

Trabalho 5

Levy Gurgel Chaves – RA: 264958

MO443/MC920 - Introdução ao Processamento de Imagem Digital
Universidade Estadual de Campinas
Prof. Hélio Pedrini

11 de Julho de 2020

1 Introdução

O objetivo deste trabalho é implementar técnicas fundamentais de processamento de imagens, onde conceitos importantes básicos como registro de imagens, detecção de pontos de interesse, imagem panorâmica e matriz de homografia são explorados de maneira prática.

Juntamente com o relatório, está sendo enviado o arquivo LEVY_MO443T5_264958.zip, contendo todo o conteúdo apresentado neste relatório. A seção 3 comenta detalhadamente sobre a entrada e as saídas do programa, enquanto a seção 4 apresenta a explicação da solução dos problemas descritos no quinto trabalho, assim como o resultado dos experimentos realizados.

2 Descrição da atividade

A atividade proposta possui a finalidade em realizar a aplicação técnicas de detecção de pontos de interesse para registrar um par de imagens e criar uma imagem panorâmica após a correspondência entre os pontos de interesse. Para tal, iremos aplicar alguns detectores e descritores, tais como SIFT([LOWE, 2004](#)), SURF([BAY et al., 2008](#)), ORB([RUBLEE et al., 2011](#)), BRIEF([CALONDER et al., 2010](#)) e STAR([AGRAWAL et al., 2008](#)).

3 O programa

O programa foi desenvolvido em Python em sua versão 3.7.5 com o auxilio das seguintes bibliotecas nas suas respectivas versões: Numpy 1.18.2; Matplotlib 3.2.1; OpenCV 3.4.2.17. O mesmo pode ser conferido com o auxílio do Jupyter notebook executando o arquivo *notebook-t5.ipynb*. A utilização de uma versão mais antiga do OpenCV deu-se pelo motivo de que nesta versão os algoritmos patenteados SIFT e SURF podem ser usados mais facilmente, sem a necessidade de compilação da biblioteca.

3.1 Entrada

O programa realiza a leitura de duas imagens coloridas em formato *JPEG* a partir da pasta */images*. Por

convenção, a Figura 1 representa um par de imagens adotado como referência para os experimentos.

3.2 Saída

As imagens relativas aos resultados dos experimentos possuem formato *JPEG* e são armazenadas na pasta */output*.

4 Soluções e decisões tomadas

4.1 Leitura das imagens

A leitura das imagens foram feitas através da função **cv2.imread** onde a mesma retorna um array Numpy possuindo as dimensões MxNx3, onde MxN representam as dimensões da imagem lida e 3 indica a quantidade de diferentes canais.

4.2 Detecção de pontos de interesse

Com o intuito de realizar a identificação de pontos de interesse nas duas imagens de entrada, temos que aplicar algoritmos apropriados para esta finalidade (detalhes sobre os algoritmos utilizados estão presentes na subseção 4.3). Como passo seguinte à leitura das imagens, estas serão transformadas para escala de tons de cinza através da função **cv2.cvtColor**.

Com as imagens em tons de cinza, pode-se realizar a aplicação de técnicas adequadas para encontrar os pontos de interesse em cada imagem. A partir dos pontos encontrados, precisa-se encontrar similaridades entre as correspondências de cada descritor. Para tal, o método de correspondência utilizada foi a de força bruta, onde para cada ponto de correspondência a distância euclidiana é verificada entre todas as características (features) e seleciona-se aquele par que apresenta a menor distância. Além disso, o algoritmo KNN (K-Nearest Neighbours) com $k = 2$ foi aplicado para verificar se as correspondências encontradas estão dentro de um limiar estabelecido e para descartar aquelas que representam falso-positivo.

4.3 Detectores e descritores

Um descritor de características é um algoritmo que recebe uma imagem e produz vetores descritores de ca-



(a) Foto A



(b) Foto B

Figura 1: Imagens referência

racterísticas. Estes descritores codificam informações relevantes em uma série de números e atuam como uma espécie de “identidade” numérica que pode ser usada para diferenciar uma característica de outra.

