

# Trabalho 4 - MO443 - Introdução ao Processamento de Imagem Digital

Aluno: Gabriel Bianchin de Oliveira RA: 230217

Junho 2019

## 1 Introdução

O objetivo desse trabalho é implementar técnicas de detecção de pontos de interesse de um par de imagens e criar uma imagem panorâmica formada pela ligação entre as imagens. As técnicas para a detecção de pontos utilizadas foram SIFT, SURF, ORB e BRIEF.

## 2 Código

O código foi implementado em Python 3.6.7, com as bibliotecas OpenCV 3.4.2 e NumPy 1.13.3.

### 2.1 Como Executar

Para a execução do código, deve-se executar o script `codigo.py`. O script recebe como argumento duas imagens no formato JPG, a técnica de detecção de pontos e o prefixo das imagens de saída. Um exemplo de execução é mostrado a seguir:

```
python3 codigo.py imagens/foto2A.jpg imagens/foto2B.jpg sift saida
```

As opções de técnicas utilizadas como parâmetro são:

- **sift**: aplica a técnica SIFT.
- **surf**: aplica a técnica SURF.
- **orb**: aplica a técnica ORB.
- **brief**: aplica a técnica BRIEF.

Caso nenhuma das opções válidas for informada, será enviada uma mensagem, via terminal, informando que não foi possível realizar a aplicação da técnica.

Durante a execução do código, será requisitado ao usuário, via terminal, o valor do limiar para encontrar correspondências. Caso não for possível, será

enviada uma mensagem, via terminal, informando que não foi possível encontrar as correspondências.

Ao final, serão salvas duas imagens, sendo que uma imagem será panorâmica e outra imagem conterá as retas entre os pontos correspondentes do par de imagens. Utilizando como exemplo a entrada acima, as imagens serão salvas como `saida-panoramica.jpg` e `saida-retas.jpg`.

## 2.2 Entrada

A entrada de dados consiste em duas imagens no formato JPG. As imagens utilizadas para testes estão na pasta `imagens`, e foram obtidas pelo seguinte diretório: [http://www.ic.unicamp.br/helio/imagens\\_registro/](http://www.ic.unicamp.br/helio/imagens_registro/)

## 2.3 Saída

A saída de dados consiste em duas imagens. Uma imagem será panorâmica após a união entre as duas imagens e a outra imagem conterá retas que ligam pontos correspondentes nas imagens. O formato das imagens de saída será JPG. As imagens serão salvas na pasta `saida`.

# 3 Detectores de Pontos de Interesse

## 3.1 SIFT

O SIFT (Scale Invariant Feature Transform) é um algoritmo que detecta as características de imagens. Esse método é invariante à translação, escala, rotação e parcialmente invariante à mudanças de iluminação [3].

Os pontos de interesse são detectados de vários pontos de referências da imagem, gerando vetores de características que descrevem regiões locais da imagem. Quando uma nova imagem é apresentada, é feito um cálculo de distância entre os pontos de interesse da nova imagem com os vetores de características da imagem original para realizar o *matching*.

## 3.2 SURF

O SURF (Speeded Up Robust Features) é um detector de características locais de imagens, que foi inspirado no SIFT, porém é mais rápido que o SIFT e mais robusto às transformações nas imagens.

Para encontrar pontos de interesse, o SURF utiliza uma matriz básica Hessiana de aproximação [1].

## 3.3 BRIEF

O BRIEF (Binary Robust Independent Elementary Features) é um descritor de pontos de características que utiliza *strings* binárias em janelas de imagens [2],

que foi construído para correspondência em tempo real para dispositivos com pouco poder computacional. Este detector é variante a escala e orientação.

Em comparações com o SURF, o detector BRIEF apresentou resultados parecidos com tempo de execução menor.

### 3.4 ORB

O detector ORB (Oriented FAST and Rotated BRIEF) foi criado com base no detector de pontos de interesse FAST e do detector BRIEF [4]. Esse detector foi criado para ser uma alternativa do SIFT e SURF, principalmente para sistemas em tempo real e máquinas com pouco poder computacional.

O ORB possui boa performance e custo computacional menor que o SIFT e SURF, além de ser livre de licenciamento, diferente dos detectores citados anteriormente.

## 4 Resultados

### 4.1 Leitura de Dados

A leitura das imagens de entrada é feita pela função `cv2.imread`, da biblioteca OpenCV<sup>1</sup>, que carrega a imagem em níveis de cinza. Além das imagens em tons de cinza, é feita a leitura das imagens coloridas, para que ao final as imagens de saída possuam cores.

Para exemplificar os resultados obtidos, serão utilizadas as imagens `foto2A.jpg` e `foto2B.jpg`. As imagens originais são mostradas na Figura 1.

