

# Sisteme avansate de codare și compresie a datelor multimedia

## Curs 2 – Curs introductiv

Sl.Dr.Ing. Camelia FLOREA

Intelligent and multimodal image processing and analysis group (IMIPA),

Communications Departament, ETTI, TUCN,

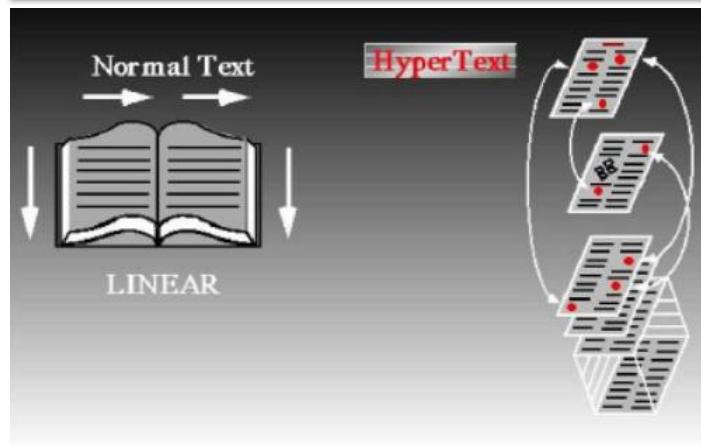
E-mail: [Camelia.Florea@com.utcluj.ro](mailto:Camelia.Florea@com.utcluj.ro),

Address: C. Daicoviciu, 15, room 431, Cluj-Napoca, RO.

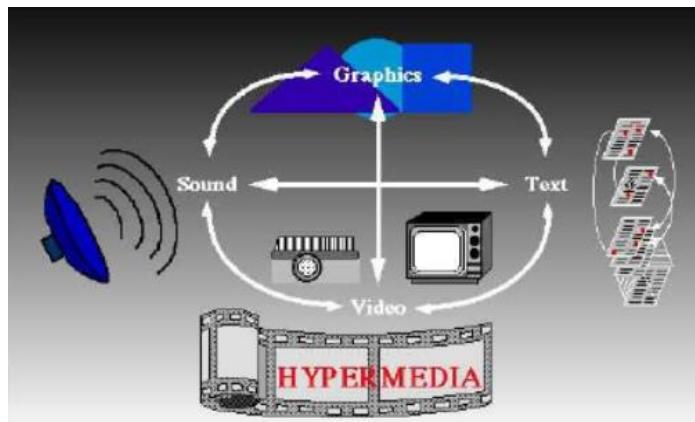
# Reprezentarea informației - Multimedia

- **multimedia** ⇔ reprezentarea informației sub diferite forme: text, imagini, audio, video etc.
- *Totuși,*
  - nu este suficientă reprezentarea informației sub diferite moduri
  - ci este necesar să fim capabili
    - să controlăm/ manipulăm/ transmitem aceste date în diferite formate
- *Astfel,* **multimedia** ⇔ domeniul care
  - se ocupă cu **integrarea comandată și controlată de calculator a diferitelor tipuri de informații:** text, grafică, imagini statice și în mișcare, animație, sunete, secvențe video
  - și în care fiecare tip de informație este reprezentată, memorată, procesată și transmisă sub formă digitală.

# HyperText și HyperMedia



- Aplicații HyperMedia
  - World Wide Web
  - Powerpoint
  - Adobe Acrobat (or other PDF software)
  - Adobe Flash
  - Video-on-Demand
  - Virtual Reality
  - Computer Games
  - Internet video streaming
  - Video conferencing
  - Digital Video Editing and Production Systems
  - Multimedia Database Systems



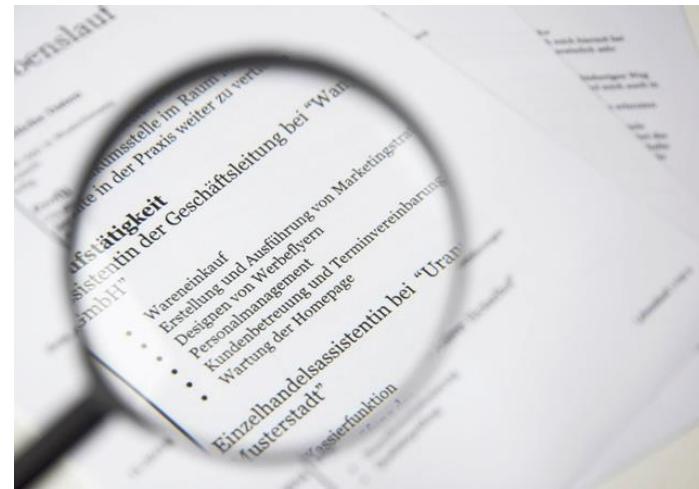
- HyperText – text care conține link-uri spre alt text/ organizarea materialului.
- HyperMedia – include imagini, grafică, video, sunet, ...

# Reprezentarea digitală a informațiilor multimedia

- producția și consumul de **materiale multimedia** => informațiile multimedia trebuie să aibă format digital -> pentru a putea fi gestionate de un sistem de calcul
  - Sistemul de calcul realizează digitizarea, manipularea și stocarea informațiilor multimedia
    - au la baza circuite cu două stări stabile (nivele de tensiune 0 și 1)  
= valori denumite **biți** -> aranjați în **octeți** -> aranjați în o succesiune liniară.
  - Modalități de **interpretare a modelelor de bit**
    - se poate face **prin asocierea cu numere** (cea mai simplă)
      - caracterele *textului* sunt reprezenta prin asocierea unui *număr unic fiecărei litere*
      - *luminozitatea* unei imagini într-un punct – valori între 0-255 (repräsentare pe 8 biti)
      - *amplitudinea instantanee* a undei de sunet, etc.
    - există **diverse modalități de implementare** a acestei asocieri,
      - singura problemă pe care trebuie să o rezolvăm este de a „programa” dispozitivele hardware/ software să fie capabile să interpreteze formatul digital pe care îl transmitem
- **STANDARDIZARE formate/ fișiere**

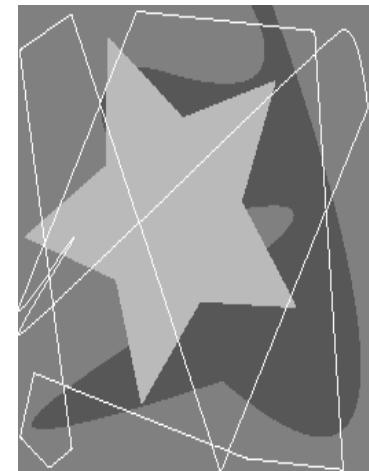
# Reprezentare: Text

- Input/generare: keyboard, speech input, optical character recognition (OCR), date stocate pe disc.
- Stocare/reprezentare caracter cu caracter:
  - 1 octet pe caracter (text), e.g. ASCII;
  - > 1 octeți pentru Unicode.
  - pentru alte tipuri de date (e.g. Spreadsheet files).
    - stocare ca text (cu formatare, e.g. CSV - Comma-Separated Values) sau stocare/codare binară
- Text formatat: Raw text sau text formatat - e.g HTML, Rich Text Format (RTF), Word or a program language source (Java, Python, MATLAB etc.)
- Compression: convenient to bundle files for archiving and transmission of larger files. E.g. Zip, RAR, 7-zip. General purpose compression programs may not work well for other media types: audio, image, video etc.



# Reprezentare: imagini statice versus grafică

- **grafică vectorială** = reține informația structurală → modelare (vectorială/ matematică) a formelor
  - sunt formate din **obiecte** (*linii, curbe, cercuri, etc.*) care au **attribute** (*grosime, scară de gri, culoare, tipar de umplere, etc.*)
  - sunt **revizuibile**, pot fi modificate, mutate, completate sau șterse.
  - obținute prin utilizarea unui **editor grafic**
  - Standarde grafice: OpenGL (Open Graphics Library), PHIGS, GKS
- **imaginile statice sau bitmap** = hartă de pixeli: o înregistrare a valorii fiecărui pixel din imagine (se atribuie o valoare pentru fiecare pixel)
  - NU sunt **revizuibile** – se poate doar ștergere și redesenare sau procesare/analiză conținut
- **dezavantajul major ale imaginilor bitmap** constă în mărimea relativ mare a fișierelor de imagine în comparație cu grafica vectorială.
- **imaginile și grafica** sunt adesea combinate
  - => o reprezentare **imagistică hibridă**.
- **conținutul semantic al desenului**
  - inclus în reprezentare



# Metode de producere a imaginilor BITMAP

- Captarea imaginilor din lumea reală (sub forma **bitmap** – imagini scanate)
  - cameră digitală
  - cameră analogică + placă de captură
  - scanare
- Create cu asistența calculatorului = ***imagini sintetizate***
  - prin generarea automata a imaginii de către un program
  - utilizarea unui program de editare specializat
  - captare ecran
  - conversia din format grafic la un format de tip imagine (ex. bitmap)
- ***Editoarele de imagine***
  - sunt programe similare cu ***editoarele grafice***, dar nu produc documente care rețin *conținutul semantic*.
  - oferă opțiuni mai sofisticate, în special pentru aplicații fotografice: procesare de imagine.
- „A picture is worth a thousand words, but it uses up three thousand times the memory.”

# Audio

---

- Input:
  - microfon
- CD Quality Audio necesita
  - 16-bit sampling la 44.1 KHz:
  - 24-bit, 96 KHz - audiophile rates
- fara compresie - 1 minut de audio
  - Mono CD quality = 5 MB.
  - Stereo CD quality = 10 MB.
- In general semnalul audio este comprimat
  - e.g. MP3, AAC, Flac, Ogg Vorbis



# Video

- Input: video camera
- O secventa video (raw) poate fi vazuta ca o serie de imagini statice
  - in general - 25, 30 sau 50 cadre pe secunda.
- Stocate comprimat:
  - Mpeg, H.26x, etc.



# Reprezentarea informației: Culoarea

- Culoarea este o senzație subiectivă produsă la nivelul creierului - ochiul uman conține două tipuri de celule receptoare:
  - **Bastonașele**
    - celulele folosite pentru vederea nocturnă - acestea *nu pot distinge culorile*
  - **Conurile**
    - folosite pentru vederea *cromatică* - sunt capabile să interpreteze culorile.
    - celulele de tip con sunt împărțite în trei subcategorii, fiecare din acestea fiind capabilă să recepționeze o anumită lungime de undă.
- => Faptul că ochiul reacționează la trei stimuli (lungime de undă diferite), conduce la ideea de **teoria vederii tricrome** adică orice culoare poate fi reprezentată prin combinația a trei **culori primare** (uzual: *roșu, verde și albastru*).
- Prin **amestecul aditiv a culorilor primare**, folosind factori de ponderare adecvați, se poate obține orice culoare din natură.
- **culorile primare** diferă în funcție de **echipament** (**RGB** afișare, **CMYK** imprimare, etc.)

# Adâncimea de culoare

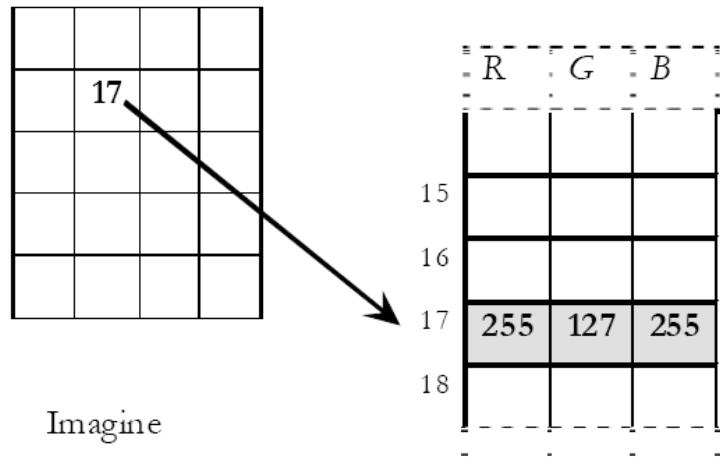
- 1bit/pixel
  - imagini binare (imaginile alb-negru, de fax, etc.)
- 4 biți/pixel (precizie mică)
  - imagini cu nivele de gri (16 nivele de gri)
  - grafică pentru calculator
- 8 biți/pixel
  - imagini atonale (“grayscale”- pe scara de gri)
  - Imagini color - 256 culori (uz general)
- 16 biți/pixel
  - utilizat de unele sisteme de calcul ( $r=5$ biți,  $g=6$ biți,  $b=5$ biți)
- 24 biți/pixel (true color)
  - sunt suficienți pentru a reprezenta toate culorile pe care ochiul uman le poate distinge
- 30, 36, 48 biți/pixel
  - aplicații profesionale

# Adâncimea de culoare

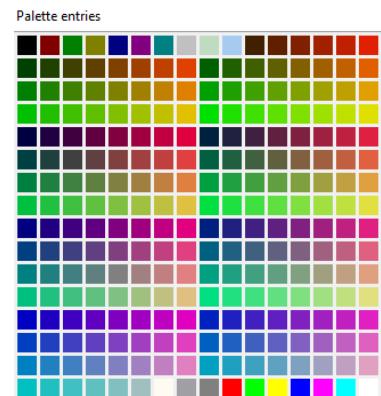
- Adâncimea de culoare
  - este factorul determinant în stabilirea **dimensiunii unui fișier**.
- Astfel, pentru stocarea informație de culoare
  - pentru 8 biți/pixel, spațiul necesar stocării este de 1 octet/pixel ( $W \times H \times 1$  octet)
  - 24 de biți/pixel este nevoie de 3 octeți/ pixel ( $W \times H \times 3$ ),
- Prin reducerea *adâncimii de culoare*
  - *putem obține o scădere substanțială a dimensiunii unui fișier*,
  - dar, se poate ajunge la o reducere a calitatii imaginii, însă nu în mod obligatoriu.
- Orice decizie în sensul reducerii adâncimii de culoare va trebui luată ținând cont
  - de imagine,
  - caracteristicile aplicației
  - și ale sistemului pe care acesta rulează.

