Lista 1 – Visão Computacional

Aluno: Rennan de Lucena Gaio DRE: 119122454

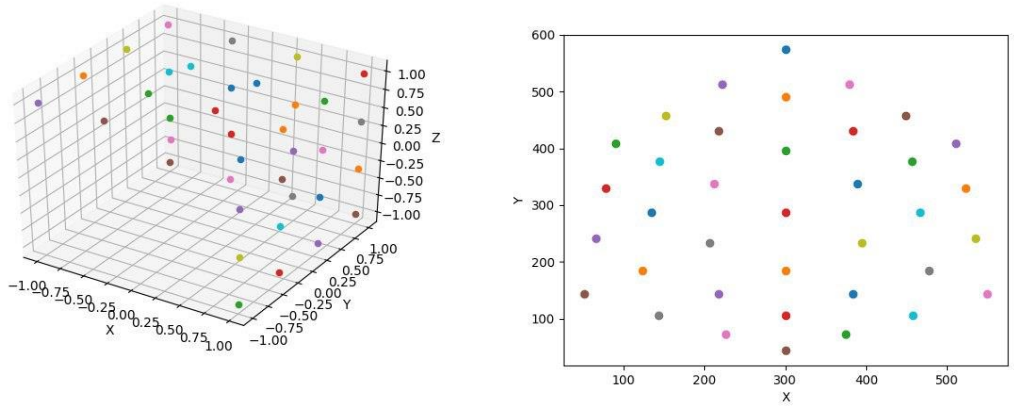
Todo o código do trabalho pode ser acessado pelo link do github:

https://github.com/RennanGaio/visao_computacional/tree/master/lista1

1 Projeções

1.1 Primeiro conjunto de dados

Primeiramente, no primeiro conjunto de imagem, observasse o seguinte posicionamento espacial dos pontos e sua respectiva projeção em 2 dimensões.



Para calcular a matriz de projeção P, foi elaborada a rotina “create_projection_dlt” no arquivo “projeções.py”, em que é utilizado como entrada os pontos do arquivo referente ao primeiro conjunto de dados. Para tal, foi criada a matriz A, assim como na sugestão, e foi encontrado o vetor p contendo todas as entradas da matriz P 3x4.

Foi encontrado o seguinte valor para a matriz:

[[-1.38922361e+02 -2.54555716e+01 8.92529105e+01 3.00500000e+02]

[4.21179680e+01 -1.56094866e+02 4.21179680e+01 2.87436119e+02]

[-8.26446762e-02 -8.47107208e-02 -8.26446762e-02 1.00000000e+00]]

Para mostrar sua corretude, foi calculado o erro quadrático médio entre os pontos originais e os pontos obtidos pela transformação projetiva dos pontos 3d. Com isso, obteve-se o erro de 3.685 e-07. Que por ser muito baixo, significa que nosso resultado foi bom.

Adicionando o ruído ro = 0.5*max3d, foi escolhido de forma aleatória se este valor iria somar ou subtrair às coordenadas dos pontos, alterando todas as entradas (x, y, z) do ponto em questão. Com isto, obteve-se a matriz de projeção:

[[-1.40960009e+02 -3.47149403e+01 8.36538339e+01 3.00826660e+02]

 [3.68946671e+01 -1.61340750e+02 3.91004681e+01 2.87839474e+02]

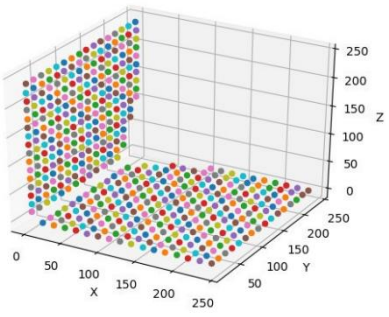
 [-9.58155247e-02 -1.11080224e-01 -9.17904865e-02 1.00000000e+00]]

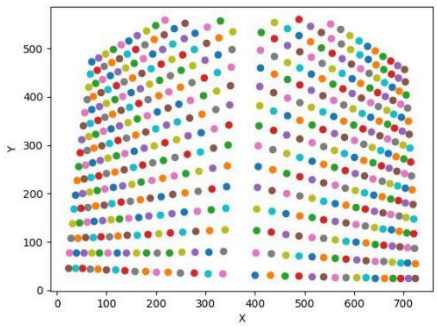
Neste experimento, foi calculado o erro de forma semelhante à anterior, porém, percebe-se um grande aumento neste valor passando a ser 6.282.

