bayer mask

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



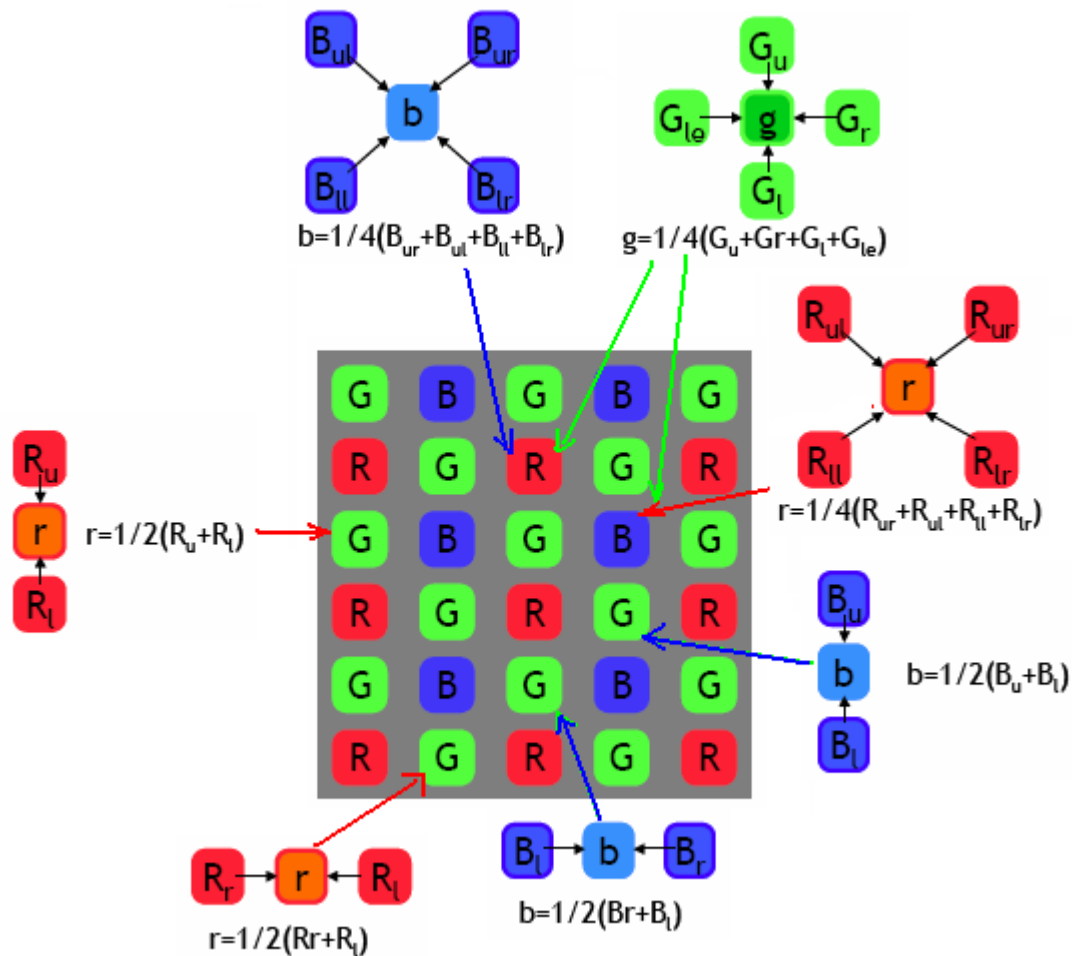
direct

Imagine RGB



Achizitia imaginilor color

Decodificarea pattern-ului Bayer



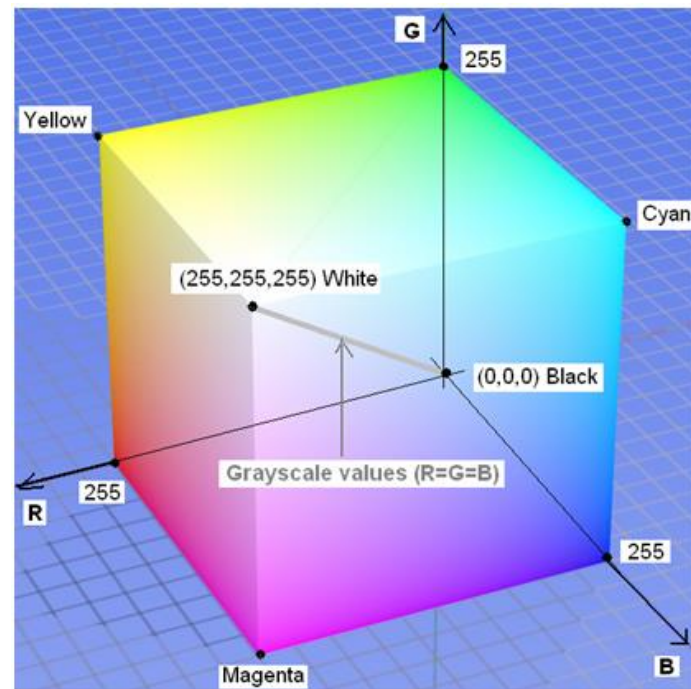
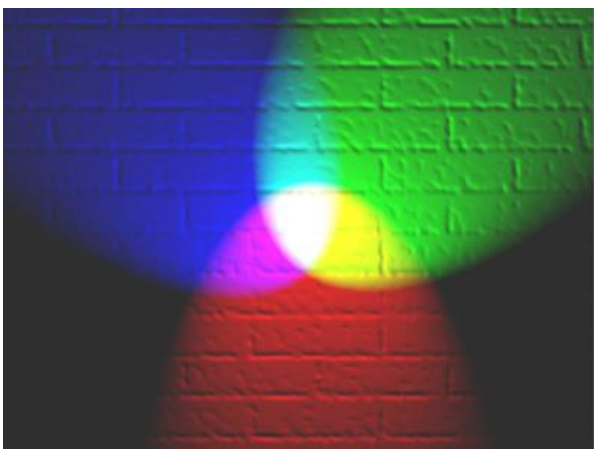
Calitatea imaginii (Bayer pattern vs. 3CCD) ???



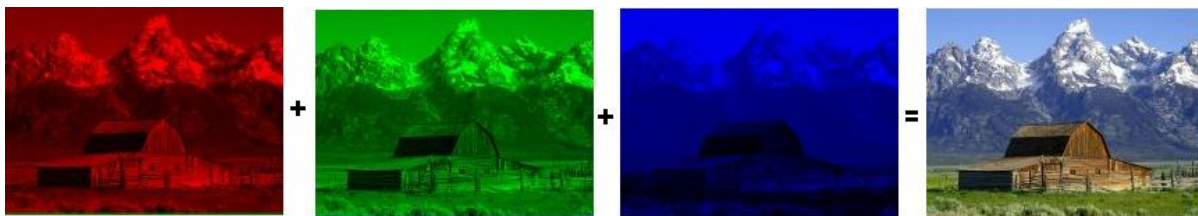
Spatiul de culoare: RGB

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 $2^8 \times 2^8 \times 2^8 = 2^{24} = 16.777.216$.