# Indexarea culorilor - Colour Look-Up Tables (LUTs)

- Folosirea **colorilor indexate** - este utilă pe sisteme de calcul de performanță scăzută; și în cazul aplicațiilor de pe internet
  - În loc să folosim cate 8 biți pentru stocarea fiecărei componente R, G, B (= 3 octetăi)
    - => vom folosi un octet pentru stocarea unui index într-un tabel, cu 256 de poziții, fiecare dintre acestea conținând o valoare RGB pe 24 de biți.
  - fiecare imagine care utilizează culorile indexate va avea propria paletă de culori.
  - Paleta de culori se memorează în fișierul imaginii
  - Formate care permit reprezentarea/indexarea culorilor prin paleta
    - BMP, PNG, GIF, TIFF și altele.



## Utilizarea paletei de culori



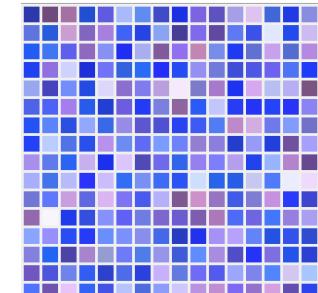
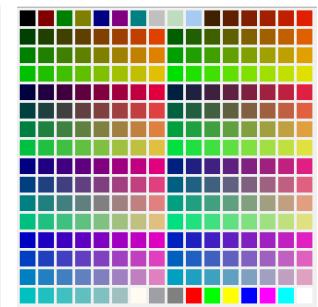
# Spațiul necesar stocării BMP

---

- Pentru o imagine de 640 x 480 pixel, este necesar:
  - Binar (1 bit) = 37.5 KB.
  - Pe nivale de gri (8 biți) = 300 KB.
  - Color (24-bit) = 921.6 KB
  - Color (8-bit, 256 culori) = 307.2 KB

# Indexarea culorilor

- Paleta de culori = culorile care sunt conținute în imagine => nu vom avea probleme deosebite legate de calitatea imaginii obținute
- fiecare imagine care utilizează culorile indexate va avea propria paletă de culori:
  - imagine subacvatică – un set de culori ce conține foarte multe nuanțe de albastru
  - imagine a unei păduri – un set de nuanțe de verde.
- chiar dacă fișierul se mărește prin adăugarea paletelor de culori, per ansamblu mărimea fișierului este de aproximativ 3 ori mai mică (practic se trece de la 24 de biți/pixel la 8 biți/pixel)



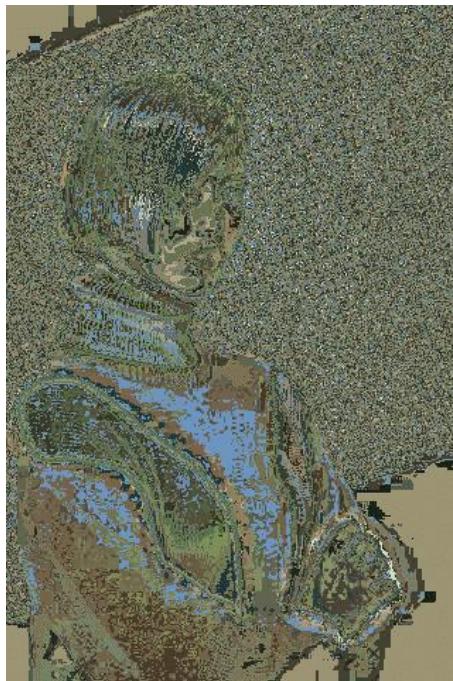


Imagine complexa (multe culori) apare efectul de contururi false.

Adăugare zgromot pseudo-aleator (dithering):



# Utilizând hărți de culoare/indexarea culorii



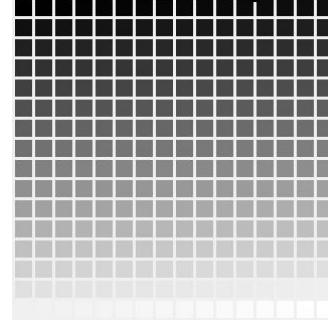
Paletă standard Web



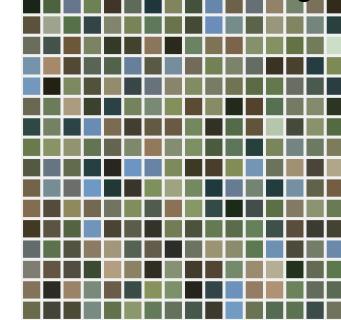
Paletă bazată pe conținut



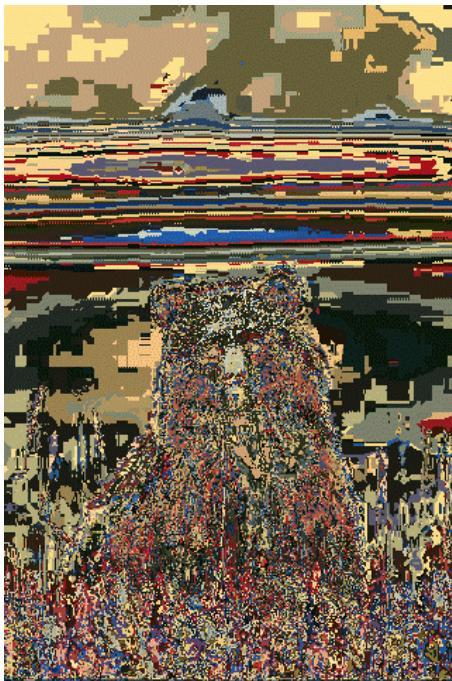
Paletă nivale de gri



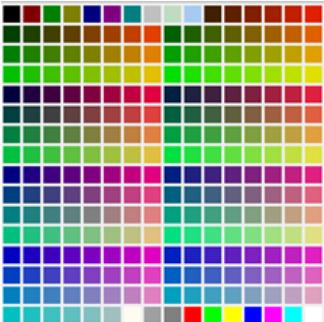
Paleta unei alte imagini



# Utilizând hărți de culoare/indexarea culorii



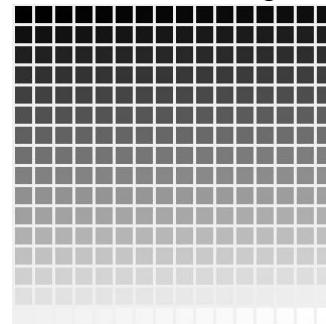
Paletă standard Web



Paleta unei alte imagini



Paletă nivale de gri

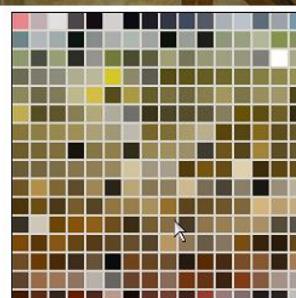
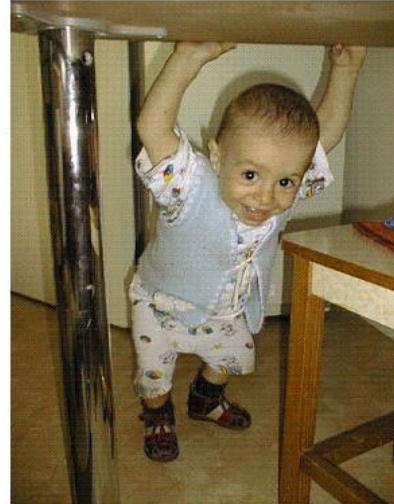


Paletă bazată pe conținut

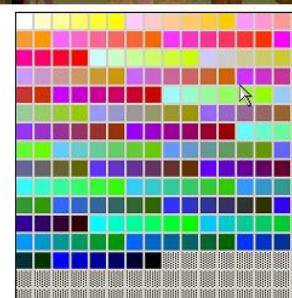


# Paleta de culori

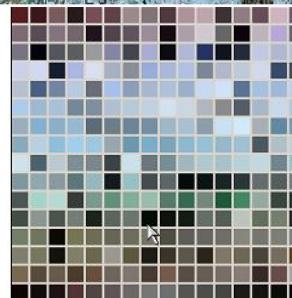
- imaginea originală cu 24 biți/pixel
- transformarea la o imagine cu 8 biți/pixel în care se folosește o paletă standard Web (figura 3.b) și se utilizează procesul de dithering
- reprezintă transformarea la nivele de gri (practic planul de luminanță al imaginii originale)
- reprezintă imaginea transformată la 8 biți/pixel utilizând o paletă (figura 3.a) cât mai apropiată de cea reală
- se utilizează din nou paleta Web din figura 3.b dar nu se folosește procesul de dithering
- prezintă imaginea transformată folosind paleta de culori 3.c a unei alte imagini



3.a



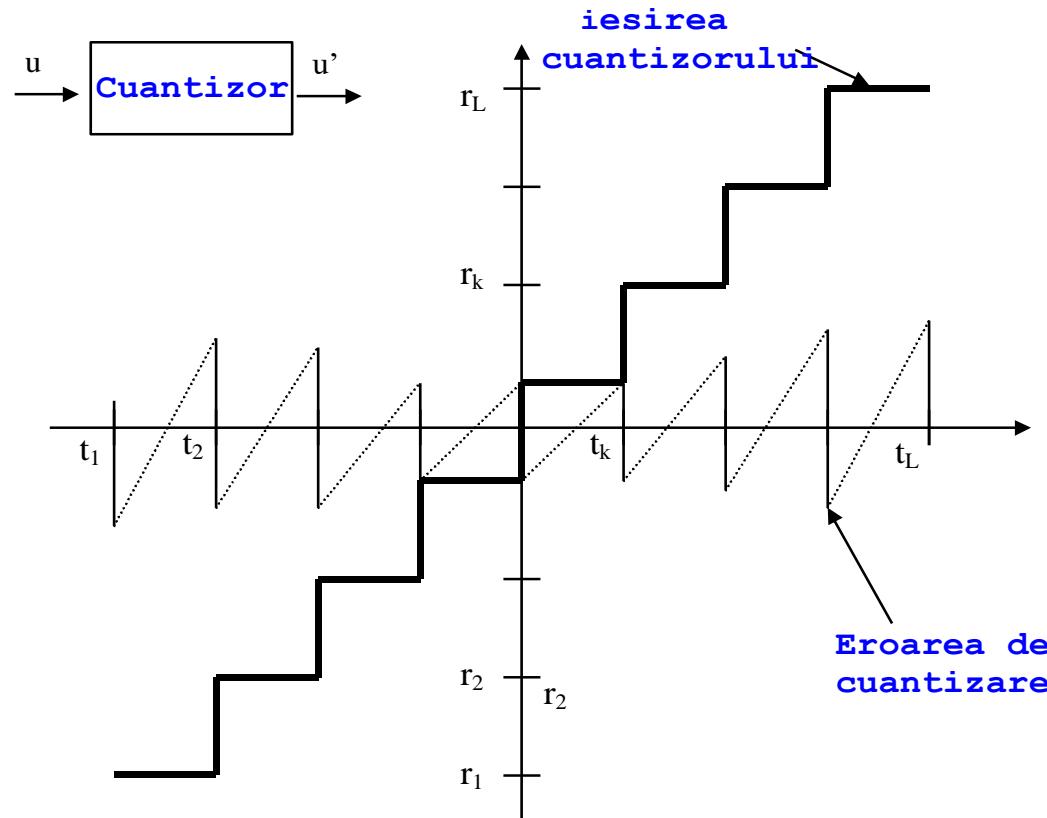
3.b



3.c

# Cuantizarea culorii/ Indexarea culorii

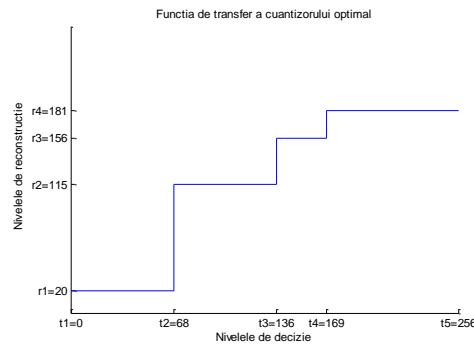
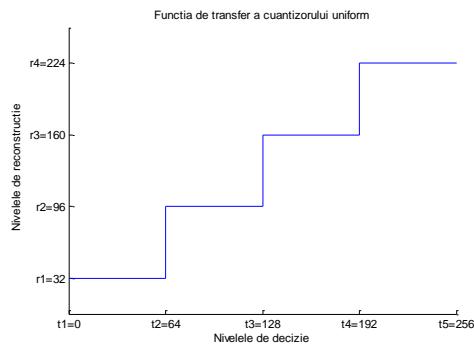
- Cuantizor uniform



