

# *Prelucrarea Imaginilor*

## *Curs 2*

*Îmbunătățirea imaginilor*

8.03.2019

# Îmbunătățirea imaginilor

- **Scopul** acestor prelucrări îl constituie *accentuarea* (punerea în evidență) a unor *caracteristici* conținute în imagine pentru a putea fi observate mai ușor.
- **Metodele** utilizate în algoritmii de îmbunătățire a imaginilor amplifică anumite *caracteristici* **fără** a mări cantitatea de informații.
- Tehnicile sunt **grupate** (după algoritmii utilizați) astfel:
  - ☒ **Operațiuni punctuale** prin care se poate realiza creșterea contrastului, reducerea zgomotului, etc.
  - ☒ **Operațiuni spațiale** care permit eliminarea zgomotului, filtrări, etc.
- Formulele se referă la imagini de tip 2 ([6]) - *nuanțe de gri*.
- Acestea se vor aplica proportional pe fiecare componentă **fără** a depăsi spatiul culorilor!

## 1. Operații punctuale

Aceste operațiuni permit trecerea de la o nuantă de gri  $u \in [0, L]$  la o altă nuantă de gri  $v \in [0, L]$ , conform unei transformări  $v = f(u)$ , unde  $f : [0, L] \rightarrow [0, L]$ .

$$P.Cul = f(P.Cul)$$

a1) Accentuarea contrastului



a2) Reducerea contrastului

b) Reducerea zgomotului

c) Binarizarea imaginilor

g) Compresia de contrast

---

d) Negativarea imaginilor

e) Operațiuni de tip fereastră



f) Extragerea unui bit

h) Scăderea imaginilor

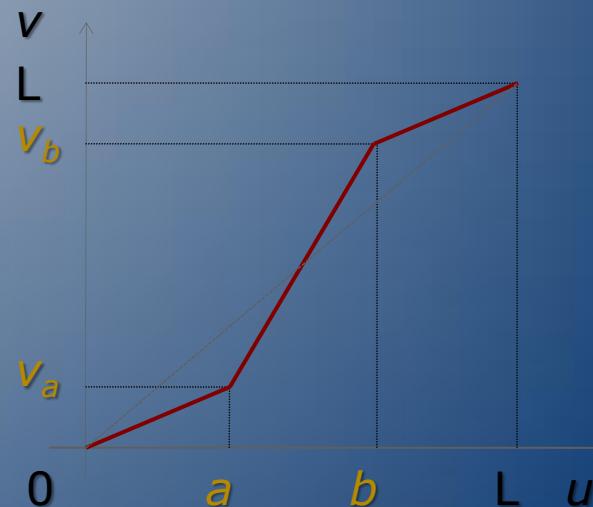
i) Modelarea imaginilor prin histograme

j) Gamma Correction

## a) Accentuarea contrastului

- Această transformare este recomandată imaginilor cu contrast scăzut (obținute de exemplu într-un mediu cu iluminare slabă).
- Fiind date două limite  $a$  și  $b$  ( $0 < a < b < L$ ) pentru care se cunosc valorile  $v_a = f(a)$  și  $v_b = f(b)$ , transformarea este:

$$f(u) = \begin{cases} u * v_a / a & \text{pentru } 0 \leq u \leq a \\ (u-a)/(b-a)*(v_b-v_a)+v_a & \text{pentru } a < u \leq b \\ (u-b)/(L-b)*(L-v_b)+v_b & \text{pentru } b < u \leq L \end{cases}$$

