



Universitatea "Politehnica" din București  
Facultatea de Electronică, Telecomunicații și  
Tehnologia Informației



## Tehnici Avansate de Prelucrarea și Analiza Imaginilor

### Curs 4 – Îmbunătățirea imaginilor prin operații punctuale

Ș.I. Bogdan IONESCU  
Prof. Constantin VERTAN  
Conf. Mihai CIUC

Master SIVA - Sisteme Inteligente și Vedere Artificială

2010-2011

## Plan Curs 4 – Operații punctuale

- 4.1. Operații punctuale pentru imagini cu niveluri de gri
- 4.2. Histograma și operațiile punctuale
- 4.3. Modificarea paletei de culoare a unei imagini

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU

1

## 4.1. Operații punctuale pentru imagini cu niveluri de gri

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU

2

### Tipuri de operații de prelucrare a imaginilor

În funcție de numărul de pixeli din imaginea inițială folosiți pentru calculul valorii unui pixel din imaginea rezultată (modificată), există:

- **operații punctuale:** fiecare pixel din imaginea modificată depinde doar de pixelul omolog din imaginea inițială (point-to-point).
- **operații de vecinătate:** fiecare pixel din imaginea modificată depinde de un anumit număr (ex. o fereastră) de pixeli din imaginea inițială, situați în vecinătatea acestuia.
- **operații integrale:** fiecare pixel din imaginea modificată depinde de toți pixelii din imaginea inițială.

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU

3

### Operații punctuale pentru niveluri de gri

- nivelul de gri al unui pixel din imaginea modificată depinde doar de nivelul de gri al pixelului omolog din imaginea inițială.

- fiecare pixel din imaginea modificată corespunde pixelului de aceeași coordonate din imaginea inițială.

↳ nu modifică relațiile spațiale din imaginea inițială.

- în general acest tip de operații sunt numite: *îmbunătățire de contrast, întindere de contrast sau transformarea nivelului de gri.*

$$B(x, y) = f(A(x, y))$$

unde A = imaginea inițială  
și B = imaginea modificată

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU

4

### Operații punctuale pentru niveluri de gri

$$B(x, y) = f(A(x, y))$$

unde A = imaginea inițială  
și B = imaginea modificată

- transformarea este specificată integral de funcția  $f(D)$  (D = nivel de gri), ce indică asocierea dintre nivelurile de gri de ieșire și cele de intrare (mapping).

↳ funcție GST = Gray-Scale Transformation function

- proprietăți  $f(D)$ :

- ↳ trebuie să păstreze gama admisibilă de valori ale imaginii  
 $\forall D \in [0; L-1] \Rightarrow f(D) \in [0; L-1]$  (L=număr niveluri de gri)
- ↳ funcția  $f()$  trebuie să fie monotonă pentru a păstra ordinea nivelurilor de gri ( $0 < 1 < \dots < L-1$ )

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU

5

## Operații punctuale pentru niveluri de gri

### aplicații uzuale:

- ↳ **calibrare fotometrică:** în anumite aplicații se dorește ca nivelurile de gri să corespundă unor proprietăți fizice (ex. densitate optică, intensitate luminoasă), astfel se poate corecta neliniaritatea senzorilor (ex. lumină neliniară, corecție: creștere uniformă).
- ↳ **îmbunătățire contrast** (contrast stretching, contrast enhancement): se poate extinde plaja de niveluri de gri ocupată de anumite zone de interes ale imaginii.
- ↳ **calibrare afișaj:** multe dispozitive de afișare nu au o dependență liniară între nivelul de gri al imaginii digitale și punctul de pe dispozitivul de afișare. De asemenea, acestea pot avea o gamă de niveluri de gri preferențială, astfel anumite detalii să nu fie vizibile (extremele) → corecție

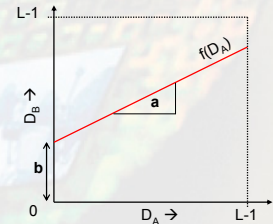
## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare

- nivelul de gri de ieșire este o funcție liniară a nivelului de gri de intrare.

$$D_B = f(D_A) = a \cdot D_A + b$$

unde:  $D_B$  este nivelul de gri de ieșire  
 $D_A$  este nivelul de gri de intrare



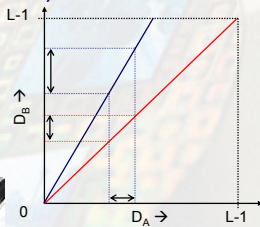
## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare (continuare)

$$D_B = f(D_A) = a \cdot D_A + b$$

### cazuri particulare:

- ↳  $a=1, b=0 \rightarrow$  identitate
- ↳  $a>1 \rightarrow$  mărire contrast



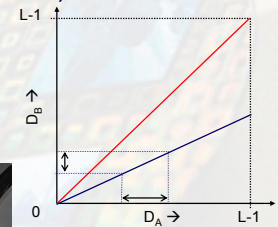
## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare (continuare)

$$D_B = f(D_A) = a \cdot D_A + b$$

### cazuri particulare:

- ↳  $a<1 \rightarrow$  diminuare contrast



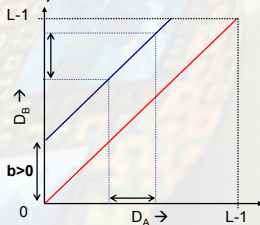
## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare (continuare)

$$D_B = f(D_A) = a \cdot D_A + b$$

### cazuri particulare:

- ↳  $a=1, b \neq 0 \rightarrow$  deplasare niveluri de gri spre max sau spre min



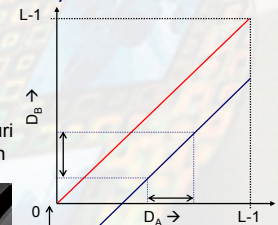
## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare (continuare)

$$D_B = f(D_A) = a \cdot D_A + b$$

### cazuri particulare:

- ↳  $a=1, b \neq 0 \rightarrow$  deplasare niveluri de gri spre max sau spre min



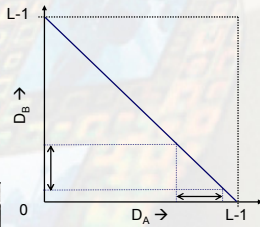
## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare (continuare)

$$D_B = f(D_A) = a \cdot D_A + b$$

cazuri particulare:

↳  $a < 0 \rightarrow$  zonele întunecate devin luminoase, și invers



## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare (continuare)

- subiectiv, în practică plaja nivelurilor de gri este cuantificată perceptual în 3 intervale semnificative:

- niveluri de gri deschise (valori apropiate de alb=255)
- niveluri de gri închise (valori apropiate de negru=0)
- niveluri de gri medii (~ plaja de mijloc, nici-nici)

idee: transformarea punctuală să se facă nu în același fel pentru toate nivelurile de gri, ci mai degrabă *liniară pe porțiuni*.