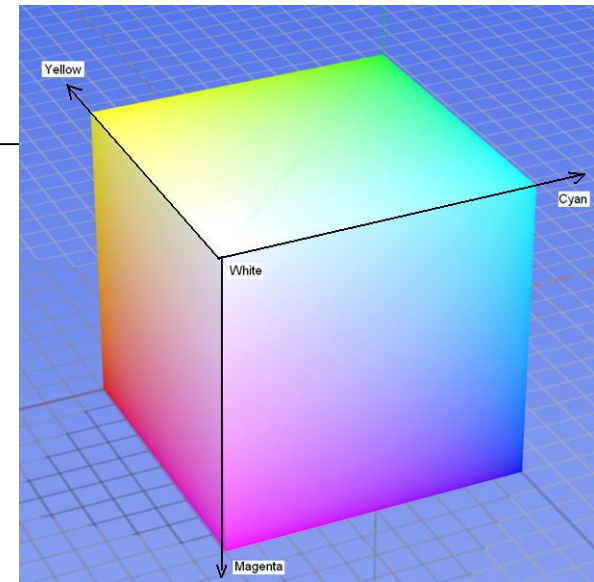
Spatiul de culoare: CMY

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: HSV (HSI, HSB, HSL)

HSI: (H, S, I), $H=0 \dots 360^\circ$, $S=0 \dots 1$, $V=0 \dots 1$

Algoritmul de conversie:

$r = R/255$; // r : componenta R normalizata
 $g = G/255$; // g : componenta G normalizata
 $b = B/255$; // b : componenta B normalizata

$M = \max(r, g, b)$;
 $m = \min(r, g, b)$;
 $C = M - m$;

