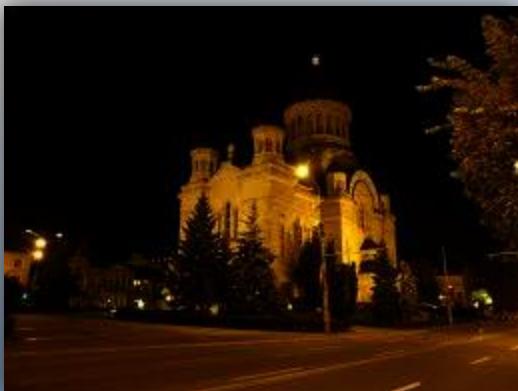


Prelucrarea Imaginilor Curs 3

... Îmbunătățirea imaginilor



Îmbunătățirea imaginilor

- 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.

... 1. Operațiuni 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 $f : [0, L] \rightarrow [0, L]$.

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

a) Accentuarea / Reducerea contrastului

b) Reducerea zgomotului

c) Binarizarea imaginilor

g) Compresia de contrast

C2

d) Negativarea imaginilor

e) Operațiuni de tip fereastră

f) Extragerea unui bit

h) Scăderea imaginilor

i) Modelarea imaginilor prin histograme

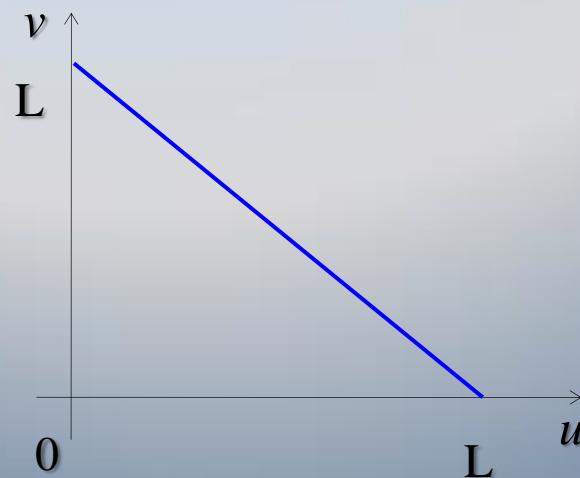
j) Gamma Correction

C3

d) Negativarea imaginilor

- ✿ Această operație de inversare a imaginilor se poate utiliza în situația în care dispunem de negativul unei imagini (de exemplu în prelucrarea imaginilor medicale).
- ✿ Operația de inversare se realizează prin transformarea următoare:

$$v = f(u) = L - u$$



d) Negativarea imaginilor

* Inversarea fiecarei componente (r,g,b):

$$(r', g', b') = (255-r, 255-g, 255-b)$$

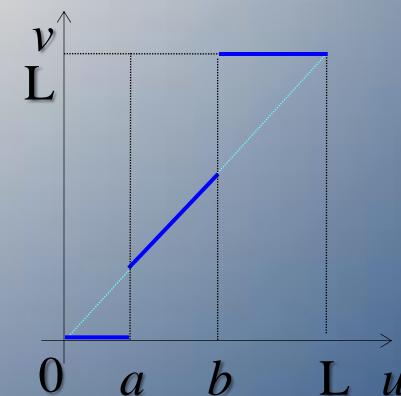
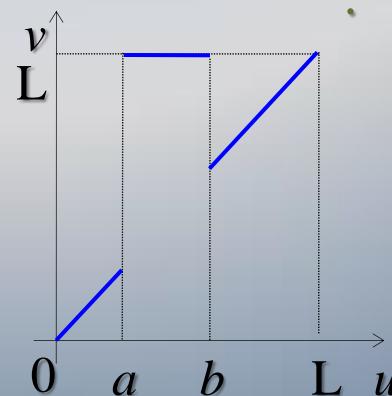
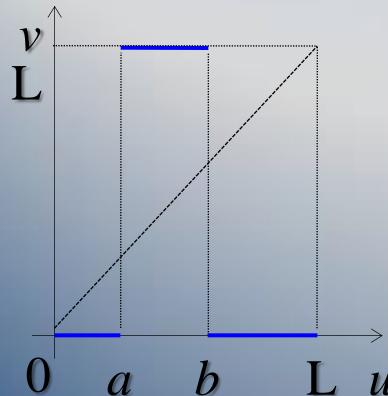


e) Operațiuni de tip fereastră

- ✿ Transformările de acest tip permit extragerea anumitor caracteristici conținute în regiuni reprezentate prin diferite nuante de gri.
- ✿ Decuparea respectiv accentuarea acestor zone (definite prin intervalul nuanțelor $[a,b]$) se poate realiza astfel:

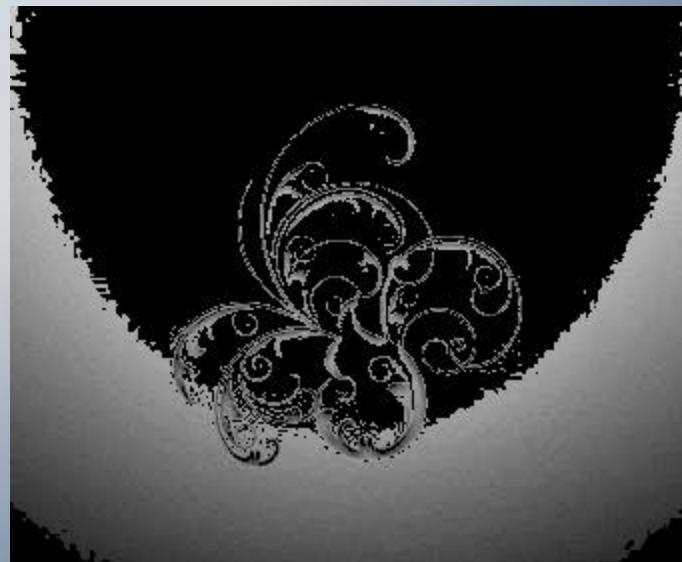
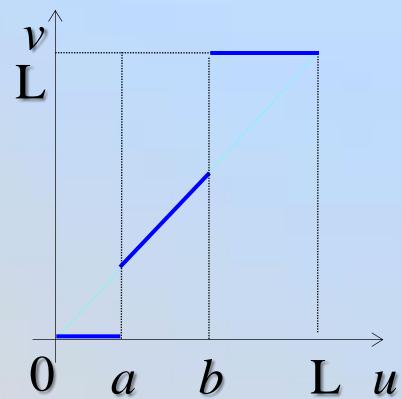
$$f_d(u) = \begin{cases} L & \text{pentru } u \in [a,b] \\ 0 & \text{pentru } u \notin [a,b] \end{cases}$$

$$f_a(u) = \begin{cases} L & \text{pentru } u \in [a,b] \\ u & \text{pentru } u \notin [a,b] \end{cases}$$



e) ... Operațiuni de tip fereastră

Exemple :



e) ... Operațiuni de tip fereastră

Exemple :



```
Im2 = new Bitmap(Im1);
int a = 100, b = 120;
for (int i = 0; i < Im1.Height; i++)
{
    for (int j = 0; j < Im1.Width; j++)
    {
        Color c = Im1.GetPixel(j, i);
        int g = c.R;
        if (g < a ) g = 0;
        else
            if (g > b ) g = 0;
        Im2.SetPixel(j, i, Color.FromArgb(2*g, 2*g, 2*g));
    }
}
pictureBox2.Image = Im2;
pictureBox2.Refresh();
```

e) ... Operațiuni de tip fereastră

- modificarea componentelor:



f) *Extragerea unui bit*

- Transformarea se aplică în determinarea biților **nesemnificativi** din punct de vedere vizual dintr-o imagine (deoarece putem spune că doar primii șase au semnificație vizuală, contribuția celorlalți fiind legată doar de redarea detaliilor fine din imagine, fără a oferi informații asupra structurii acesteia).
- Operația de extragere a bitului k corespunzătoare codificării

$$u = b_n 2^n + b_{n-1} 2^{n-1} + \dots + b_k 2^k + \dots + b_1 2 + b_0$$

se realizează prin transformarea:

$$f_k(u) = \begin{cases} 1 & \text{pentru } b_k = 1 \\ 0 & \text{rest} \end{cases}$$

f) ... Extragerea unor biti

• Setarea bitilor:



h) Scăderea imaginilor

- Operația de scădere a două imagini este necesară la compararea acestora dacă deosebirile dintre ele sunt relativ mici față de complexitatea imaginilor.
- Prin scăderea celor două imagini se pun în evidență diferențele dintre cele două imagini.



- ✿ Diferenta / compararea



... Click

- Diferenta / compararea



... Click

i) Modelarea imaginilor prin histograme

- Prin histogramă înțelegem reprezentarea grafică a *frecvenței de apariție* a culorilor (nuanțelor de gri) conținute într-o imagine. Prin această metodă (modelarea histogramei) se obține o imagine cu o histogramă dorită. O imagine cu contrast scăzut (cu o histogramă îngustă) se poate transforma printr-un procedeu de egalizare a *histogramei* într-o imagine cu o histogramă uniformă.

