

Capítulo 4

Operações em Imagens

Computação Gráfica - Vol. 2 - Cap. 4

Capítulo 4

- 4.1. Operações Pontuais**
- 4.2. Operações Locais**
- 4.3. Operações Globais**
- 4.4. Transformações Geométricas**

Computação Gráfica - Vol. 2 - Cap. 4

Operações em imagens são realizadas:

- pontualmente nos *pixels*;
- em partes da imagem; e,
- em toda a imagem

Computação Gráfica - Vol. 3 - Cap. 4

4.1. Operações Pontuais

4.1.1. Operações Aritméticas

4.1.2. Operações Lógicas

Computação Gráfica - Vol. 4 - Cap. 4

4.1. Operações Pontuais

O *pixel*, na posição (x_i, y_i) , da imagem resultante depende apenas do *pixel* na imagem original.

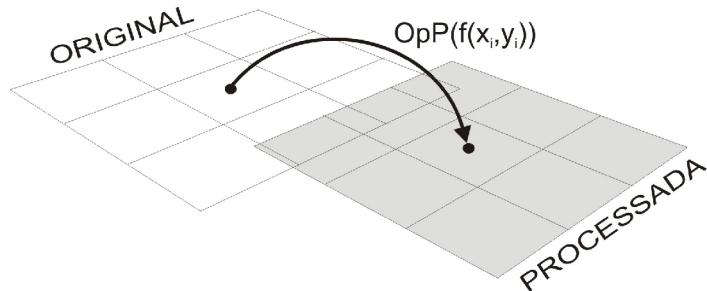


Figura 4.1 – Esquema de operações pontuais em imagens.

Computação Gráfica - Vol. 3 - Cap. 4

Algumas operações que alteram apenas características de cor ou luminância:

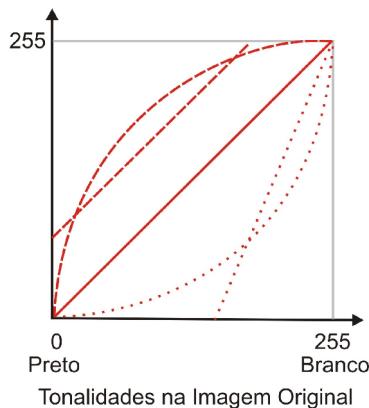


Figura 4.2 – Esquema de Mudanças de Tons para Imagem em 256 tons de cinza.

Computação Gráfica - Vol. 3 - Cap. 4

O processamento pode levar em consideração dados globais da imagem, como por exemplo, o histograma.



As operações locais *pixel-a-pixel* de duas imagens podem ser descritas pela expressão:

$$Z = (X \text{ } OpP \text{ } Y) \quad (4.3)$$

OpP é um operador aritmético ou lógico.

Computação Gráfica - Vol. 2 - Cap. 4

4.1.1. Operações Aritméticas

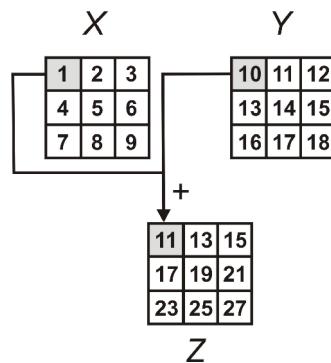
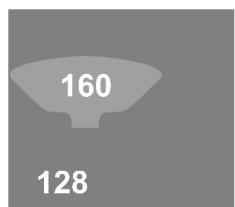
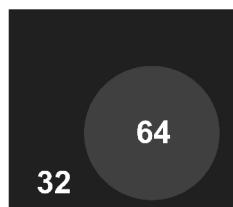


Figura 4.3 – Exemplo de operação aritmética de soma

Computação Gráfica - Vol. 2 - Cap. 4

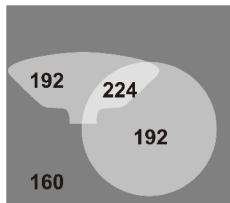


(a) X



(b) Y

Figura 4.4 – Imagens X e Y utilizadas como exemplos.



(a) $X+Y$



(b) $X-Y$

Figura 4.5 – Exemplos de operações aritméticas com as imagens da Figura 4.4.