# Cuantizare uniforma/ Cuantizare optimală



Imaginea necuantizata



Histograma imaginii necuantizate

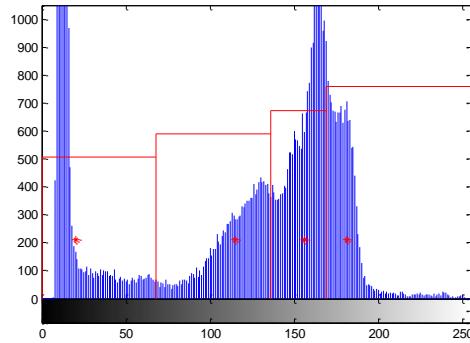
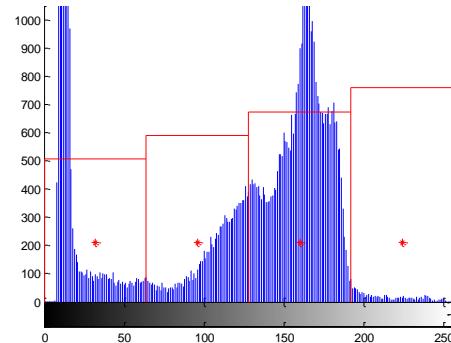
$$B=2 \Rightarrow L=4$$



Imaginea cuantizata uniform / optimal  
MSE=15



MSE=9.6



# Metode de cuantizare vizuala

- In general – daca  $B < 6$  (la cuantizarea uniforma) sau  $B < 5$  (la cuantizarea optimala) => apare fenomenul de "conturare" (apar contururi false) in imaginea cuantizata.
- "Conturarea" = grupuri de pixeli vecini sunt cuantizati la aceeasi valoare => apar regiuni de nivele de gri constante; contururile acestor regiuni sunt contururile false specifice "conturarii".
- Contururile false introduse de cuantizare nu contribuie semnificativ la MSE, dar sunt foarte deranjante vizual => este important sa reducem vizibilitatea erorii de cuantizare, nu doar MSE.
  - ⇒ Solutii: scheme de cuantizare vizuala, care sa mentina eroarea de cuantizare sub un nivel de vizibilitate acceptabil.
  - ⇒ O metoda foarte uzuală: cuantizarea cu zgomot pseudo-aleatoriu (dithering)



Cuantizare uniforma, B=4



Cuantizare optimala, B=4



Cuantizare uniforma, B=6

# Cuantizarea culorii/ Indexarea culorii



Img. Orig. B=24  
(16777216 culori)

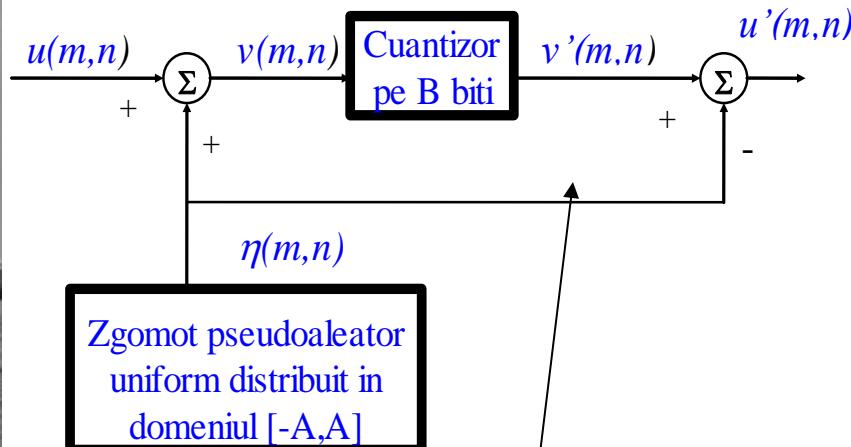
B=1 (2 culori)

B=2 (4 culori)

B=4 (16 culori)

B=8 (256 culori)

# Cuantizarea cu zgomot pseudo-aleator (“dither”)



Zgomot de amplitudine mare



Cuantizare uniforma,  $B=4$

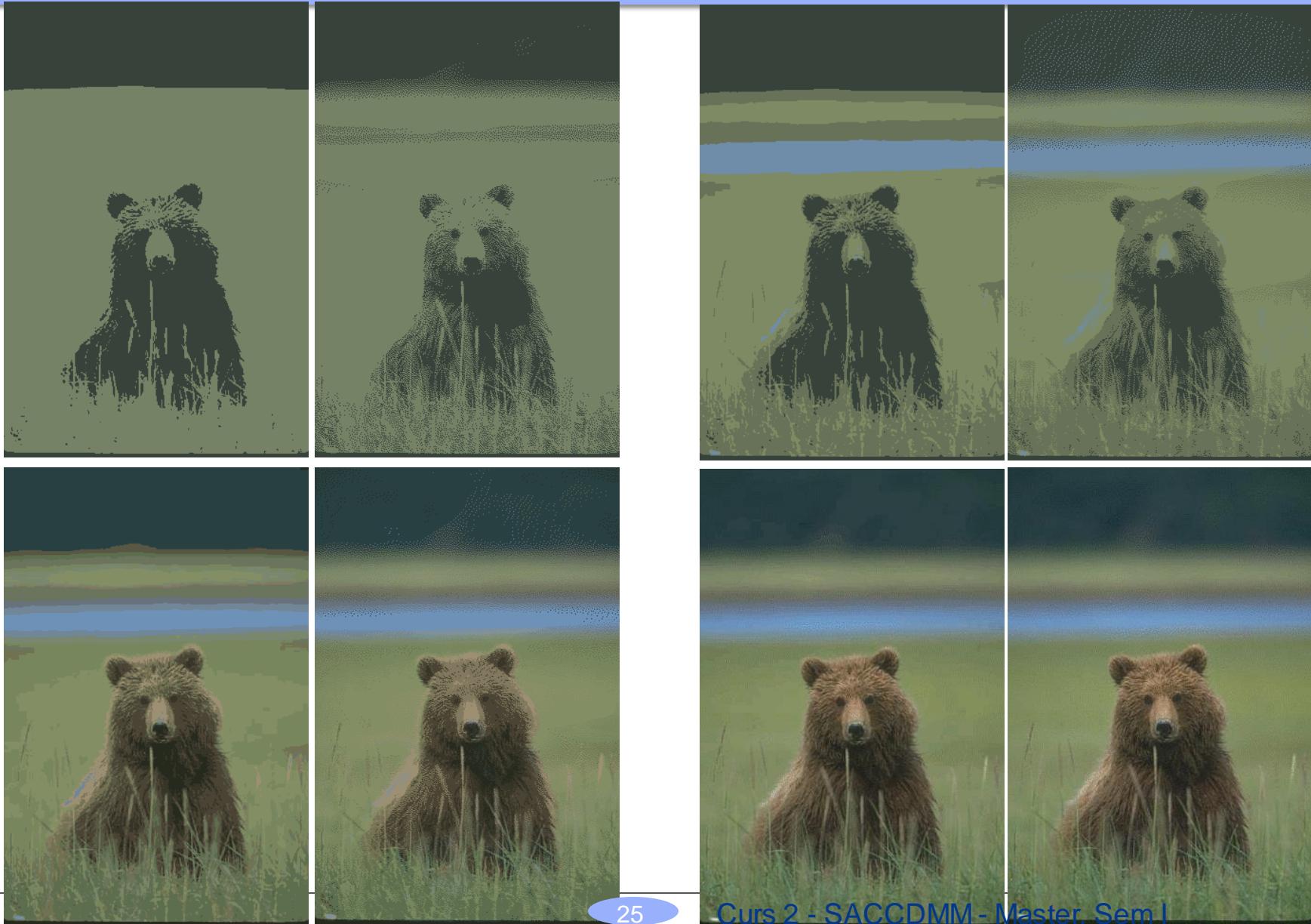


Inainte de scaderea zgomotului



Zgomot de amplitudine mica

# Cuantizare fără și cu zgomot pseudoaleator



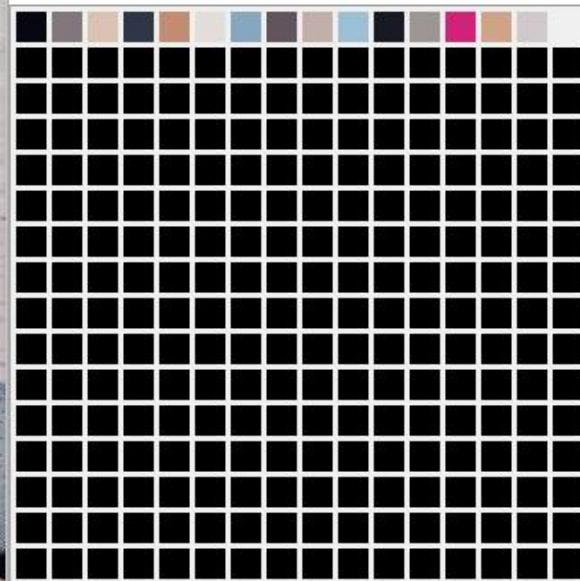
# Cuantizare fără și cu zgomot pseudoaleator





640 x 425 x 8 BPP | 4/12 | 102 % | 266.68 KB / 266.66 KB | 03.10.2017 / 14:47:15

Palette entries



Click on a color to select; double-click to edit

Index: Value:

Hint: click into the image (main window) to select the index here.

Refresh

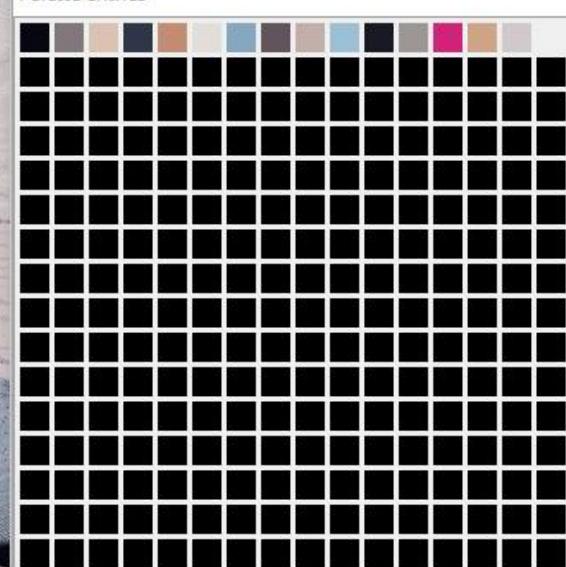
OK



640 x 425 x 8 BPP | 4/12



Palette entries



Click on a color to select; double-click to edit

Index: Value:

Hint: click into the image (main window) to select the index here.

Palette entries

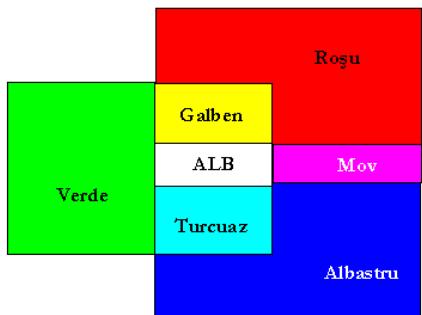


Click on a color to select; double-click to edit

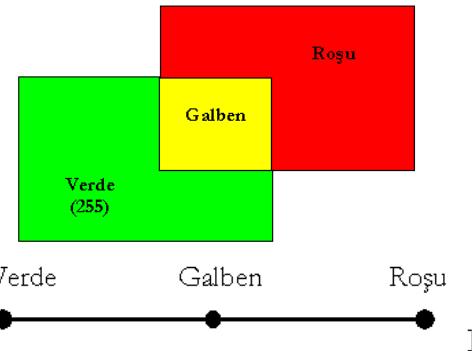
Index: Value:

Hint: click into the image (main window) to select the index here.