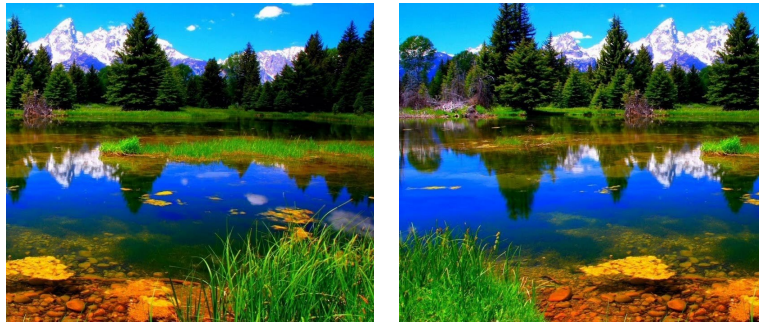


Figura 1: Figuras utilizadas para exemplificar os resultados

### 4.2 Detecção de Pontos de Interesse e Descritores

Para a utilização dos detectores de pontos de interesse SIFT, SURF e BRIEF, foi necessária a instalação dos módulos extras do OpenCV, chamado `opencv-contrib`.

---

<sup>1</sup><https://docs.opencv.org/3.0-beta/index.html>

Os detectores de pontos de interesse SIFT, SURF e ORB possuem a função **detectAndCompute**, que computam os pontos de interesse e os descritores. Para a utilização do detector BRIEF, foi necessário utilizar o detector STAR para encontrar os pontos de interesse e a função **compute** para gerar os descritores.

### 4.3 Seleção de Melhores Correspondências

Para selecionar as melhores correspondências, foram utilizadas as funções de *matching* **cv2.FlannBasedMatcher** e **cv2.FlannBasedMatcher.knnMatch**.

Após aplicada essas funções, é verificado se a quantidade de correspondências são maiores que o limiar informado pelo usuário. Caso contrário, é enviada uma mensagem, via terminal, indicando que não foi possível concluir a aplicação das funções de correspondência e o algoritmo se encerra.

### 4.4 Matriz de Homografia e Alinhamento de Imagens

Para realizar o cálculo da matriz de homografia, são necessários no mínimo 4 pontos de correspondência. Caso isso não ocorra, é enviada uma mensagem, via terminal, indicando que não foi possível aplicar as funções de correspondência e algoritmo é encerrado. A função que faz o cálculo da matriz de homografia é **cv2.findHomography**, retornando uma máscara que especifica *inliers* e *outliers*.

Para criar o alinhamento das imagens, foi utilizada a função de projeção de perspectiva **cv2.warpPerspective**, utilizada também para criar uma imagem panorâmica.

### 4.5 Comparação entre as Técnicas

Em todos os testes entre as técnicas, o limiar adotado foi 30 e as imagens de testes utilizadas são mostradas na Figura 1.

A Figura 2 mostra as linhas de correspondências do detector SIFT, enquanto a Figura 3 mostra a imagem panorâmica. A Figura 4 mostra as linhas de correspondências do detector SURF e a Figura 5 mostra a imagem panorâmica do detector SURF. As Figuras 6 e 7 mostram as linhas de correspondências e imagem panorâmica do detector ORB, enquanto as Figuras 8 e 9 mostram as linhas de correspondências e imagem panorâmica do detector BRIEF, respectivamente.



Figura 2: Linhas de correspondências do detector SIFT



Figura 3: Imagem panorâmica do detector SIFT



Figura 4: Linhas de correspondências do detector SURF

## 5 Conclusão

No comparativo entre os quatro detectores, cada um deles utilizou pontos de interesse diferentes, portanto nenhuma imagem que mostra as linhas de correspondência são iguais. Os detectores ORB e BRIEF utilizaram mais pontos das árvores das imagens originais (Figura 1), enquanto os detectores SURF e SIFT utilizaram as árvores e elementos do lago, como pedras e a água.



Figura 5: Imagem panorâmica do detector SURF



Figura 6: Linhas de correspondências do detector ORB



Figura 7: Imagem panorâmica do detector ORB

Porém, quando comparadas as imagens panorâmicas, as quatro imagens resultantes são idênticas, mostrando que os quatro detectores chegaram em um mesmo resultado, independente dos pontos de interesses diferentes.



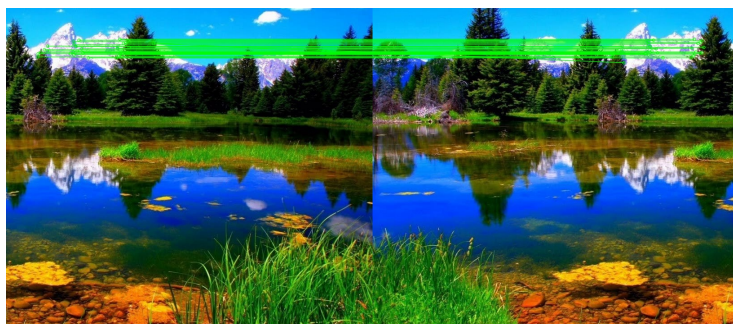


Figura 8: Linhas de correspondências do detector BRIEF



Figura 9: Imagem panorâmica do detector BRIEF

## Referências

- [1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.
- [2] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*, pages 778–792. Springer, 2010.
- [3] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.
- [4] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. ORB: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer, 2011.