Value:

$$V = M;$$

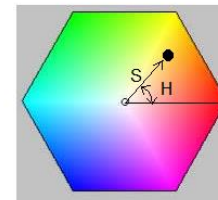
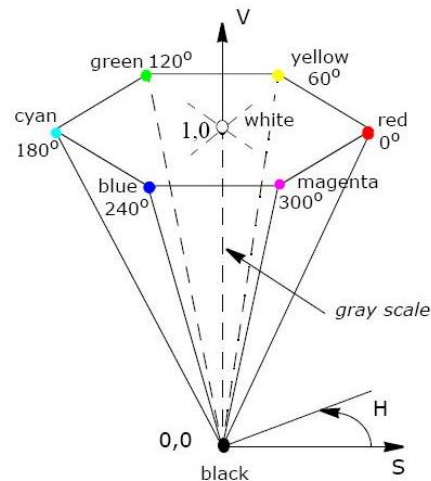
Saturation:

If (C)

$$S = C / V;$$

Else // grayscale

$$S = 0;$$





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

HSI: (H, S, I), $H=0 \dots 360^\circ$, $S=0 \dots 1$, $V=0 \dots 1$

Algoritmul de conversie:

Hue:

If (C) {

*if (M == r) H = 60 * (g - b) / C;*

*if (M == g) H = 120 + 60 * (b - r) / C;*

*if (M == b) H = 240 + 60 * (r - g) / C;*

}

Else // grayscale

H = 0;

If (H < 0)

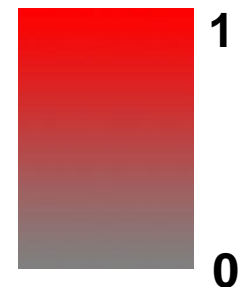
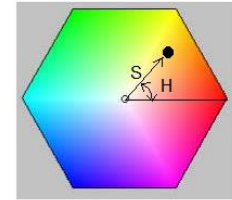
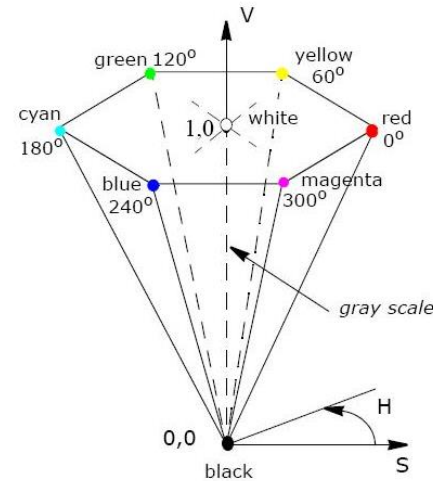
H = H + 360;

Valorile pt. H, S si V calculate cu formulele de mai sus vor avea următoarele domenii de valori:

$H = 0 \dots 360$

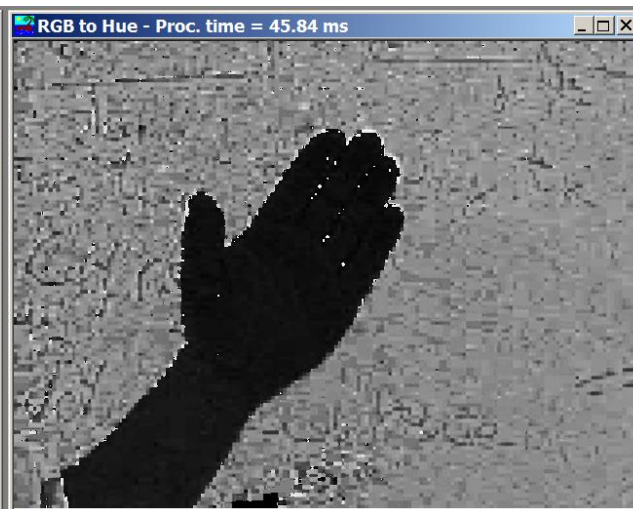
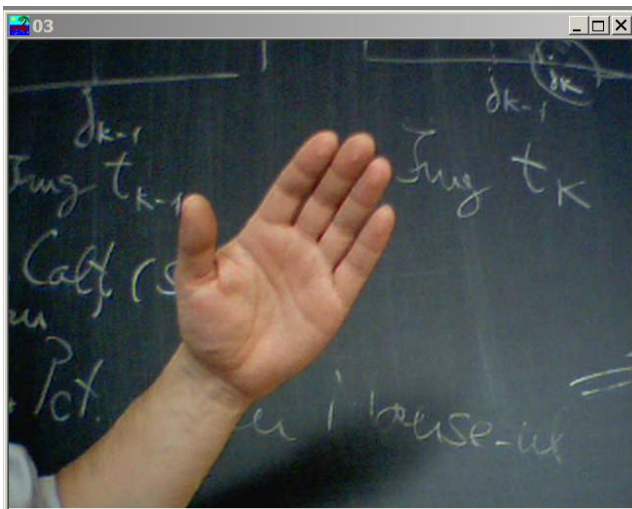
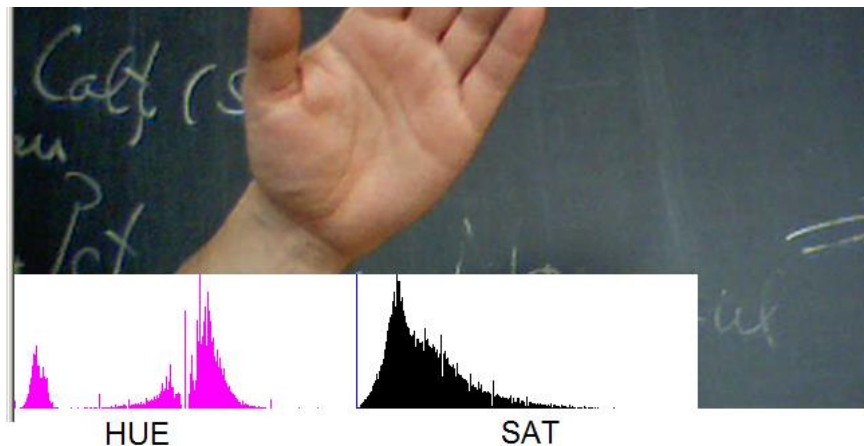
$S = 0 \dots 1$