- Notăm:

- cu $t(u)$ numărul de pixeli de culoare u (frecvența nuanței u , $u \in \{0, 1, \dots, L\}$),
 - cu T numărul total de pixeli din imagine ($T = t(0) + t(1) + \dots + t(L)$),

- Probabilitatea ca un punct să fie de nuanță $\leq u$ este:

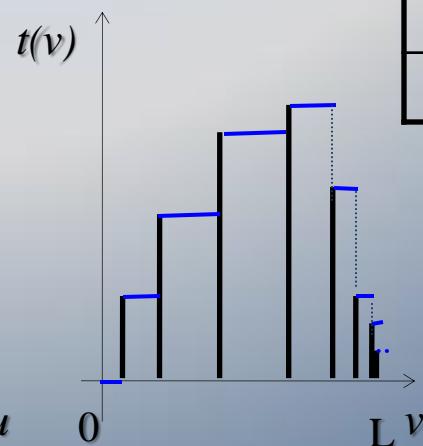
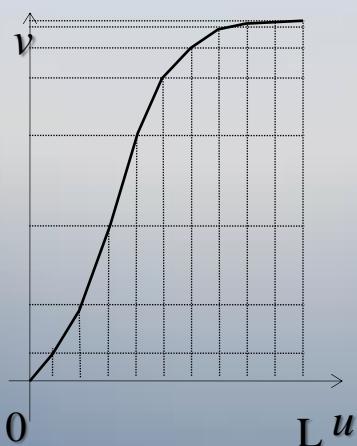
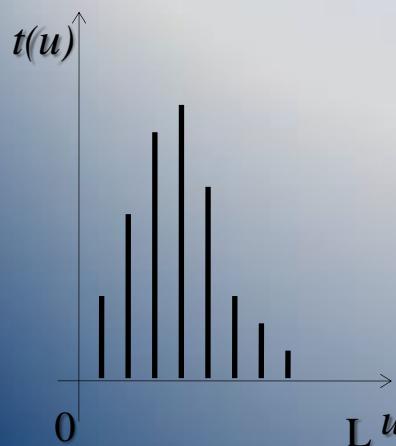
$$p(u) = (t(0) + t(1) + \dots + t(u)) / T$$

- Se observă că $0 \leq p(u) \leq 1$, deci transformarea este $v = f(u) = p(u) * L$

i) ... Modelarea imaginilor prin histograme

- În exemplul de mai jos se poate vedea histograma inițială a), graficul transformării b) și histograma egalizată obținută c) rezultate din datele din tabelul alăturat pentru L=100.
- Modificarea histogramei se realizează de fapt și prin compresia de contrast prezentată anterior g) și de asemenea se mai poate realiza prin alte transformări definite în intervalul $[0, L]$ cu valori în același interval, de exemplu:

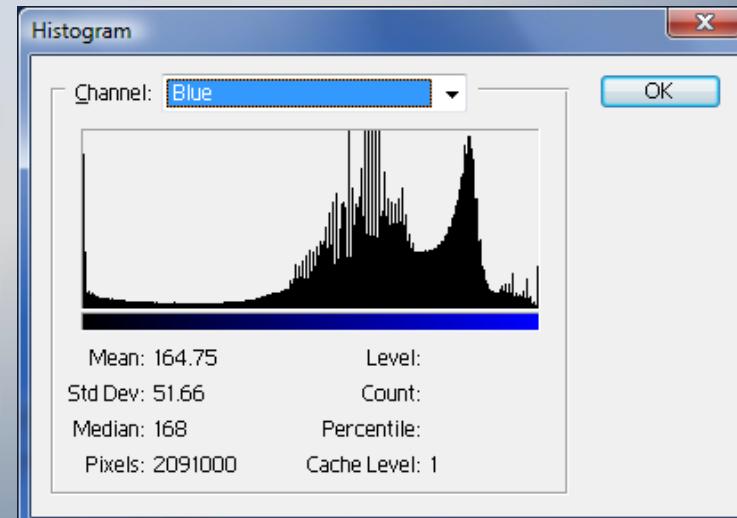
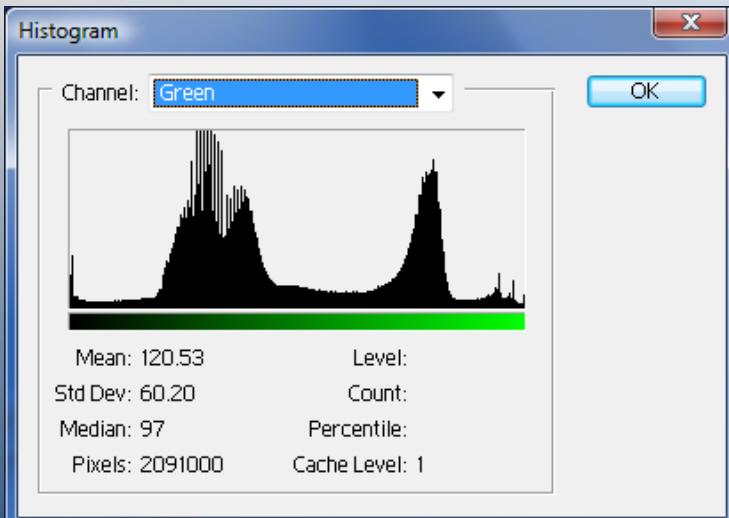
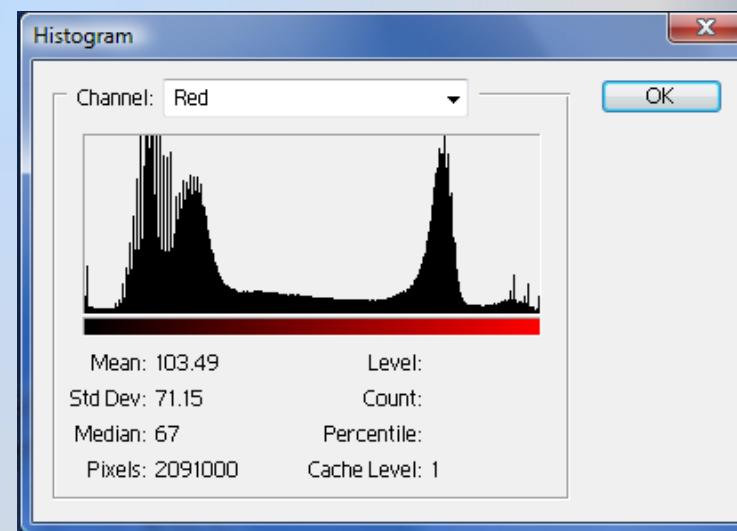
$$f(u) = \text{Sqrt}(u) * \text{Sqrt}(L)$$



u	$t(u)$	$p(u)$	$p(u)*L$	$f(u)$
0	0	0.00	0.00	0
10	30	0.07	7.32	7
20	60	0.22	21.95	22
30	90	0.44	43.90	44
40	100	0.68	68.29	68
50	70	0.85	85.37	85
60	30	0.93	92.68	93
70	20	0.98	97.56	98
80	10	1.00	100.00	100
90	0	1.00	100.00	100
100	0	1.00	100.00	100

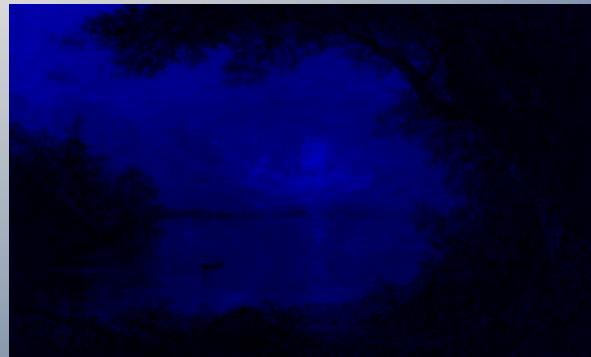


i) ... Modelarea imaginilor prin histograme



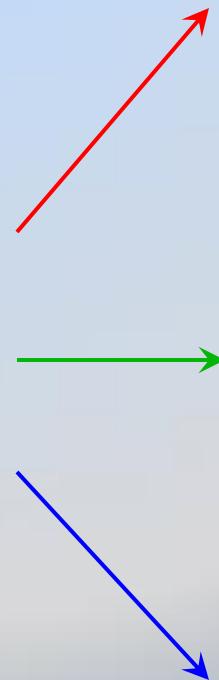
Grayscale

Color → R, G, B



... Grayscale

Color → Gray (R,G,B)



Coeficientii reprezintă
perceptia umană a culorilor.



$g_1:$



$g_2:$



Oamenii sunt mai sensibili
la verde și mai puțin sensibili
la albastru.