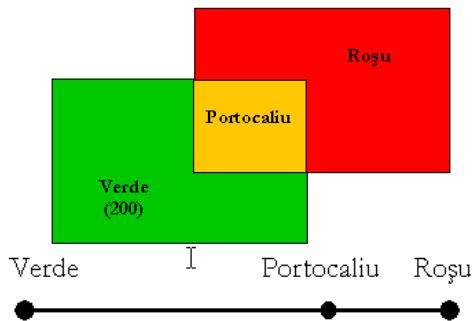
# Spațiul culorilor RGB



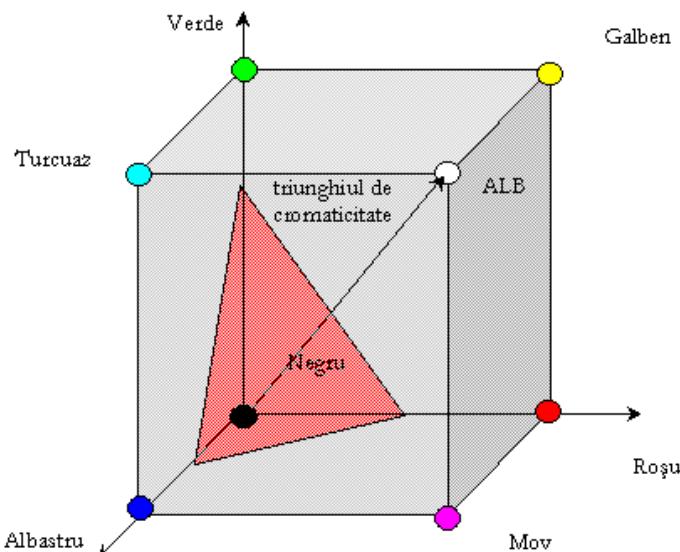
Amestec aditiv dintre **verde, roșu** și **albastru**



Amestec aditiv dintre **verde = 255** și **roșu = 255**  
(proportii egale – maxim)



Amestec aditiv dintre **verde = 200** și **roșu = 255**



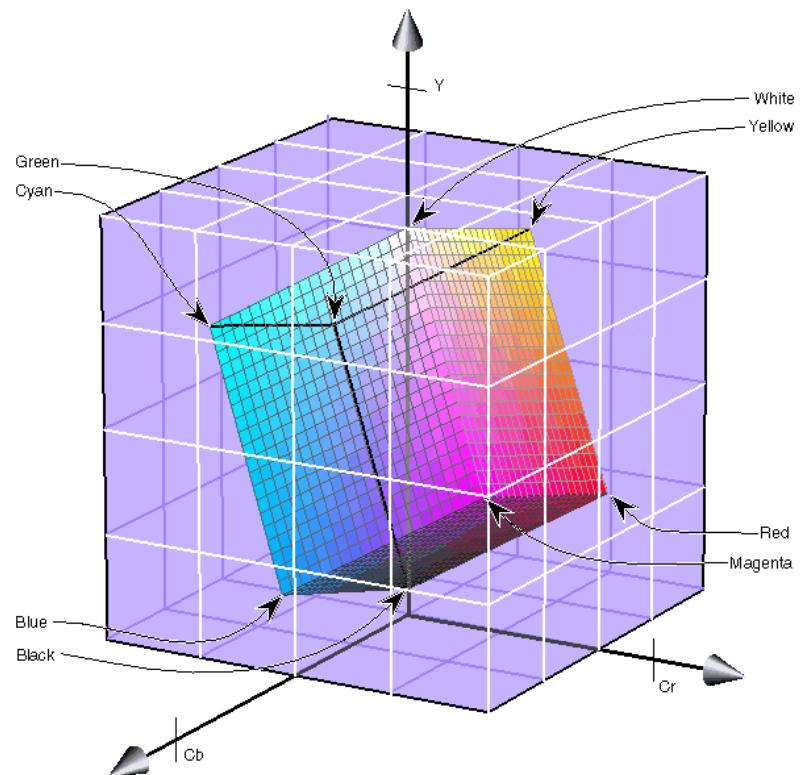
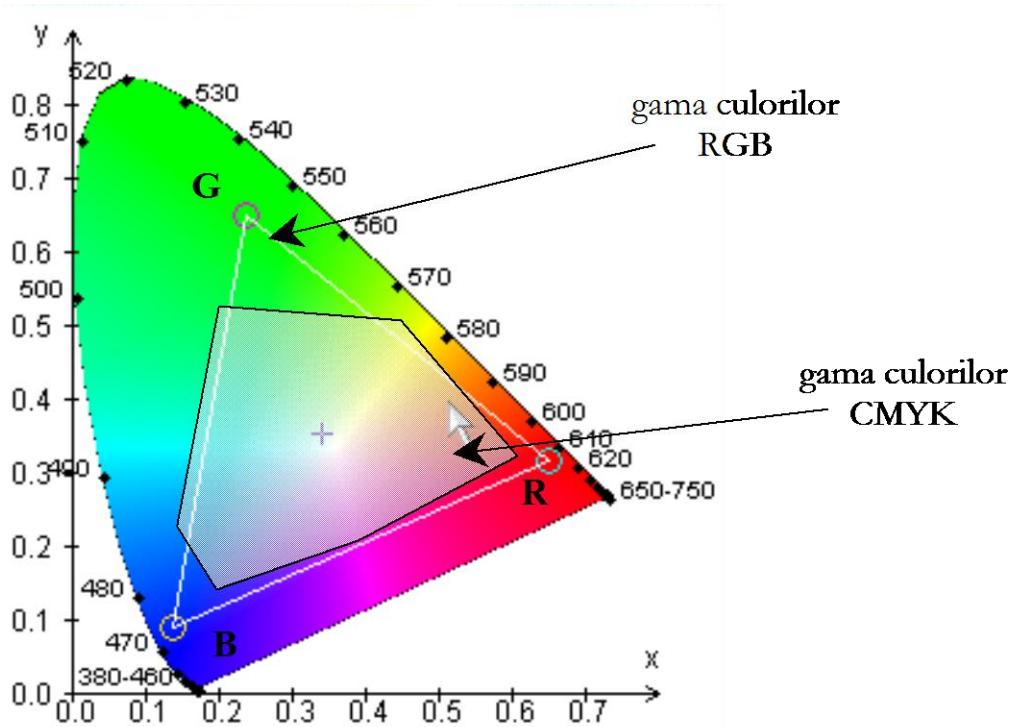
Componentele primare de culoare: R, G și B  
**sunt puternic corelate** ⇔ nu se pot analiza/  
procesa separat pentru descrierea culorii.

Imagine monocromă:

- toate punctele din imagine au  $R=G=B$   
⇒ cad pe diagonala cubului ( $R, G, B$ ).

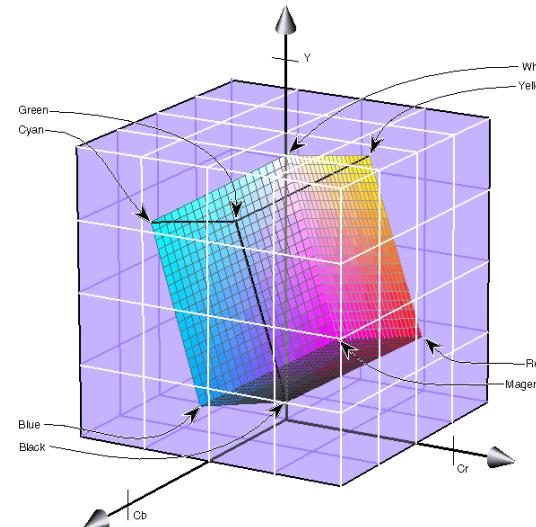
# Spații de culoare

- Tipuri de transformări:
  - liniare: YUV, YCbCr, YIQ, XYZ, etc.
  - neliniare: HSV, CIE L\*a\*b\*, CIE L\*u\*v\*, etc.

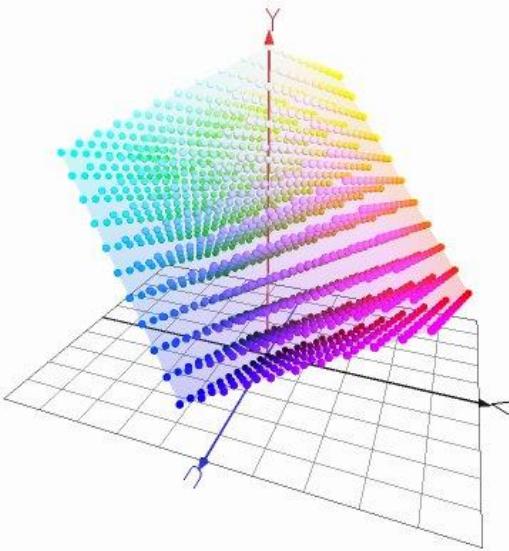


# Spații de culoare

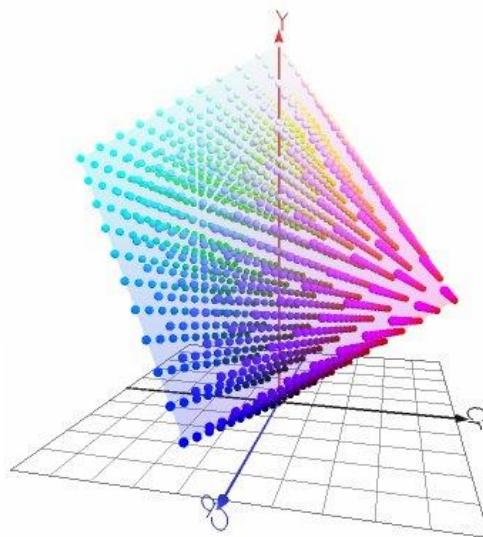
- **Tipuri de transformări:**
  - **liniare:** YUV, YCbCr, YIQ, XYZ, etc.
    - Besides the RGB representation, YIQ and YUV are the two commonly used in image/video.
  - **neliniare:** HSV, CIE L\*a\*b\*, CIE L\*u\*v\*, etc.
- Use Luminance and Chrominance to better encode how Humans 'see' colour.



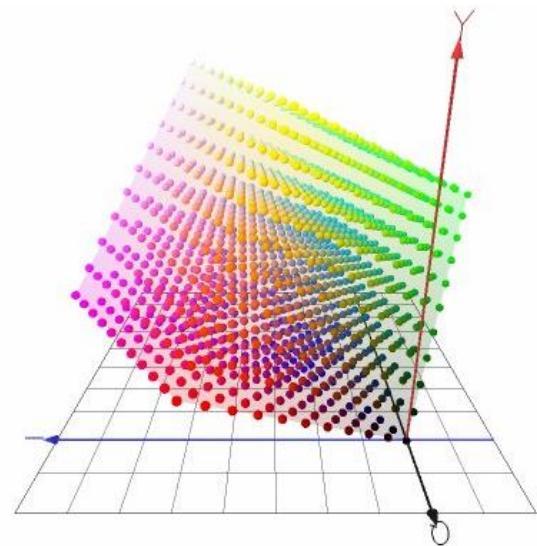
# Transformari liniare ale spatiului RGB



$$\begin{cases} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ U &= -0.147 \times R - 0.289 \times G + 0.436 \times B \\ V &= 0.615 \times R - 0.515 \times G - 0.100 \times B \end{cases}$$



$$\begin{cases} Y &= 0.2989 \times R + 0.5866 \times G + 0.1145 \times B \\ Cb &= -0.1688 \times R - 0.3312 \times G + 0.5000 \times B \\ Cr &= 0.5000 \times R - 0.4184 \times G - 0.0816 \times B \end{cases}$$

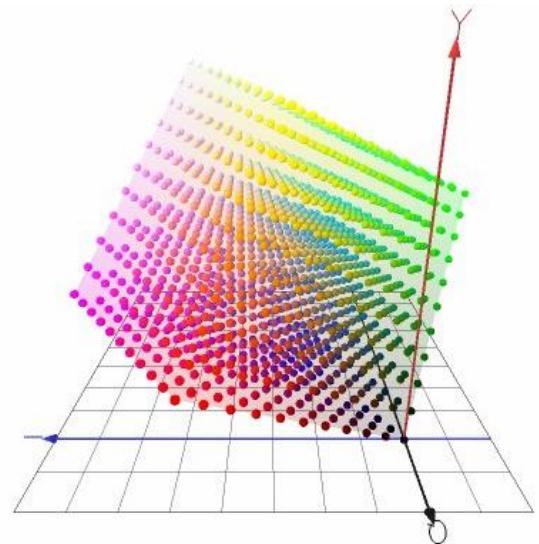


$$\begin{cases} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ I &= 0.596 \times R - 0.274 \times G - 0.322 \times B \\ Q &= 0.212 \times R - 0.523 \times G + 0.311 \times B \end{cases}$$

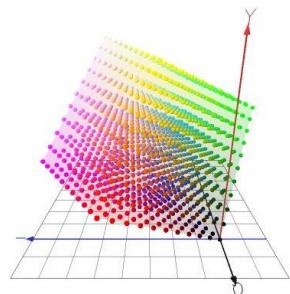
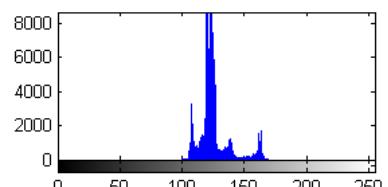
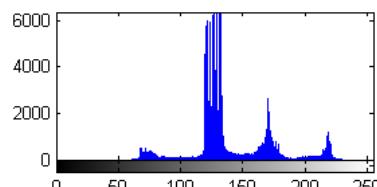
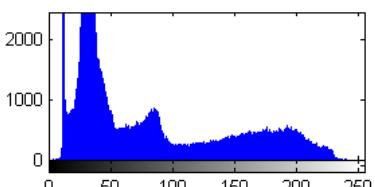
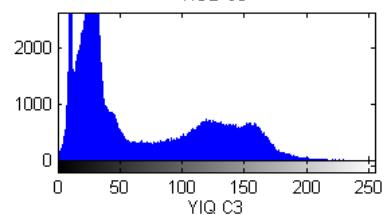
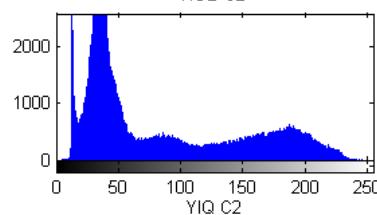
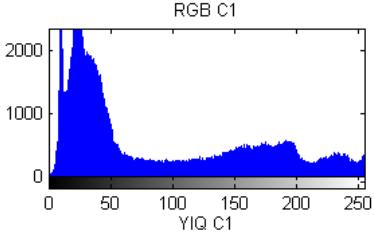
# YIQ Colour Model