$V = 0 \dots 1$





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



Reprezentare normalizata (in intervalul 0 .. 255) a valorilor lui Hue si a Saturatiei



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

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

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

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

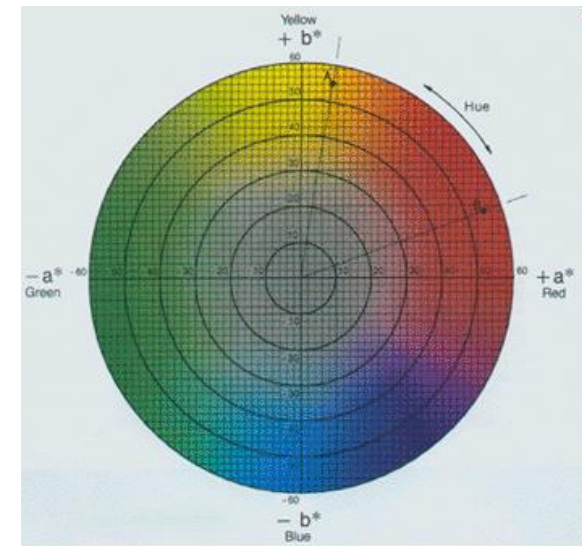
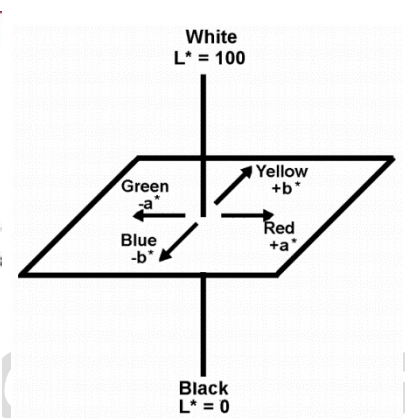
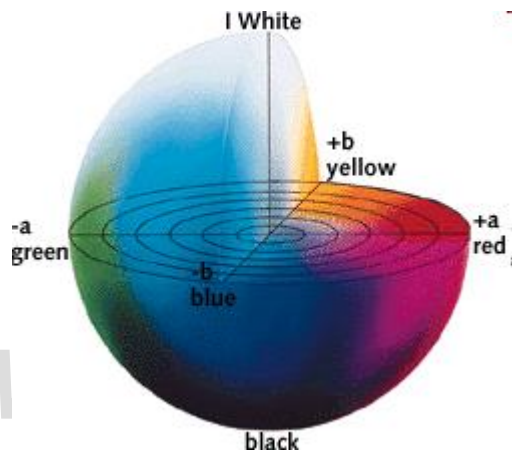
L – componenta de intensitate

a, b - componentele de culoare cu variatie liniara

CIE(Luv) space

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

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





Proprietati ale trasaturilor cromatice

Invarianta la translatie si scalare (la variatii de iluminare)

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 scalarea uniforma 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. $R, G, B \approx 0$

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

Exemple:

$$H(1,0,0) = 0, H(0,1,0) = 120^\circ$$

$$H(1,1,0) \Rightarrow H(2,1,0) : \delta H = 30^\circ$$

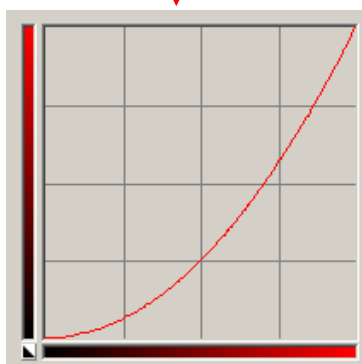
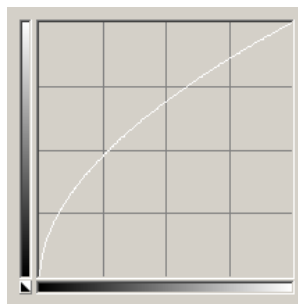
Concluzie: calcularea H in zone cu intensitatea mica \Rightarrow erori numerice