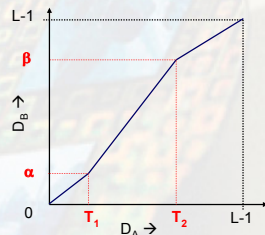
→ în acest fel pot modifica doar anumite plaje de niveluri de gri.

## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare pe porțiuni

$$D_B = f(D_A) =$$

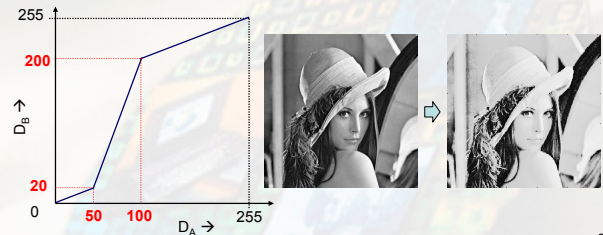
$$\begin{cases} \frac{\alpha}{T_1} \cdot D_A & 0 \leq D_A < T_1 \\ \alpha + \frac{\beta - \alpha}{T_2 - T_1} \cdot (D_A - T_1) & T_1 \leq D_A < T_2 \\ \beta + \frac{L-1-\beta}{L-1-T_2} \cdot (D_A - T_2) & T_2 \leq D_A < L \end{cases}$$



-parametrii de control sunt punctele  $(T_1, \alpha)$  și  $(T_2, \beta)$  care împreună cu punctele  $(0,0)$  și  $(L-1, L-1)$  definesc trei segmente de dreaptă.

## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare pe porțiuni (continuare)



se aplică regulile anterioare, doar că de această dată pe porțiuni:

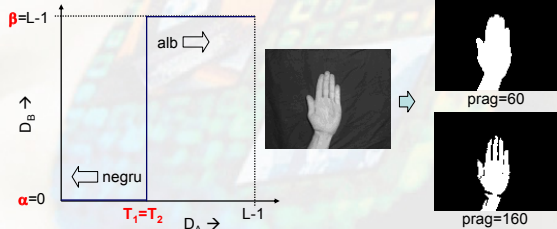
- $[0;50]$  compactat la  $[0;20] \rightarrow$  scade contrastul pentru negru
- $[50;100]$  extins la  $[20;200] \rightarrow$  crește contrastul pentru mediu
- $[100;255]$  compactat la  $[200;255] \rightarrow$  scade contrastul pentru alb

## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare pe porțiuni (continuare)

cazuri particulare:

↳ binarizarea imaginii (thresholding):  $T_1=T_2$  și  $\alpha=0$ ,  $\beta=L-1$  contrast maxim

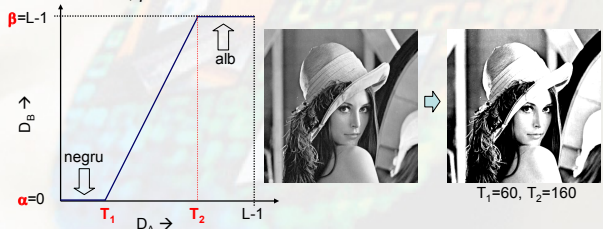


## Operații punctuale pentru niveluri de gri

### ↳ operații punctuale liniare pe porțiuni (continuare)

cazuri particulare:

↳ întinderea maximă a contrastului (contrast stretching):  $\alpha=0$ ,  $\beta=L-1$



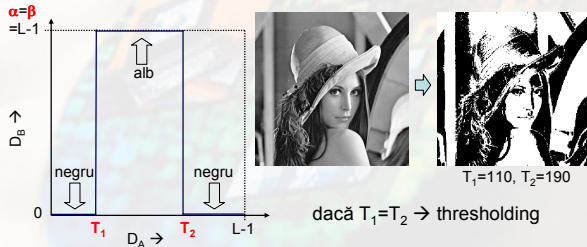


### Operații punctuale pentru niveluri de gri

#### ↳ operații punctuale liniare pe porțiuni (continue)

cazuri particulare:

#### ↳ transformarea de decupare (slicing): $\alpha=\beta=L-1$ , $T_1 \neq T_2$



### Operații punctuale pentru niveluri de gri

#### ↳ operații punctuale liniare pe porțiuni (continue)

**Observație:** se poate folosi un număr arbitrar (ex. >3) de segmente de dreaptă și astfel de puncte  $(T, \alpha)$ .

#### ↳ operații punctuale neliniare

- nivelul de gri de ieșire este o funcție neliniară a nivelului de gri de intrare.