- YIQ is used in colour TV broadcasting
- Y (luminance) is the CIE Y primary.
- I is red-orange axis, Q is roughly orthogonal to I.
- Eye is most sensitive to Y, next to I, next to Q.  
⇒ Analog Video Compression Scheme:
  - 4 MHz is allocated to Y,
  - 1.5 MHz to I,
  - 0.6 MHz to Q.

$$\begin{cases} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ I &= 0.596 \times R - 0.274 \times G - 0.322 \times B \\ Q &= 0.212 \times R - 0.523 \times G + 0.311 \times B \end{cases}$$



# RGB vs YIQ



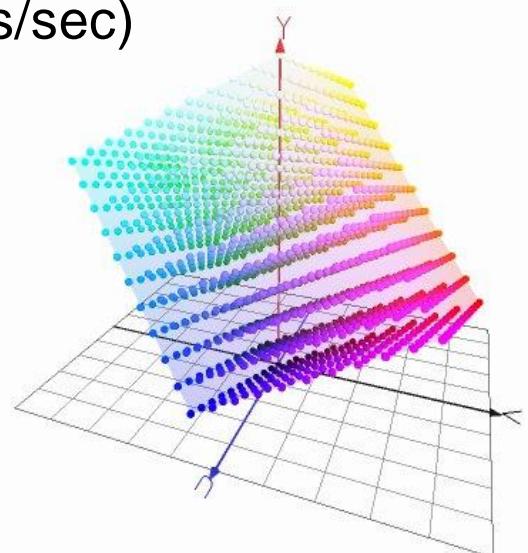
# YUV Color Model

- Digital video standard established in 1982
- Video is represented by a sequence of fields (odd and even lines). Two fields make a frame.
- Works in PAL (50 fields/sec) or NTSC (60 fields/sec)
- Uses the Y, U, V colour space

$$Y = 0.299R + 0.587G + 0.114B,$$

$$U = B - Y,$$

$$V = R - Y$$



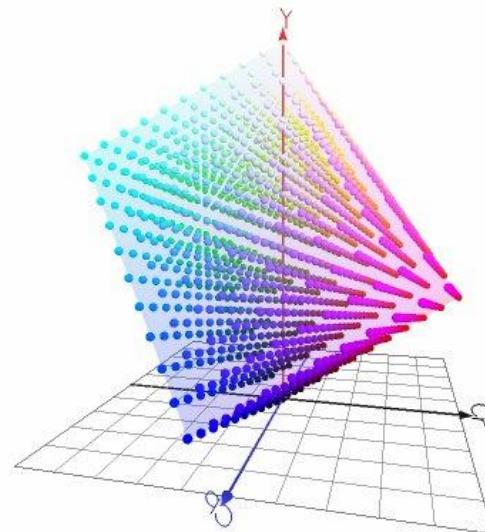
$$\begin{cases} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ U &= -0.147 \times R - 0.289 \times G + 0.436 \times B \\ V &= 0.615 \times R - 0.515 \times G - 0.100 \times B \end{cases}$$

# YCrCb Colour Model

- Similar to YUV
- YUV is normalised by a scaling

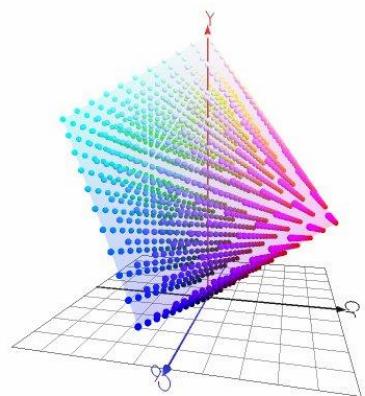
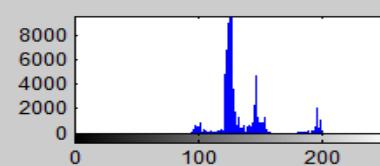
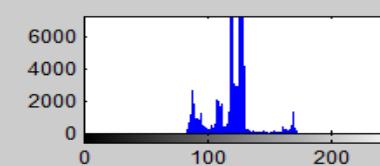
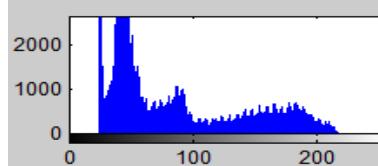
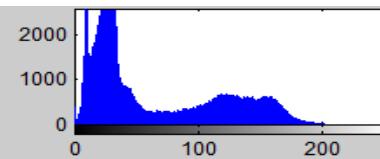
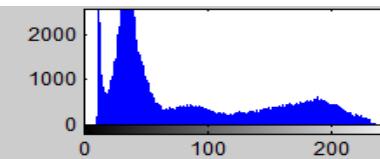
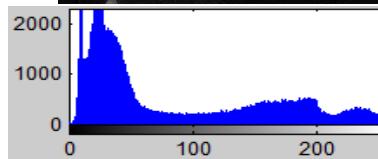
$$Cb = (B - Y)/1.772$$

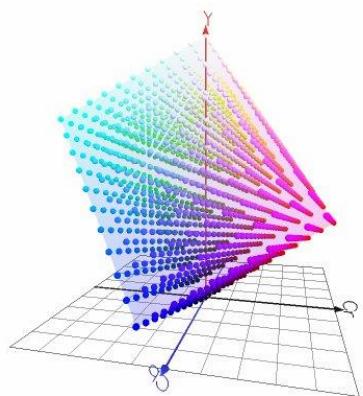
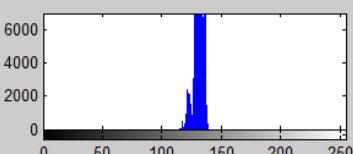
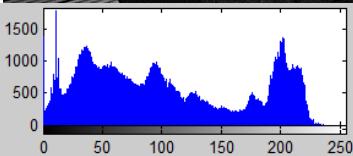
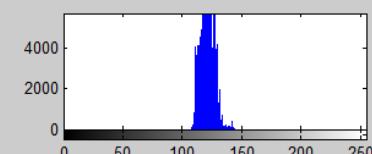
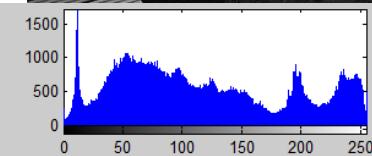
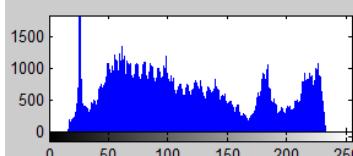
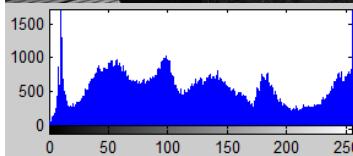
$$Cr = (R - Y)/1.402$$

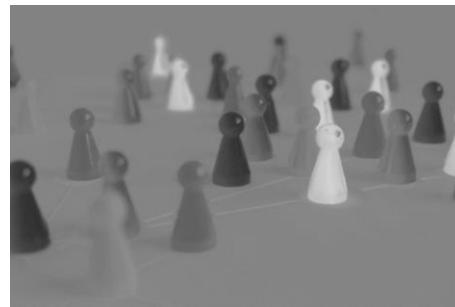
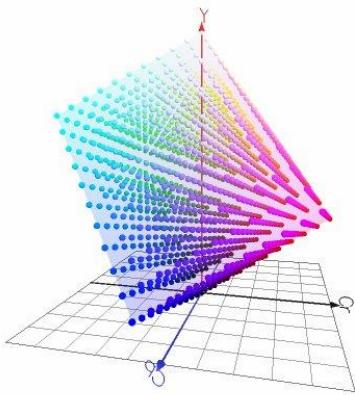
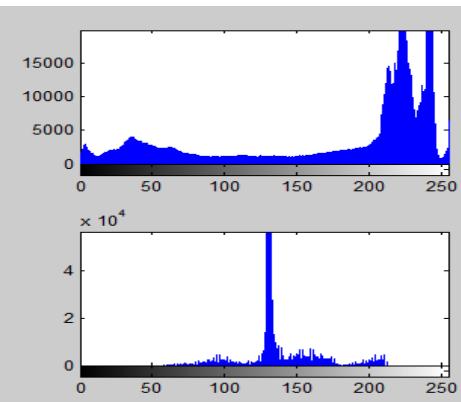
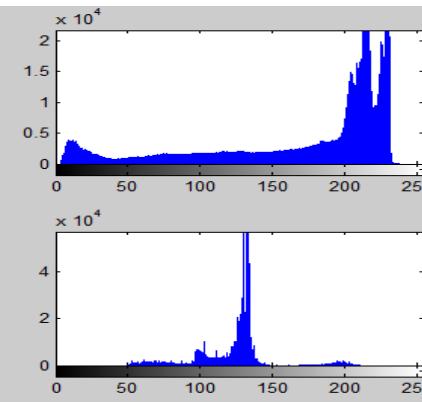
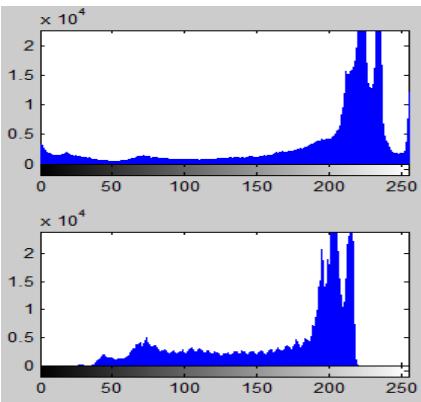


$$\begin{cases} Y &= 0.2989 \times R + 0.5866 \times G + 0.1145 \times B \\ Cb &= -0.1688 \times R - 0.3312 \times G + 0.5000 \times B \\ Cr &= 0.5000 \times R - 0.4184 \times G - 0.0816 \times B \end{cases}$$

# RGB vs YCbCr

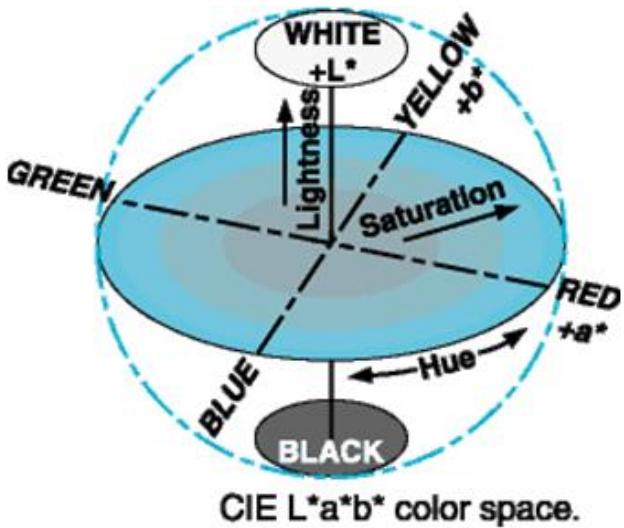




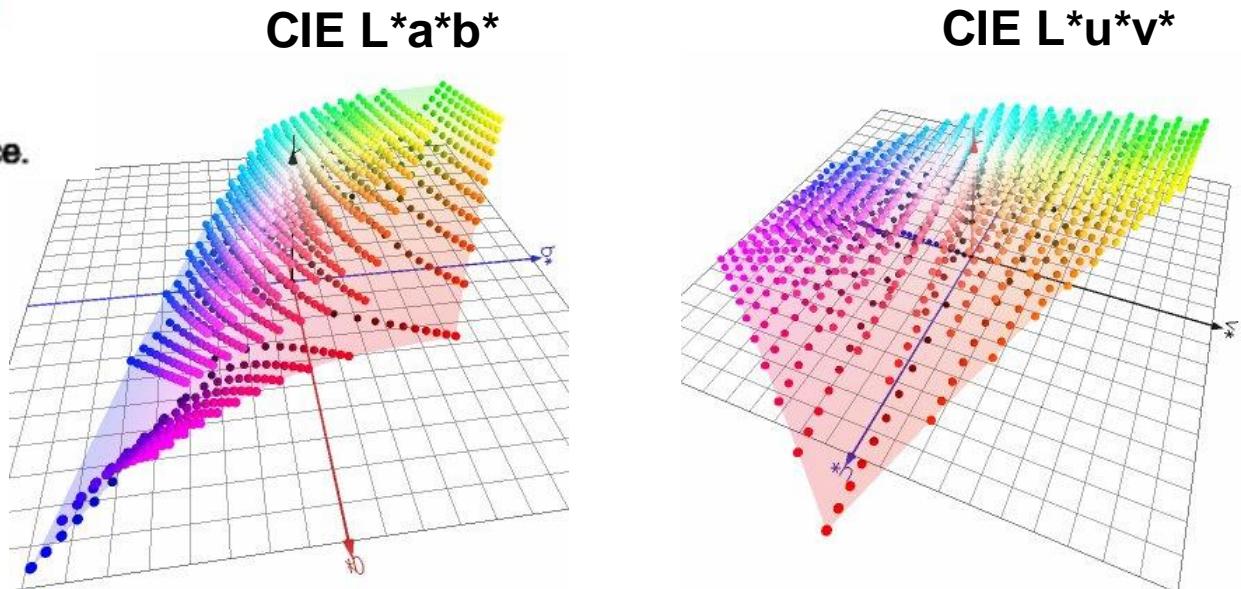


# Spatii de culoare - transformari neliniare

- Transformarea **CIE L\*a\*b\*** (similar – CIE L\*u\*v\*):
  - Conduce la un spatiu uniform perceptual