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

3. Supresia non-maximelor

4. Thresholding cu histereza



Metoda Canny (imagini color)

Pas2 :

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

Gradient pt. fiecare componenta de culoare \Rightarrow **Jacobian** (matricea derivatelor parțiale 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:

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

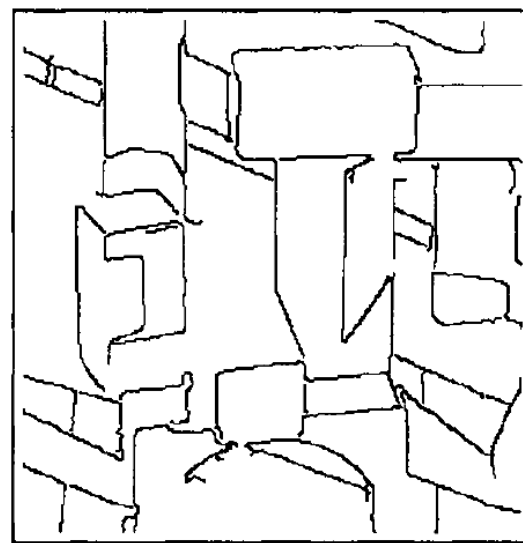
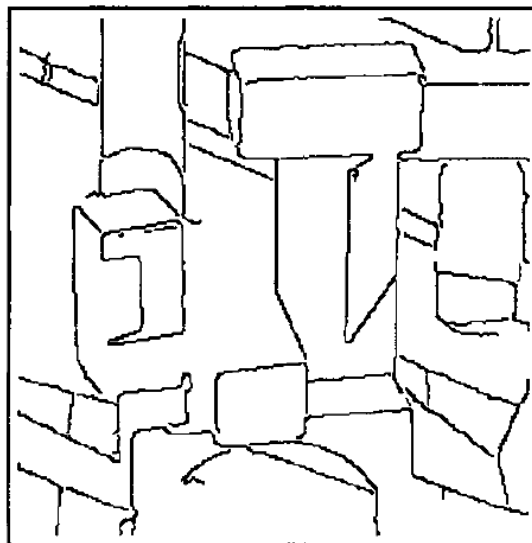
$$\mathbf{C}_x = (R_x, G_x, B_x)$$

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]





Segmentarea imaginilor color

Segmentare := identificare zone omogene din imagine

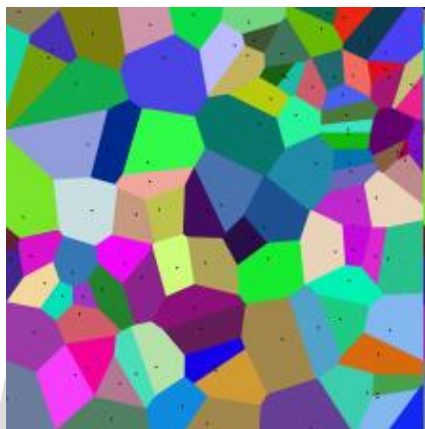
Segmentare imaginii color := identificare regiuni (componente conexe) care satisfac anumite criterii de omogenitate, bazate pe trasaturi derivate din componentele spectrale. Aceste componente sunt definite in spatiul de culoare considerat.

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

(a) Culoarea pixelului este intr-un semispatiu definit de un plan;

(b) Culoarea pixelului se incadreaza intr-un poliedru;

(c) Culoarea 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 notiunea de regiune) := 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)

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



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 (vezi L3)

The image displays a software interface for medical image processing, specifically for elastography. The main window shows a grayscale ultrasound image with a color-coded overlay representing stiffness. A color scale on the right ranges from 'Soft' (red) to 'Hard' (blue). The interface includes a menu bar (File, View, Scale, Processing, ROI, Measurements, Convert, Image, Geometry, Filters, Movie, Window, Help) and a toolbar with various tools. A 'Filtered Hue Histogram' dialog box is open, showing a histogram with several peaks, indicating the distribution of hue values. The histogram has an x-axis from 0 to 360 and an 'OK' button. The software interface also displays patient information: HITACHI UMF CLUJ CLIN. RADIOLOGICA, FR:17, TN, 001, P:H, MI, 05-APR-07 13:01:36. Technical parameters at the bottom include BG:15, L54M, 70I-/2/5/1/A/6, 26%, 2I-/1-/2/3/4 T-Elasto, BG:15, 70I-/2/5/1/A/6, L54M, 10M, Superficial, 50mm, L54M, and a series of numbered buttons (1-7). A coordinate string 'x:759, y:458; R:13, G:13, B:25, A:17, H:240' is visible at the bottom.



