

## Trabalho 4

### 1 Especificação do Problema

Este trabalho está dividido em duas partes, a primeira que realiza transformações geométricas de escala e rotação em uma imagem e a segunda que aplica técnicas de detecção de pontos de interesse para registrar um par de imagens e criar uma imagem panorâmica formada pela ligação entre as imagens após sua correspondência.

#### 1.1 Transformações Geométricas

O fator de escala e o valor do ângulo de rotação das transformadas devem 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 os métodos de interpolação descritos a seguir.

##### 1.1.1 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.

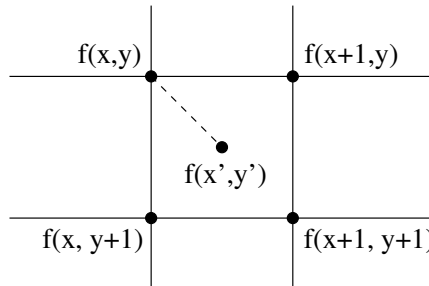


Figura 1: Interpolação pelo vizinho mais próximo.

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.

### 1.1.2 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

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

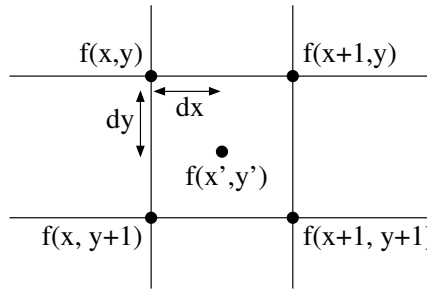


Figura 2: Interpolação bilinear.

### 1.1.3 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

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

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

### 1.1.4 Interpolação por Polinômios de Lagrange

Outro método que utiliza uma vizinhança de  $4 \times 4$  pontos para calcular o valor de intensidade de um pixel  $(x', y')$  é a interpolação por polinômios de Lagrange, definida como

$$f(x', y') = \frac{-dy(dy-1)(dy-2)L(1)}{6} + \frac{(dy+1)(dy-1)(dy-2)L(2)}{2} + \frac{-dy(dy+1)(dy-2)L(3)}{2} + \frac{dy(dy+1)(dy-1)L(4)}{6}$$

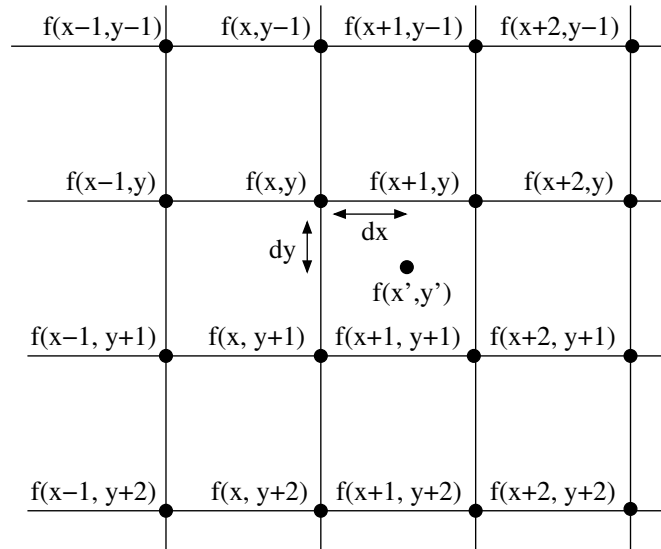


Figura 3: Interpolação bicúbica.

sendo

$$L(n) = \frac{-dx(dx-1)(dx-2)f(x-1, y+n-2)}{6} + \frac{(dx+1)(dx-1)(dx-2)f(x, y+n-2)}{2} + \frac{-dx(dx+1)(dx-2)f(x+1, y+n-2)}{2} + \frac{dx(dx+1)(dx-1)f(x+2, y+n-2)}{6}$$

### 1.1.5 Especificação do Programa

Uma sugestão para execução do programa é:

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

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

A cada execução, o programa deve realizar apenas uma transformação geométrica (escala ou rotação) na imagem. Os códigos não poderão utilizar comandos disponíveis em bibliotecas para realizar as operações descritas anteriormente, mas implementá-las de acordo com as formulações apresentadas.