Neste projeto, optou-se por testar algumas combinações entre detectores e descritores de features disponíveis na biblioteca OpenCV, tais como:

- **SIFT(Scale Invariant Feature Transform)**: é um dos métodos responsáveis por detectar pontos de interesses. Este utiliza a ideia que os vetores de features extraídas precisam ser invariantes à escala, rotação, translação e parcialmente à iluminação. Neste projeto, o SIFT foi utilizando tanto para realizar a detecção de pontos de interesse quanto para a descrição destes pontos. Para tal, a função `cv2.xfeatures2d.SIFT_create` foi utilizada.
- **SURF(Speed Up Robust Feature)**: método que possui o SIFT como base, porém realiza modificações pontuais para tornar execução mais rápida. Para sua utilização como descritor e detector, fez-se o uso da função `cv2.xfeatures2d.SURF_create`.
- **STAR(Censure)**: é um método para realizar a detecção de pontos de interesse, contudo este não oferece nenhum método para a descrição. Com o intuito de testar combinações diferentes, optou-se por utilizar o método STAR como detector e BRIEF como descritor, respectivamente, através das funções `cv2.xfeatures2d.StarDetector_create` e `cv2.xfeatures2d.BriefDescriptorExtractor_create`.
- **ORB(Oriented FAST, Rotated BRIEF)**: representa uma alternativa ao SURF e SIFT quanto ao custo computacional, além de não ser patenteado. Conceitualmente, é uma união entre o detector FAST e o descritor BRIEF, porém com certas melhorias para o aumento de performance computacional. Para a utilização deste, recorreu-se a função `cv2.ORB_create`.

4.4 Seleção de correspondências

Agora com uma série de correspondências obtidas através de um descritor de características, precisa-se selecionar aquelas que representam as melhores. Para tal, aplicou o teste de Lowe, onde as correspondências selecionadas foram aquelas que apresentavam 70% de proximidade. Após este passo de seleção, o respectivo resultado no par de imagens referência pode ser visto na Figura 2.

4.5 Imagem panorâmica

Com as melhores correspondências selecionadas entre as duas imagens de entrada, agora pode-se encontrar a respectiva matriz de homografia. Após assegurarmos da obtenção de, pelo menos, 4 correspondências, pode-se realizar o cálculo da matriz utilizando o método de RANSAC através da função `cv2.findHomography` do OpenCV. É importante notar que neste passo pode-se encontrar duas matrizes diferentes, dependendo da ordem de correspondência entre os pontos de similaridades. Por motivo arbitrário, escolheu-se que na panorâmica final a imagem B ficará mais à esquerda e a imagem A ficará mais à direita.

De posse da respectiva matriz de homografia, pode-se, então, aplicar o método de ajuste de perspectiva utilizando a função `cv2.warpPerspective`. Para isso, a imagem mais à direita foi desenhada em uma imagem resultante possuindo largura final como a soma das larguras entre as duas imagens de entrada e a altura final será a altura máxima entre as respectivas imagens.

Aplicando os passos descritos anteriormente, para a imagem de referência adotada, consegue-se como resultado a imagem 3.

A metodologia citada anteriormente foi aplicada aos 5 pares de imagens disponíveis em https://www.ic.unicamp.br/~helio/imagens_registro/. Para cada par disponível, tentou-se aplicar os 4 descritores ditos anteriormente, porém apenas alguns resultados foram colados neste relatório e podem ser verificados nas Figuras 4, 5, 6, 7, 8, 9, 10 e 11. Todos os resultados podem



Figura 2: Melhores correspondências com a aplicação do ORB



Figura 3: Imagem panorâmica resultante

ser conferidos na pasta */output*.

