Curso: Bacharelado em Ciência da Computação

Disciplina: Processamento de Imagens

Conteúdo dos Slides:

- Transformações Geométricas
  - Translação
  - Rotação
  - Escala
  - Cisalhamento
  - Interpolação

Profa. Regina Célia Coelho - rccoelho@unifesp.br

Introdução

- Transformações geométricas podem ser usadas para:
  - Alterar orientação de cenários (ex: fotografias);
  - Registro de Imagens (ex: imagens de satélite e médicas);
  - Gerar imagens artísticas (ex: montagem de comercial);
  - Inserir efeitos de distorção de objetos (ex: distorcer face e inserir sorriso).

Introdução

- Numa transformação geométrica, nem sempre os *pixels* transformados coincidem com a grade da imagem.

- Neste caso é necessário transformar as coordenadas pseudodiscretas em discretas, ou seja, é preciso calcular os valores dos pixels finais em função dos transformados.

Introdução (cont.)

- Para chegar ao resultado final é preciso:

  - Determinar as coordenadas de cada *pixel(i,j)* pertencente à grade original e à grade transformada (*i',j'*). Em geral, estes *pixels* transformados não são valores inteiros e, portanto, não coincidem com *pixels* da grade destino.

  - Calcular os valores dos *pixels* (*x,y*) finais na grade destino a partir dos valores conhecidos dos *pixels* (*i',j'*).

Introdução *(cont.)*

- O primeiro passo depende de definir a transformação a ser aplicada, enquanto o 2º corresponde a uma operação de interpolação.

Introdução (cont.)

- As operações geométricas podem ser aplicadas de forma direta ou inversa.

- Na aplicação direta, calcula-se as coordenadas da imagem transformada (f,g) em função das coordenadas da imagem original (x, y), percorrendo-se os pontos da imagem original e depois arredondando, se necessário, (f,g) para as coordenadas inteiras (u,v) mais próximas.

Introdução (cont.)

- Na aplicação inversa, calcula-se as coordenadas reais (f,g) na imagem original a partir das coordenadas discretas da imagem transformada (u,v), arredondando (f,g) para as coordenadas inteiras (x,y) mais próximas, se necessário.

- A operação direta pode gerar mais de um ponto com as mesmas coordenadas discretas transformadas.

- Neste caso, alguns pixels na imagem transformada ficam sem correspondência gerando lacunas.

Introdução *(cont.)*

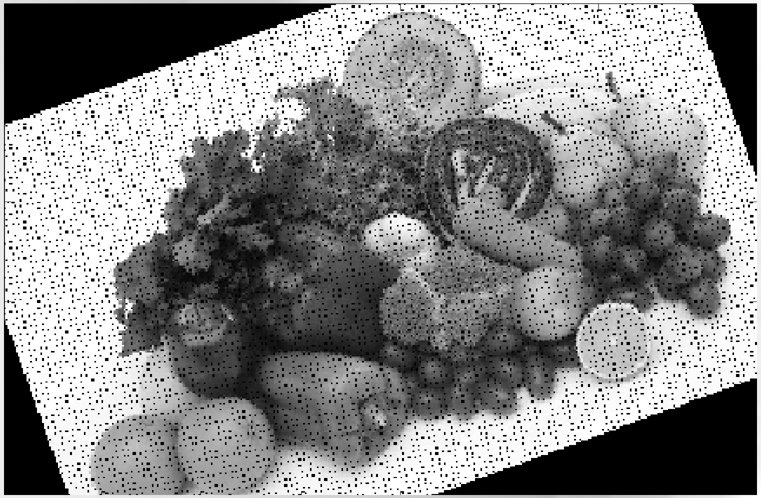
- Portanto, nesta aula utilizaremos as transformações inversas.

- Existem outras maneiras de determinar a intensidade dos pixels da imagem transformada com relação ao arredondamento de (*f*,*g*) para as coordenadas inteiras (*x*,*y*) mais próximas, que serão mostrados nessa aula.

Introdução (cont.)

