



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale  
2007-2013



# Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

**AEACD**

## **8. Algoritmi generali de binarizare: Alte efecte**

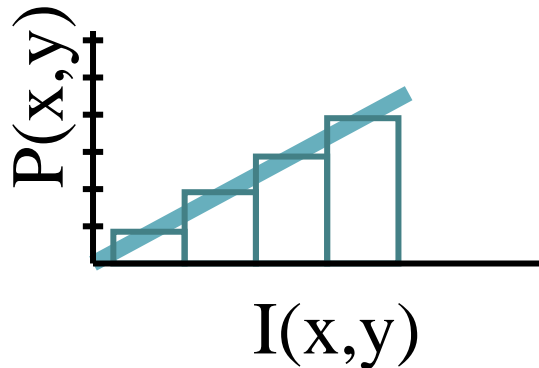
# AGB - alte efecte

- Dithering
- Halftone
- Floyd-Steinberg
- Variatie threshold
- Color uniformization

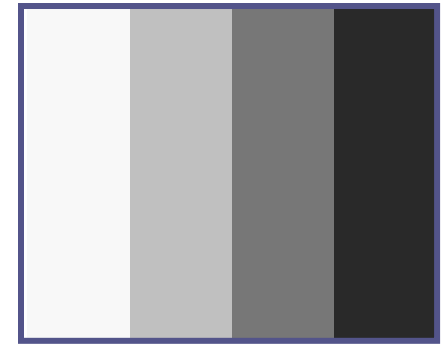
# Image quantization

- Reduce rezolutia unei imagini:
  - Imaginile sunt reprezentate cu un numar limitat de biti per pixeli

$$P(x, y) = \text{trunc}(I(x, y) + 0.5)$$



$I(x,y)$



$P(x,y)$

(2 bits per pixel)

256 bpp



32 bpp



8 bpp



1 bpp



# Dithering

- In domeniul graficii pe calculator, dithering este o tehnica folosita pentru a crea iluzia unei culori ce nu exista intr-un domeniu limitat
- Astfel, o culoare ce nu exista in domeniu este aproximata printr-o difuzie de pixeli din domeniul de culoare
- Ochiul uman va percepe aceasta colectie de pixeli ca o culoare inexistentă in paleta de culori disponibile
- Metoda este similara cu adaugare unui zgomot slab pentru a imbunatati calitatea

## Dithering (2)

- Reduce efectul cuantizarii imaginii
- Printr-o paleta redusa de culori se pot reprezenta pentru ochiul uman culori inexistente in acea paleta



Dithered to 256 colors



# Tipuri de dithering

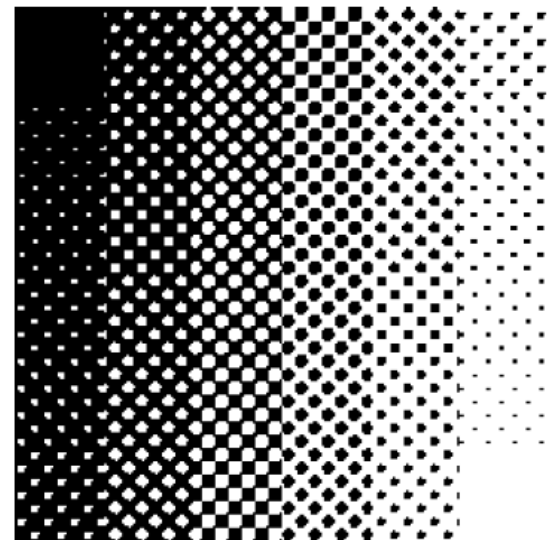
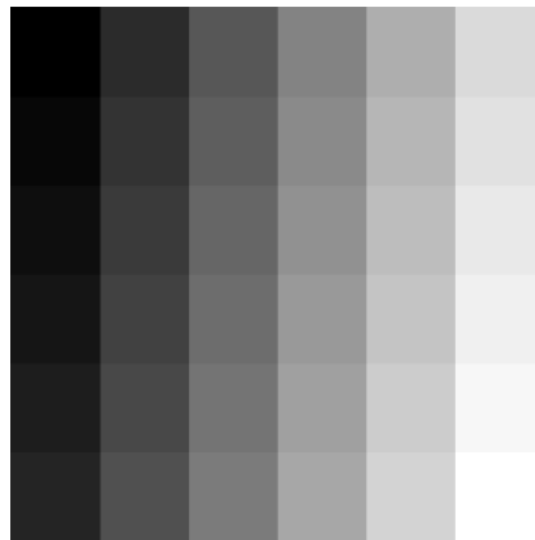
- Exista mai multe tipuri de dithering in functie de metoda folosita:
  - **Thresholding** – fiecare pixel este comparat cu un prag; cea mai simpla, dar si cea mai slaba metoda
  - **Dithering aleator** – pixelii sunt comparati cu praguri alese aleator
  - **Folosind un tipar** – pentru fiecare pixel pragul este reprezentat de valoare tiparului in acel punct; ex: Halftone, Ordered
  - **Error-diffusion** – eroarea acumulata prin adaugarea zgomotului este transmisa la pixelii invecinati; ex: Floyd-Steinberg