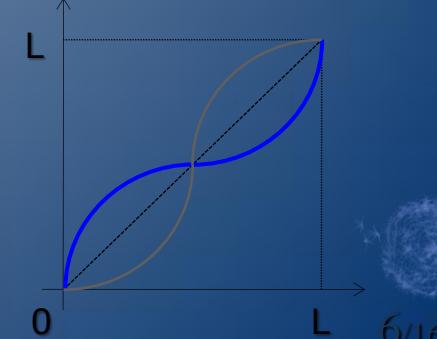
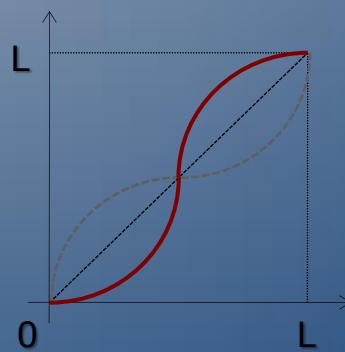
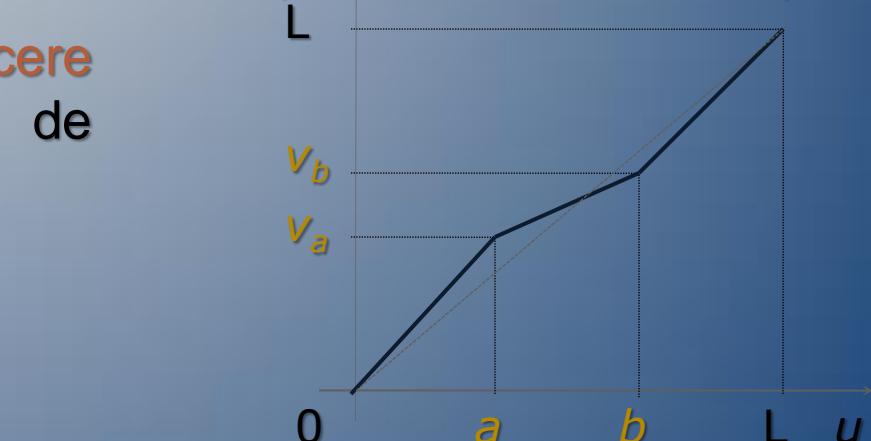


## a) Reducerea contrastului

### \* Observații.

- Valorile  $a$  și  $b$  se pot obține din histograma imaginii;
- Parametrii  $a$  și  $b$  precizează amplificarea contrastului;
- Pentru  $a > b$  se obține transformarea inversă (Reducerea contrastului).
  - Pentru o **accentuare** respectiv **reducere** netedă transformările sunt date de următoarele funcții:

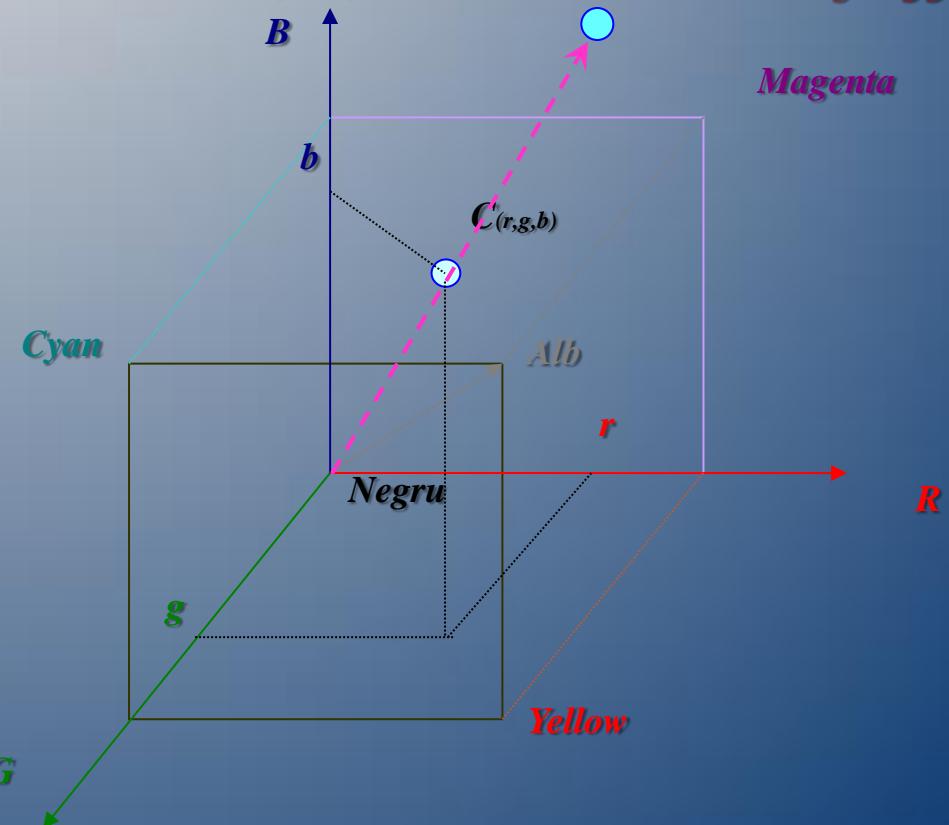
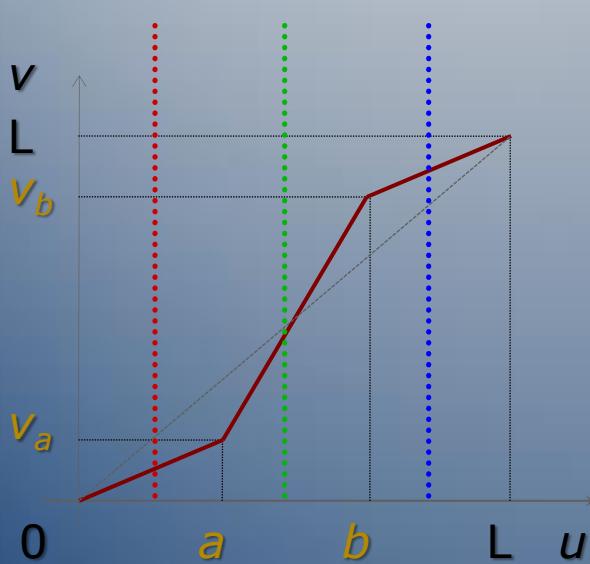
$$\begin{cases} f_1(u) = (\sin(\pi*u/L - \pi/2) + 1) / 2*L \\ f_2(u) = (\arcsin(2u/L - 1) + \pi/2) / \pi*L \end{cases}$$



