Relatório do trabalho 4

Allan Nozomu Fukasawa RA:163527

10/06/2019

1 Introdução

O objetivo deste trabalho é aplicar técnicas de detecção de pontos de interesse para, a partir de um par de imagens, criar uma imagem panorâmica pela ligação entre os pontos de interesse, formando uma correspondência.

Os passos seguidos para a execução do trabalho foram extraídos de sua especificação. [1]

2 Componentes

Está sendo enviado junto a este relatório os seguintes arquivos e diretórios:

- arquivo Trabalho 4.ipynb: contém todo o código executado durante este trabalho.
- quatro imagens utilizadas durante o processamento (todas disponíveis pelo professor no seguinte link *http:www.ic.unicamp.br/helioimagens_registro*. No arquivo do Jupyter Notebook, está sendo utilizada as imagens *foto4A.jpg* e *foto4B.jpg*, as imagens respectivamente 1 e 2.
- resultados em formato de imagem, tanto a imagem final (panorâmica) quanto a imagem de relação entreos pontos em comum entre as duas imagens respctivamente as imagens 4 e 3.

Figura 1: Imagem A

Figura 2: Imagem B

2.1 O Programa

O programa foi implementado com Jupyter Notebooks, usando Python 3.7.1. As bibliotecas utilizadas no desenvolvimento do programa foram, com suas respectivas versões:

- numpy (1.15.4): para manipulação dos vetores.
- matplotlib (3.0.2): visualização dos dados, resultados finais e intermediários.
- opencv (3.4.2): realização da leitura e escrita das imagens, transformação das cores (tanto em escala de cinxa quanto na coloração de RGB para BGR).

2.2 Formato das imagens

As imagens de entrada estão no formato .jpg. As imagens de saída, tanto as finais quanto as intermediárias se encontram no formato .jpeg.

3 Leitura, escrita e plotagem das imagens

3.1 Leitura das imagens

A imagem de entrada é lida com função **cv2.imread** que armazena a imagem em um **numpy.ndarray** de 3 dimensões (MxNx3).

Depois de lida, uma cópia da imagem é convertida para níveis de cinza pela função **cv2.cvtColor**, pois os algoritmos de detecção de pontos de interesse tem de estar em escala de cinza. A imagem original também teve que sofrer mudanças quanto a conversão de cores, pois por padrão, o OpenCV lê a imagem e armazena-a em formato de cores BGR. Portanto, foi-se utilizado a conversão para RGB para visualização dos dados usando matplotlib e depois convertida novamente para BGR na hora de savar os resultados.

3.2 Escrita das imagens

Também foi feito uma função auxiliar para facilitar na saída das imagens utilizando a função **cv2.imwrite**. Assim como dito anteriormente, antes de salvar as imagens, foi convertida novamente do formato RGB para BGR.

3.3 Plotagem das imagens

Foi utilizado para visualização dos resultados as funções de plotagem de imagens em **matplotlib.pyplot**.

4 Solução

4.1 Pontos de interesse

Para a detecção dos pontos de interesse das imagens, primeiramente foi-se convertida em escala de cinza. Depois, foram utilizadas 3 algoritmos diferentes, sendo todos apresentando resultados bem similares:

- SIFT (Scale Invariant Feature Transform): utilizando *cv2.xfeatures2d.SIFT_create()*
- SURF (Speed Up Robust Feature): utilizando *cv2.xfeatures2d.SURF_create()*
- ORB (OrientedFAST, Rotated BRIEF).): utilizando *cv2.ORB_create()*

Depois de identificado cada um dos pontos de interesse, foi-se computado as distâncias (similariedades) entre cada descritor das duas imagens. Para isso, utilizou-se uma correspondência utilizando um método de força bruta, onde para cada ponto é verificato a distância euclidiana entre todas as features e verifica-se qual é o par que apresenta a menor distância.

Além disso, é aplicado o algoritmo de KNN (K-Nearest Neighbours), com 2 vizinhos para verificar se a distância esta no limiar e também para evitar pontos falso-positivos. Depois, para cada relação, é aplicado o teste de relação de Lowe. No caso, foi-se utilizado um fator de correspondência de 0.75.