- Exemplo de rotação de uma imagem por transformada direta e por transformada inversa.

imagem original



rotação direta

rotação inversa

Transformações Elementares

- As transformações sempre usam coordenadas homogêneas.
- Um ponto *(x,y)* é representado em coordenadas homogêneas como *(Sx,Sy,S)*, sendo *S* um fator de escala.
- Ex: o ponto *(12,6)* em coordenadas cartesianas, pode ser representado como *(24,12,2)* ou *(6,3,0.5)*.
- Por conveniência, esse fator de escala é sempre considerado com o valor 1.

Translação

- Consiste em deslocar o *pixel (i,j)* na imagem original para a posição $(i-d_x, j-d_y)$, mantendo a distância entre eles.

$$x = i - d_x$$
$$y = j - d_y$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -d_x \\ 0 & 1 & -d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}$$

Translação (cont.)

- Alguns pixels podem cair fora dos limites da imagem original após ela ser transladada.

Rotação

- Consiste em rotacionar o *pixel* (u,v) de um ângulo θ com relação à origem de coordenadas na imagem (ponto de coordenadas $(0,0)$).

- Toda a rotação pode ser reduzida a uma translação de (d_x,d_y), seguida de uma rotação de um ângulo ϴ em torno do ponto (0, 0) da imagem transladada, seguida de uma translação de $(-d_x,-d_y)$.

Rotação

- Consiste em rotacionar o *pixel (u,v)* de um ângulo θ com relação à origem de coordenadas na imagem (ponto de coordenadas (*0,0*)).

$$f = u*\cos(\theta) - v*\mathrm{sen}(\theta)$$

$$g = u*\mathrm{sen}(\theta) + v*\cos(\theta)$$

$$\begin{bmatrix} f \\ g \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Rotação *(cont.)*

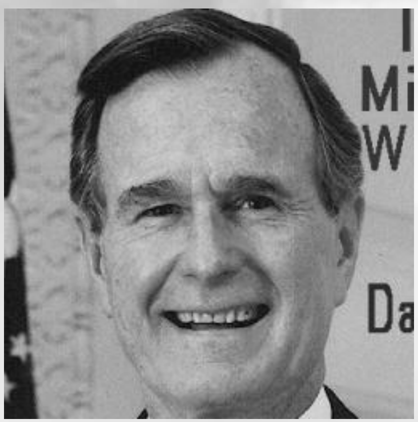
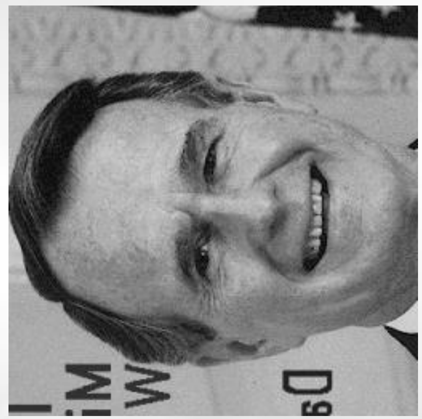
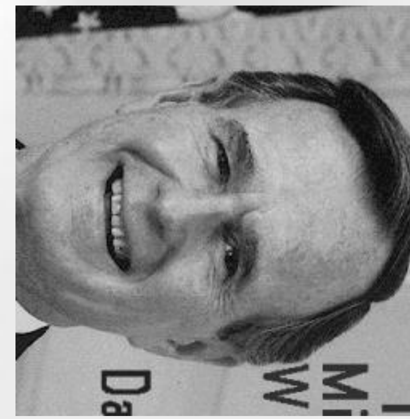


Imagem original



Rebatimento pela diagonal



Rotação de 90 graus em torno de (R/2,C/2)

Rotação *(cont.)*

- A rotação gera pontos que caem fora da imagem. Pode-se aumentar o tamanho da imagem para evitar perder informações.

- Rotação por transformada inversa em ângulo Θ=0.2 radianos.

Escala

- Consiste em variar o tamanho da imagem original (contrair ou expandir) em relação à origem (ponto de coordenadas(0,0)).
- Esta variação pode ser na largura e/ou altura.
- Fatores entre *0* e *1* reduz o tamanho da imagem.
- Fatores maiores que *1* ampliam a imagem.

$$f = u/S_x$$
$$g = v/S_y$$

$$\begin{bmatrix} f \\ g \\ 1 \end{bmatrix} = \begin{bmatrix} 1/S_x & 0 & 0 \\ 0 & 1/S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Escala

- A contração pode gerar efeitos indesejáveis se for realizada sem interpolação.