# a) Spatiul culorilor în sistemul *RGB*

## • Observații.

- Modificarea pe fiecare componentă separat poate afecta componenta culorii!
  - Modificarea culorii trebuie să conserve proporția initială!
  - La factor unic se poate ieși din spațiu (cub)!
- Ex<sub>1</sub>:  $P(50, 100, 200) \rightarrow P'(25, 100, 225)$  -  $fs(0.50, 1.00, 1.125) \Rightarrow$  modifica nuanta!
- Ex<sub>2</sub>:  $P(150, 150, 240) \rightarrow P'(25, 100, 225)$  -  $fs=v(180)/180=200/180 \Rightarrow P'.B=250 < 256!$   
 $fs=v(180)/180=210/180 \Rightarrow P'.B=263 > 255!$



# Teme

- 1. Pentru imagini gri cresterea/diminuarea, contrastului liniar/neted.



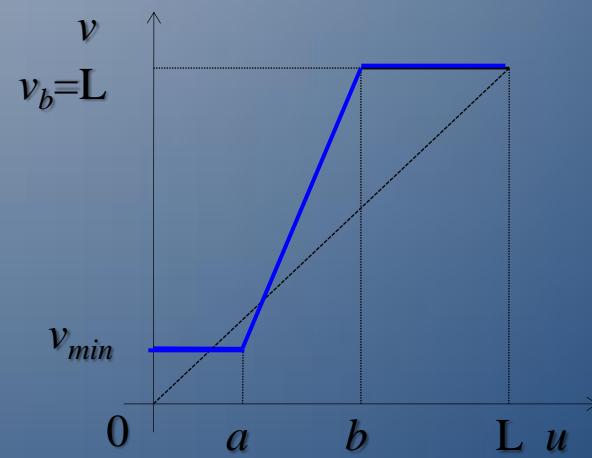
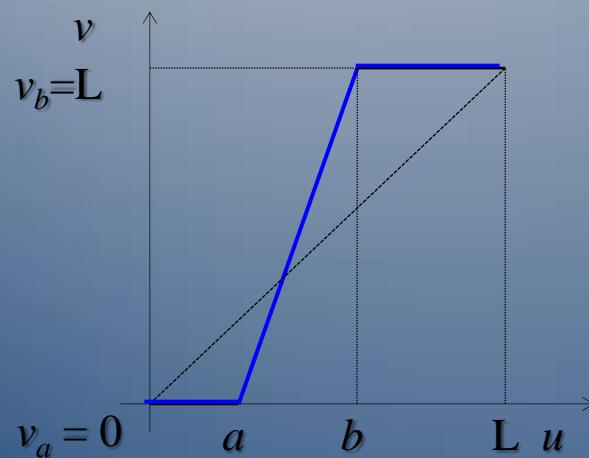
- 2. Pentru imagini color cresterea/diminuarea, contrastului liniar/neted.