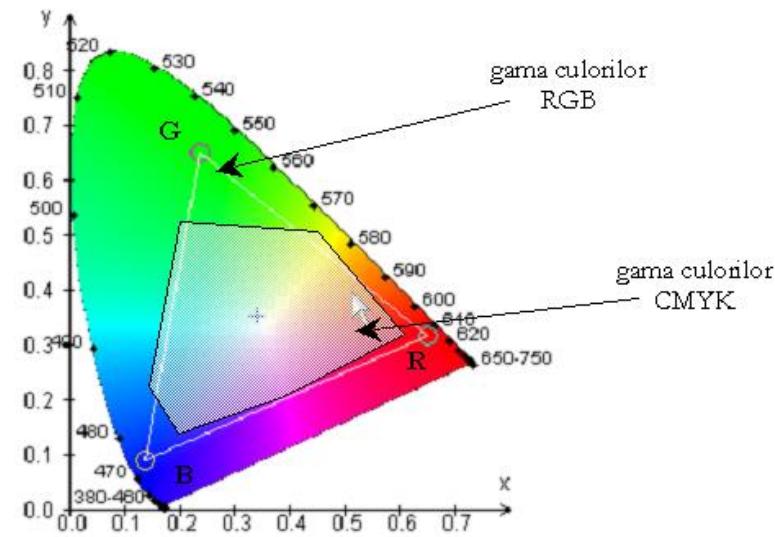
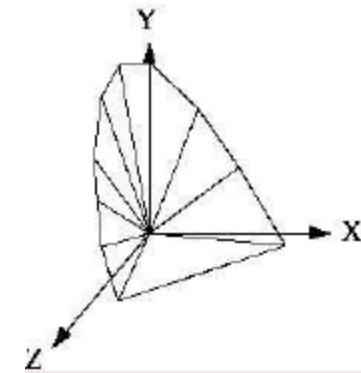


- Luminance: L; Chrominance: a,b
  - a - ranges from green to red,
  - b - ranges from blue to yellow



# CIE Colour Model

- Human focussed Colour Models:
  - In 1931, the CIE defined three standard primaries ( $X$ ,  $Y$ ,  $Z$ )
    - The  $Y$  primary was intentionally chosen to be identical to the luminous-efficiency function of human eyes (Perceptual Model).
  - All visible colours are in a horseshoe shaped cone in the  $X$ - $Y$ - $Z$  space. Consider the plane  $X+Y+Z=1$  and project it onto the  $X$ - $Y$  plane, we get the CIE chromaticity diagram.
  - The edges represent the pure colours
  - White is at the dot
  - Vector-based colour model:
    - when added, any two colours (points on the CIE diagram) produce a point on the line between them.



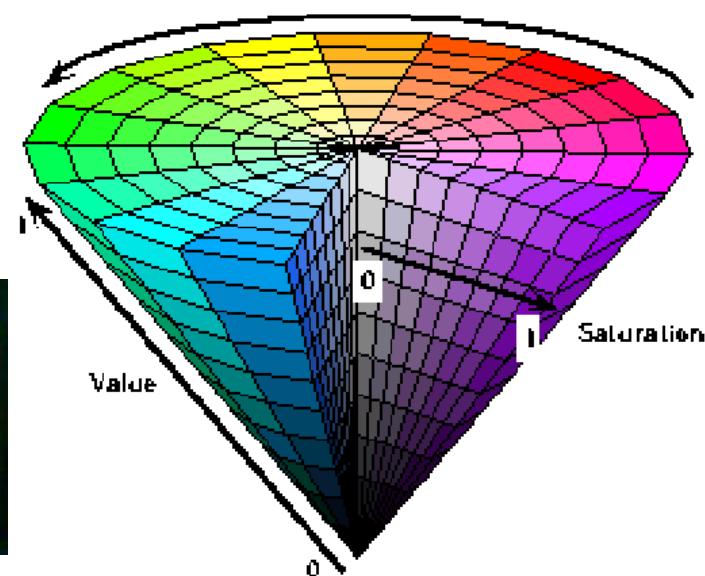
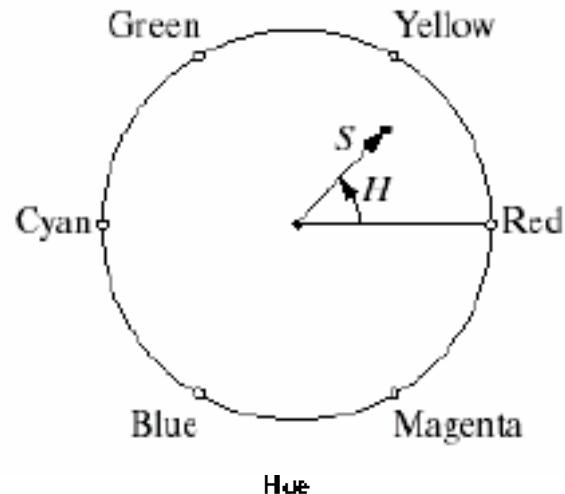
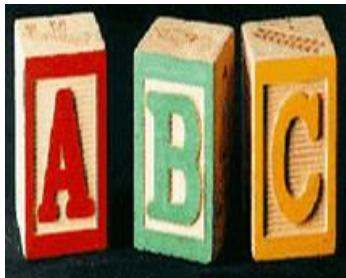
# Atributele perceptuale ale culorii

- **Atributele perceptuale ale culorii**

- atributele folosite de catre sistemul vizual uman pentru descrierea unei culori in procesul de perceptie si distinctie a culorii.

- **Sunt 3 atribute perceptuale ale culorii:**

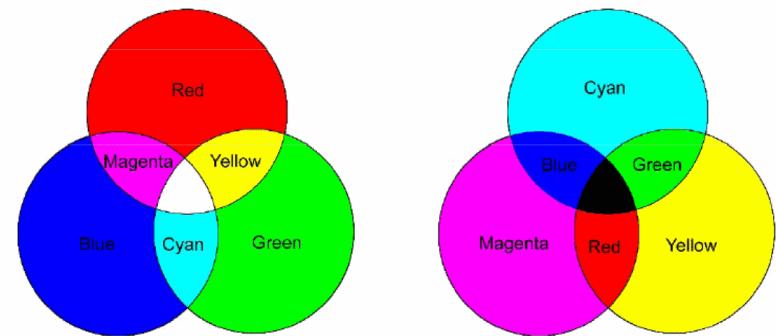
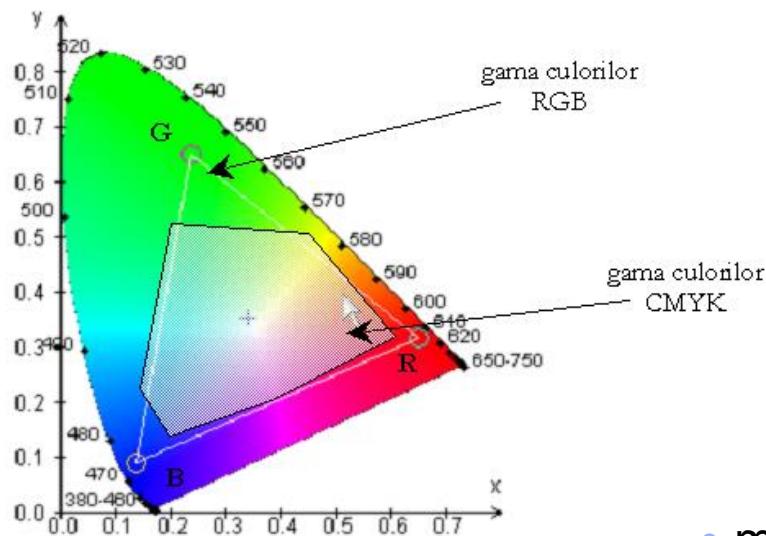
- Nuanța - culoarea de bază a unui obiect (familia de culori din care face parte) - primul criteriu prin care se deosebesc culorile
- Saturația (Puritatea)
- Luminanța (Strălucirea)



# Spațiul de culoare – CMYK

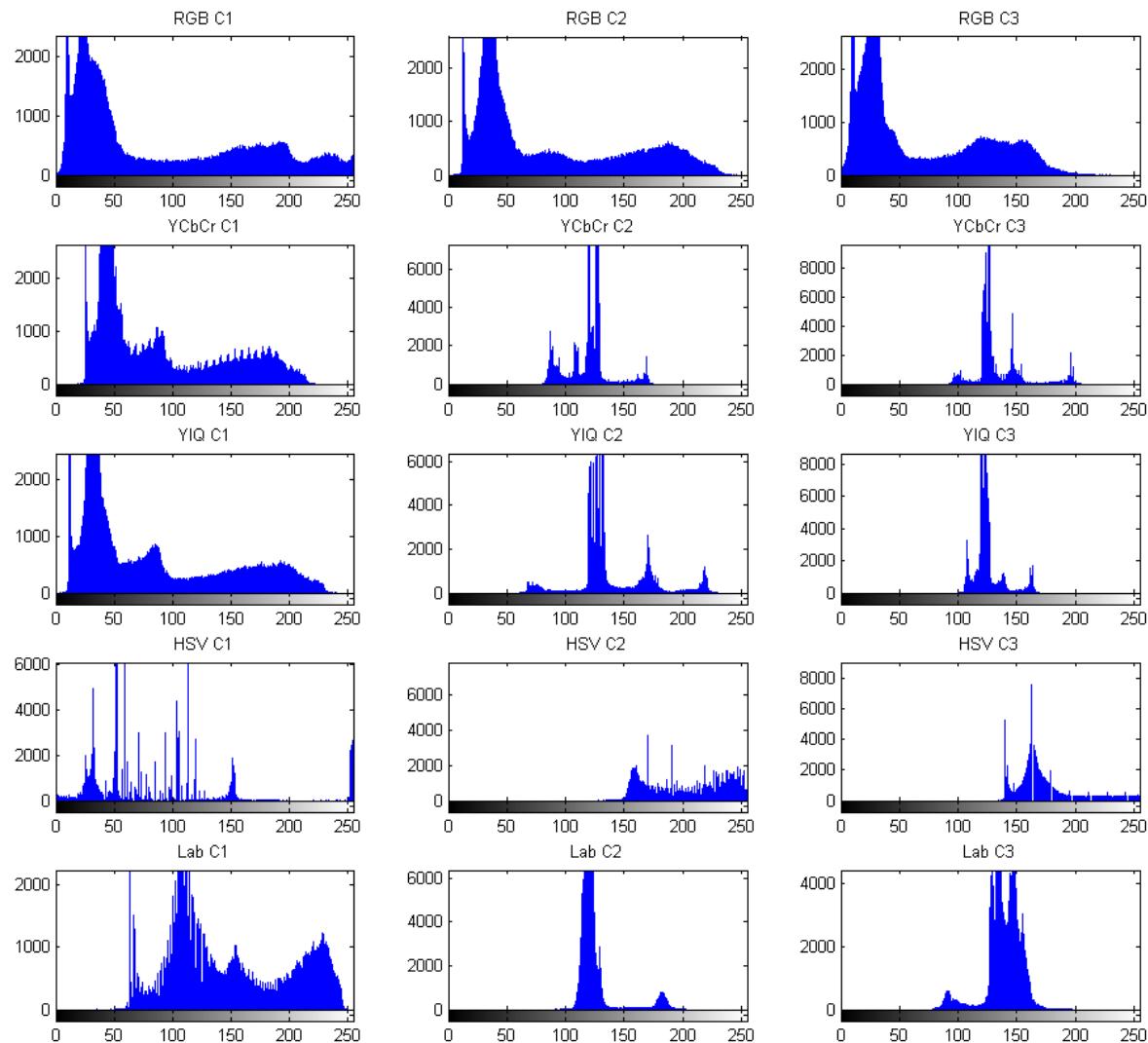
- culori complementare - descompunere din lumina albă
  - roșu – turcuaz (C – Cyan)
  - verde – mov (M – Magenta)
  - albastru – galben (Y – Yellow)
  - completare culoarea NEAGRĂ – BLACK