- Por exemplo, uma imagem listrada na qual cada linha contém apenas um pixel de espessura perde sua textura ao se reduzir sua resolução.

Escala

- Exemplo de alteração de escala com $S_x = 1.5$ e $S_y = 0.8$.

Escala *(cont.)*

- Um dos efeitos mais relevantes da escala é o *zoom*.
- Um método simples e intuitivo de realizar o *zoom* consiste em repetir os valores do *pixels* para criar um efeito de blocos.

Original

10	10
20	30

Ampliação por fator 3

10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10
20	20	20	30	30	30
20	20	20	30	30	30
20	20	20	30	30	30

Escala *(cont.)*

- Outro método consiste em encontrar o valor médio entre os *pixels* e usar este valor como o valor do *pixel* entre os dois. Este procedimento pode ser realizado em 2 passos: primeiro na linha e depois na coluna.

Escala (cont.)

Exemplo (amplia a imagem *NxN* para *(2N-1)*x*(2N-1)*)

$$\begin{bmatrix} 10 & 6 & 10 \\ 6 & 10 & 6 \\ 10 & 4 & 10 \end{bmatrix}$$

Imagem original

$$\begin{bmatrix} 10 & 8 & 6 & 8 & 10 \\ 6 & 8 & 10 & 8 & 6 \\ 10 & 7 & 4 & 7 & 10 \end{bmatrix}$$

Imagem com as linhas expandidas

$$\begin{bmatrix} 10 & 8 & 6 & 8 & 10 \\ 8 & 8 & 8 & 8 & 8 \\ 6 & 8 & 10 & 8 & 6 \\ 8 & 7.5 & 7 & 7.5 & 8 \\ 10 & 7 & 4 & 7 & 10 \end{bmatrix}$$

Imagem com as linhas e colunas expandidas

Redução

- Por eliminação de pixel
- Por Média

Original

10	10	10	10	10	10
13	14	16	17	18	19
17	19	21	23	25	28
20	23	27	30	33	37
23	27	33	37	41	46
27	32	38	43	48	55

Redução por fator 3

14	18
27	41

Nesta redução o valor considerado foi o do meio de cada submatriz.

Cisalhamento

- A operação de cisalhamento consiste em aplicar uma força em uma das extremidades de um objeto em uma direção fixando as outras extremidades, com exceção da extremidade oposta à força aplicada.

- Como resultado, uma imagem retangular fica na forma de um paralelogramo.

- Em uma imagem, o cisalhamento é aplicado na direção vertical das extremidades da direita, ou na direção horizontal das extremidades inferiores.

Cisalhamento

- Dado um pixel de coordenadas (u, v) em uma imagem cisalhada no eixo x por c_v, as coordenadas (x, y) na imagem original são os inteiros mais próximos dados por:

$$f = u - v * c_v$$

$$g = v$$

- Do mesmo modo, dado um pixel de coordenadas (u, v) em uma imagem cisalhada no eixo y por c_h, as coordenadas (x, y) na imagem original são os inteiros mais próximos dados por:

$$f = u$$

$$g = v - u * c_h$$

Cisalhamento

- O cisalhamento também pode ocorrer em ambas as direções, ao mesmo tempo. Neste caso, temos:

$$f = u - v * c_v$$

$$g = v - u * c_h$$

Cisalhamento

- O cisalhamento pode ser representado em forma de matriz de coordenadas homogêneas:

$$\begin{bmatrix} f \\ g \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -c_v & 0 \\ -c_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Para o cisalhamento vertical, basta atribuir $c_h = 0$ e para o cisalhamento horizontal, basta atribuir $c_v = 0$.

Cisalhamento

$c_v=0.2$

$c_h=0.25$

$c_v=0.25, \quad c_h=-0.1$

- Exemplos de cisalhamento vertical ($c_v=0.2$), horizontal ($c_h=0.25$) e composto ($c_v=0.25$, $c_h=-0.1$)

Transformações Afim

- Todas as operações geométricas vistas podem ser combinadas em uma única função matemática. Para isto, basta multiplicar as matrizes de transformação na ordem inversa em que se deseja realizar as operações.