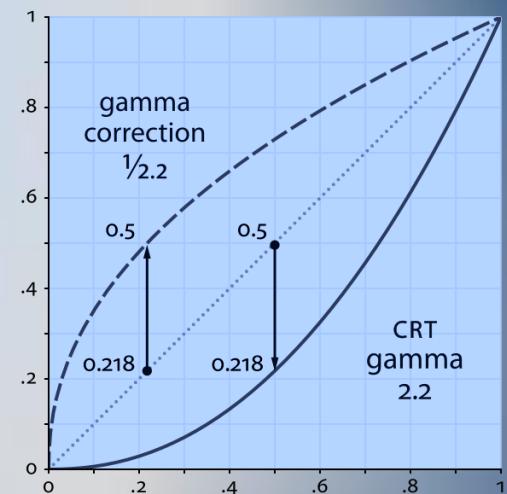
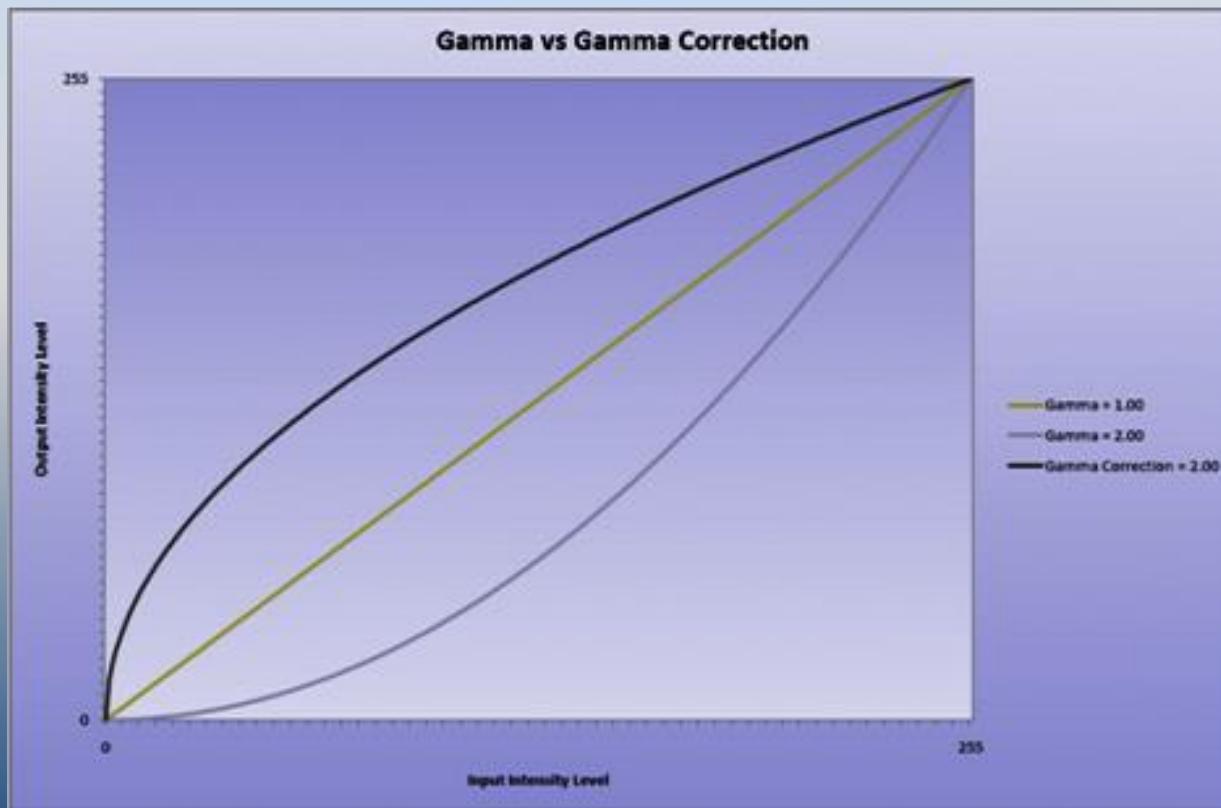
$$g_1 = 0.299 * R + 0.587 * G + 0.114 * B$$