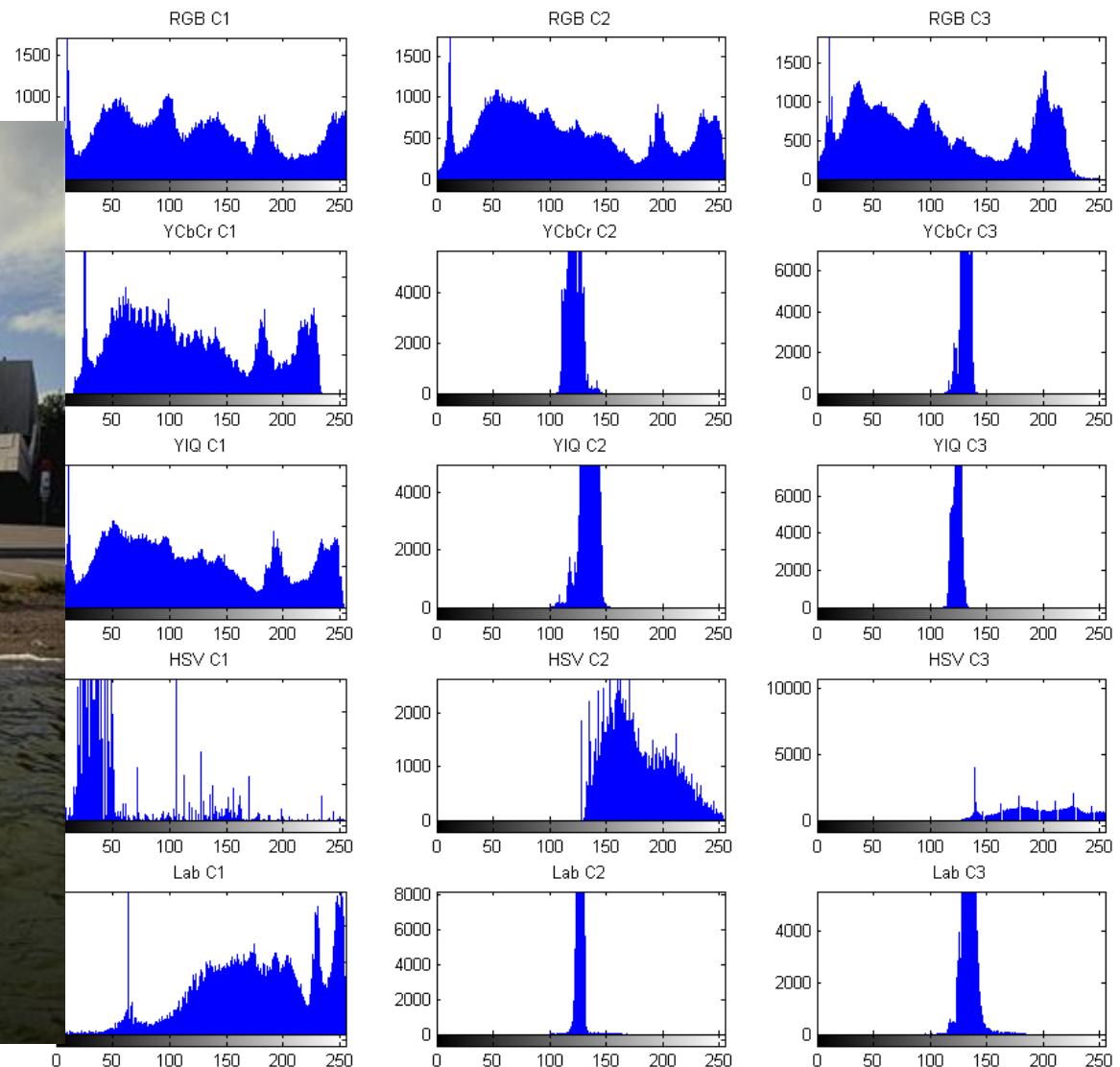
$$C = G + B = W - R$$
$$M = R + B = W - G$$
$$Y = G + R = W - B$$

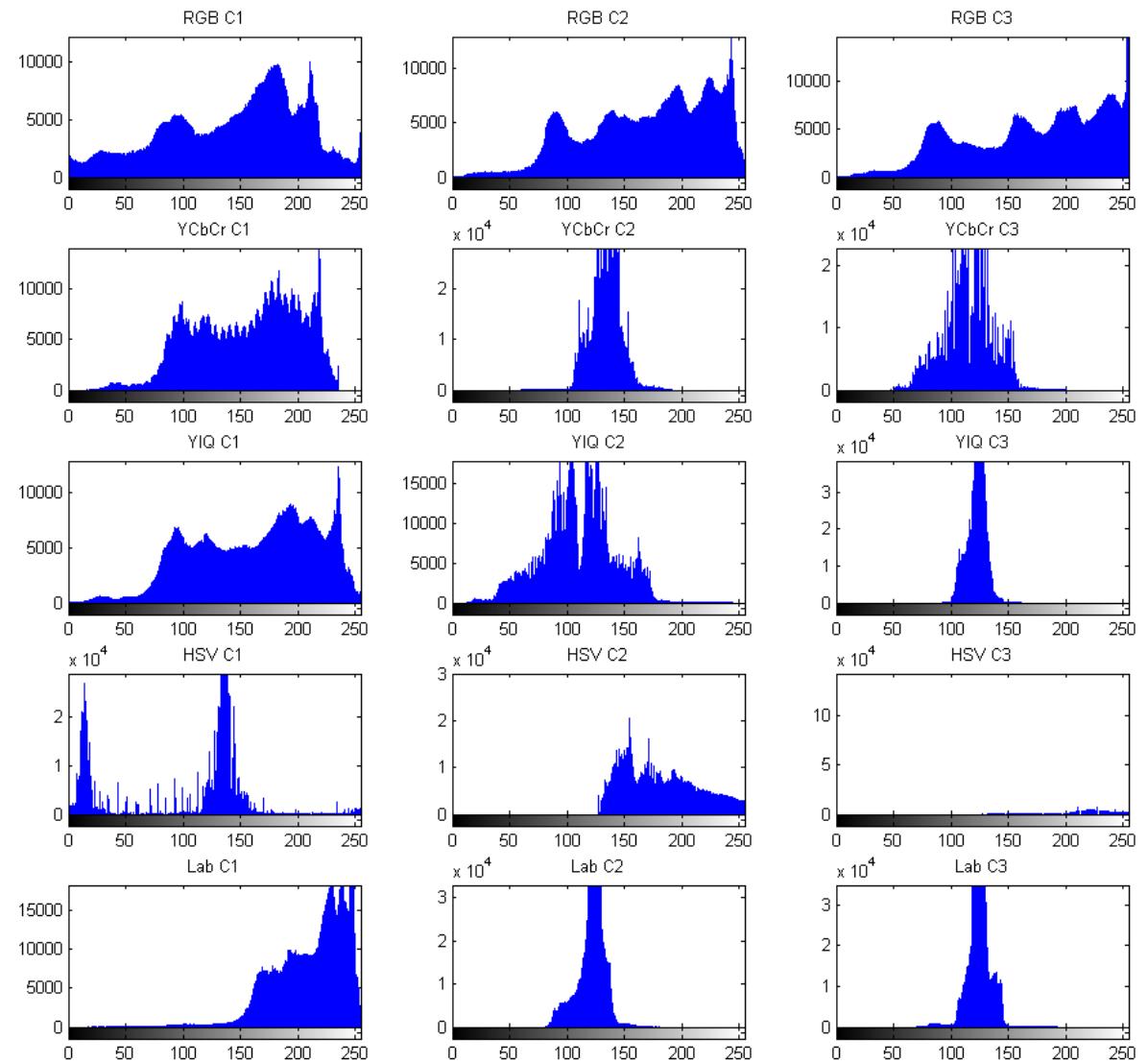


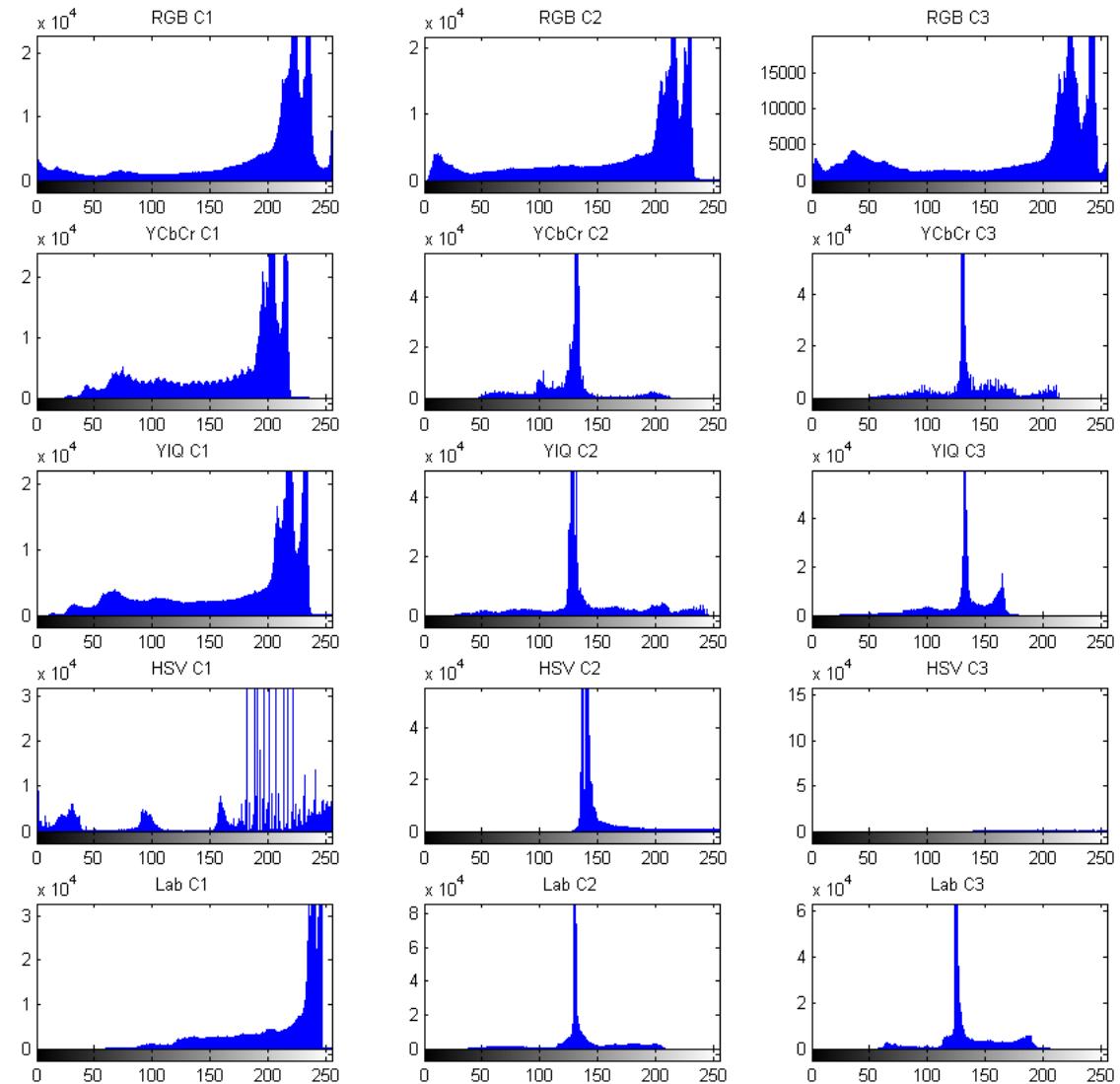
- mai apropiat de formarea culorii în realitate
- CMYK - culori de procesare pentru imprimare

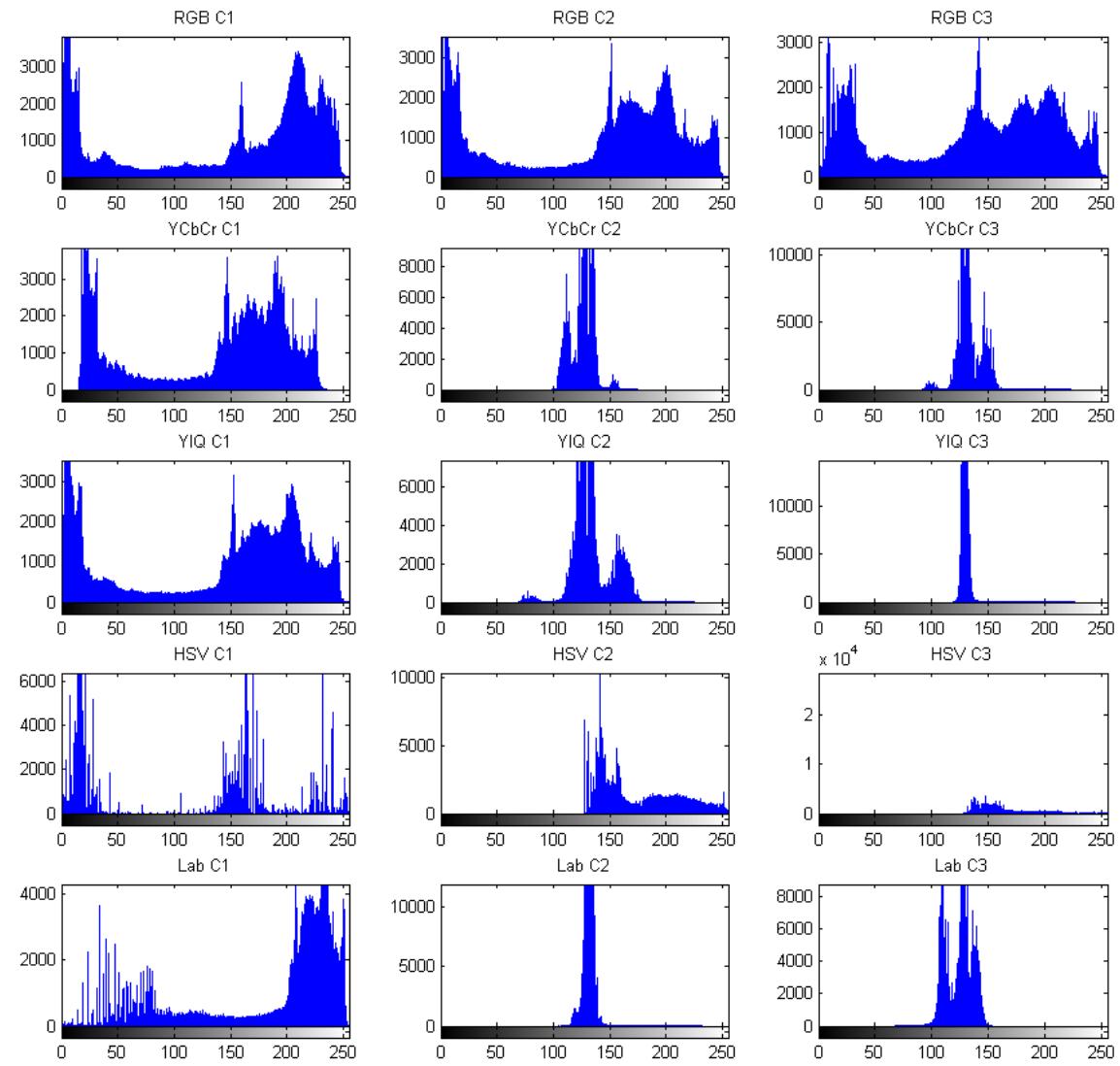
# Exemplificare Spații de Culoare











# Summary of Colour

- Colour images are encoded as triplets of values.
- Three common systems of encoding in video are
  - RGB, YIQ, YUV, and YCrCb.
  - besides the hardware-oriented colour models (i.e., RGB, CMY, YIQ, YUV),
    - HSB (Hue, Saturation, and Brightness, e.g., used in Photoshop) and
    - HLS (Hue, Lightness, and Saturation) are also commonly used.
- YUV/ YCrCb/ YIQ uses properties of the human eye to prioritise information
  - Y is the black and white (luminance) component,
  - U and V are the colour (chrominance) components
- YUV/YCrCb is a standard for digital video that specifies image size, and decimates the chrominance images (for 4:2:2 video)