- A matriz resultante realiza uma transformação denominada **transformação afim**.

Transformações Afim (cont.)

- Dado um pixel de coordenadas (*u*, *v*) em uma imagem transformada por uma operação afim, as coordenadas (x, y) na imagem original são os inteiros mais próximos dados por:

$$\begin{bmatrix} f \\ g \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{21} & t_{31} \\ t_{12} & t_{22} & t_{32} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Transformações Afim (cont.)

- Exemplo de transformação afim composta de translação d_x=-$N/2$, d_y=-$M/2$, seguida de rotação de ângulo Θ=0.2 e de translação d_x=$N/2$ e d_y=$M/2$.

- M e N são o número de colunas e linhas da imagem respectivamente.

- Esta transformação equivale a uma rotação no centro da imagem.

Interpolação

- Interpolação é o processo de usar dados conhecidos para estimar dados em locais onde são desconhecidos.

- A interpolação é utilizada, por exemplo, para estimar o valor de intensidade de uma determinada coordenada (u, v) após uma transformação geométrica.

- O valor de um *pixel* é calculado como uma função dos *pixels* vizinhos.

Interpolação

- Existem três métodos mais utilizados para interpolar:
  - Vizinho mais próximo;
  - Bilinear;
  - Bicúbico.

Interpolação pelo vizinho mais próximo

- Uma forma de efetuar este cálculo é tomar o valor do *pixel* mais próximo entre os 4 vizinhos que o rodeia na imagem (daí o nome). Para isso basta usar o cálculo da distância euclediana.

- Sejam *p(i,j)*, *p(i+1,j)*, *p(i,j+1)* e *p(i+1,j+1)* os 4 vizinhos de *p(x,y)*, sendo que *p(x,y)* é o *pixel* que se deseja como resultado e (x,y) são valores em ponto flutuante. Verifica qual dos vizinhos está mais próximo de *p(x,y)* e toma o valor deste *pixel*.

Interpolação pelo vizinho mais próximo (cont.)

- Isto nada mais é que arredondar as coordenadas e tomar o brilho deste ponto como o valor resultante do *pixel* da imagem transformada.

Interpolação pelo vizinho mais próximo (cont.)

- Desvantagens:
  - efeito de blocos;
  - tende a perder detalhes da imagem.

Interpolação pelo vizinho mais próximo (cont.)

- Vantagens:
  - processamento rápido;
  - não cria novos níveis de cinza (NC).

Interpolação Bilinear

- A interpolação bilinear define que a intensidade nas coordenadas (u,v) de uma imagem transformada é calculada como:

$$I(u,v) = I(x,y) = ax + by + cxy + d.$$

- *a, b, c* e *d* são escalares calculados por um sistema de quatro equações na forma:

$$I(x',y') = ax' + by' + cx'y' + d,$$

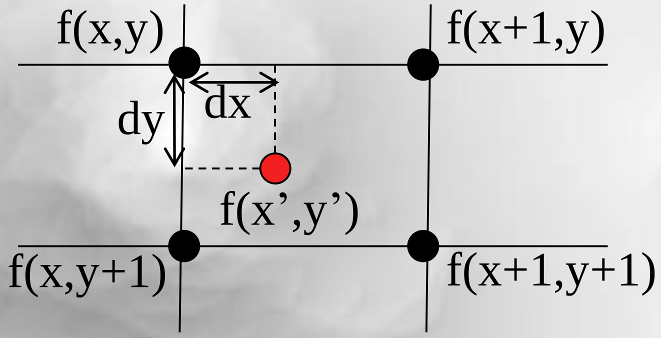
sendo que (x', y') são as quatro coordenadas inteiras mais próximas às coordenadas reais (x, y).

Interpolação Bilinear *(cont.)*

- Assim, a interpolação bilinear utiliza uma média ponderada dos quatro pixels vizinhos mais próximos para determinar a intensidade de cada pixel (x',y') na imagem transformada.

- Segundo Hélio Pedrini, ela pode ser calculada como:

  $$I(x',y') = (1-dx)(1-dy)\,f(x,y) + dx\,(1-dy)\,f(x+1,y) + (1-dx)\,dy\,f(x,y+1) + dx\,dy\,f(x+1,y+1).$$



Interpolação Bilinear (cont.)