#### ↳ compandarea domeniului cu o curbă logaritmică:

$$D_B = f(D_A) = \frac{L-1}{\log L} \cdot \log(1 + D_A)$$

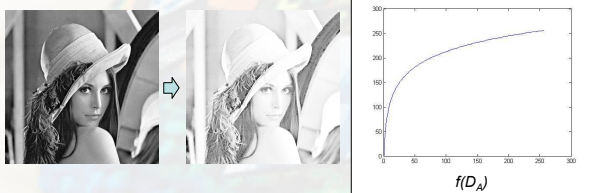
= contrastul variază neuniform de-a lungul nivelurilor de gri, mărindu-se la capătul inferior (negru) și diminuându-se la capătul superior (alb)

### Operații punctuale pentru niveluri de gri

#### ↳ operații punctuale neliniare (continue)

#### ↳ compandarea domeniului cu o curbă logaritmică:

$$D_B = f(D_A) = \frac{L-1}{\log L} \cdot \log(1 + D_A)$$



### Operații punctuale pentru niveluri de gri

#### ↳ operații punctuale neliniare

#### ↳ expandarea domeniului cu o curbă exponențială:

$$D_B = f(D_A) = (L-1) \frac{e^{D_A} - 1}{e^{L-1} - 1} \text{ sau } D_B = f(D_A) = L^{D_A/L} - 1$$

= contrastul variază neuniform de-a lungul nivelurilor de gri, mărindu-se la capătul superior (alb) și diminuându-se la capătul inferior (negru)

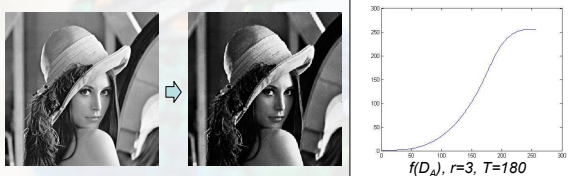


### Operații punctuale pentru niveluri de gri

#### ↳ operații punctuale neliniare pe regiuni

$$D_B = f(D_A) = \begin{cases} T \cdot \left(\frac{D_A}{T}\right)^r & D_A \in [0; T] \\ L-1 - (L-1-T) \cdot \left(\frac{L-1-D_A}{L-1-T}\right)^r & D_A \in [T; L-1] \end{cases}$$

unde  $(T, T)$  este un punct fix iar  $r$  este un parametru de reglaj.



### Implementare operații punctuale pentru niveluri de gri

operațiile punctuale se pot implementa practic în două variante:

↳ imaginea este parcursă pixel cu pixel (în general, de la stânga la dreapta și de sus în jos) iar nivelul de gri este modificat aplicând  $f(A(x,y))$  ( $A()$  reprezintă imaginea).

→ redundantă, aceleași valori sunt calculate de mai multe ori

↳ considerând nivelurile de gri o paletă de culoare, putem calcula noile valori pe baza funcției  $f(D_A)$ , cu  $D_A=0, \dots, L-1$  și apoi aplicăm noua paletă imaginii inițiale

→ mai eficient, dezavantaj folosire paletă de culoare

## 4.2. Histograma și operațiile punctuale

### Histograma unei imagini

o măsură statistică ce contabilizează numărul de apariții în imagine a fiecărui nivel de gri (sau culori).

$$h(D) = \frac{1}{X \cdot Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} \delta(A(x, y) - D) \rightarrow h(D) = \frac{N_{\text{pixel}=D}}{N_{\text{total}}}$$

unde  $D$  este un nivel de gri (sau culoare),  $D=0, \dots, L-1$  (sau ia valori în paleta de culori considerată),  $X, Y$  dimensiunea imaginii,  $A()$  este funcția imagine, iar:

$$\delta(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases}$$

statistic are sens de *densitate de probabilitate discretă*, probabilitatea ca un pixel să aibă valoarea  $D$ .

### Histograma unei imagini

dacă  $h(D)$  este densitate de probabilitate  $\rightarrow$  se poate exprima și ca funcție de repartiție discretă a variabilei aleatoare pixel:

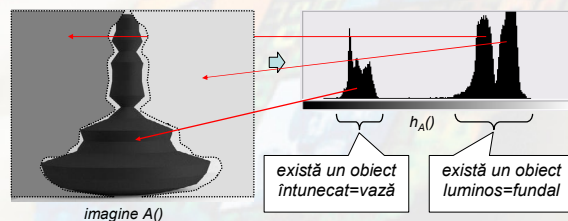
$$H(D) = \sum_{d=0}^D h(d) \rightarrow \text{histograma cumulată}$$

unde  $d, D$  este un nivel de gri (sau culoare),  $d=0, \dots, D$ , iar  $h()$  este histograma imaginii  $A()$ ,

statistic reprezintă probabilitatea ca un pixel să aibă o valoare inferioară lui  $D$  (relația de ordine între valori/culori este dată de relația de ordine din paleta considerată)

### Histograma unei imagini

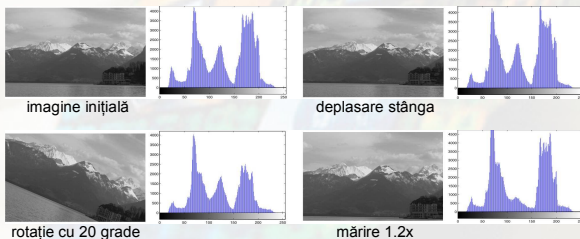
o modalitate eficientă de caracterizare a distribuției globale de niveluri de gri din imagine (estimare număr obiecte, imagine întunecată/luminoasă, etc.).



### Histograma unei imagini

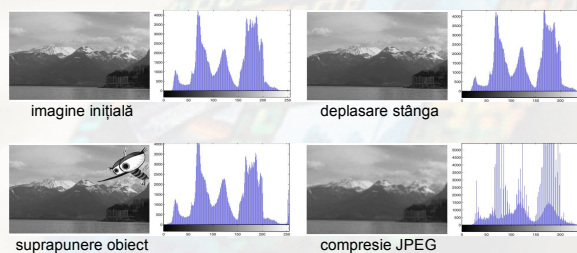
**avantaj principal:** histograma prezintă *invarianță totală sau parțială* la anumite transformări geometrice ale imaginii (ex. rotații, translații, mărire imagine, compresie, ocluzie obiecte, etc.)

$\rightarrow$  în timp ce complexitatea de calcul este redusă.



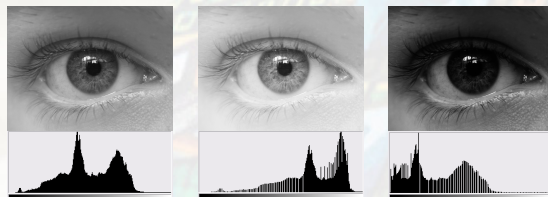
### Histograma unei imagini

**avantaj principal:** histograma prezintă *invarianță totală sau parțială* la anumite transformări geometrice ale imaginii.