Adicionando mais ruído aos nossos dados, desta vez sendo escolhidos 20% dos pontos que receberiam um valor aleatório entre [-max3d/2, max3d/2], foi proposto reproduzir os mesmos experimentos anteriores. Porém, neste caso, a nossa função de SVD não convergiu para os pontos oferecidos. Isso significa que a projeção ficou impossibilitada devido ao ruído que impactou muito no método. Foram executadas algumas vezes para averiguar se era uma coincidência do fator aleatório, porém em nem uma das execuções o algoritmo convergiu.

1.2 Segundo conjunto de dados

No segundo conjunto de imagem, observasse o seguinte posicionamento espacial dos pontos e sua respectiva projeção em 2 dimensões.





Para calcular a matriz de projeção P, foi elaborada a rotina “create_projection_dlt” no arquivo “projeções.py”, em que é utilizado como entrada os pontos do arquivo referente ao segundo conjunto de dados. Para tal, foi criada a matriz A, assim como na sugestão, e foi encontrado o vetor p contendo todas as entradas da matriz P 3x4.

Foi encontrado o seguinte valor para a matriz:

[[3.16674972e+00 3.95151068e-01 -1.41642705e+00 3.65312215e+02]

 [2.33281127e-02 3.36313475e+00 1.69996226e-01 -1.91031160e+01]

[2.27548072e-03 7.95349953e-04 2.34214062e-03 1.00000000e+00]]

Para mostrar sua corretude, foi calculado o erro quadrático médio entre os pontos originais e os pontos obtidos pela transformação projetiva dos pontos 3d. Com isso, obteve-se o erro de 0.957. Apesar de ser superior ao do primeiro experimento, ainda se obtém um valor baixo.

Adicionando o ruído ro = 0.5*max3d, foi escolhido de forma aleatória se este valor iria somar ou subtrair às coordenadas dos pontos, alterando todas as entradas (x, y, z) do ponto em questão. Com isto, obteve-se a matriz de projeção:

[[-2.86857424e+02 -7.12727460e+01 3.04530426e+00 3.88038726e+02]

[-4.36922681e+00 -1.84574925e+02 -1.14599128e+01 2.12655961e+02]

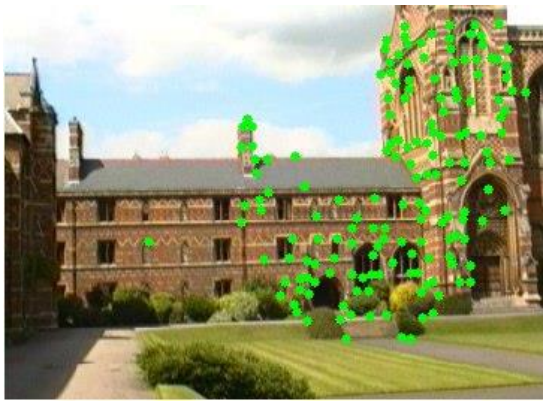
[-3.78429140e-01 -1.82032955e-01 -3.76008719e-01 1.00000000e+00]]

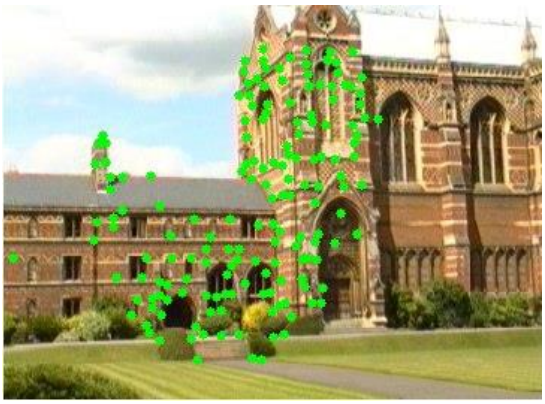
Neste experimento, foi calculado o erro de forma semelhante à anterior, porém, percebe-se que desta vez que o valor absoluto do erro ficou muito alto atingindo 178.679.