# Halftone

- Acest tip de dithering simuleaza culoriile folosind puncte de dimensiuni diferite si aflate la distante diferite
- Privite de la departare aceste colectii de puncte vor rezulta pentru ochiul uman intr-o culoare ce nu poate fi reprezentata cu o anumita paleta
- Astfel daca dimensiunea si rezolutia imaginii sunt destul de mari pixelii vor fi atat de mici incat vor crea efectul unei noi culori

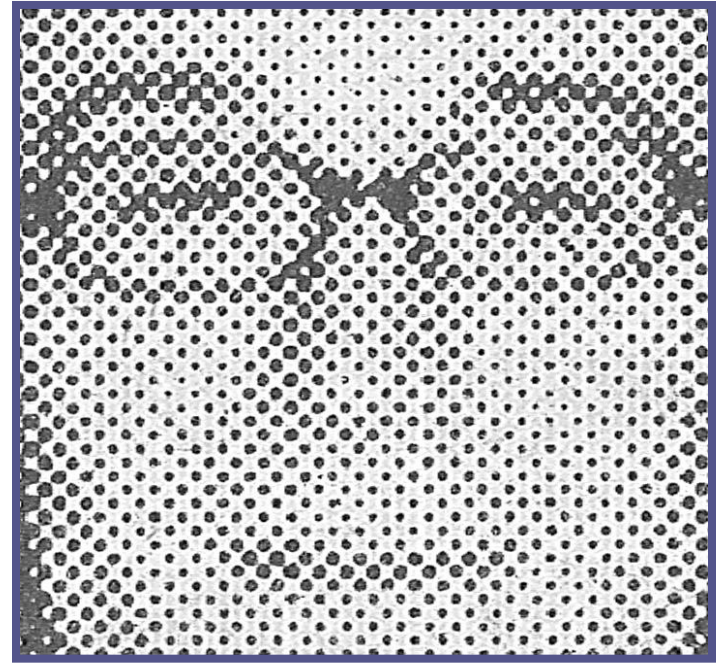
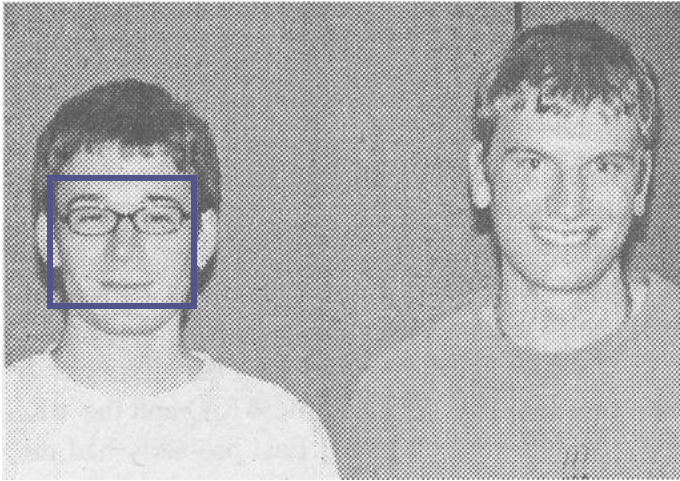
# Halftone (2)

- In imaginea de mai jos este prezentat un model de reprezentare a nuanțelor de gri prin puncte albe pe fundal negru sau puncte negre pe fundal alb
- Se observa si cum dimensiunea punctelor variaza