## Histograma unei imagini

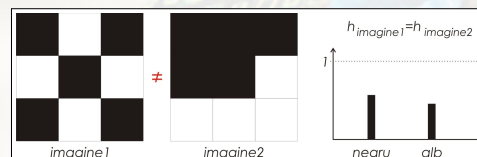
limitări: → foarte sensibilă la variații importante ale intensității luminoase.



> **ameliorare** (cazul imaginilor color): histograma este calculată doar pentru componentele de cromaticitate (ex. YCbCr, Lab) ce nu sunt afectate de variații ale intensității.

## Histograma unei imagini

limitări: → nu ține cont de informația spațială din imagine, și anume de poziția spațială a pixelilor



→ histograme identice pot aparține unor imagini complet diferite,

→ doi pixeli de aceeași valoare nu au neapărat aceeași semnificație, dimpotrivă pot reprezenta informații spațiale diferite (contur vs. zone omogene)

## Histograma unei imagini

limitări: → nu ține cont de informația spațială din imagine, și anume de poziția spațială a pixelilor

> **ameliorare: histograma ponderată**

$$h_p(D) = \frac{1}{X \cdot Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} w(x,y) \cdot \delta(A(x,y) - D)$$

unde  $w(x,y)$  reprezintă ponderea pixelului de coordonate  $x,y$  la valoarea histogramei pentru nivelul/culoarea  $D$ .

$w(x,y)$  este calculat de regulă pentru o vecinătate a pixelului  $(x,y)$  depinzând astfel de informația spațială locală, exemple:

- ~ gradient
- ~ probabilitatea de apariție a nivelului  $D$  în vecinătate

## Histograma unei imagini

limitări: → nu ține cont de informația spațială din imagine, și anume de poziția spațială a pixelilor

> **ameliorare: histograma acumulativă**, imaginea este divizată în mai multe blocuri de pixeli ( $b=0, \dots, N-1$ ), și pentru fiecare regiune se calculează o histogramă clasică,  $h_b()$  →

$$\tilde{h}(D) = \sum_{b=0}^{N-1} f(h_b(D))$$

unde  $f()$  este o funcție neliniară ce evidențiază repartiția geometrică a nivelurilor de gri sau a culorilor.

cu cât sunt acumulate mai multe valori (niveluri de gri/culori) cu atât modulele/valorile histogramei sunt mai diferențiable.

→ eficientă pentru caracterizarea conținutului imaginilor color

## Histograma unei imagini

histograme fuzzy

> în cazul clasic, valoarea unui pixel intervine doar la calculul valorii histogramei în acest punct.

> în cazul fuzzy, se ia în calcul similaritatea dintre nivelurile de gri (culori) prin distribuirea unei anumite funcții de apartenență fuzzy a fiecărui pixel către toate nivelurile de gri (culorile din paletă).

$$h_f(A) = [f_0, f_1, \dots, f_{L-1}] \quad \text{unde} \quad f_i = \sum_{j=1}^N \mu_{ij} \cdot P_j$$

unde:

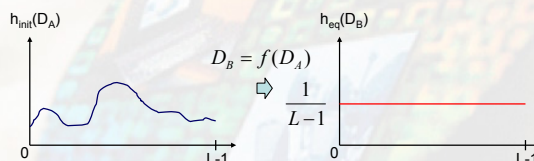
- $i$  = nivelul de gri (sau indicele culorii),
- $j$  = indicele pixelului curent din imagine,  $j=1, \dots, N$ ,  $N$  numărul total de pixeli,
- $P_j$  = probabilitatea de apariție a pixelului  $j$  în imagine,
- $\mu_{ij}$  = valoarea funcției de apartenență a pixelului  $j$  pentru nivelul  $i$

## Histograma și operațiile punctuale

↳ **egalizarea histogramei (histogram equalization)**

are ca scop obținerea unei imagini, și astfel a unei histograme teoretic ideale:

- toată plaja nivelurilor de gri este folosită
- fiecare nivel de gri este echiprobabil



→ histogramă uniformă = avantajoasă pentru alte etape de prelucrare, ex. comparare sau segmentare.

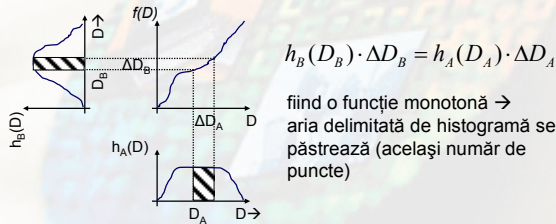


### Histograma și operațiile punctuale

#### ↳ egalizarea histogramei (continuare)

⇒ trebuie găsită funcția  $f(D)$  care transformă imaginea astfel încât fiecare nivel de gri să fie echiprobabil.

>care este relația dintre histograma imaginii transformate printr-o funcție monotonă ( $f, B, h_B()$ ) și histograma imaginii inițiale ( $A, h_A()$ )?



### Histograma și operațiile punctuale

#### ↳ egalizarea histogramei (continuare)

$$h_B(D_B) \cdot \Delta D_B = h_A(D_A) \cdot \Delta D_A$$

$$h_B(D_B) = \frac{h_A(D_A)}{\Delta D_B / \Delta D_A}$$

$$\frac{\Delta D}{D_B = f(D_A)} \text{ suficient de mic} \Rightarrow h_B(D_B) = \frac{h_A(D_A)}{\frac{d}{dD_A} f(D_A)} = \frac{h_A(D_A)}{f'(D_A)}$$

$$\text{histogramă uniformă} \Rightarrow h_B(D) = \frac{1}{L-1} \forall D = 0, \dots, L-1$$

### Histograma și operațiile punctuale

#### ↳ egalizarea histogramei (continuare)

$$f'(D_A) = \frac{h_A(D_A)}{h_B(D_B)} = (L-1) \cdot h_A(D_A)$$

integrăm de la 0 la  $D$ :

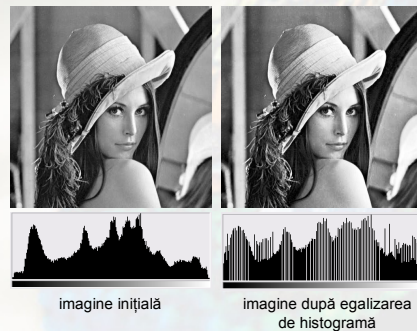
$$f(D) = (L-1) \cdot \int_0^D h_A(u) du = (L-1) \cdot H_A(D)$$

unde  $H_A(D)$  este histograma cumulată (funcția de repartiție a variabilei aleatoare pixel)

funcția de transformare a nivelurilor de gri în vederea egalizării de histogramă **este chiar histograma cumulativă**.

