

# Projeto 5

## MC920 - Introdução ao Processamento de Imagem Digital

Caio Augusto Alves Nolasco - RA:195181

### 1. Introdução e especificação do problema

O objetivo do projeto 5 é unir duas imagens para formar só imagem panorâmica, através de técnicas de detecção de pontos de interesse e descritores que ligam os pontos comuns entre as imagens.

### 2. Estrutura geral do algoritmo

O programa lê do disco do usuário as duas imagens que serão unidas em um panorama, assim como parâmetros importantes que ditam o funcionamento do algoritmo como o tipo de algoritmo de detecção de descritores e o limiar máximo para distância entre descritores das duas imagens. Cada imagem é convertida para a escala de cinza e submetida aos algoritmos de detecção de pontos de interesse e descritores (SURF, SIFT, ORB ou BRIEF). O conjuntos de descritores das duas imagens são comparados para encontrar as melhores correspondências. O conjunto de descritores escolhidos é usado para calcular a matriz de homografia, que é usada para alinhar as duas imagens. Por fim, as imagens são combinadas para formar a imagem panorâmica final.

### 3. Pressupostos e decisões de design

O programa em Python recebe uma imagem de entrada para aplicar os operadores morfológicos e encontrar as partes com texto. Sobre essa imagem são feitos os seguintes pressupostos:

- Duas imagens são fornecidas, via seu caminho endereço, pelo usuário;
- As imagens estão salvas no disco do computador que executa o código;
- O usuário informa qual algoritmo de detecção de pontos de interesse e descritores deve ser usado (SURF, SIFT, ORB, BRIEF);
- O usuário fornece o limiar limite de distância entre descritores para selecionar as melhores;
- As imagens estão em formato '.jpg';
- O local de origem dos arquivos é dado pelo usuário (ex: /home/cnolasco/foto1A.jpg e /home/cnolasco/foto1B.jpg);
- O local de destino das imagens intermediárias do processo e a imagem final é dado pelo usuário (ex: /home/cnolasco/Outputs/);
- As imagens resultantes são salvas em formato '.jpg'.

- **A versão usada da biblioteca OpenCV para testes é 3.4.2. Versões mais recentes, como a versão 4.3.0, não suportam alguns dos algoritmos de detecção por serem patenteados.**

#### 4. Padrões de implementação

A junção das imagens é feita usando principalmente funções da biblioteca OpenCV.

As imagens são lidas, com o endereço dado pelo usuário, como objetos da imagem OpenCV pelo método **cv2.imread()**.

As imagens então são convertidas para escala de cinza usando **cv2.cvtColor()**, e um objeto de detector de pontos de interesse e descritores é criado de acordo com o parâmetro lido (**cv2.xfeature2d.SIFT\_create()**, **cv2.xfeature2d.SURF\_create()**, **cv2.ORB\_create()** para SIFT, SURF, ORB respectivamente). O algoritmo BRIEF contém somente um detector de descritores, então outro algoritmo, STAR, é usada para complementar a detecção de pontos de interesse (**cv2.xfeature2d.BriefDescriptorExtractor\_create()** e **cv2.xfeature2d.StarDetector\_create()**). Os pontos de interesse são desenhados na imagem original para visualização, e junto com os descritores, são submetidos a comparação entre os resultados das duas imagens.

A comparação entre pontos de interesse das duas imagens é feita por um objeto BFMatcher da biblioteca OpenCV (**cv2.BFMatcher()**). Os descritores são analisados, e a distância entre os objetos correspondentes são calculados. Caso a distância respeite o limiar dado pelo usuário, o ponto de interesse é salvo em uma lista de pontos selecionados. A distâncias selecionadas são desenhadas para visualização.

O algoritmo requer que sejam encontrados ao menos quatro pontos de correspondência. Com os pontos selecionados, é calculada a matriz de homografia com a função **cv2.findHomography()**, baseado no algoritmo RANSAC. A matriz de homografia é impressa no terminal.

Por fim, as imagens são alinhadas com a função **cv2.warpPerspective()** a partir da matriz de homografia, e unidas atribuindo os valores em uma só matriz representando da imagem panorâmica

#### 5. Saída

O programa salva as seguintes imagens, para um algoritmo de detecção “feat”: as duas imagens com os pontos de interesse desenhados (“featAkeypointsDrawn.jpg” e “featBkeypointsDrawn.jpg”); um imagem com as linha que ligam os pontos de correspondências entre as imagens (“featMatches.jpg”); e a imagem panorâmica (“featfinalPanoramic.jpg”).

#### 6. Resultados intermediários e imagem panorâmica final

Para os algoritmos SIFT, SURF e BRIEF, o limiar usado é 0.3. Para o algoritmo ORB, o valor usado é 0.6.