$$g_2 = 0.2126 * R + 0.7152 * G + 0.0722 * B$$



j) *Gamma Correction* = $1/\text{Gamma}$

$$f(u) = c * 255 * (u / 255)^{\gamma_c} \quad (c=1)$$



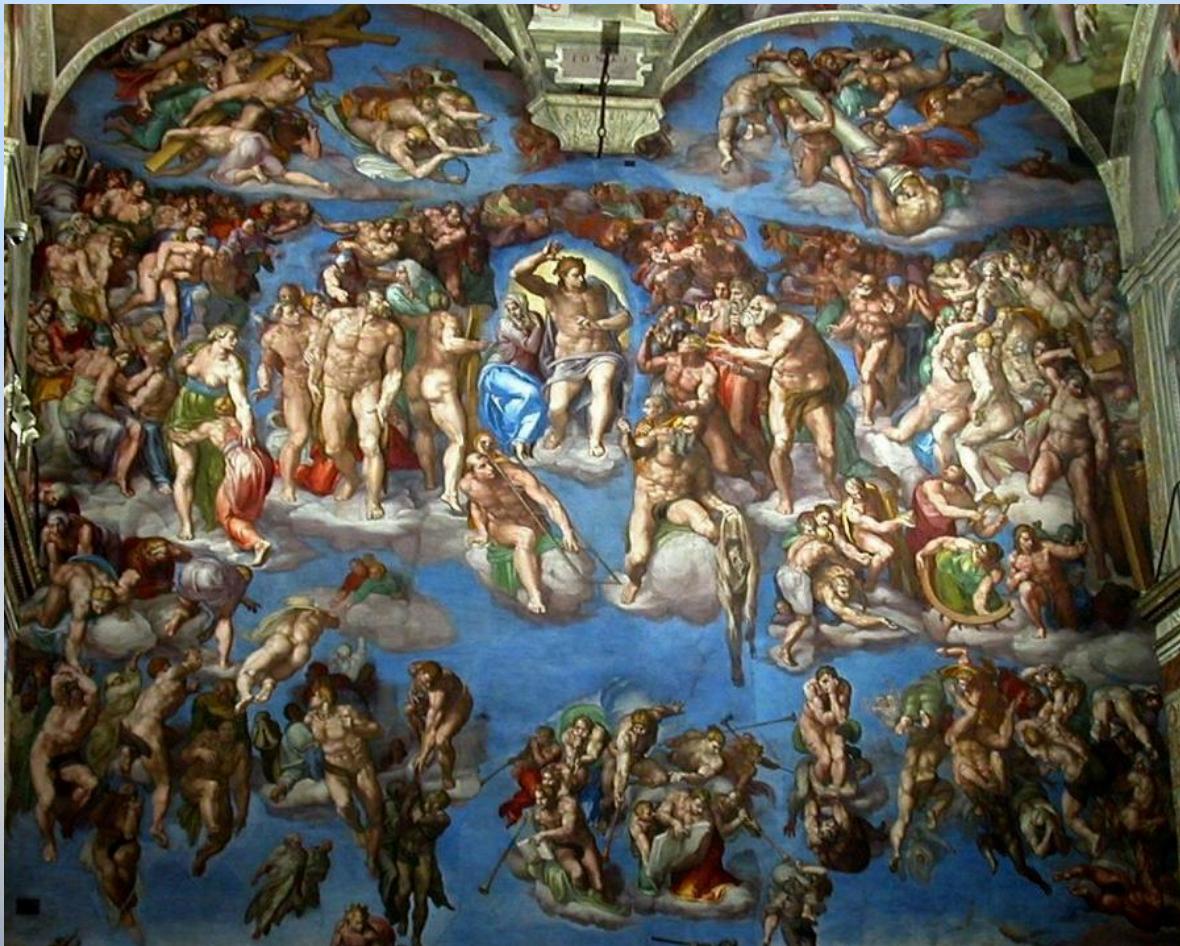
