

Lista 3 – Visão Computacional

Aluno: Rennan de Lucena Gaio

DRE: 119122454

Todo o código do trabalho pode ser acessado pelo link do github:

https://github.com/RennanGaio/visao_computacional/tree/master/lista3

1 SIFT

1.1 Implementação

Diante de todas as imagens testadas, os procedimentos base de implementação foram os mesmos. Os procedimentos do exercício 1 são todos encontrados no arquivo “sift.py” no diretório indicado. Para a utilização do SIFT foi utilizada uma implementação já pronta em python presente no arquivo “pysift.py” que foi importado para o programa principal. Nesta biblioteca é possível retornar os pontos característicos assim como seus descritores.

Para associar os pontos característicos das 2 imagens foi utilizado o módulo “FlannBasedMatcher.KnnMatch” do OpenCV, que associa os pontos com menor distancia entre os descritores das 2 imagens. Para eliminar os falsos negativos, foi criado um filtro que só aceita distancias entre os 2 descritores de 0.7.

Os resultados para cada imagem serão apresentados a seguir.

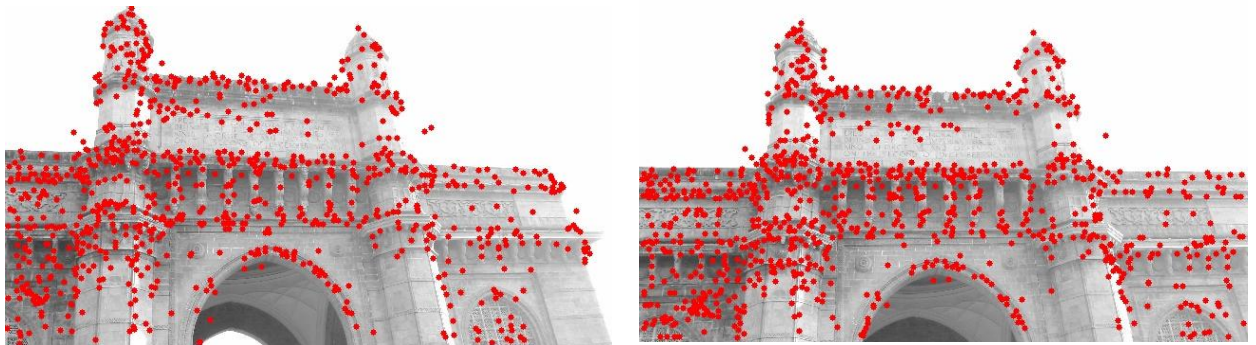
1.2 Resultados

Para representar os pontos característicos das imagens obtidos pelo SIFT de forma comparativa não foi desenhado à direção do ângulo do descritor, pois a imagem fica extremamente poluída. Para obter este atributo, nos pontos marcados estão salvos em “keypoints.angle”, e ele é utilizado internamente no algoritmo do SIFT.

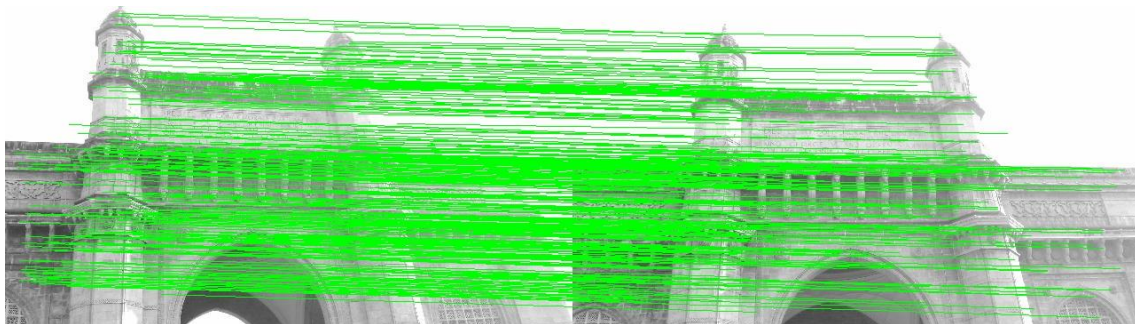
A seguir serão apresentados os resultados dos pontos encontrados nos pares de imagens e suas correspondências logo em seguida.

1.2.1 imagem goi

Pontos característicos encontrados:



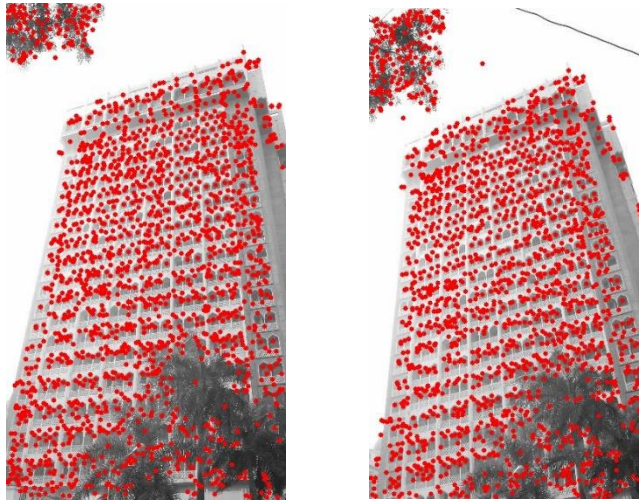
Correspondências entre imagens:



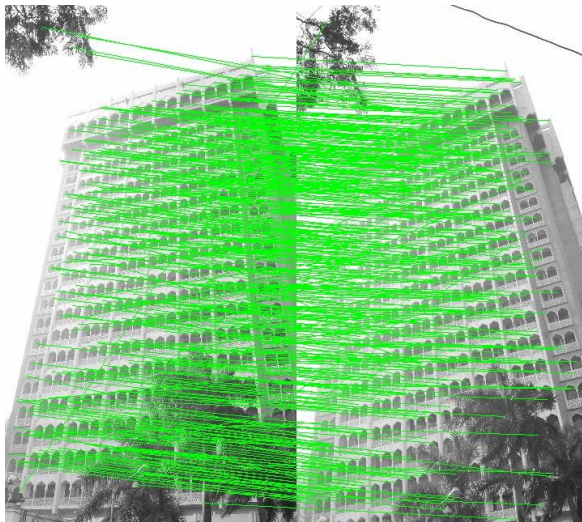
Em relação aos resultados encontrados nos resultados da lista 2, sem a utilização do SIFT, as correspondências foram realmente muito boas. Este resultado mostra que o mapeamento dos pontos foi feito de forma muito efetiva, mesmo com a utilização de muitos pontos identificados pelo algoritmo.

1.2.2 Imagens tajnew

Pontos característicos encontrados:



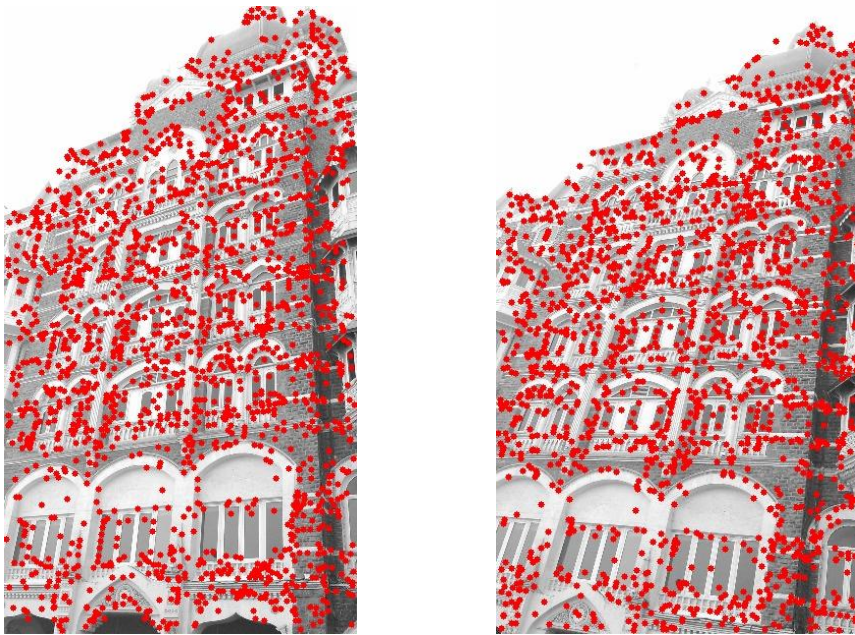
Correspondências entre imagens:



Assim como na imagem anterior, é perceptível os bons resultados encontrados pelo paralelismo entre as linhas da imagem.

1.2.3 Imagens tajold

Pontos característicos encontrados:



Correspondências entre imagens:



Em relação às imagens anteriores, esta possuiu um pouco mais de outliers, porém a associação ainda permanece muito boa em geral.

Após a análise de todos os resultados em conjunto. Caso fosse utilizados todos os pontos para se criar a melhor homografia entre as imagens, devido a quantidade de pontos, a matriz poderia tentar “acerta” correspondências que não são muito boas, e distorcer muito a imagem. Isso pode causar um resultado inferior do que utilizando menos pontos que sejam melhores representantes da imagem em questão. Porém, com a alta qualidade de associação dos pontos obtidos, os resultados da homografia ainda ficou muito bom, e pode ser visto no diretório de resultados do trabalho.

2 RANSAC

2.1 Implementação

A partir da implementação do RANSAC utilizando a biblioteca do OpenCV, foi possível retornar a matriz de homografia que melhor se adequa à transformação de perspectiva entre as 2 imagens. Na seleção aleatória de pontos são escolhidos 4 aleatórios pois essa é a quantidade mínima de pontos necessária para se fazer uma homografia entre 2 imagens.

A partir da geração das homografias do RANSAC, foram consideradas *inliers* as correspondências que possuíam erro de reprojeção menor do que 5 em relação a reta do par de pontos selecionado. Este comportamento pode ser expressado pela formula:

$$\| \text{dstPoints}_i - \text{convertPointsHomogeneous}(H * \text{srcPoints}_i) \| > \text{ransacReprojThreshold}$$

Em que `ransacReprojThreshold` é igual a 5 no experimento feito.

A partir da melhor homografia encontrada, foi reconstruída a primeira imagem com relação a segunda imagem. Os resultados estão presentes no subcapítulo abaixo.

2.2 Resultados

Todos os resultados abaixo são referentes a transformação por homografia da primeira imagem, comparando-a com a segunda.

2.2.1 Imagens goi



2.2.2 Imagens tajnew



2.2.3 Imagens tajold



Avaliando os resultados encontrados, percebe-se que as transformações ficaram bem razoáveis dado as 2 imagens. Não houve resultados muito estranhos, e o posicionamento das imagens que resultaram na homografia ficaram bem de acordo com o esperado.