4.6 Discussões

Nota-se que os resultados obtidos, de maneira geral, foram satisfatórios. Observando as Figuras 4 e 5 boas panorâmicas foram obtidas. Isto deve-se principalmente porque as features a serem correspondidas estão na mesma perspectiva, sem a necessidade de alteração de escala ou rotação. Já em outros casos, por exemplo nas Figuras 8, 9, 11, constata-se que muitas das correspondências foram encontradas de maneira correta, o que contribuiu para o resultado satisfatório das imagens panorâmicas.

Também é possível observar que os métodos SURF e SIFT são os que mais encontram correspondências, enquanto o ORB e STAR são os que menos encontram. Um caso onde isso pode ser ilustrado é que utilizando o método STAR para o quinto par de imagens menos de 4 correspondências são encontradas, impedindo a geração da matriz de homografia. Contudo, houve um caso, com o mesmo par de imagens, ao tentar utilizar o SIFT, onde não foram encontrados 4 correspondências.

Na Figura 9 pode-se observar que a panorâmica obtida não teve resultado visual tão satisfatório quanto

as demais. Uma explicação para este caso deve-se às correspondências encontradas pelo método utilizado. Se for realizada uma inspeção visual nas melhores correspondências obtidas, percebe-se que em muitos casos que o método encontrou correspondências erradas, sendo este fator o responsável pelo resultado deturpado.

4.7 Limitações

Devido à natureza das imagens dadas como entradas, aconteceu para algum determinado par de imagens a não detecção de pelo menos 4 correspondências. Sendo assim, pode-se, por exemplo, ajustar o limiar de correspondência para um valor mais flexível para estes casos.

Neste trabalho adotou-se em construir a imagem panorâmica final colocando a imagem B como a imagem mais à esquerda, enquanto a imagem A ficará mais à direita.

Uma outra limitação do trabalho é que apenas foi desenvolvido o teste de seleção de Lowe para a seleção das melhores correspondências.



(a) Melhores correspondências



(b) Panorâmica obtida

Figura 4: Primeira foto com ORB aplicado

Referências

AGRAWAL, M.; KONOLIGE, K.; BLAS, M. R. Censure: Center surround extremas for realtime feature detection and matching. In: SPRINGER. **European Conference on Computer Vision.** [S.l.], 2008. p. 102–115.

BAY, H.; ESS, A.; TUYTELAARS, T.; VAN GOOL, L. Speeded-up robust features (surf). **Computer vision and image understanding**, Elsevier, v. 110, n. 3, p. 346–359, 2008.

CALONDER, M.; LEPESTIT, V.; STRECHA, C.; FUÀ, P. Brief: Binary robust independent elementary features. In: DANIILIDIS, K.; MARAGOS, P.; PARAGIOS, N. (Ed.). **Computer Vision – ECCV 2010**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 778–792. ISBN 978-3-642-15561-1.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, Springer, v. 60, n. 2, p. 91–110, 2004.

RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. Orb: An efficient alternative to sift or surf. In: IEEE. **2011 International conference on computer vision**. [S.l.], 2011. p. 2564–2571.