... j) *Gamma Correction = 1/Gamma*

$$f(u) = c * 255 * (u / 255)^{\gamma c} \quad (c=1)$$



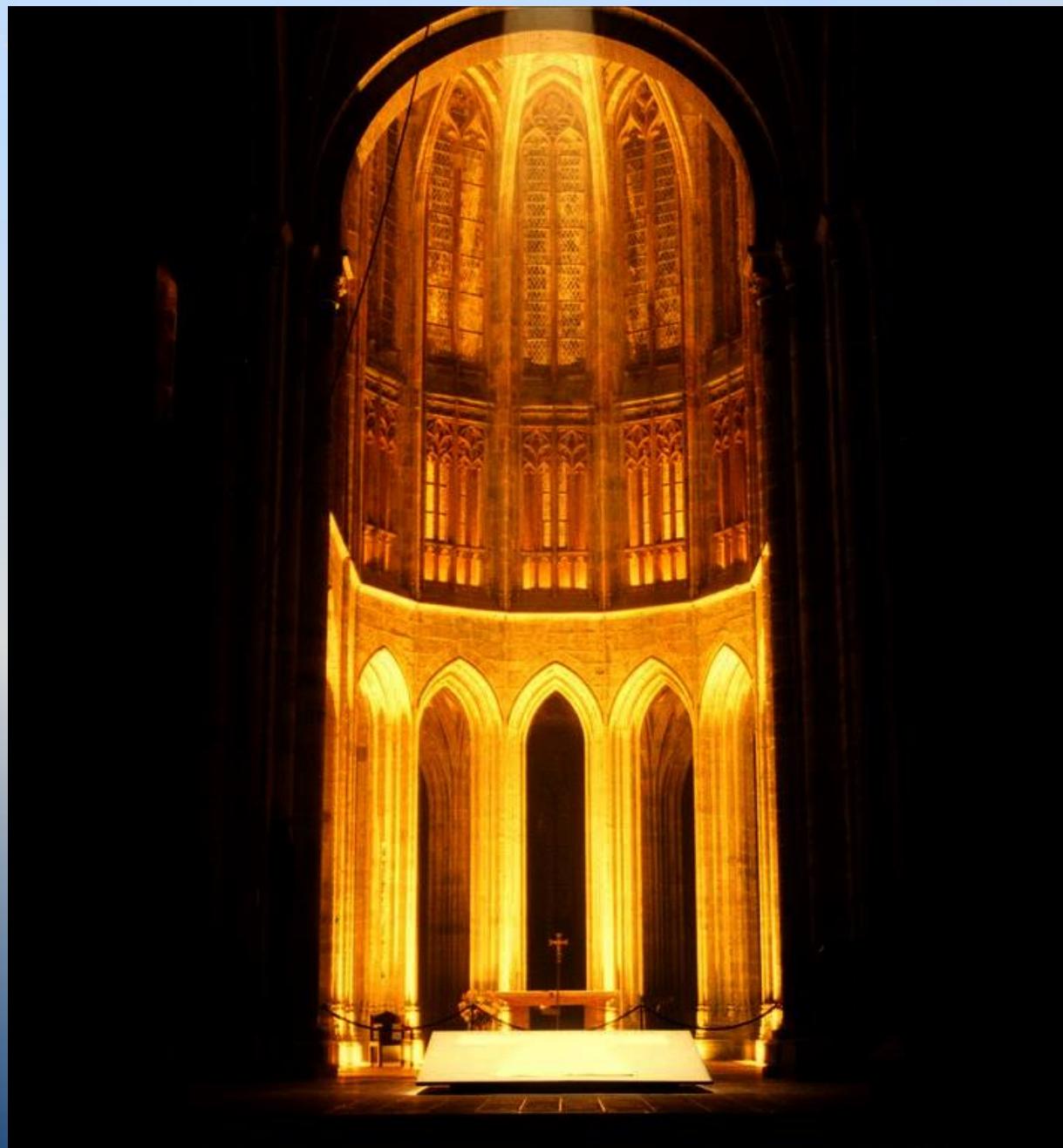
... j) *Gamma Correction* = $1/\text{Gamma}$