1.2 Clustering in spatiul de culoare

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

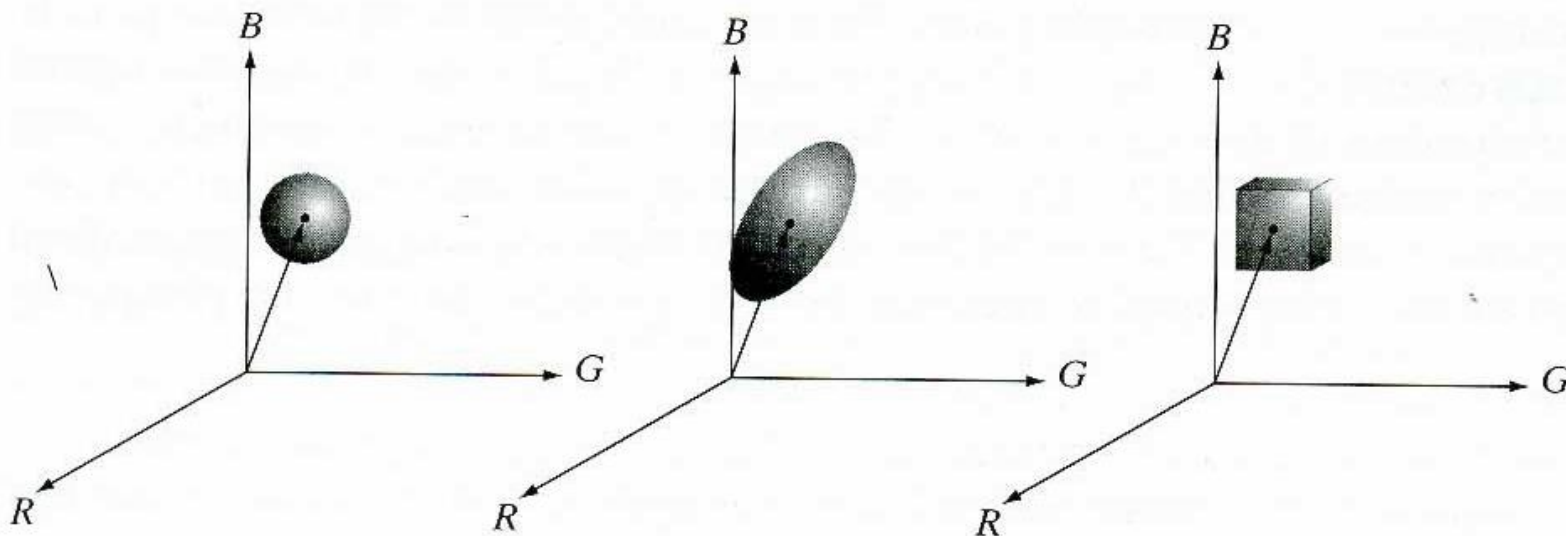
- **cuantizare/posterizare** (impartirea spatiului trasaturilor in subspatii de dimensiuni fixe);

- **clustering**:

- info. apriori: se stiu nr. clusterelor / pozitiile centrelor (de exemplu varfurile histogramelor)