(a) Melhores correspondências



(b) Panorâmica obtida

Figura 5: Primeira foto com SIFT aplicado

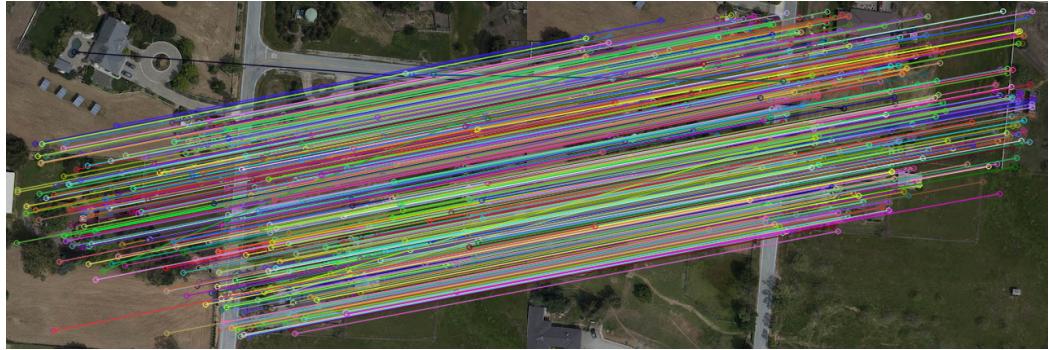


(a) Melhores correspondências



(b) Panorâmica obtida

Figura 6: Resultados para o terceiro par após a aplicação do STAR e BRIEF



(a) Melhores correspondências

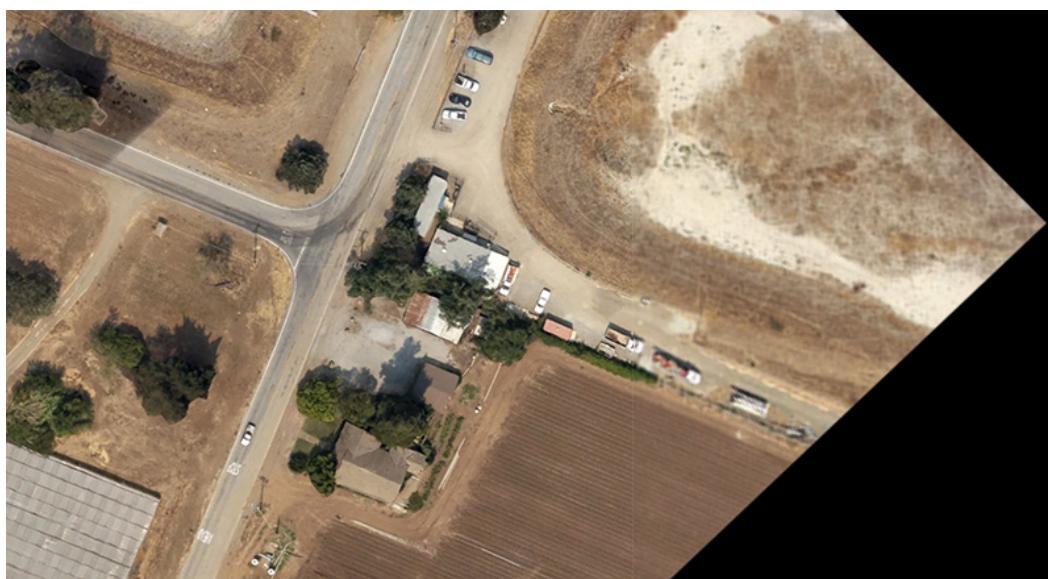


(b) Panorâmica obtida