Computação Gráfica - Vol. 9 - Cap. 4

4.1.1.1. Limites Inferior e Superiores nas Operações

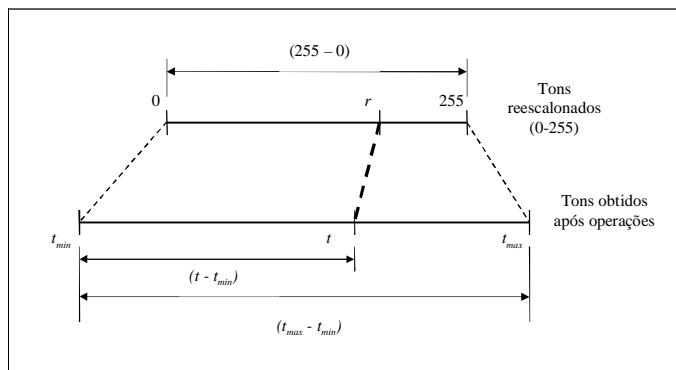


Figura 4.6 – Reescalonamento em casos de *underflow* e de *overflow*.

Computação Gráfica - Vol. 10 Cap. 4

4.1.2. Operações Lógicas

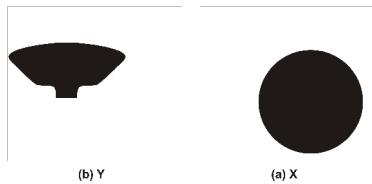


Figura 4.7 – Objetos X e Y utilizados como modelo.

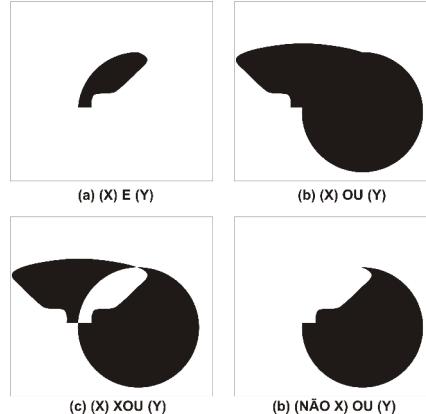


Figura 4.8 - Exemplos de operações lógicas com objetos da figura 4.10.

Computação Gráfica - Vol. 21 Cap. 4

4.2. Operações Locais

Um *pixel* da imagem resultante depende de uma vizinhança do mesmo *pixel* na imagem original

$$\text{OpL}(f(x_i, y_i), f(x_i-1, y_i-1), f(x_i+1, y_i+1), f(x_i-1, y_i+1), f(x_i+1, y_i-1), \dots)$$

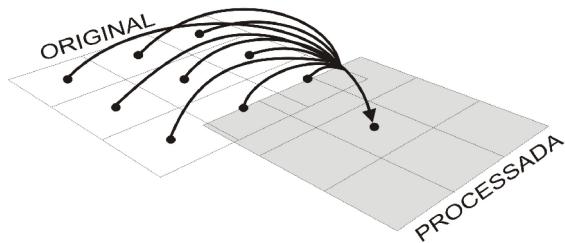


Figura 4.9 – Exemplo de uma operação local em uma área em torno do *pixel* (x_i, y_i) .

Computação Gráfica - Vol. 22 Cap. 4



(a)

(b)

(c)

Figura 4.10 - Redução de ruídos na imagem. (a) Imagem com ruído.
(b) Redução do ruído usando filtro de *blur*. (c) Imagem original.

Computação Gráfica - Vol. 23 Cap. 4

4.2.1. Forma de atenuar o efeito de *aliasing*

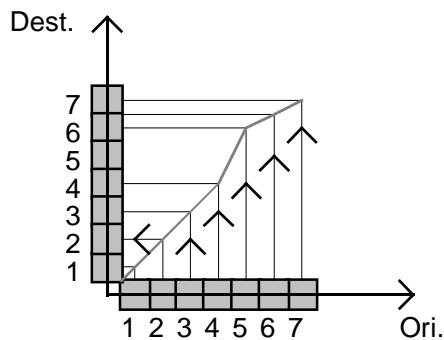


Figura 4.11 - Efeito da expansão e contração no domínio discreto.