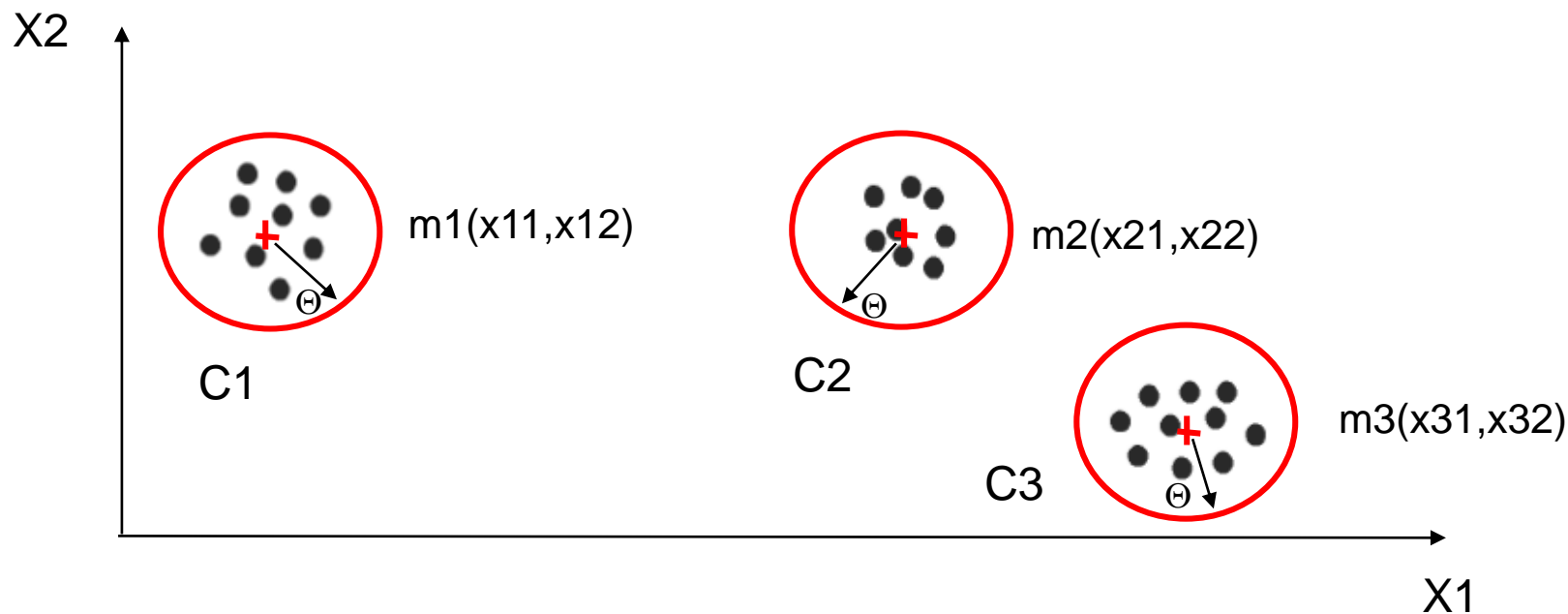
- fara informatii apriori

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





Exemplu: clustering in R^2



X_1, X_2 – pot fi cele doua axe de coordonate corespunzatoare componentelor de culoare: (H,S) sau (a,b) sau (u,v)



K-means clustering

K-means

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. pt. un spatiu bidimensional: $\mathbf{x}_i = (a_i, b_i)$)

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 reactualizeaza 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 pasii 2 si 3. Oprire cand se atinge convergenta (nu mai sunt schimbari sau se atinge un anumit numar de pasi/repetitii).



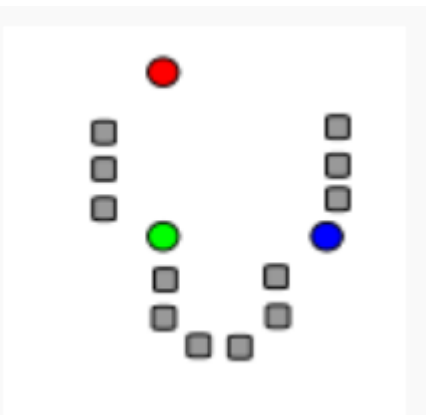
K-means clustering

Metode de initializare

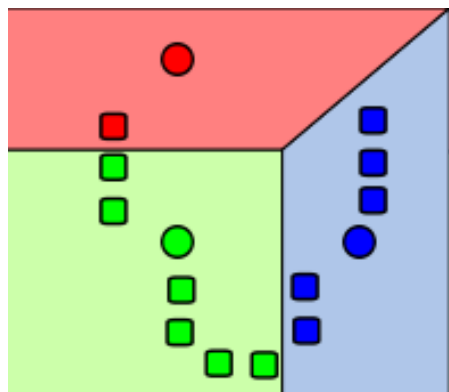
- Forgy** – aleg aleator k observatii \Rightarrow medii initiale
- Random partition** - initial asigneaza aleator un cluster la fiecare observatie
 - Mediile initiale: centroidele unor puncte alese aleator pt. fiecare cluster

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

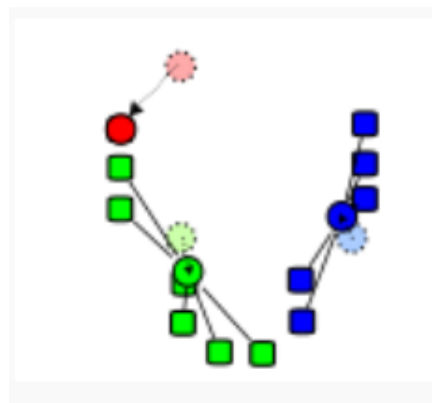
Exemplu:



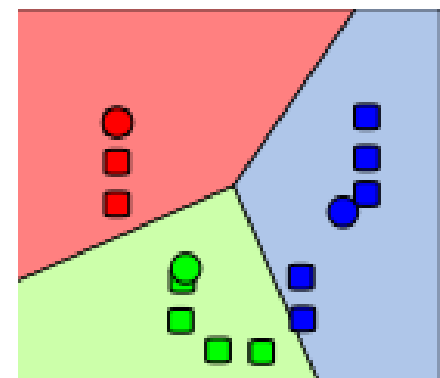
Pas 1: Initializare
k=3, centre alese
aleator



Pas 2:
Partitionare
Voronoi



Pas 3:
Recalculare
centroide



Repeta pasii 2 si 3
pana la
convergenta



Metrici de distanta

Exemple pt. cazul 2D:

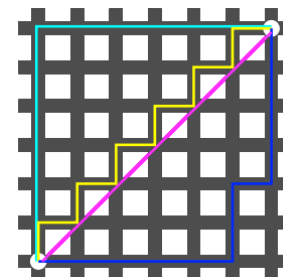
- doua puncte $P_1 = (x_1, y_1)$ si $P_2 = (x_2, y_2)$

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:

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

Mahalanobis – distanta geometrica normalizata

$$d_{Mahalanobis}(P_1, P_2) = \sqrt{\frac{(x_1 - x_2)^2}{\sigma_x^2} + \frac{(y_1 - y_2)^2}{\sigma_y^2}}$$



2. Segmentare la nivel de regiune

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

Se pot aplica atat pe imagini color (componente de culoare) cat si pe grayscale (intensitate)

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 ≥ 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 marcheaza ca si parcurse – ex. primesc eticheta regiunii) 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 algoritmul BFS și coadă FIFO (similar cu algoritmul omonim de etichetare):

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 etichetă unică pentru acea regiune
2. Cât timp coada nu este vidă, repeta:
 - Extrage primul punct din coada
 - Găsește toți vecinii acestui punct care satisfac condiția de similaritate
 - Marchează în imagine vecinii acestui punct cu eticheta seed pointului initial
 - Pune coordonatele acestor puncte în coadă
 - Continuă cu următorul punct din coadă
3. Continuă de la pasul 1 cu următorul seed point (neparcurs încă).

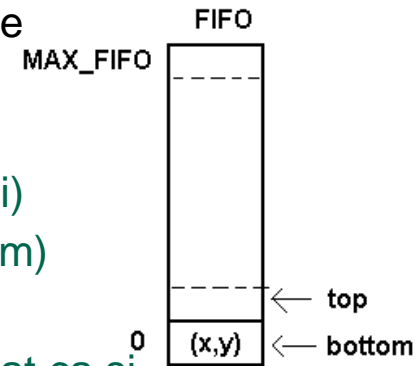


Exemplu de implementare (grayscale)

Pt. fiecare punct (seed-point) nemarcat (x,y):

1. Se adauga elementul de start in lista FIFO si se seteaza indecsii de inceput si sfarsit ai listei (top , bottom), valorile initiale ale contoarelor pentru numarul curent de elemente din lista FIFO si pentru numarul curent de elemente (pixeli) din regiunea curenta:

- $bottom = 0$; //indexul celui mai vechi element din lista
- $top = 1$; // indexul la prima pozitie libera (din partea de sus a listei)
- $contor = 1$; // numarul de elemente din lista ($contor = top - bottom$)
- $N = 1$; // numarul curent de pixeli din regiune
- $Labels(x,y)=k$; // pixelul primeste eticheta k si simultan este marcat ca si parcurs/procesat ($Labels(x,y)=>0$)



2. repeat

- pentru fiecare vecin (i,j) al pixelului din pozitia „bottom” a listei:
 - daca $Labels(i,j)=0$ (pixel neprocesat inca) si $abs(I(i,j)-I_{avg}) < T$:
 - adauga pixelul (i,j) in lista FIFO in pozitia top: $top++$; $contor++$; $N++$;
 - pixelul (i,j) primeste eticheta k (este adaugat la regiunea curenta si este marcat ca procesat: $Labels(i,j)=k$
 - se actualizeaza valoarea medie a regiunii:
$$I_{avg} = \frac{N * I_{avg} + I(i, j)}{N + 1}$$
- sterge elementul de pe pozitia bottom a listei FIFO: $bottom++$; $contor--$;

until (contor = 0) //lista FIFO devine goala (top=bottom)



Region Growing - examples

RG in spatiul (H,S) cu 1
seed point selectat manual



RG in spatiul (R,G,B)





- [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