Figura 7: Resultados para o terceiro par após a aplicação do SURF



(a) Melhores correspondências

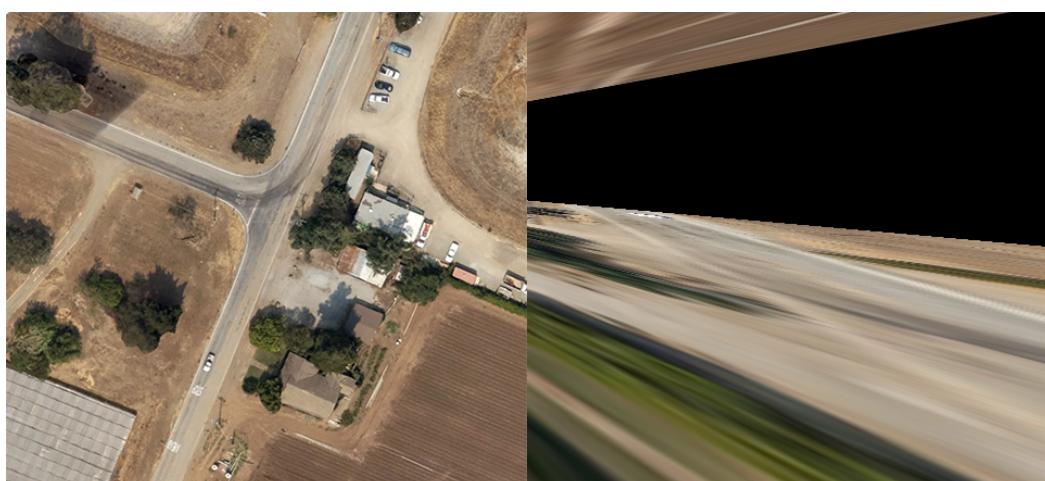


(b) Panorâmica obtida

Figura 8: Resultados para o quarto par após a aplicação do ORB

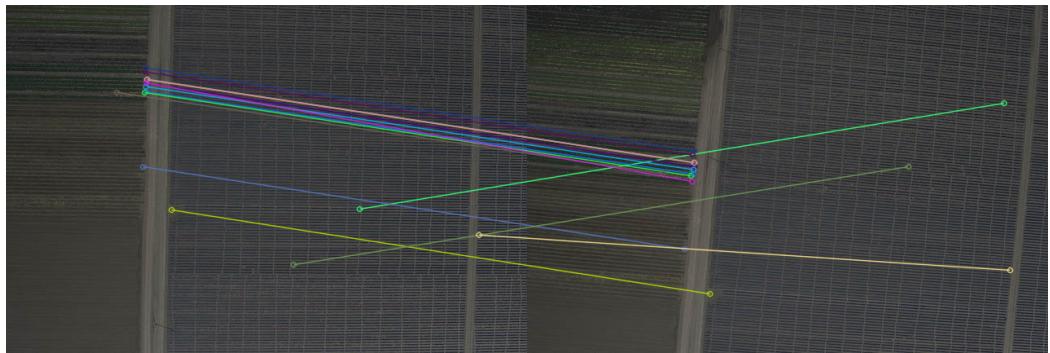


(a) Melhores correspondências

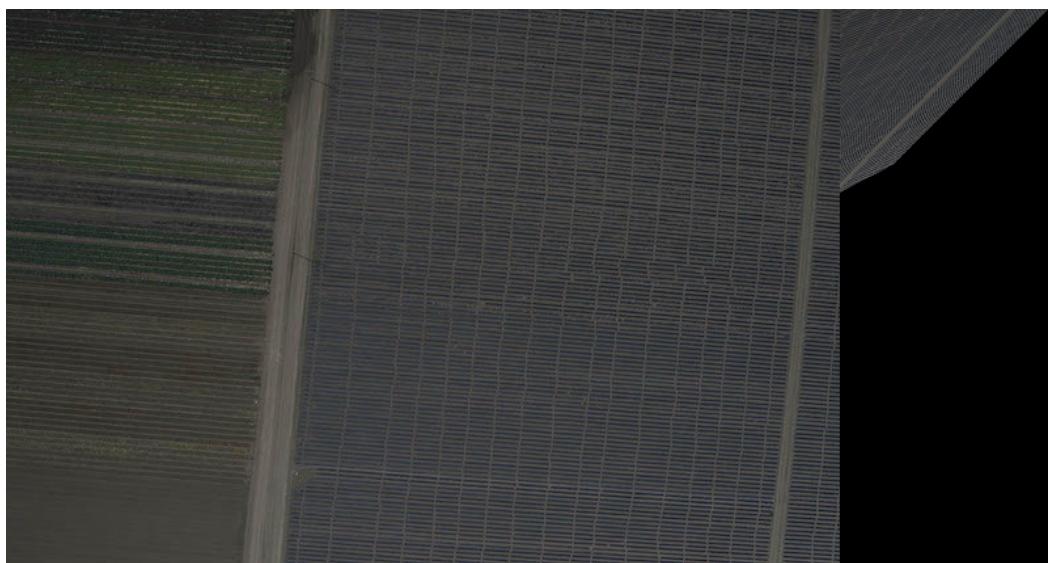


(b) Panorâmica obtida