Computação Gráfica - Vol. 24 Cap. 4

Filtro de média:

$$g(x_i, y_i) = \frac{1}{9} \left(f(x_{i-1}, y_{i-1}) + f(x_i, y_{i-1}) + f(x_{i+1}, y_{i-1}) + \right. \\ \left. f(x_{i-1}, y_i) + f(x_i, y_i) + f(x_{i+1}, y_i) + \right. \\ \left. f(x_{i-1}, y_{i+1}) + f(x_i, y_{i+1}) + f(x_{i+1}, y_{i+1}) \right) \quad (4.2)$$



(c)

(d)

Figura 4.12 – Imagem com Aliasing (c) e o efeito da aplicação de filtro de média (d).

Computação Gráfica - Vol. 25 Cap. 4

4.3. Operações Globais

Um *pixel* da imagem resultante depende de um processamento realizado em todos os *pixels* da imagem original.

Transformadas de: Fourier, Wavelet, Hough, Cosenos (usada para codificação) e funções interativas ou fractal.

Computação Gráfica - Vol. 26 Cap. 4

4.4. Transformações Geométricas

Levam o tom do *pixels* na posição (x_o, y_o) da *imagem origem*, para outra posição (x_d, y_d) do espaço em uma *imagem destino*

Computação Gráfica - Vol. 27 Cap. 4

4.4. Transformações Geométricas

4.4.1. Translação, Rotação e Escala

4.4.2 Espelhamento ou reflexão

4.4.3. Deformações e *Morphing*

Computação Gráfica - Vol. 28 Cap. 4

4.4.1. Translação, Rotação e Escala



(a)

(b)

Figura 4.13 – Exemplo de translação da imagem. (a)
Imagen Original. (b) Imagen Transladada.

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

(4.3)

Computação Gráfica - Vol. 19 Cap. 4



(a)

(b)

(c)

Figura 4.14 – Exemplo de ampliação e redução da
imagem. (a) Imagem Original. (b) Imagem Ampliada 2
vezes. (c) Imagem Reduzida pela metade

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

(4.4)

Computação Gráfica - Vol. 20 Cap. 4



$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

(4.5)

Figura 4.15 – Exemplo de Rotação de 90º no sentido horário

Computação Gráfica - Vol. 21 Cap. 4

Rotação em duas passadas:

$$x_d = x_o \cos \theta - y_o \sin \theta$$

$$y_d = x_o \sin \theta + y_o \cos \theta$$

$$y_d = x_d \tan \theta + y_o \cos \theta$$

$$x_d = x_o \cos \theta - y_d \tan \theta$$

(4.6)

(4.7)



(a)



(b)



(c)

Figura 4.16 - Rotação de 45 graus da imagem. (a) Imagem original. (b) 1^a passada. (c) Resultado final (2^a passada).

Computação Gráfica - Vol. 22 Cap. 4

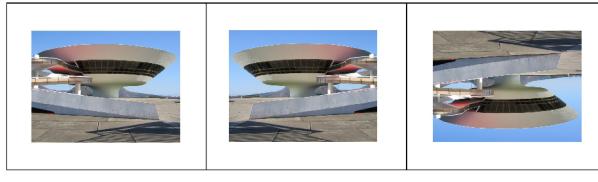
4.4.2 Espelhamento ou reflexão

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

(4.8)

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}$$

(4.9)



(a)

(b)

(c)

Figura 4.17 – Exemplo de espelhamento. (a) Imagem Original. (b) Flip Horizontal. (c) Flip Vertical.

Computação Gráfica - Vol. 23 Cap. 4

4.4.3. Deformações e *Morphing*

Deformação: considera que a imagem está por segmentos de reta com seus respectivos vértices e altera a forma dos objetos a partir de mudanças de posição dos segmentos envolventes.

Morphing: considera o processo de deformação aliado à decomposição de suas cores.

Computação Gráfica - Vol. 24 Cap. 4

4.4.3.1 Deformações

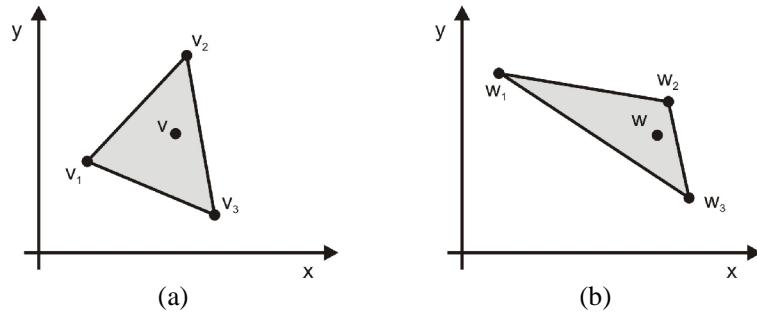


Figura 4.18 - Deformação simples de uma região triangular com pontos de vértice não-colineares. (a) $v = c_1v_1 + c_2v_2 + c_3v_3$. (b) $w = c_1w_1 + c_2w_2 + c_3w_3$