## b) Reducerea zgomotului

- **Rz** se realizează prin limitarea culorilor imaginii, stiind că acestea se află în domeniul  $[a,b]$ . Prin această metodă se pot pune în evidență nuante greu vizibile. Operatia este un caz particular al celei precedente ( a ) pentru  $v_a = 0$  și  $v_b = L$ . Valorile  $a$  si  $b$  se pot determina din histograma imaginii. Transformarea se poate realiza conform unuia din cele două grafice de mai jos :

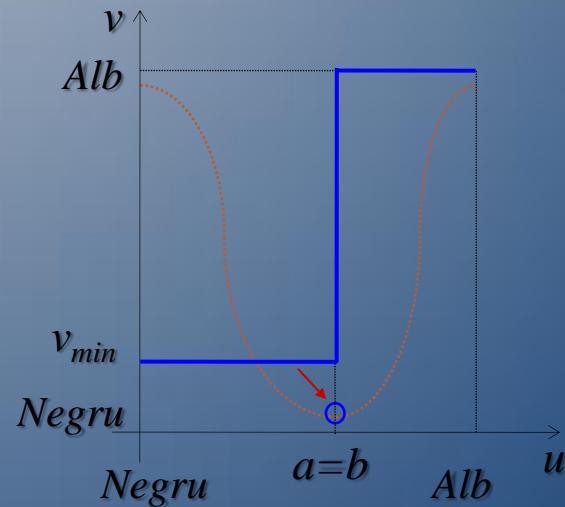
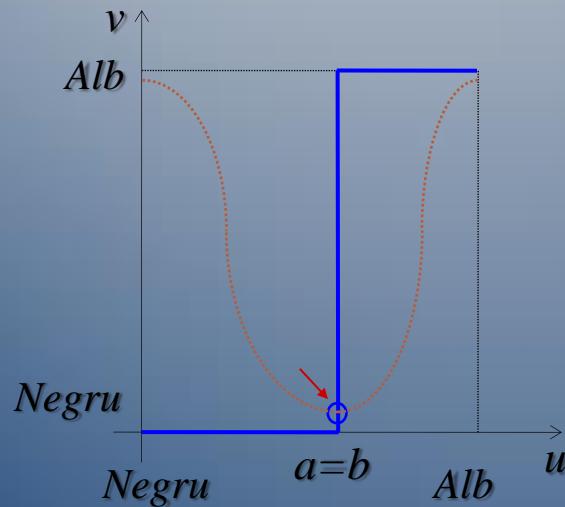
$$f(u) = \begin{cases} 0 & (\text{sau o valoare minima } v_{min}) \\ & \text{pentru } 0 \leq u \leq a \\ (u-a) / (b-a) * L & \text{pentru } a < u \leq b \\ L & \text{pentru } b < u \leq L. \end{cases}$$



## c) Binarizarea imaginilor

- Această operatie are ca obiectiv obtinerea unei imagini alb-negru dintr-o imagine care contine și alte nuante nedorite provenite din diverse motive tehnice (de exemplu copiere). Aceste zgomote apărute în imagine vor fi eliminate prin studierea histogramei imaginii. Transformarea este un caz particular al celei precedente (b) ) pentru  $a=b =$  valea histogramei (vezi graficul de mai jos):

$$f(u) = \begin{cases} 0 \text{ (sau o valoare minima } v_{\min} \text{ )} & \text{pentru } 0 \leq u \leq a \\ L & \text{pentru } a < u \leq L. \end{cases}$$