- A forma mais simples de se realizar a interpolação bilinear para escala é mostrada a seguir.

Original

10	10
20	30

Interpolação nas linhas

Passos de níveis de cinza:

10 a 10: 0

20 a 30: (30-20)/3 = 3,3

Ampliação por fator 3

10	10	10	10	10	10
20	23	27	30	33	37

Interpolação Bilinear *(cont.)*

Original

10	10
20	30

Interpolação nas colunas

Passos de níveis de cinza:

10 a 20: (20-10)/3 = 3,3

10 a 23: (23-10)/3 = 4,3

10 a 27: (27-10)/3 = 5,7

...

Ampliação por fator 3

10	10	10	10	10	10
13	14	16	17	18	19
17	19	21	23	25	28
20	23	27	30	33	37
23	27	33	37	41	46
27	32	38	43	48	55

Interpolação Bilinear *(cont.)*

- Desvantagens:
  - Custo computacional maior;
  - Produz borramento.

- Vantagem:
  - Preserva a intensidade média;
  - Reduz contraste (reduz efeito serrilhado);
  - Apresenta um resultado melhor que o método anterior, pois produz um efeito de suavização devido à operação de média.

Interpolação Bilinear (cont.)

Interpolação pelo vizinho mais próximo

Interpolação Bilinear

Interpolação Bilinear (cont.)

- Exemplo: Ampliação por fator 10

Original

Replicação

Interpolação

Interpolação Bicúbica

- A interpolação bicúbica define que a intensidade nas coordenadas (u, v) de uma imagem transformada é:

  $I(u, v) = I(f, g) = \Sigma_{(i=0 \text{ to } 3)}[\ \Sigma_{(j=0 \text{ to } 3)}(\ a_{ij} * f^i * g^j\ )\ ].$

- a_{ij} são escalares calculados por um sistema de 16 equações na forma:

  $I(x', y') = \Sigma_{(i=0 \text{ to } 3)}[\ \Sigma_{(j=0 \text{ to } 3)}(\ a_{ij} * (\ x')^i * (y')^j\ )\ ],$

  em que (x', y') são as 16 coordenadas inteiras mais próximas das coordenadas (f, g).

Interpolação Bicúbica (cont.)

- Uma função comumente usada para determinar as intensidades do pixel na imagem interpolada é a função B-Spline cúbica, definida como:

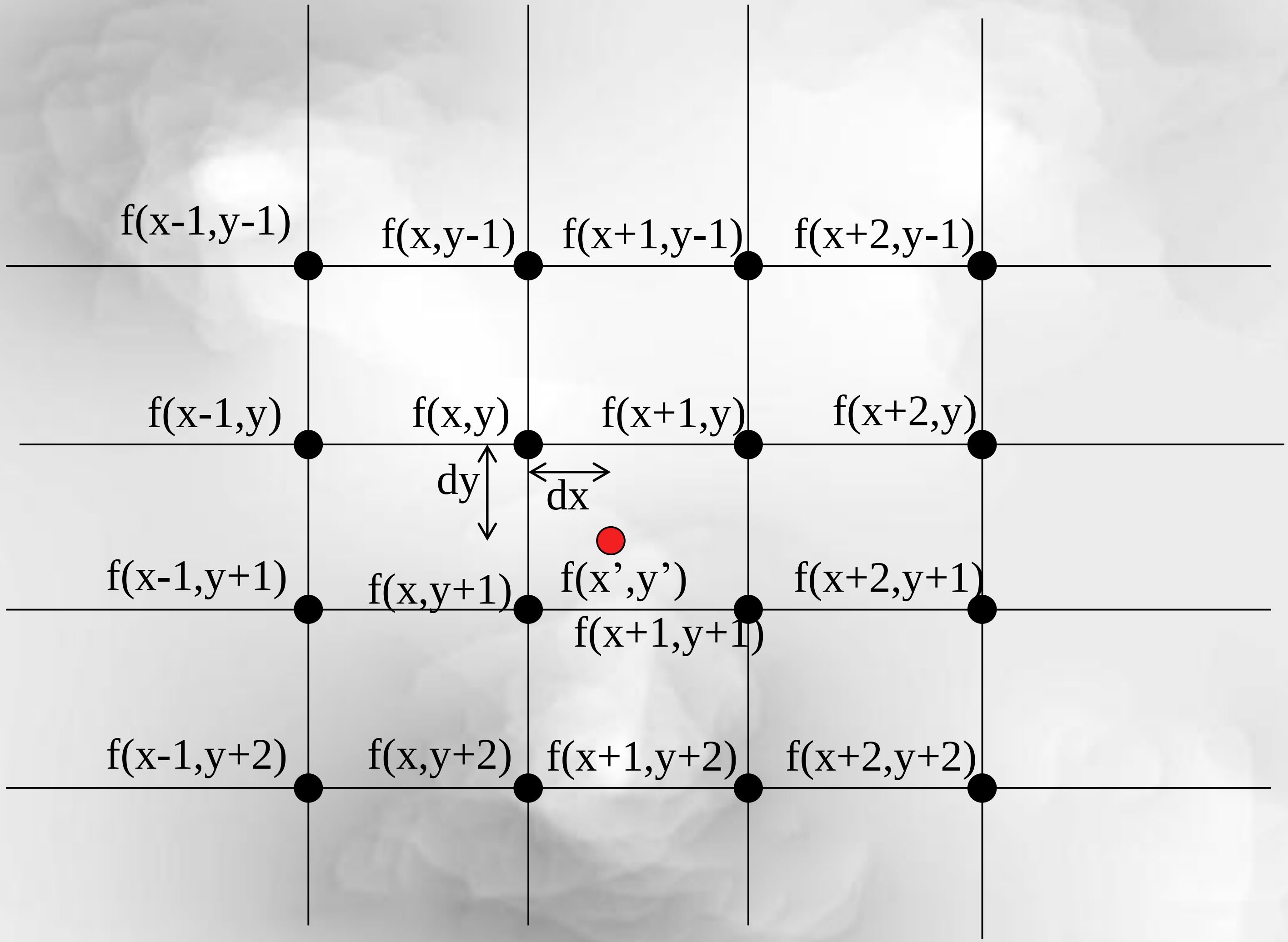
$$I(x', y') = \Sigma_{(m=-1 \text{ to } 2)}\Sigma_{(n=-1 \text{ to } 2)}\, f(x+m,y+n)\, R(m-dx)\, R(dy-n)$$

sendo

$$R(s) = [P(s+2)^3 - 4P(s+1)^3 + 6P(s)^3 - 4P(s-1)^3]/6$$

$$P(t) = \begin{cases} t, & t > 0 \\ 0 & c.c. \end{cases}$$

Interpolação Bicúbica (cont.)



Interpolação Bicúbica *(cont.)*

- Apresenta um resultado melhor que as outras duas interpolações.
- Considera 16 pontos vizinhos do *pixel* que se está interpolando.
- Desvantagens:
  - Computacionalmente mais cara;
  - Introduz novos níveis de cinza.
- Vantagem:
  - Apresenta um resultado melhor, pois produz imagem com aparência mais natural (melhora o efeito de borramento).