### Histograma și operațiile punctuale

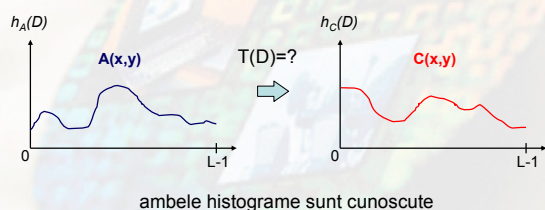
#### ↳ egalizarea histogramei (continuare)



### Histograma și operațiile punctuale

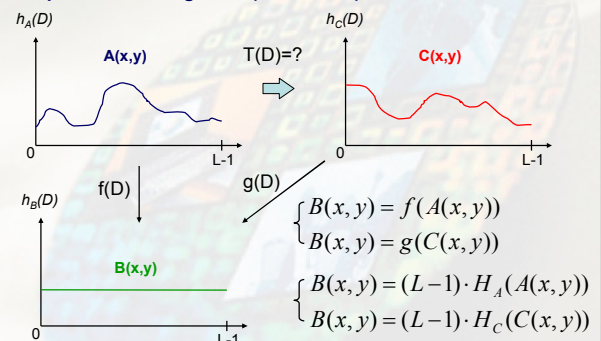
#### ↳ potrivirea histogramei (histogram matching)

se dorește transformarea unei imagini astfel încât histograma imaginii modificate să corespundă unei histograme cunoscute sau unei funcții.



### Histograma și operațiile punctuale

#### ↳ potrivirea histogramei (continuare)



### Histograma și operațiile punctuale

↳ **potrivirea histogramei (continuare)**

$$\begin{cases} B(x, y) = (L-1) \cdot H_A(A(x, y)) \\ B(x, y) = (L-1) \cdot H_C(C(x, y)) \end{cases}$$

$$(L-1) \cdot H_C(C(x, y)) = (L-1) \cdot H_A(A(x, y))$$

$$C(x, y) = H_C^{-1}(H_A(A(x, y)))$$

$$T(D) = H_C^{-1}(H_A(D)) = g^{-1}(f(D))$$

→ potrivirea histogramei este avantajoasă în cazul în care se dorește compararea aceleiași imagini "achiziționate" în condiții diferite de luminozitate.

### Histograma și operațiile punctuale: concluzii

> operațiile punctuale descrise anterior se pot aplica și imaginilor color folosind abordările prezentate în cursul anterior (marginală, transformări colorimetrice, etc.)  
→ nefiind metode vectoriale native, rezultatele pot fi mediocre

> operațiile punctuale asupra histogramei pot fi extinse în cazul imaginilor color dacă se consideră o paletă de culoare fixă (vezi definiția histogramei)

> există bineînțeles și o serie de tehnici vectoriale (egalizarea histogramelor color, decimarea histogramei, explozia histogramei etc.)  
→ chiar dacă sunt vectoriale acestea pot introduce culori false (ordinea culorilor este relativă)

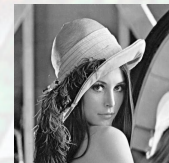
## 4.3. Modificarea paletei de culoare a unei imagini

### Modificarea paletei de culoare

↳ **pseudo-culori**

în anumite aplicații ce implică obținerea de imagini cu niveluri de gri, sau mai general scalare (ex. medicale, satelitare...), se dorește evidențierea anumitor detalii ale imaginii  
→ acest lucru se realizează prin colorarea acestora cu *pseudoculori*.

- **abordarea 1:** se folosește o reprezentare a imaginii indexată (paletă de culoare sau LUT) pentru a facilita schimbarea culorilor (în totalitate sau doar anumite valori).



| Index culoare | R   | G   | B   |
|---------------|-----|-----|-----|
| 0             | 0   | 0   | 0   |
| 1             | 1   | 1   | 1   |
| ...           |     |     |     |
| L-1           | 255 | 255 | 255 |

LUT niveluri de gri

### Modificarea paletei de culoare

↳ **pseudo-culori**

- **abordarea 1:** modificare LUT (paletă de culoare) parțială



imagine inițială



imagine cu pseudo-culori

| Index culoare | R   | G | B |
|---------------|-----|---|---|
| 0             | 0   | 0 | 0 |
| 1             | 1   | 1 | 1 |
| ...           |     |   |   |
| L-1           | 255 | 0 | 0 |

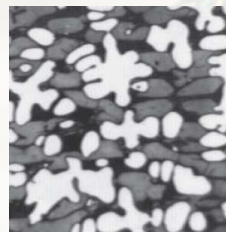
LUT

### Modificarea paletei de culoare

[J.C. Russ 2006]

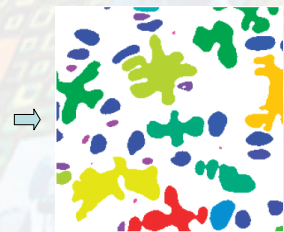
↳ **pseudo-culori**

- **abordarea 1:** modificare LUT (paletă de culoare) parțială



LUT gri

imagine microscopică secțiune metal



pseudocolorare în funcție de dimensiune

LUT modif.



