



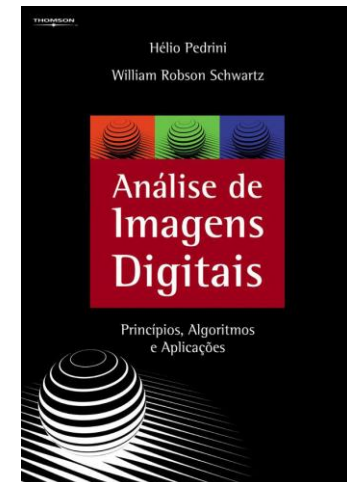
PROCESSAMENTO DE IMAGENS

DITHERING

Prof. Msc. Giovanni Lucca França da Silva
E-mail: giovanni-lucca@live.com

SOBRE A DISCIPLINA

- Bibliografia principal:
 - GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento digital de imagens.** Pearson, 2011.
- Bibliografia complementar:
 - PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações.** Thomson Learning, 2008.



NA AULA PASSADA...

- Fundamentos de cores.
- Modelos de cores.

ROTEIRO

- Dithering.

DITHERING

- O que é dithering?



DITHERING

- Denominação de método para produzir a sensação visual de tons de cinza em dispositivo de visualização capaz apenas de representação dois níveis de intensidade, em geral, preto e branco.
- Procura minimizar os efeitos visuais de contornos artificiais formados pela redução do número de intensidades da imagem.

DITHERING

256 gray levels



Preto e branco



DITHERING

- Algoritmo básico.

```
threshold = (WHITE + BLACK) / 2;
```

```
if( pixel[x][y] < threshold )
```

```
    pixel[x][y] = BLACK;
```

```
else
```

```
    pixel[x][y] = WHITE;
```



DITHERING

- Algoritmo básico.

```
for (int i = 0; i < width; i++){  
    for (int j = 0; j < height; j++){  
  
        ImageType::IndexType pixelIndex = {{i, j}};  
  
        int pixel = image->GetPixel(pixelIndex);  
  
        (pixel > 127) ? pixel = 255 : pixel = 0;  
  
        ditheringImage->SetPixel(pixelIndex, pixel);  
  
    }  
}
```

DITHERING

- O surgimento dos contornos de quantização está associado ao erro de quantização, mais precisamente, à correlação do erro de quantização em relação a pixels vizinhos.
- Solução: introdução de um ruído (perturbação aleatória) para desconcorrelacionar os pixels.

DITHERING

- Algoritmo da modulação aleatória.

```
threshold = (BLACK + WHITE) / 2;  
  
temp = pixel[x][y] + random();  
  
if( temp < threshold )  
    o pixel[x][y] = BLACK;  
else  
    o pixel[x][y] = WHITE;
```

random() - função que gera um valor aleatório entre $[-\text{threshold}, +\text{threshold}]$.



DITHERING

- Algoritmo da modulação aleatória.

```
for (int i = 0; i < width; i++){  
    for (int j = 0; j < height; j++){  
  
        ImageType::IndexType pixelIndex = {{i, j}};  
  
        int random = rand() % 256;  
  
        int pixel = image->GetPixel(pixelIndex) + random;  
  
        (pixel > 127) ? pixel = 255 : pixel = 0;  
  
        ditheringImage->SetPixel(pixelIndex, pixel);  
  
    }  
}
```

DITHERING



DITHERING

- A técnica de dithering está associada a descorrelacionar o erro de quantização, evitando, ou pelo menos tentando minimizar, os contornos de quantização.
- Explora a característica do sistema visual humano de integrar (compor um valor médio) os estímulos luminosos recebidos por determinado ângulo sólido.

DITHERING

- Algoritmos de dithering ordenados procuram introduzir perturbação determinísticas (não aleatórias) para a descorrelacionar o erro de quantização.
- Classes de algoritmos ordenados.

Aperiodico	Floyd-Steinberg	
Periodico	Dispersed-Dot Ordered	Clustered-Dot Ordered
	Dispersão	Aglomerção

DITHERING

- Classes de algoritmos ordenados.
 - Periódico – gerados por processos determinísticos baseados em grade de amostragem regulares.
 - Aperiódico – procuram minimizar o erro de quantização distribuindo-o globalmente ao longo da imagem.
 - Dispersos – resultam da realização dos níveis de cinza por pontos individuais uniformemente distribuídos.
 - Aglomerados – concentração de pontos em pequenos grupos do mesmo valor.

DITHERING

- Algoritmo ordenado periódico.

- Estrutura básica.

```
i = x % N;    /* resto da divisao por N */  
j = y % N;
```

```
if( pixel[x][y] > D[i][j] )  
    o pixel[x][y] = WHITE;  
else  
    o pixel[x][y] = BLACK;
```