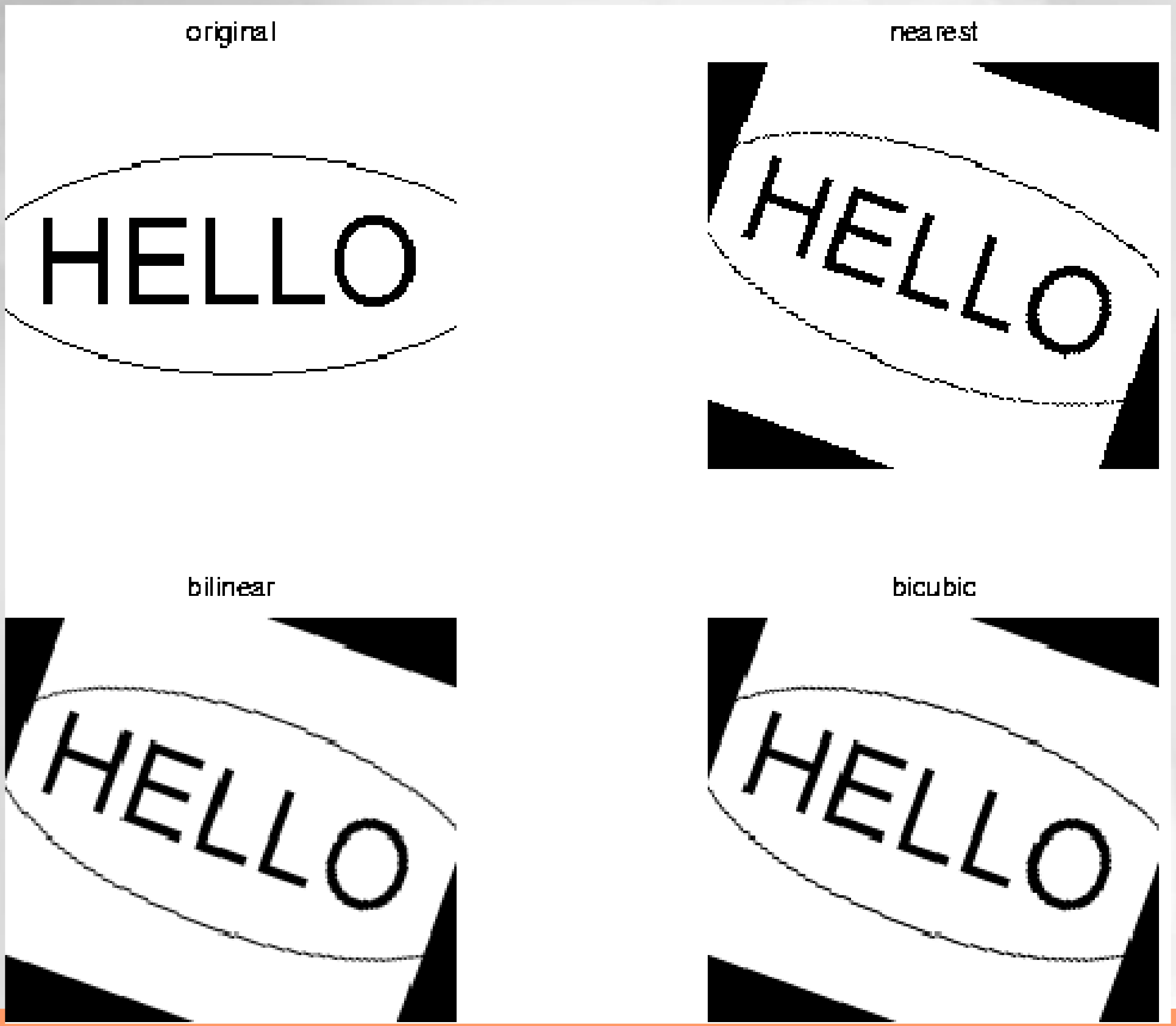
Interpolação Bicúbica (cont.)

Interpolação pelo vizinho mais próximo

Interpolação Bilinear

Interpolação Bicúbica

Outros Exemplos de Interpolação



Mapeamentos espaciais

- Quando as transformações aplicadas nas imagens são conhecidas, a recuperação da imagem original a partir da imagem transformada é simples. Basta aplicar as transformações inversas na imagem transformada.

- No entanto, nem sempre esta transformação é conhecida. Para estes casos, o método mais usado é formular a realocação espacial dos *pixels* pelo uso de <u>pontos de amarração</u>.

Mapeamentos espaciais *(cont.)*

- Pontos de amarração são pontos que são conhecidos na imagem distorcida e na imagem corrigida.

- Considere que uma imagem *f*, com coordenadas de *pixels* *(x,y)* sofra distorção geométrica produzindo uma imagem *g* com coordenadas *(x',y').* Essa transformação pode ser expressa como:

$$x' = r(x,y) \qquad y' = s(x,y)$$

Mapeamentos espaciais *(cont.)*

- Suponha que o processo de distorção geométrica considerando os 4 vizinhos possa ser modelado como:

$$r(x,y)= c_1x + c_2y + c_3xy + c_4$$

$$s(x,y)= c_5x + c_6y + c_7xy + c_8$$

ou seja,

$$x'= c_1x + c_2y + c_3xy + c_4$$

$$y'= c_5x + c_6y + c_7xy + c_8$$

Mapeamentos espaciais *(cont.)*

- Havendo quatro pontos de amarração conhecidos, essas equações são facilmente resolvidas para os 8 coeficientes (4 coeficientes para x e 4 para y) c_i, $i=1, 2, ..., 8$. Os coeficientes constituirão o modelo utilizado para transformar todos os *pixels* dentro da região quadrilátera caracterizados pelos pontos de amarração utilizados para se obter os coeficientes.