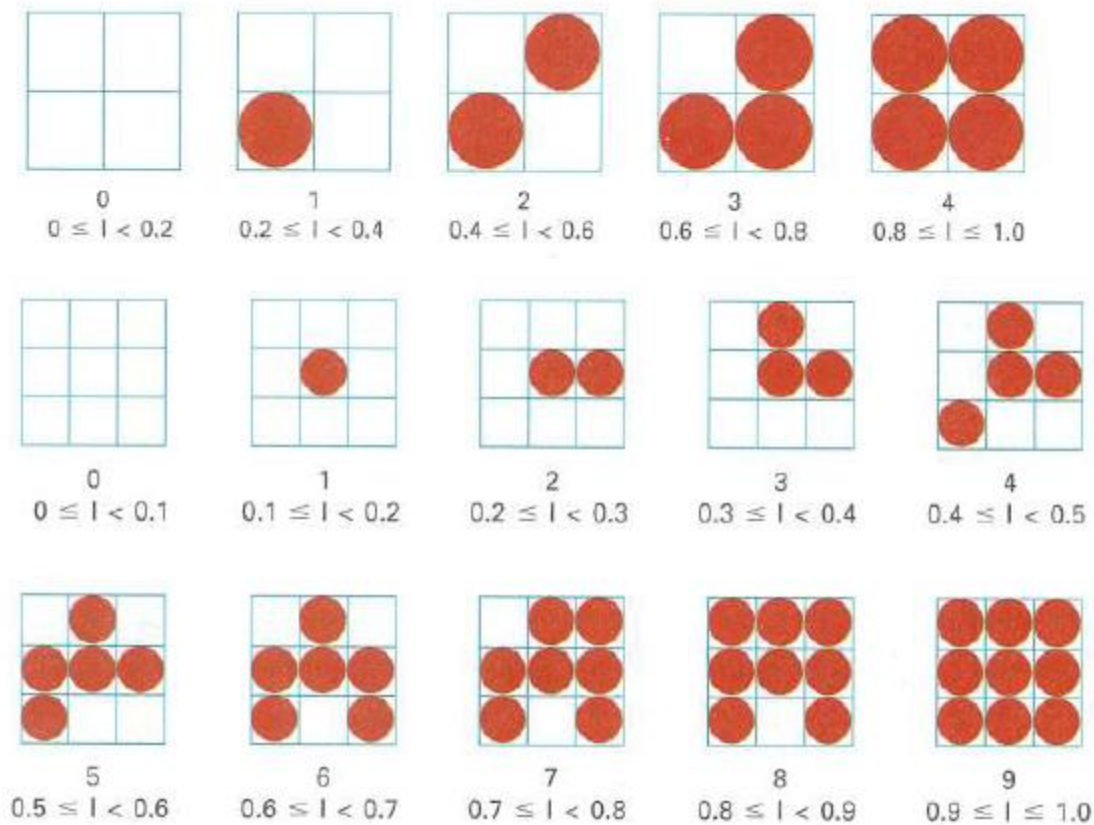
# Halftone (3)



Newspaper image from *North American Bridge Championships Bulletin*, Summer 2003

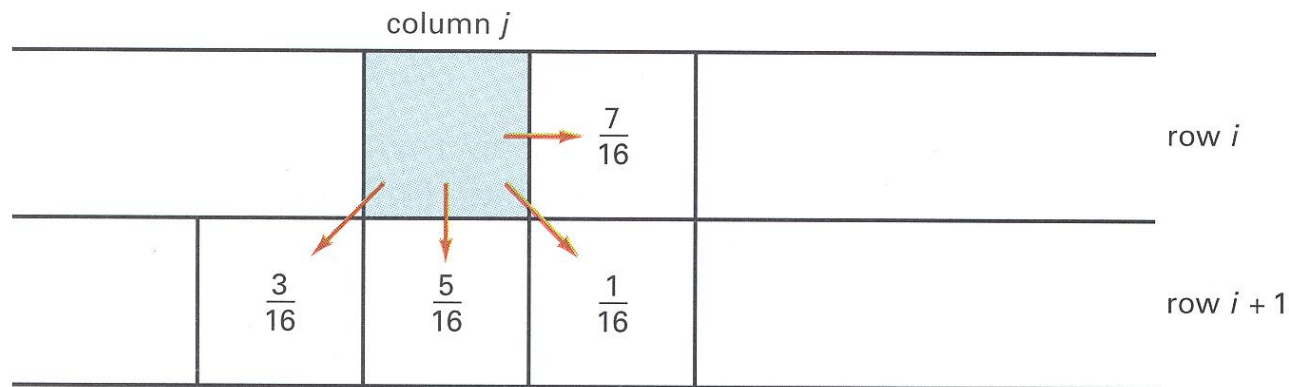
# Halftone (4)

- Sabloane de reprezentare a intensitatilor



# Error Diffusion Dither

- Difuzarea erorilor de cuantizare catre pixelii vecini
  - Dispersia erorilor catre pixelii de jos si din dreapta