Computação Gráfica - Vol. 25 Cap. 4

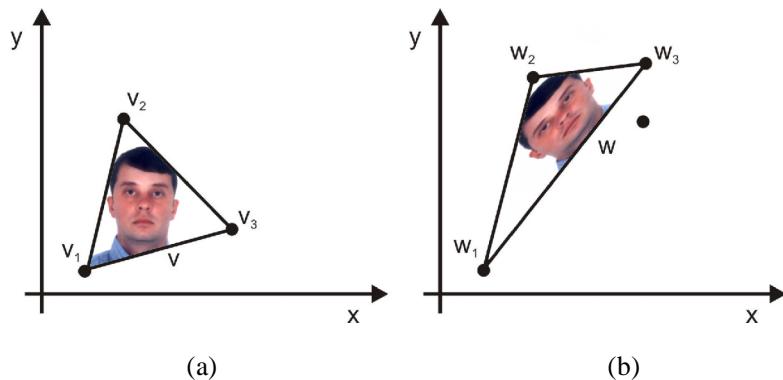


Figura 4.19 - Triângulo inicial com uma imagem. (a) Imagem original: $v = c_1v_1 + c_2v_2 + c_3v_3$. (b) Imagem transformada: $w = c_1w_1 + c_2w_2 + c_3w_3$.

Computação Gráfica - Vol. 26 Cap. 4

Pode-se repartir uma imagem em várias regiões triangulares e deformar cada região de uma maneira diferente:

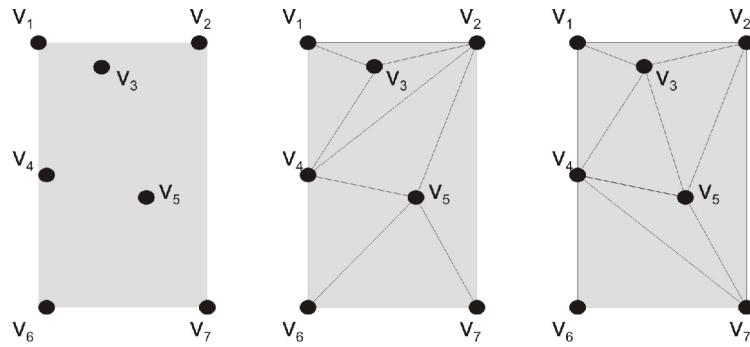


Figura 4.20 – Fazendo Triangulações

Computação Gráfica - Vol. 27 Cap. 4

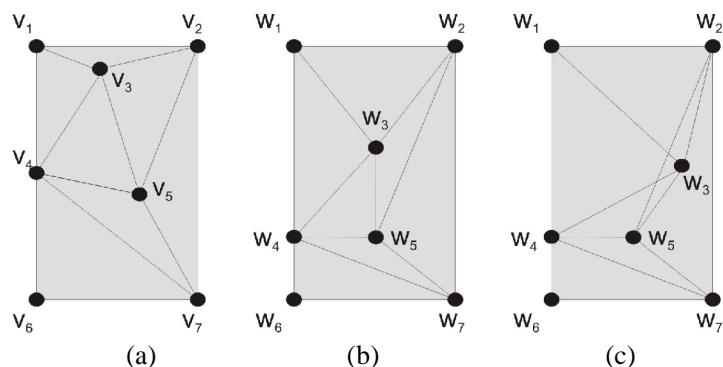


Figura 4.21 – Deformações por movimento de pontos dos vértices permitida(b) e não permitida (c)

Computação Gráfica - Vol. 28 Cap. 4

4.4.3.2. Deformações Dependentes do Tempo

É um conjunto de deformações geradas quando os pontos de vértice da imagem inicial são movidos continuamente ao longo do tempo desde suas posições originais até posições finais especificadas.