### 1.1.6 Entrada de Dados

As imagens de entrada estão no formato PNG (*Portable Network Graphics*). Alguns exemplos encontram-se disponíveis no diretório: [http://www.ic.unicamp.br/~helio/imagens\\_png/](http://www.ic.unicamp.br/~helio/imagens_png/).

### 1.1.7 Saída de Dados

As imagens de saída devem estar no formato PNG (*Portable Network Graphics*). Resultados intermediários podem ser também exibidos na tela.

## 1.2 Registro de Imagens

Nesta etapa, uma imagem panorâmica deve ser gerada a partir da correspondência de um par de imagens. Pontos de interesse devem ser detectados e associados para o registro das imagens.

Os principais passos do processo de correspondência e geração da imagem panorâmica são listados a seguir:

- (1) converter as imagens coloridas de entrada em imagens de níveis de cinza.
- (2) encontrar pontos de interesse e descritores invariantes locais para o par de imagens.
- (3) computar distâncias (similaridades) entre cada descritor das duas imagens.
- (4) selecionar as melhores correspondências para cada descritor de imagem.
- (5) executar a técnica RANSAC (*RANdom SAmple Consensus*) para estimar a matriz de homografia (`cv2.findHomography`).
- (6) aplicar uma projeção de perspectiva (`cv2.warpPerspective`) para alinhar as imagens.
- (7) unir as imagens alinhadas e criar a imagem panorâmica.
- (8) desenhar retas entre pontos correspondentes no par de imagens.

No passo (2), explore e compare diferentes detectores de pontos de interesse e descritores, tais como SIFT (*Scale Invariant Feature Transform*), SURF (*Speed Up Robust Feature*), BRIEF (*Robust Independent Elementary Features*) e ORB (*Oriented FAST, Rotated BRIEF*). No passo (4), uma correspondência será considerada se o limiar definido estiver acima de um valor especificado pelo usuário. No passo (5), o cálculo da matriz de homografia requer o uso de, no mínimo, 4 pontos de correspondência.

A Figura 4 mostra um par de imagens de entrada e seus respectivos resultados. A matriz de homografia  $H$  para o exemplo mostrado é:

$$H = \begin{bmatrix} 1.15573599e + 00 & 1.09468342e + 00 & -2.19708966e + 02 \\ -1.09488996e + 00 & 1.15552234e + 00 & 2.13056040e + 02 \\ 3.70911895e - 07 & 3.34356720e - 07 & 1.00000000e + 00 \end{bmatrix}$$

### 1.2.1 Entrada de Dados

As imagens de entrada estão no formato JPEG (*Joint Photographic Experts Group*). Exemplos de pares de imagens podem ser encontrados em [http://www.ic.unicamp.br/~helio/imagens\\_registro/](http://www.ic.unicamp.br/~helio/imagens_registro/).

### 1.2.2 Saída de Dados

As imagens de saída, após o processo de registro e geração da imagem panorâmica, devem estar no formato JPEG (*Joint Photographic Experts Group*).

## 2 Especificação da Entrega

- A entrega do trabalho deve conter os seguintes itens:
  - código fonte: o arquivo final deve estar no formato *zip* ou no formato *tgz*, contendo todos os programas ou dados necessários para sua execução.
  - relatório: deve conter uma descrição dos algoritmos e das estruturas de dados, considerações adotadas na solução do problema, testes executados, eventuais limitações ou situações especiais não tratadas pelo programa.
- O trabalho deve ser submetido por meio da plataforma *Google Classroom*.
- Data de entrega: 24/06/2024.



(a) imagem A



(b) imagem B



(c) linhas de correspondência



(d) imagem panorâmica

Figura 4: Imagens de entrada e respectivos resultados.

### 3 Observações Gerais

- Os programas serão executados em ambiente Linux. Os formatos de entrada e saída dos dados devem ser rigorosamente respeitados pelo programa, conforme definidos anteriormente.
- Os seguintes aspectos serão considerados na avaliação: funcionamento da implementação, clareza do código, qualidade do relatório técnico.