Figura 9: Resultados para o quarto par após a aplicação do STAR e BRIEF

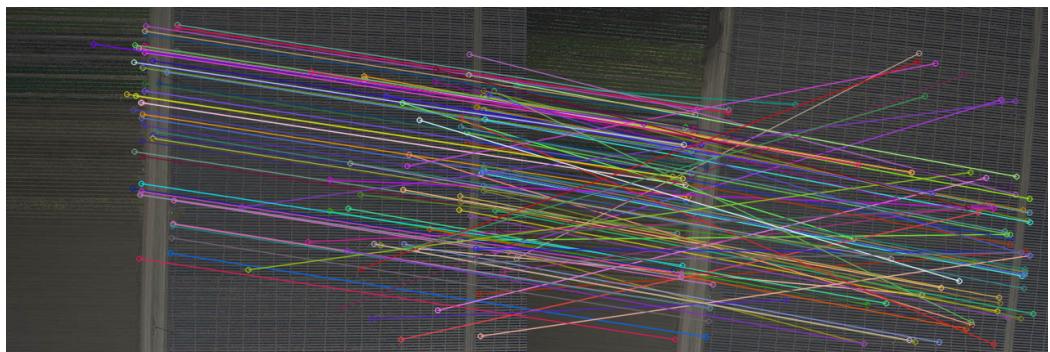


(a) Melhores correspondências

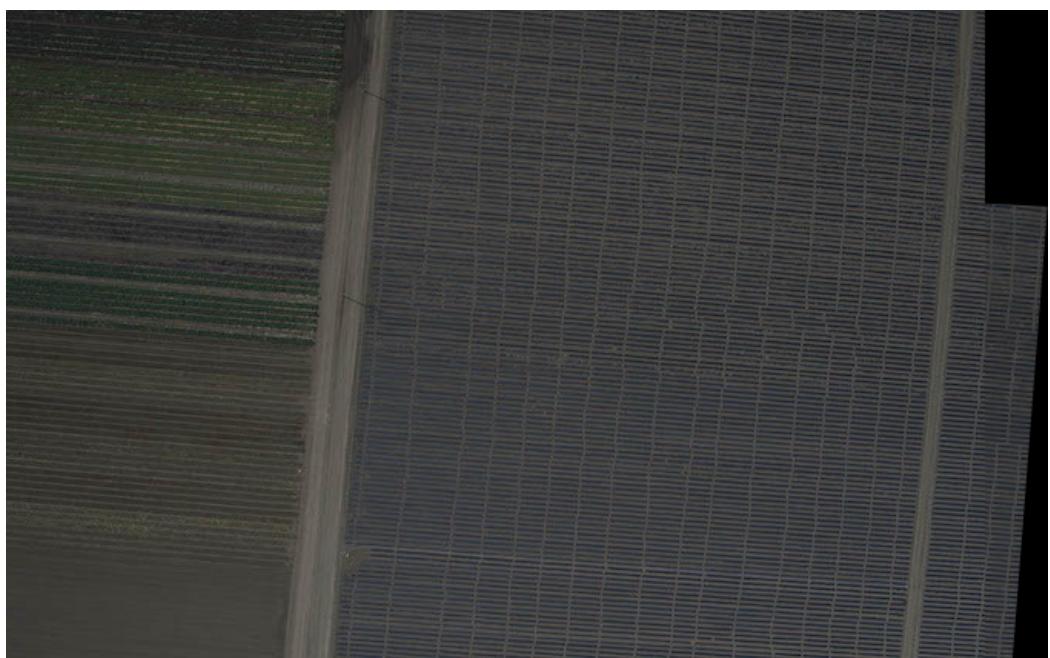


(b) Panorâmica obtida

Figura 10: Resultados para o quinto par após a aplicação do ORB



(a) Melhores correspondências



(b) Panorâmica obtida

Figura 11: Resultados para o quinto par após a aplicação do SURF