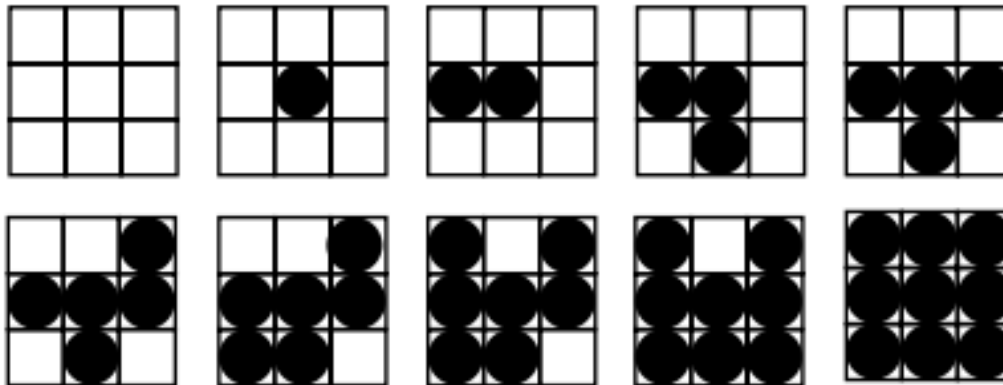
D[][] - matriz de dithering (NxN)

N - ordem da matriz

$$\begin{bmatrix} 8 & 3 & 4 \\ 6 & 1 & 2 \\ 7 & 5 & 9 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 7 & 4 \\ 5 & 8 & 3 \\ 6 & 2 & 9 \end{bmatrix}$$

DITHERING

- Algoritmo ordenado periódico (com aglomeração de pontos).
 - Apropriado para dispositivos que não possuem boa precisão na definição de pontos isolados, como, por exemplo, impressora laser.



$$\begin{bmatrix} 1 & 7 & 4 \\ 5 & 8 & 3 \\ 6 & 2 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 8 & 3 & 4 \\ 6 & 1 & 2 \\ 7 & 5 & 9 \end{bmatrix}$$

DITHERING

- Algoritmo ordenado periódico (com aglomeração de pontos).

```
int dither[3][3] = { 8, 3, 4, 6, 1, 2, 7, 5, 9 };

for (int i = 0; i < width; i++){
    for (int j = 0; j < height; j++){

        ImageType::IndexType pixelIndex = { { i, j } };

        int pixel = image->GetPixel(pixelIndex);

        int m = i % 3;
        int n = j % 3;

        float temp1 = ((pixel * 1.0) / maximum);
        float temp2 = ((dither[m][n] * 1.0) / 10);

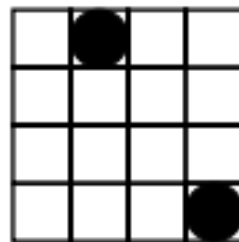
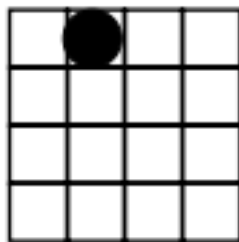
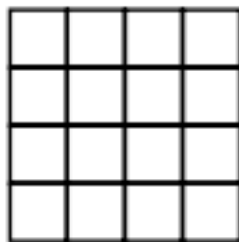
        (temp1 > temp2) ? pixel = 255 : pixel = 0;

        ditheringImage->SetPixel(pixelIndex, pixel);

    }
}
```

DITHERING

- Algoritmo ordenado periódico (com dispersão de pontos).
 - Apropriado para dispositivos com boa precisão de posicionamento de pontos, como, por exemplo, monitores de vídeos.
 - Dispersão -> pontos isolados -> áreas não conectadas.



$$\begin{bmatrix} 2 & 3 \\ 4 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 16 & 3 & 13 \\ 10 & 6 & 11 & 7 \\ 4 & 14 & 1 & 15 \\ 12 & 8 & 9 & 5 \end{bmatrix}$$

DITHERING

- Algoritmo ordenado periódico (com dispersão de pontos).

```
int ditherBayer[2][2] = { 2, 3, 4, 1};

for (int i = 0; i < width; i++){
    for (int j = 0; j < height; j++){

        ImageType::IndexType pixelIndex = { { i, j } };

        int pixel = image->GetPixel(pixelIndex);

        int m = i % 2;
        int n = j % 2;

        float temp1 = ((pixel * 1.0) / maximum);
        float temp2 = ((ditherBayer[m][n] * 1.0) / 5);

        (temp1 > temp2) ? pixel = 255 : pixel = 0;

        ditheringImage->SetPixel(pixelIndex, pixel);

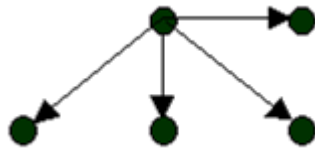
    }
}
```

DITHERING



DITHERING

- Algoritmo aperiódico (Floyd-Steinberg).



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

if($f[x][y] < 128$):

$p[x][y] = 0$

else:

$p[x][y] = 255$

$erro = p[x][y] - f[x][y]$

$f[x+1][y] = f[x+1][y] + erro * \frac{7}{16}$

$f[x][y+1] = f[x][y+1] + erro * \frac{5}{16}$

$f[x+1][y+1] = f[x+1][y+1] + erro * \frac{1}{16}$

$f[x-1][y+1] = f[x-1][y+1] + erro * \frac{3}{16}$

DITHERING

- Algoritmo aperiódico (Floyd-Steinberg).



REFERÊNCIAS

- GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento digital de imagens**. Pearson, 2011.
- PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações**. Thomson Learning, 2008.
- SILVA, Aristófanés. **Notas de aula da disciplina Processamento de Imagens da Universidade Federal do Maranhão**. 2018.
- BRAZ Jr, Geraldo. **Notas de aula da disciplina Visão Computacional da Universidade Federal do Maranhão**. 2018.