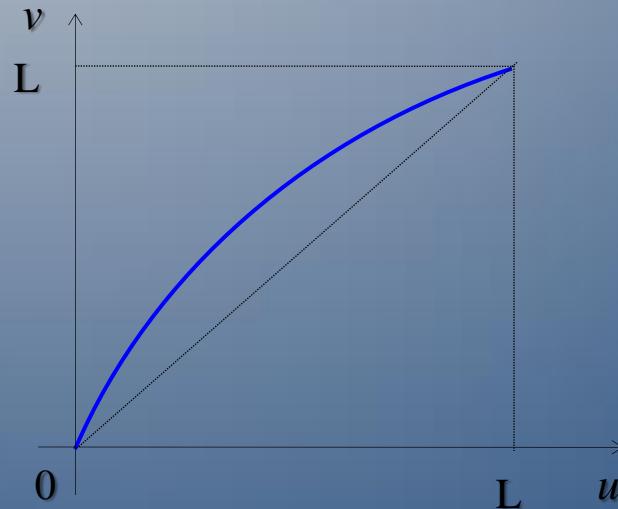
## g) Compresia de contrast

- În situația în care dinamica datelor din imagine este foarte largă este recomandată o astfel de operatie prin care se îmbunătățeste vizibilitatea punctelor de amplitudine relativ mică față de ceilalți.
- Operatia de compresie se realizează printr-o transformare logaritmică de tipul:

$$v = f(u) = \text{cst} * \log(1+|u|)$$

- Pentru transformarea reprezentată alăturat, formula de calcul este următoarea:

$$v = f(u) = L / \ln(L+1) * \ln(1+u)$$

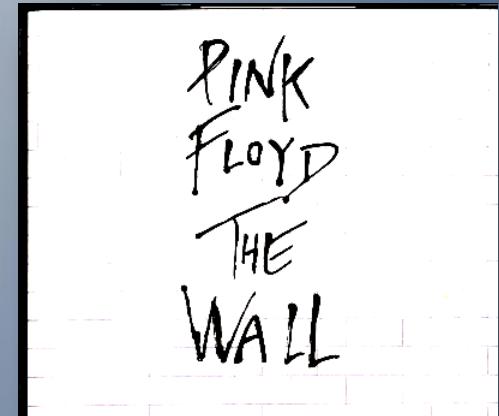
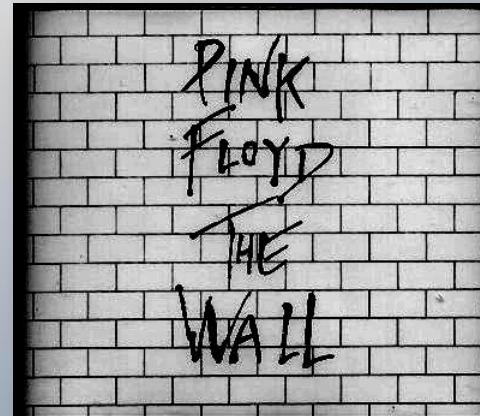


**g) ... Compresia de contrast**



# Teme

- 3. Pentru imagini gri reducerea zgomotului si binarizare.



- 4. Pentru imagini color compresia de contrast.



# Ex. Converting color to grayscale

- Then, add together 30% of the red value, 59% of the green value, and 11% of the blue value (these weights depend on the exact choice of the RGB primaries, but are typical).
- The formula  $(11 * R + 16 * G + 5 * B) / 32$  is also popular since it can be efficiently implemented using only integer operations.



# Ex. Converting color to grayscale

```
public partial class Form1 : Form
{
    Bitmap Im2, Im3;
    private Image loadedImage;
    public Form1()
    {
        InitializeComponent();
        Im2 = new Bitmap(pictureBox2.Image);
        Im3 = new Bitmap(pictureBox3.Image);
    }
    ...
    Im3.SetPixel(i/3+325,j/3+150,Color.FromArgb(255,f(c.R, p.R), ...G, ...B)));
    saveFileDialog1.ShowDialog();
    Im3.Save(saveFileDialog1.FileName);
}

private void Gri_Click(object sender, EventArgs e)
{
    openFileDialog1.ShowDialog();
    loadedImage =
        Image.FromFile(openFileDialog1.FileName);
    Bitmap Im1 = new Bitmap(loadedImage);
    pictureBox1.Image = Im1;
    pictureBox1.Refresh();
}
```



*Succes!*