$$f(u) = c * 255 * (u / 255)^{\gamma_c} \quad (c=1)$$



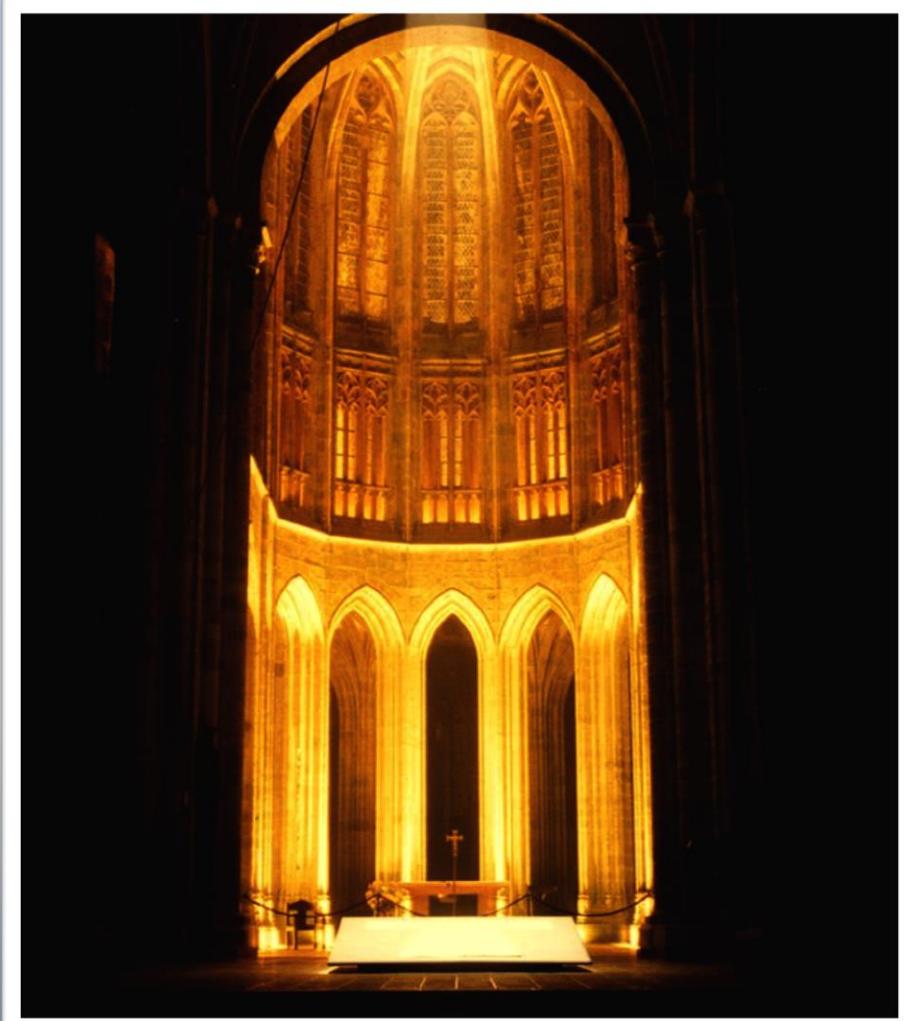
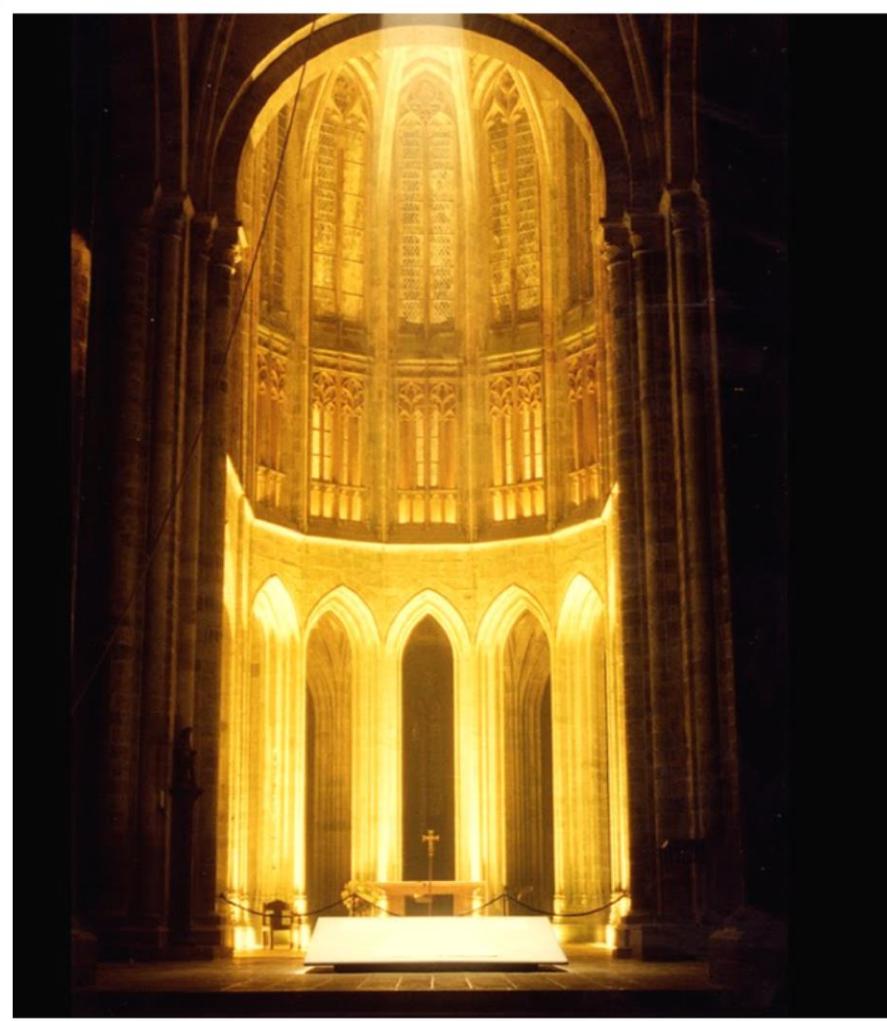
... j) Gamma Correction = $1/\text{Gamma}$

$$f(u) = c * 255 * (u / 255)^{\gamma c} \quad (c=1)$$



... j) Gamma Correction = $1/\text{Gamma}$

$$f(u) = c * 255 * (u / 255)^{\gamma c} \quad (c=1)$$



Teme

Aplicati *Operațiuni punctuale* pentru:

- d) Negativarea imaginilor
- e) Operațiuni de tip fereastră
- f) Extragerea unui bit
- h) Scăderea imaginilor
- i) Modelarea imaginilor prin histograme
- j) Gamma Correction