## Modificarea paletei de culoare

[W.K. Pratt 2007]

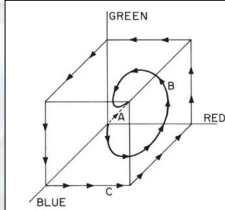
### ↳ pseudo-culori

abordarea 2: modificarea nivelurilor de gri se realizează pe baza unei funcții liniare sau neliniare.

$$\begin{cases} R(x, y) = f_R(A(x, y)) \\ G(x, y) = f_G(A(x, y)) \\ B(x, y) = f_B(A(x, y)) \end{cases} \quad \text{unde } A(x, y) \text{ este imaginea cu niveluri de gri iar } R(), G() \text{ și } B() \text{ sunt componentele de culoare obținute}$$

$$A_{\text{modif}}(x, y) = [R(x, y), G(x, y), B(x, y)]$$

un exemplu de asociere al culorilor:  
calea A = niveluri de gri  
calea B = pornește de la negru și ajunge la alb

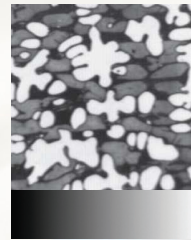


## Modificarea paletei de culoare

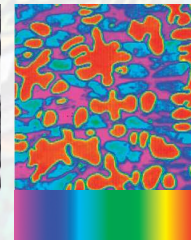
[J.C. Russ 2006]

### ↳ pseudo-culori

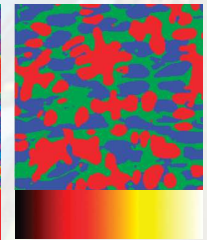
- abordarea 2: modificare LUT (paletă de culoare) totală



imagine microscopică  
secțiune metal



pseudocolorare



pseudocolorare  
după segmentare

## Modificarea paletei de culoare

### ↳ culori false

colorarea falsă a imaginii se aplică imaginilor color (vectoriale) și constă într-o operație punctuală de transformare a componentelor de culoare cu o funcție liniară sau neliniară.

$$\begin{cases} R(x, y) = f_R(A_1, A_2, \dots) \\ G(x, y) = f_G(A_1, A_2, \dots) \\ B(x, y) = f_B(A_1, A_2, \dots) \end{cases} \quad \text{unde } A_1, A_2, \dots \text{ sunt componentele de culoare ale imaginii inițiale (ex. } R, G, B, \text{ multispectrală, etc.) iar } R(), G() \text{ și } B() \text{ sunt componentele de culoare obținute}$$

$$A_{\text{modif}}(x, y) = [R(x, y), G(x, y), B(x, y)]$$

**motivații:** - ca și în cazul pseudo-culorilor, punerea în evidență prin colorarea distinctă a anumitor informații din imagine.

## Modificarea paletei de culoare

### ↳ culori false

**motivații:** - colorarea imaginii în concordanță cu sesibilitatea sistemului vizual uman, ex. ochiul este mai sensibil la verde, sau la schimbări ale luminii albastre, etc. (→ observare mai bună a detaliilor)

- producerea unei reprezentări color naturale a imaginilor multispectrale (ex. imagini ce furnizează valori în afara spectrului vizibil)



IR → culori false [J. Cramer]



optic → culori false



3 benzi → culori false

## Modificarea paletei de culoare

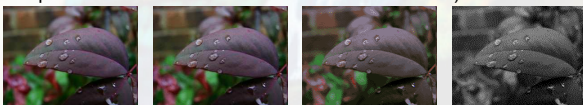
### ↳ cuantizarea culorilor

procesul prin care se reduce numărul de culori folosite de imagine și astfel paleta de culoare a acesteia.

→ modificare cu pierdere de informație

> având la dispoziție  $N$  vectori într-un spațiu  $M$  dimensional, cum facem să selectăm  $K$  vectori,  $K \ll N$ , astfel încât eroarea indusă de cuantizare să fie minimă (cuantizare vectorială, imagini  $M=3$ )

> **motivație:** inițial, impusă de limitările hardware (încă valabile pentru dispozitivele mobile sau ecranele de mici dimensiuni)



true color

256 culori

16 culori

4 culori

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

> **motivație:** recentă, necesară în diversele procese de analiză și prelucrare a imaginilor (reducere complexitate de calcul, facilitare anumite operații ex. histograma color, etc.)

→ compresie, segmentare color, watermarking, indexare,...

în general presupune următoarele etape:

1. analiza, de regulă statistică, a conținutului imaginii,
2. pe baza acesteia, determinarea unei noi paletă de culoare,
3. determinarea unei relații între culorile din imagine și culorile din paleta determinată (mapping)
4. cuantizarea propriu-zisă și construirea noii imaginii.

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

astfel, cuantizarea culorilor presupune:

- determinarea unei paletă de culoare ("palette design", etapa 1-2),
- aplicarea acesteia în imagine ("pixel mapping", etapa 3-4).

clasificare metode existente:

- cuantizarea la o **paletă fixă de culoare** (universală): metode independente de imagine,
  - de regulă *rapide, calitate vizuală în general scăzută* (paleta fixă nu este suficient de vastă pentru a permite substituirea oricărei culori din imaginea originală),
  - calitatea vizuală poate fi totuși redresată prin: cunoașterea domeniului de aplicație, alegerea unei paletă mai vaste, etc.
  - eficiente în cazul comparării imaginilor, datorită faptului că pixelii iau valori în aceeași mulțime (paletă).

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

clasificare metode existente:

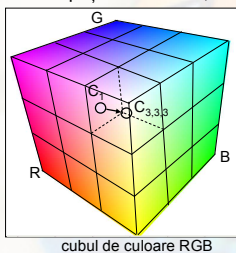
- cuantizarea la o **paletă de culoare adaptivă**: metode dependente de conținutul imaginii,
  - oferă cea mai bună fidelitate vizuală în detrimentul complexității de calcul fiind folosite în aplicații unde calitatea vizuală este esențială (reprezentare imagini, analiză conținut...)
  - iau în calcul conținutul imaginii,
  - fiecare imagine va dispune de o paletă proprie,
  - mult mai studiate în literatura de specialitate decât metodele cu paletă fixă datorită aplicațiilor mult mai vaste.

