

## Trabalho I

### 1 Especificação do Problema

Escreva um programa para realizar as transformações geométricas de escala e rotação em uma imagem. O fator de escala e o valor do ângulo de rotação deve permitir valores contínuos (ou seja, valores em ponto flutuante).

Um modo de se ampliar uma imagem é mapear cada posição dos pontos da imagem de saída a partir da posição correspondente dos pontos na imagem de entrada. Por exemplo, se o fator de escala é 2.25, então a posição de saída do pixel  $P_o = (10, 23)$  seria mapeado para  $P_i = P_o/s = (10/2.25, 23/2.25) = (4.444, 10.222)$  na imagem de entrada. Para determinar o valor do pixel em  $P_i$ , utilize pelo menos três métodos de interpolação, por exemplo, através do *vizinho mais próximo*, *bilinear* e *bicúbica*.

#### 1.1 Interpolação pelo Vizinho Mais Próximo

Na interpolação pelo vizinho mais próximo, o valor da intensidade a ser atribuído ao pixel  $(x', y')$  na imagem reamostrada terá o mesmo valor do pixel que estiver mais próximo da posição ocupada pelo pixel  $(x, y)$  na imagem original, conforme mostrado na figura 1.

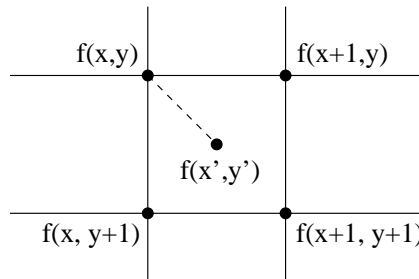


Figura 1: Interpolação pelo vizinho mais próximo.

A interpolação pode ser expressa como

$$f(x', y') = \begin{cases} f(x, y), & \text{para } dx < 0.5 \text{ e } dy < 0.5 \\ f(x+1, y), & \text{para } dx \geq 0.5 \text{ e } dy < 0.5 \\ f(x, y+1), & \text{para } dx < 0.5 \text{ e } dy \geq 0.5 \\ f(x+1, y+1), & \text{para } dx \geq 0.5 \text{ e } dy \geq 0.5 \end{cases} \quad (1)$$

em que  $dx$  e  $dy$  correspondem às distâncias nas direções  $x$  e  $y$ , respectivamente, entre os pontos  $(x', y')$  e  $(x, y)$ , ou seja, apenas as partes fracionárias do ponto  $(x', y')$

$$\begin{aligned} dx &= x' - x = x' - \lfloor x' \rfloor \\ dy &= y' - y = y' - \lfloor y' \rfloor \end{aligned} \quad (2)$$

A equação 1 pode ser reescrita como

$$f(x', y') = f(\text{round}(x), \text{round}(y)) \quad (3)$$

em que *round* é uma função que aproxima um número para seu valor inteiro mais próximo.

## 1.2 Interpolação Bilinear

A interpolação bilinear utiliza uma média ponderada de distância dos quatro pixels vizinhos mais próximos para determinar a intensidade de cada pixel  $(x', y')$  na imagem transformada, como mostrado na figura 2.

A interpolação é dada por

$$f(x', y') = (1 - dx)(1 - dy) f(x, y) + dx(1 - dy) f(x + 1, y) + (1 - dx)dy f(x, y + 1) + dxdy f(x + 1, y + 1) \quad (4)$$

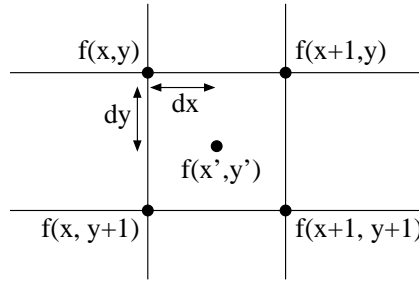


Figura 2: Interpolação bilinear.