# Types of Colour Video Signals

- **Component video** - each primary is sent as a separate video signal.
  - The primaries can either be RGB or a luminance-chrominance transformation of them (e.g., YIQ, YUV).
  - Best colour reproduction
  - Requires more bandwidth and good synchronisation of the three components
- **Composite video**
  - Colour (chrominance) and luminance signals are mixed into a single carrier wave.
  - Some interference between the two signals is inevitable.
- **S-Video** (Separated video, e.g., in S-VHS)
  - A compromise between component analog video and the composite video.
  - It uses two lines, one for luminance and another for composite chrominance signal.

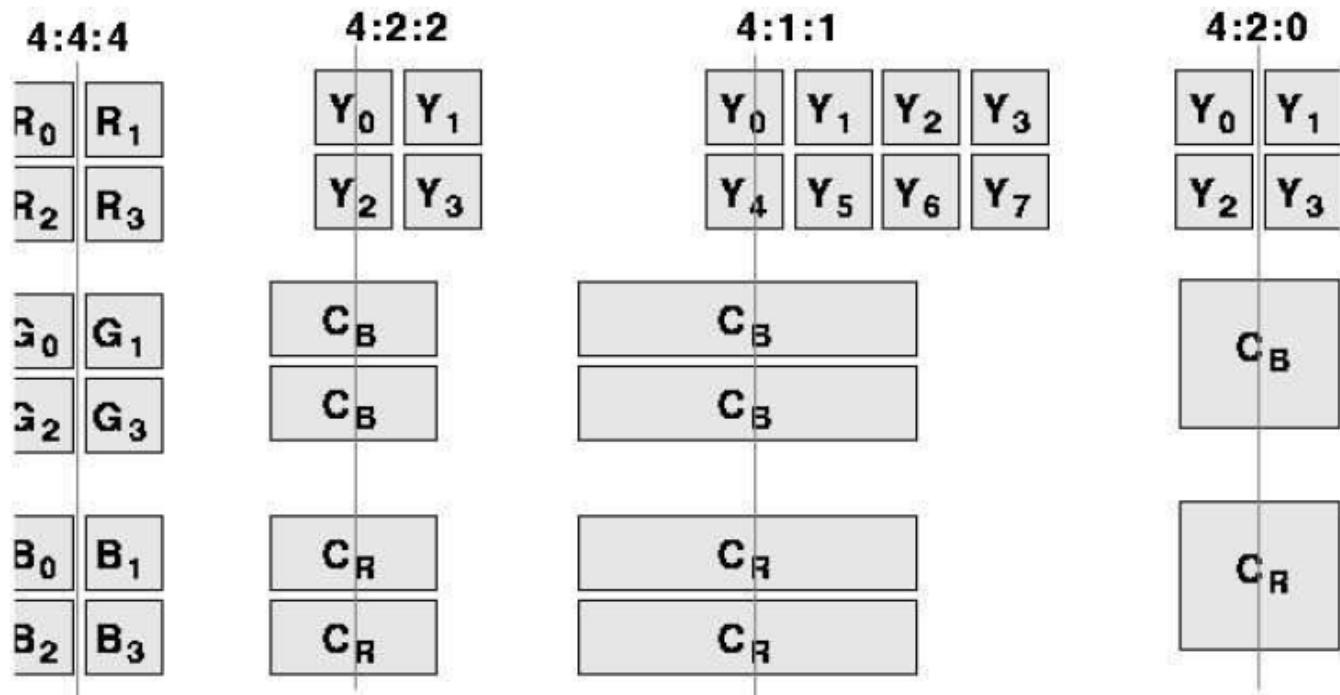
# Chroma Subsampling

- A method that stores an image's (or video frame's) colour information at lower resolution than its intensity information.
  - Main Application: COMPRESSION
  - Used in JPEG Image and MPEG Video Compression
    - One (of two) primary Lossy sources.
- Exploit traits of Human visual system (HVS)
  - HVS more sensitive to variations in brightness than colour.
  - So devote more bandwidth to Y than the color difference components Cr/I and Cb/Q.
    - HVS is less sensitive to the position and motion of color than luminance
    - Bandwidth can be optimised by storing more luminance detail than color detail.
  - Reduction results in almost no perceivable visual difference.

# How to Chroma Subsample?

- Operate on color difference components - the signal is divided into:
  - Luma (Y): the intensity component and
  - Chroma: two color difference components which we subsample in some way to reduce its bandwidth
- How to subsample for chrominance?
  - The subsampling scheme is commonly expressed as a three part ratio (e.g. 4:2:2)
- Chroma Subsample 3 Part Ratio - Each part is respectively:
  1. Luma (Y) or Red (R): Horizontal sampling reference (originally, as a multiple of 3.579 MHz in the NTSC analog television system | rounded to 4)
  2. Cr/I/G: Horizontal factor (relative to first digit)
  3. Cb/Q/B: Horizontal factor (relative to first digit), except when zero.
    - Zero indicates that Cb (Q/B) horizontal factor is equal to second digit, and,
      - Both Cr (I/G) and Cb (Qb) are subsampled 2:1 vertically.

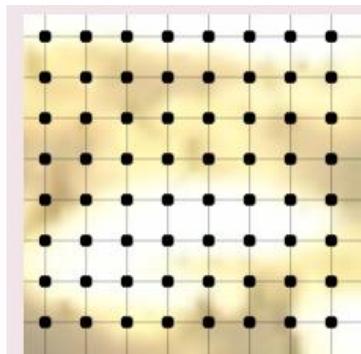
# Chroma Subsampling Examples



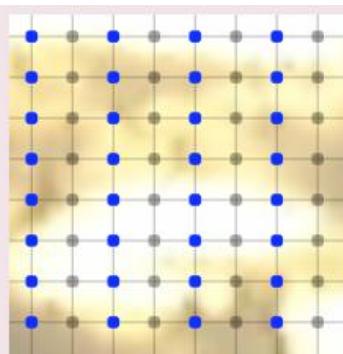
- 4:4:4 — no subsampling in any band — equal ratios.
- 4:2:2 → Two chroma components are sampled at half the sample rate of luma, horizontal chroma resolution halved.
- 4:1:1 → Horizontally subsampled by a factor of 4.
- 4:2:0 → Subsampled by a factor of 2 in both the horizontal and vertical axes

# Chroma Subsampling: How to Compute?

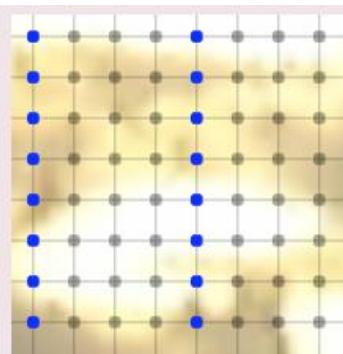
- Simple Image sub-sampling:
    - Simply different frequency sampling of digitised signal
    - Digital Subsampling: For 4:4:4, 4:2:2 and 4:1:1
- Perform 2x2 (or 1x2, or 1x4) chroma subsampling
- Subsample horizontal and, where applicable, vertical directions
  - i.e. Choose every second, fourth pixel value.



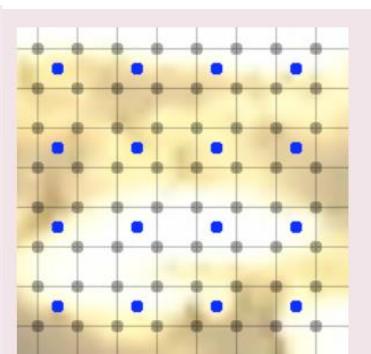
4:4:4



4:2:2



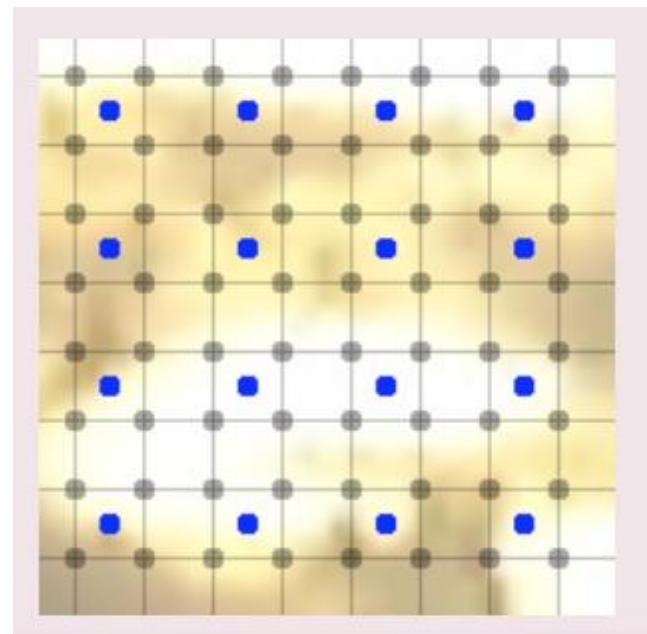
4:1:1



4:2:0

# Chroma Subsampling: How to Compute?

- 4:2:0 Subsampling:
  - For 4:2:0, Cr and Cb are effectively centred vertically halfway between image rows.:
  - Break the image into 2x2 pixel blocks and
  - Stores the average color information for each 2x2 pixel group.



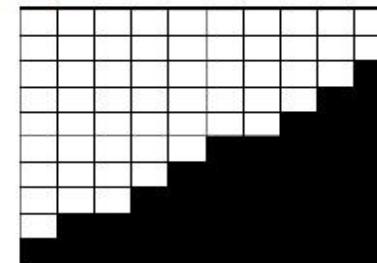
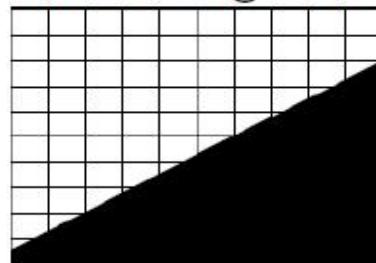
- This sampling process introduces two kinds of errors:
  - A minor problem is that colour is typically stored at only half the horizontal and vertical resolution as the original image - subsampling.
    - This is not a real problem:
      - Recall: The human eye has lower resolving power for colour than for intensity.
      - Nearly all digital cameras have lower resolution for colour than for intensity, so there is no high resolution colour information present in digital camera images.

# Chroma Subsampling Errors - Integer Rounding Errors

- Another issue: The subsampling process demands two conversions of the image:
  - From the original RGB representation to an intensity+colour (YIQ/YUV) representation , and
  - Then back again (YIQ/YUV {>} RGB) when the image is displayed.
  - Conversion is done in integer arithmetic – some round-off error is introduced.
  - This is a much smaller effect,
  - But (slightly) affects the colour of (typically) one or two percent of the pixels in an image.
- Do not compress/recompress videos too often:
  - edit the original!

# Aliasing in Images

- **Stair-stepping:** Stepped or jagged edges of angled lines, e.g., at the slanted edges of letters.



- **Image Zooming:** changing resolution or not acquiring image in adequate resolution, e.g. digital zoom on cameras, digital scanning.  
(see [zoom\\_alias.m](#))



**Explanation:** Simply Application of Nyquist's Sampling Theorem:  
Zooming in by a factor  $n$  divides the sample resolution by  $n$

# Factorul de scalare

---

- de obicei - afişarea pe un dispozitiv cu factorul de scalare 1
- Factor de scalare supraunitar
  - rezultă supraeşantionare
  - pierderea calităţii
- Factor de scalare subunitar
  - subeşantionare
  - unii pixeli lipsesc/sunt eliminati
  - pot dispara detalii
- Trebuie gasite tehnici adecvate
  - pentru supraeşantionare / subeşantionare