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

cuantizarea la o paletă fixă

- cea mai simplă abordare constă în cuantizarea uniformă a unui spațiu de culoare, ex. RGB.



cubul de culoare RGB

- fiecare axă de culoare este împărțită în  $N$  intervale egale  
→  $N \times N \times N$  cuburi
- fiecare culoare dintr-un cub este aproximată cu, culoarea centrală

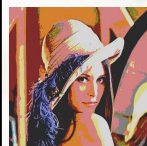
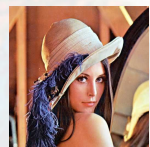
$C_1$  din imaginea inițială este înlocuit cu  $C_{3,3,3}$  în imaginea cuantizată

## Modificarea paletei de culoare

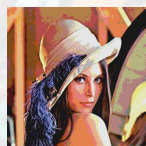
### ↳ cuantizarea culorilor (continuare)

cuantizarea la o paletă fixă

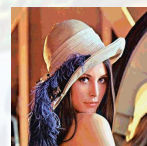
- cuantizarea uniformă RGB (continuare)



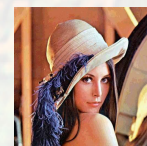
cuantizare RGB  
3x3x3=27



cuantizare RGB  
5x5x5=125



cuantizare RGB  
9x9x9=729



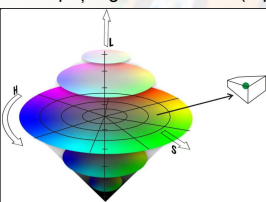
cuantizare RGB  
15x15x15=3375

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

cuantizarea la o paletă fixă

- în spațiul RGB nu există o separare perceptuală a culorilor, idee: să realizăm cuantizarea uniformă a unui spațiu perceptual ex. spațiul geometric HSL (reprezentare cilindrică)



gamutul de culoare HSL

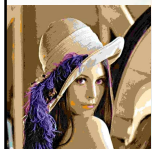
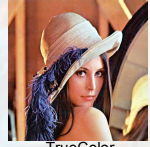
- axele spațiului sunt cuantizate uniform într-un număr diferit de intervale (ochiul uman este mai sensibil la nuanță și luminanță decât la saturație)
- nivelurile de gri sunt adăugate separat deoarece nu reies din cuantizare.

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

cuantizarea la o paletă fixă

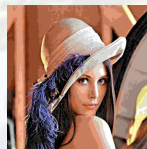
- cuantizarea uniformă a spațiului geometric HSL (continuare)



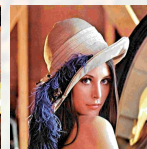
cuantizare HSL  
5x2x5=55



cuantizare HSL  
7x5x7=245



cuantizare HSL  
10x4x10=400



cuantizare HSL  
20x10x20=4000

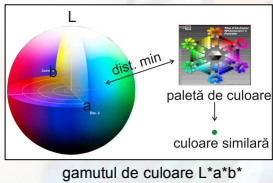


## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

cuantizarea la o paletă fixă

→ altă abordare, stabilirea paletei "a priori". Culorile din imagine pot fi înlocuite pe baza distanței percepțuale dintre culori.



- alegem o paletă de culoare cu o diversitate de culoare suficientă pentru aplicația vizată,
- fiecare culoare din imagine este înlocuită cu acea culoare din paletă pentru care se obține distanța Euclidiană minimă în spațiul  $L^*a^*b^*$ .

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

cuantizarea la o paletă fixă

→ alegerea culorilor pe baza distanței în  $L^*a^*b^*$  (continuare)



TrueColor



cuantizare cu dist.  $L^*a^*b^*$

!complexitate de calcul ridicată, optimizare prin calculul în prealabil al distanțelor.



TrueColor



cuantizare dist.  $L^*a^*b^*$



paleta folosită Webmaster 216 culori

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

cuantizarea la o paletă adaptivă

→ **metode cu preclasificare** (preclustering): bazate în general pe analiza statistică a distribuției de culoare. Paleta de culoare este determinată o singură dată.

**metode cu divizare:** pornesc de la o singură clasă de pixeli (întreaga imagine) iar aceasta este divizată recursiv până la numărul de clase dorite.

- exemple: median-cut, octree, bazate pe varianță, binary splitting, center-cut, etc.

**metode aglomerative:** pornesc de la un număr de clase egal cu numărul de pixeli din imagine (un pixel pe clasă) și apoi clasele sunt fuzionate progresiv până când se ajunge la numărul de clase dorite.

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

cuantizarea la o paletă adaptivă

→ **metode cu postclasificare** (postclustering): acestea determină o paletă de culoare inițială și apoi aceasta este îmbunătățită în mod iterativ.

- implică calcul iterativ sau optimizare stochastică, putând furniza astfel o calitate vizuală superioară metodelor cu preclasificare, dar în defavoarea complexității de calcul (7),
- se poate folosi în esență orice algoritm de clasificare a datelor,

> exemple de algoritmi de clasificare (clustering) adaptați problemei cuantizării culorilor: k-means, minmax, competitive learning, fuzzy c-means, BIRCH sau self-organizing maps SOM

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

metoda median cut (preclasificare)

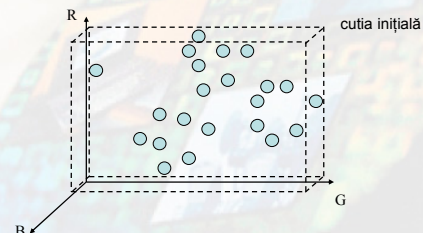
→ se urmărește ca fiecare clasă de culoare obținută să conțină un număr egal de pixeli:

1. în spațiul de culoare considerat, se determină cea mai mică "cutie" ce include toate culorile din imagine,
2. culorile din aceasta sunt sortate după axa cea mai lungă,
3. cutia este împărțită în două după valoarea mediană (median cut),
4. procesul se repetă până se obțin 256 de regiuni,
5. paleta de culoare este dată de culoarea medie a fiecărei cutii (centroidul clasei).

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

metoda median cut (preclasificare)





**Modificarea paletei de culoare**  
↳ **cuantizarea culorilor (continuare)**  
metoda median cut (preclasificare)

- valorile sunt ordonate după axa G (cea mai lungă)  
- ex. [0 4 4 5 100] valoare mediană?