Depois, caso tenha-se encontrado ao menos 4 pontos de interesse em comum, é calculado a matrix de Homografia, utilizando a função do opencv *findHomography* e passando os pontos como parâmetro bem como a técnica empregada que foi a RANSAC (RANdom SAmple Consensus).

Nota-se que há uma limitação na solução apresentada em que a correspondência e também a formação da imagem panorâmica foi só possível ser realizada a partir da imagem da esquerda para a imagem da direita.

4.2 Resultados

Seguindo os passos, foi-se estimada a seguinte matriz de homeografia:

$$\begin{bmatrix} 1.15562971 + 00 & 1.09434802 + 00 & -2.19606581 + 02 \\ -1.09444476 + 00 & 1.15562133 + 00 & 2.12948548 + 02 \\ 5.29573677 - 07 & 2.21007038 - 07 & 1.00000000 + 00 \end{bmatrix}$$

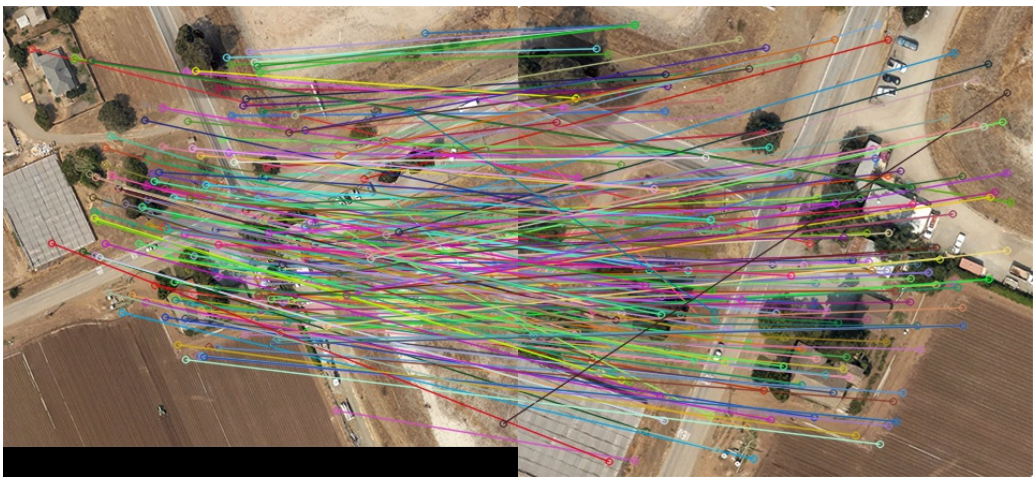

Figura 3: Linhas de correspondência entre as duas imagens, utilizando-se SURF

Além disso, utilizando a função *drawMatchesKnn* foi possível desenhar os pontos de correspondência entre as duas imagens disponíveis na imagem 3 obtidos através das correspodências obtidas.

Utilizando a função *warpPerspective* foi possível também unir as duas imagens formando uma panorâmica. Para isso, primeiramente a imagem a direita foi desenhada em uma imagem resultante de no maximo a soma das duas larguras e a maior altura. Depois, sobreescreveu-se a imagem a esquerda, obtendo o resultado da imagem 4

Figura 4: Resultado da imagem panorâmica formada pelas duas imagens

5 Conclusão

Após a execução de todos os procedimentos foi possível formar a imagem panorâmica, mas com a limitação de formar a imagem da esquerda para a direita. Sendo assim, apenas é possível sobreescrever uma imagem sobre a outra e não o contrário.

Podemos perceber que, tanto no SIFT quanto no SURF, o número de pontos por padrão foi substancialmente maior do que o do ORB. Porém é possível alterar adicionando novos parâmetros para a detecção de novos pontos de interesse no ORB. Porém, em todos, os números de pontos foi suficiente para se realizar a imagem panorâmica e também para o cálculo da matriz de homogrfia.

Além disso, as imagens a serem unidas podem apresentar diferentes contrastes, níveis de brilho e cores, o que leva a clara detecção da união das imagens. Portanto, para melhores resultados, ainda é necessário fazer um pós processamente para nivelar os níveis de cores e contrastes principalmente na união entre elas.

Referências

[1] H. Pedrini, *Trabalho 4*. Introdução ao Processamento de Imagens (MC920 / MO443), 2019.