## 1.3 Interpolação Bicúbica

A interpolação bicúbica utiliza uma vizinhança de  $4 \times 4$  pontos ao redor do ponto em questão para calcular seu valor de intensidade, como mostrado na figura 3. Uma função comum para calcular as intensidades do pixel na imagem interpolada é a função B-spline cúbica, definida como

$$f(x', y') = \sum_{m=-1}^2 \sum_{n=-1}^2 f(x + m, y + n) R(m - dx) R(dy - n) \quad (5)$$

sendo

$$R(s) = \frac{1}{6} [P(s + 2)^3 - 4P(s + 1)^3 + 6P(s)^3 - 4P(s - 1)^3] \quad (6)$$

$$P(t) = \begin{cases} t, & t > 0 \\ 0, & t \leq 0 \end{cases} \quad (7)$$

## 2 Especificação do Programa

Algumas opções do programa estão especificadas abaixo

```
prog [-a ângulo]
      [-e fator de escala]
      [-d largura altura]
      [-m interpolação]
      [-i imagem]
      [-o imagem]
```

onde os parâmetros são:

- a ângulo de rotação medido em graus no sentido anti-horário
- e fator de escala
- d dimensão da imagem de saída em pixels

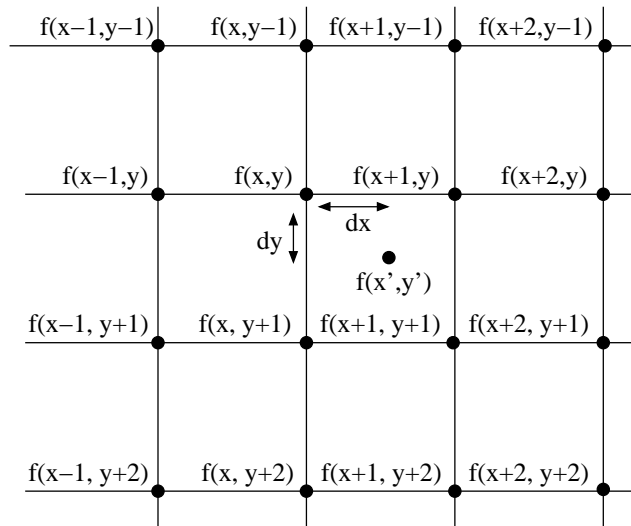


Figura 3: Interpolação bicúbica.

- m método de interpolação utilizado
- i imagem de entrada no formato PGM
- o imagem de saída no formato PGM (após a transformação geométrica)

O programa deve realizar apenas uma transformação geométrica (escala ou rotação) na imagem.

### 3 Especificação da Entrega

- A entrega do trabalho deve conter os seguintes itens:

- código fonte: a submissão dos programas deve ser realizada através da execução do seguinte *script*:

```
~helio/pi/bin/entregar <arquivo>.tgz
```

O arquivo final deve estar no formato *tar* compactado com *gzip*, contendo todos os programas necessários para sua execução. Favor incluir um *Makefile* para facilitar a compilação dos programas. A geração do arquivo pode ser realizada através do seguinte comando, considerando que todos os programas estão no diretório corrente:

```
tar -cvzf <arquivo>.tgz *
```

Envie o arquivo a partir da conta de usuário de um dos membros do grupo. Caso o arquivo seja enviado mais de uma vez, a última cópia irá se sobrepôr à anterior. Após o envio do arquivo, o *script* emitirá uma mensagem confirmando o sucesso da operação.

- relatório impresso: deve conter uma descrição dos algoritmos e estruturas de dados, considerações adotadas na solução do problema, testes executados, eventuais limitações ou situações especiais não tratadas pelo programa.

- Data de entrega: 13 de abril de 2007

**Observações Gerais:**

- O programa será compilado com *gcc* ou *g++* em ambiente Linux. Os formatos de entrada e saída dos dados devem ser rigorosamente respeitados pelo programa, conforme definidos anteriormente. Trabalhos entregues com atraso terão 10% da nota descontada por dia de atraso. Não serão aceitos trabalhos após 5 dias da data de entrega.
- Serão considerados os seguintes aspectos na avaliação: funcionamento da implementação, clareza, modularidade do código, uso de comentários descritivos no código.