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU 66

**Modificarea paletei de culoare**  
↳ **cuantizarea culorilor (continuare)**  
metoda median cut (preclasificare)

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU 67

**Modificarea paletei de culoare**  
↳ **cuantizarea culorilor (continuare)**  
metoda median cut (preclasificare)

valorile medii pentru fiecare clasă (centroid)

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU 68

**Modificarea paletei de culoare**  
↳ **cuantizarea culorilor (continuare)**  
metoda octree (preclasificare)

→ culorile sunt reprezentate cu arbori de tip "octree" = un arbore în care fiecare nod are până la 8 vecini (oct)

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU 69

**Modificarea paletei de culoare**  
↳ **cuantizarea culorilor (continuare)**  
metoda octree (preclasificare)

→ motivație: reprezentarea culorilor pe 8 biți:

cub RGB reprezentare octree

-întregul cub RGB corespunde rădăcinii (nivelul 0),  
- se înjumătățește după fiecare axă → 2x2x2 cuburi (nivelul 1),  
- se repetă procesul până când cuburile obținute reprezintă o singură valoare → frunzele arborelui (nivelul 8 )  
- 8 niveluri =  $2^{3 \times 8}$   
→  $2^{24}$  = True color

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU 70

**Modificarea paletei de culoare** [Dr.Dobb]  
↳ **cuantizarea culorilor (continuare)**  
metoda octree (preclasificare)

→ algoritmul este următorul:  
1. imaginea este scanată și se construiește arborele octree.

|   | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| G | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| B | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

o culoare RGB ce trebuie inserată

L1:101 = 5  
L2:011 = 3  
L8:111 = 7

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU 71

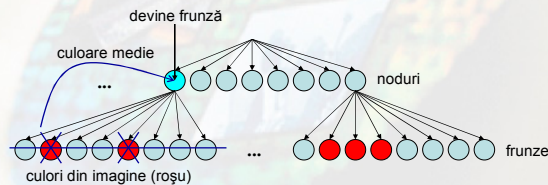
## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

metoda octree (preclasificare)

→ algoritmul este următorul (continuare):

2. dacă numărul de frunze obținute este mai mare decât numărul de culori dorite (evident pentru cuantizarea culorilor), atunci arborele este "curățat" până la obținerea numărului de frunze dorit.



## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

metoda octree (preclasificare)

→ algoritmul este următorul (continuare):

3. imaginea este parcursă încă o dată iar culorile din aceasta sunt înlocuite cu noile frunze din arbore.

>indiferent de modul în care se determină paleta de culoare redusă a imaginii cuantizate, aceasta presupune o eroare (diferența dintre culoarea inițială și cea obținută prin cuantizare)

→ efectul este cu atât mai vizibil cu cât imaginea conține mai multe schimbări fine (ex. gradient de culoare)



## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

difuzia erorii

→ cel mai popular algoritm este Floyd-Stenberg,

→ acesta folosește o mască de filtrare care se aplică fiecărui pixel din imaginea ce urmează a fi cuantizată:

$$\frac{1}{16} \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

Diagram showing the 3x3 Floyd-Stenberg error diffusion mask. The center pixel is labeled 'pixel curent' and the surrounding pixels are labeled 'pixeli vecini'.

→ algoritmul este următorul:

- imaginea se parcurge secvențial de la stânga la dreapta și de sus în jos,

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

difuzia erorii

→ algoritmul Floyd-Stenberg (continuare):

- pentru fiecare pixel curent  $c=(c_x, c_y, c_z)$ , unde  $c_x$ ,  $c_y$  și  $c_z$  reprezintă coordonatele de culoare dintr-un anumit spațiu (ex.  $x=R, y=G, z=B, \dots$ ), se determină, folosind unul dintre algoritmii de cuantizare, noua valoare  $c'=(c'_x, c'_y, c'_z)$ ,

- se estimează erorile de aproximare pentru fiecare componentă,  $E_x, E_y$  și  $E_z$ :

$$E_i = |c_i - c'_i|, \text{ unde } i \in \{x, y, z\}$$

$$\frac{1}{16} \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

difuzia erorii

→ algoritmul Floyd-Stenberg (continuare):

- folosind masca de filtrare erorile cuantificării sunt distribuite vecinilor astfel:

$$\begin{aligned} c_{est}^i &= c_{est}^i + \frac{7}{16} \cdot E^i & c_{sud-est}^i &= c_{sud-est}^i + \frac{1}{16} \cdot E^i \\ c_{sud}^i &= c_{sud}^i + \frac{5}{16} \cdot E^i & c_{sud-vest}^i &= c_{sud-vest}^i + \frac{3}{16} \cdot E^i \end{aligned}$$

$$i \in \{x, y, z\}$$

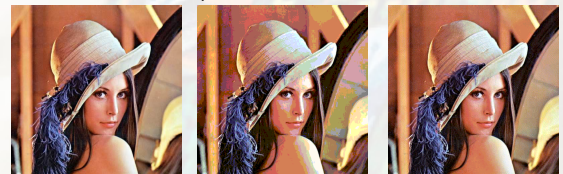
## Modificarea paletei de culoare

### ↳ cuantizarea culorilor (continuare)

difuzia erorii

→ în ciuda aparențelor, aplicarea măștii nu modifică culorile determinate în urma cuantizării (noile culori) !

→ efectul vizual este important:



**Modificarea paletei de culoare**  
↳ **cuantizarea culorilor (continuare)**

difuzia erorii

TrueColor

cuantizare cu dist.  $L^*a^*b^*$

cuantizare cu dist.  $L^*a^*b^*$  + difuzie eroare

256 niveluri de gri

difuzia erorii → alb-negru

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU 78

**Modificarea paletei de culoare**  
↳ **cuantizarea culorilor (continuare)**

difuzia erorii → cum funcționează ? având în vedere că nu adaugă culori noi ?

difuzia erorii → alb-negru

gri ?

păcăleşte ochiul uman prin alăturarea culorilor disponibile pentru a crea senzația altor culori, ex. negru+alb=gri, roșu+negru=roșu închis

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU 79

Sfârșit Curs

Tehnici avansate de prelucrarea și analiza imaginilor, Ș.I. Bogdan IONESCU 80