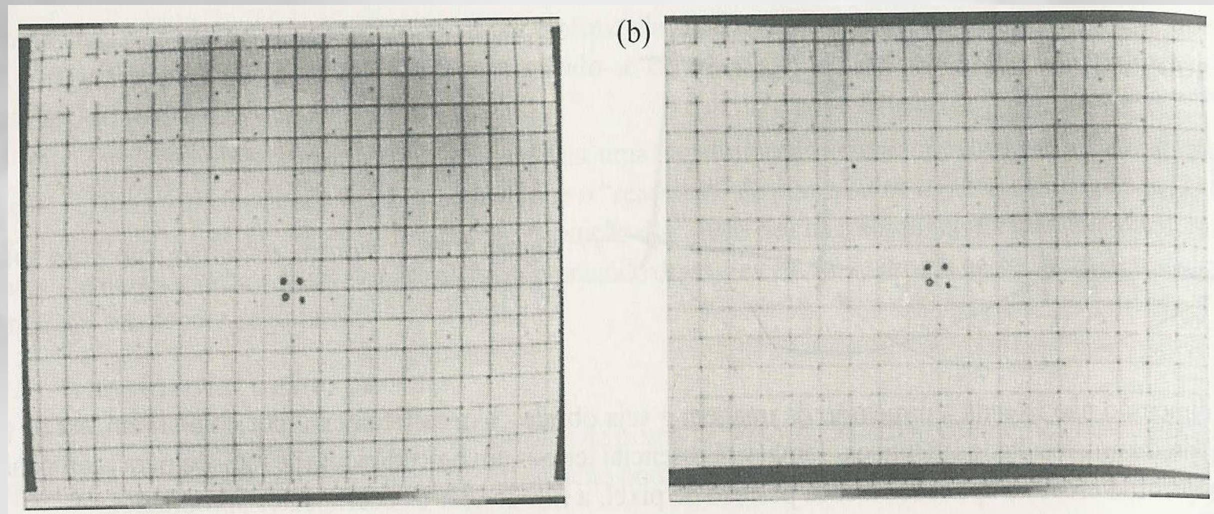
Mapeamentos espaciais *(cont.)*

- Gerando a imagem corrigida:
  - Para gerar *g(0,0)*, substitua *(x,y) = (0,0)* nas equações anteriores e obtenha um par de coordenadas *(x',y')*.
  - Faça *g(x',y') = f(0,0)*.
  - A seguir, substitua *(x,y) = (0,1)* nas equações anteriores e obtenha um outro par de coordenadas *(x',y')*.
  - Faça *g(x',y') = f(0,1)* para estes novos valores, e assim por diante.
- Observe que os níveis de cinza não são alterados.

Mapeamentos espaciais *(cont.)*

- A diferença desta interpolação das outras vistas anteriormente é que, em vez de usar o valor do nível de cinza do vizinho mais próximo, é interpolado um valor de posição e é utilizado esse valor para atribuir o nível de cinza.

Importância da Interpolação na Aquisição de Imagens



- A primeira imagem está distorcida, exibindo a distorção em "barril" encontrada em muitas câmeras de imageamento. A grade retilínea é severamente distorcida, particularmente perto das bordas da imagem e não é uma distorção uniforme.
- A segunda imagem foi corrigida por mapeamento espacial e interpolação bilinear. Os pontos de amarração são os pontos escuros espalhados na imagem.

Referência

- Slides
  - Prof. Fábio Cappabianco – UNIFESP
- Livros
  - Pedrini, Hélio. *Análise de Imagens Digitais – Princípios, Algoritmos e Aplicações*, Thomson, 2008.