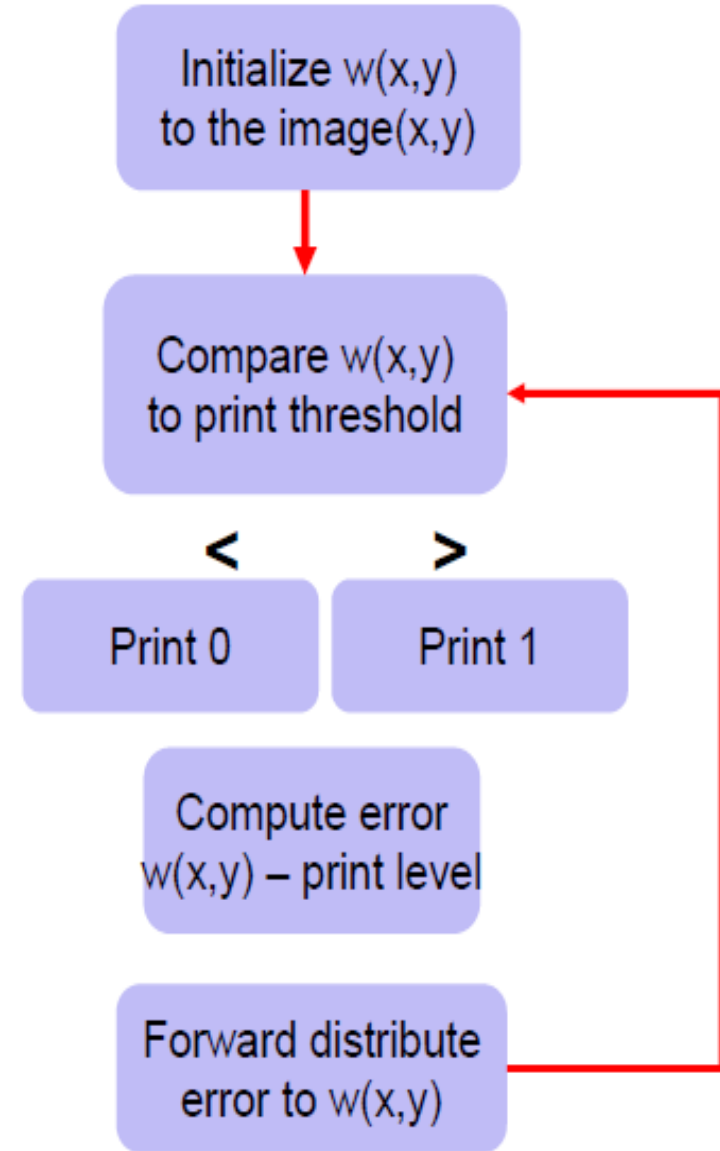
$$\alpha + \beta + \gamma + \delta = 1.0$$

- Cea mai cunoscuta metoda de difuzie a erorilor este Floyd-Steinberg

# Floyd-Steinberg

- Un algoritm folosit des in conversie, mai ales in trecerea de la un format ce permite folosirea mai multor culori la unul mai restrans
- Algoritmul transmite zgomotul acumulat (eroarea) de la un pixel la pixelii invecinati folosind distributia:

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



# Floyd-Steinberg (2)

- Algoritmul scaneaza imaginea de la stanga la dreapta si de sus in jos, modificand fiecare pixel, astfel incat pixelii deja vizitati nu vor mai fi modificati
- In acest fel, in medie, eroarea acumulata se apropie de valoarea 0



# Floyd-Steinberg : Algorithm

```
for (x = 0; x < width; x++) {  
    for (y = 0; y < height; y++) {  
        P(x,y) = trunc(I(x,y) + 0.5)  
        e = I(x,y) - P(x,y)  
        I(x,y+1) += a*e;  
        I(x+1,y-1) += b*e;  
        I(x+1,y) += c*e;  
        I(x+1,y+1) += d *e;  
    }  
}
```

# Floyd-Steinberg (3)



Original  
(8 bits)



Uniform  
Quantization  
(1 bit)



Floyd-Steinberg  
Dither (1 bit)



Floyd-  
Steinberg  
Dither (1 bit)  
(pure B&W)

# Variatie threshold

- Putem observa ca un algoritm ofera pentru un anumit tip de documente rezultate bune dar acestea sunt prea luminate sau prea intunecate
- Putem varia procentual pragul (thresholdul)  $T$  obtinut pentru a mai intuneca sau a mai lumina imaginea binara obtinuta



# Color uniformization

- Ideea de baza este de a “difuza” diferentele de culoare, ponderate in functie de distantele dintre pixelii vecini
- Fiecare pixel obtinut este calculat folosind urmatoarea suma, pentru o dimensiune predefinita pentru fereastra pixelilor vecini

$$\sigma(x, y) = \sum_{(x', y') \in W} \frac{color(x', y') - color(x, y)}{\sqrt{(x - x')^2} + \sqrt{(y - y')^2}}$$

fiecare pixel de coordonate  $(x', y')$  se afla in fereastra centrata in  $(x, y)$

- in final sumele sunt scalate pentru a apartine valorilor de la 0-255