---

## Pontos de interesse







a) Pontos de interesse SURF







b) Pontos de interesse SIFT





c) Pontos de interesse ORB



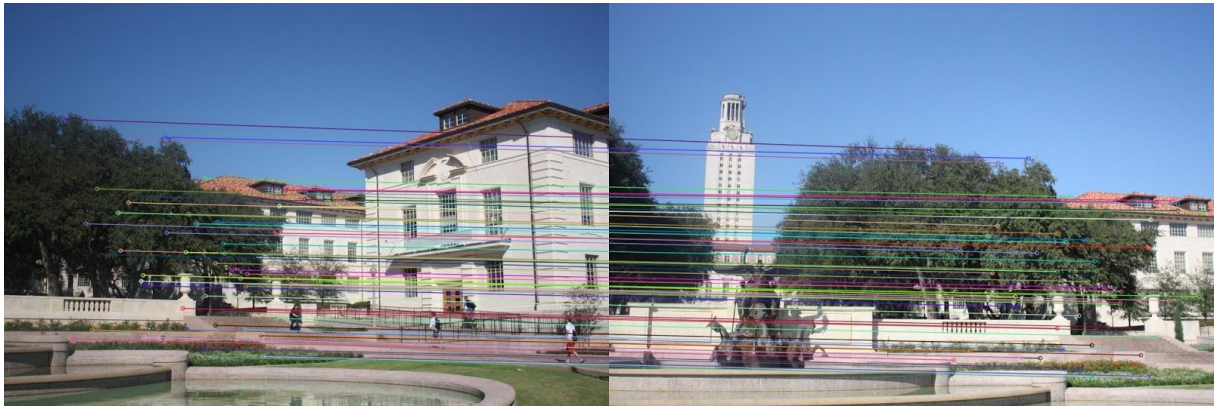


d) Pontos de interesse STAR & BRIEF



---

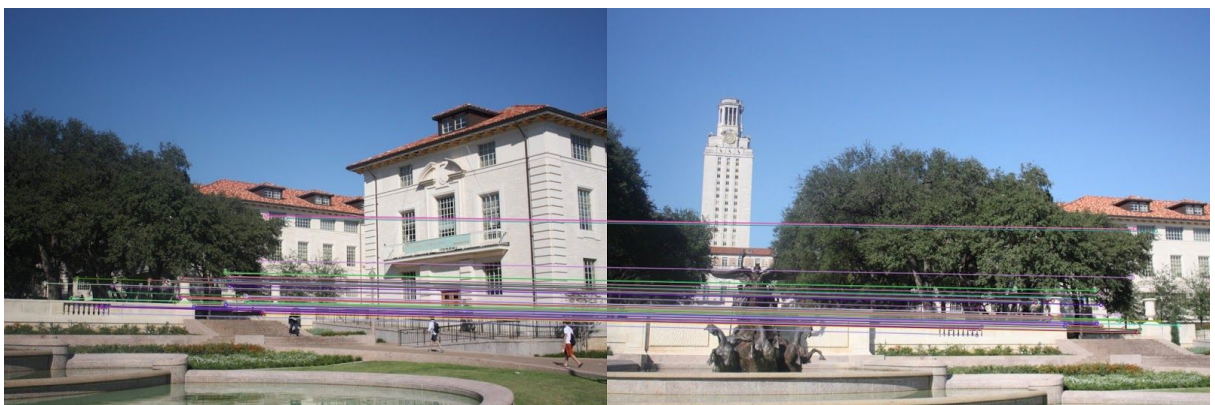
## Linhas entre pontos de correspondências selecionados



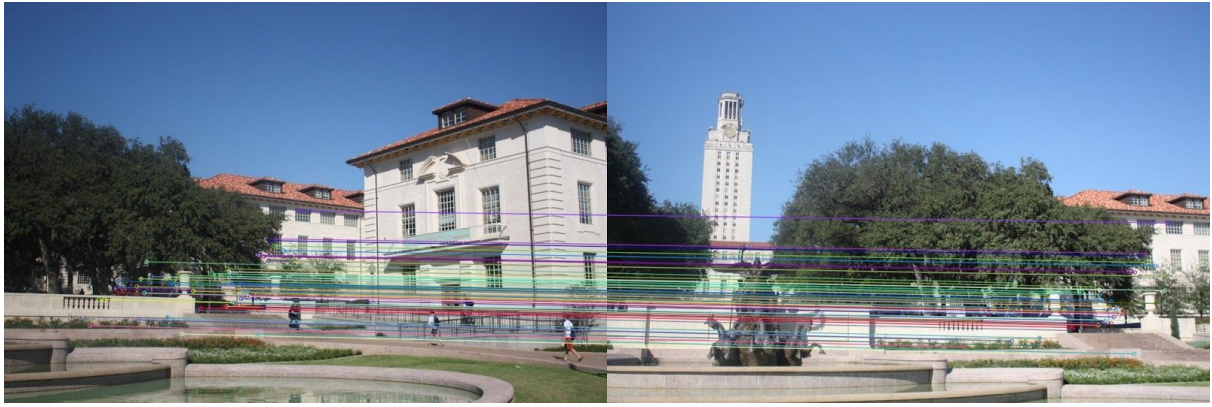
a) Correspondências SURF



b) Correspondências SIFT



c) Correspondências ORB



d) Correspondências BRIEF

---

## Panoramas finais



a) Panorama SURF



b) Panorama SIFT





c) Panorama ORB



d) Panorama BRIEF