$$u_i(t) = (I - t)v_i + t w_i \quad (4.15)$$

Computação Gráfica - Vol. 29 Cap. 4

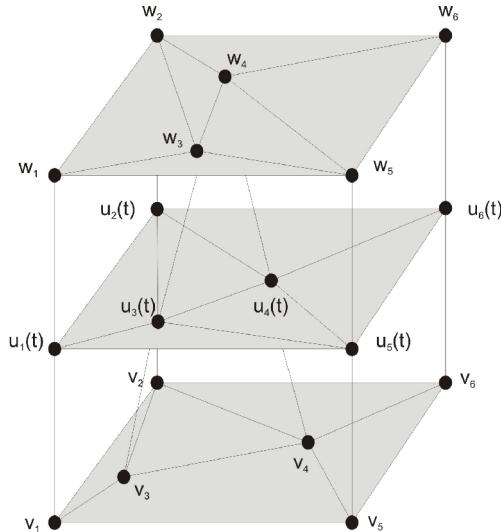


Figura 4.22 – Triangulação dependente do tempo de uma região

Computação Gráfica - Vol. 29 Cap. 4

4.3.3.3. *Morphing*

O termo *morph* tem como origem a palavra grega *morhos* que significa forma, sendo a ciência que estuda as formas chamada de *Morfologia*. *Morphing* é uma redução da palavra *metamorfose*.

O *morphing* envolve simultaneamente dois tipos de transformação:

- (a) de deformação (*warping*);
- (b) de tons (*cross-dissolve ou decomposição cruzada*).

Computação Gráfica - Vol. 31 Cap. 4

```
novo_red  =(r_origem+r_destino)/2  
novo_green=(g_origem+g_destino)/2  
novo_blue =(b_origem+b_destino)/2
```

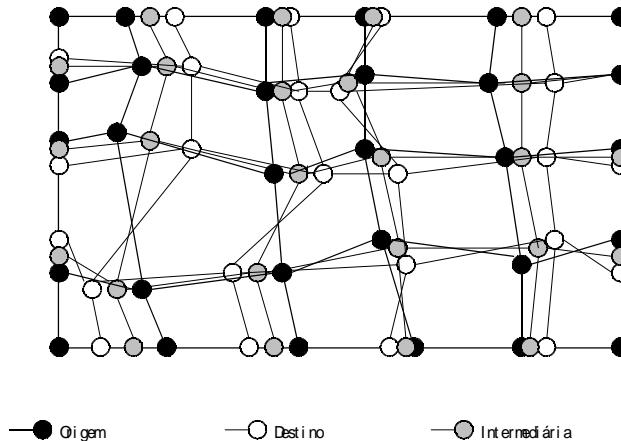


Figura 4.23 - Transformação de *pixels* origem em destino.

Computação Gráfica - Vol. 32 Cap. 4

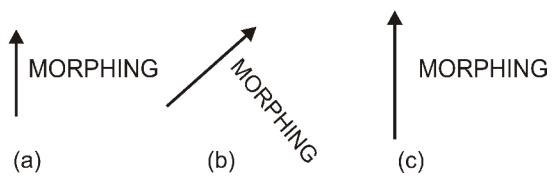


Figura 4.24 - Efeitos decorrentes de alterações em uma única linha de controle.

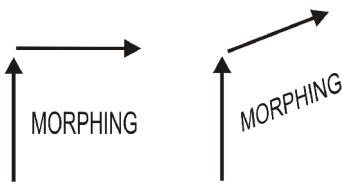


Figura 4.25 - Efeito de duas linhas de controle.

Computação Gráfica - Vol. 23 Cap. 4

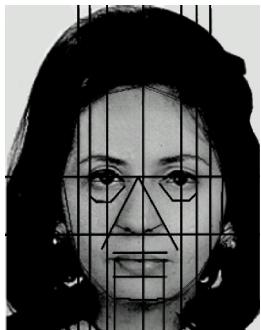


Figura 4.26 - Segmentos de Controle e linhas auxiliares.

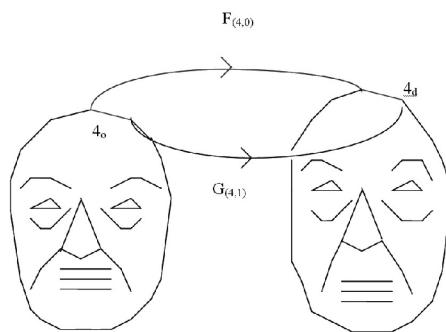


Figura 4.27 - Exemplo de interpolação de segmentos.

Computação Gráfica - Vol. 24 Cap. 4