Adicionando mais ruído aos nossos dados, desta vez sendo escolhidos 20% dos pontos que receberiam um valor aleatório entre [-max3d/2, max3d/2], foi proposto reproduzir os mesmos experimentos anteriores. Porém, assim como no primeiro conjunto de dados, o SVD não convergiu devido ao ruído.

2 Homografias

Fazendo a visualização das imagens com os respectivos pontos marcados, obtemos os seguintes resultados:





Para calcular a matriz de transformação de homografia entre as duas imagens foi utilizada a técnica de dlt normalizada. Pois esta possui resultados superiores à dlt normal. Para isto, foi criado o módulo “create_homographi_dlt” no arquivo “homografias.py”, em que ele

recebe como entrada os conjuntos de pontos mapeados nas duas imagens. Com base na execução do experimento, foram obtidas as seguintes matrizes de homografia:

H12 = [[1.03940731e+00 -3.20063455e-02 -9.77973998e+01]

[4.53110315e-02 1.01434726e+00 4.61709275e+00]

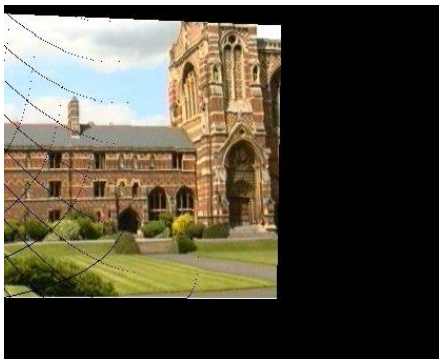
[1.41747009e-04 -5.42067927e-05 1.00000000e+00]]

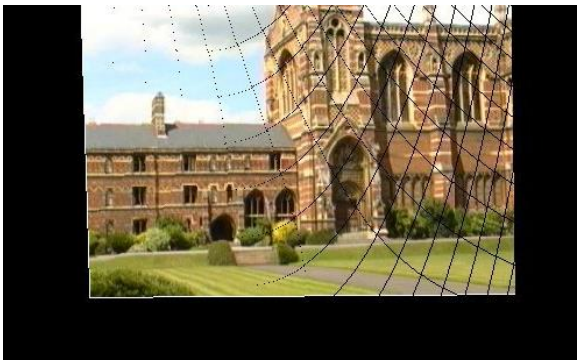
H21 = [[9.60903505e-01 3.48519922e-02 9.38578755e+01]

[-4.21881427e-02 9.97360412e-01 -8.75191738e+00]

[-1.38190793e-04 4.73514245e-05 1.00000000e+00]]

Aplicando a transformação de homografia em cada uma das imagens, obtiveram os seguintes resultados (h12 na imagem 1 e h21 na imagem 2 respectivamente).





A imagem foi contruida a partir de um grid preto maior do que a imagem original para se buscar capturar mais detalhes independente de translações.

Observando os resultados, podemos perceber que ouve uma tentativa de ajuste entre as duas imagens escolhidas. Pelo fato de não ter sido realizado uma interpolação dos pontos, seguindo as linhas pretas observa-se como a matriz alterou o plano da imagem para se adaptar a imagem original. Como estas linhas marcam bem os ajustes que a matriz H fez na imagem, optou-se por mantelos no resultado. Outro fato que pode ser observado, é que no caso da primeira imagem a torre está muito mais para a direita do que na imagem 2, logo em sua homografia, a matriz H desloca bastante a torre para a esquerda, se aproximando da posição da torre na imagem 2. O mesmo acontece na imagem 2 com relação à imagem 1, porém neste caso a translação é feita para a direita.