

Lista 4 – 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/lista4

Calibração de Câmeras

A partir dos dados fornecidos pela lista, foi gerado a visão tridimensional dos pontos (**imagem 1**), e a visão projetada do cubo vista pela câmera do robô (**imagem 2**).

Imagem 1 – Visão tridimensional dos vértices do cubo.

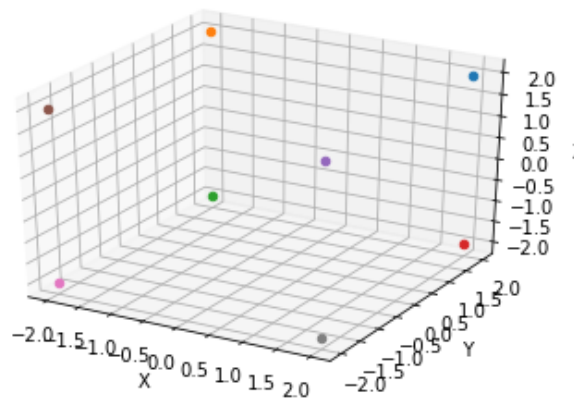
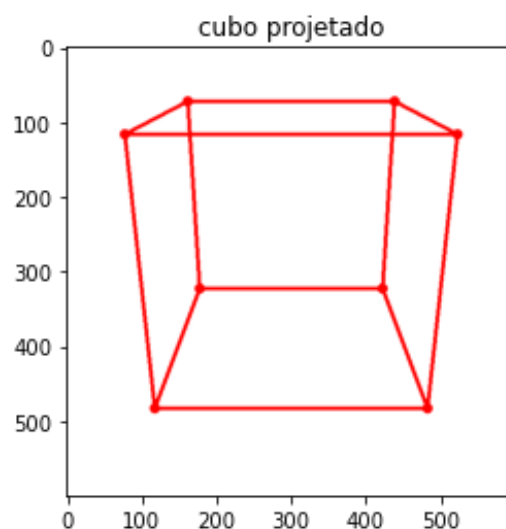


Imagem 2 – Visão projetada do cubo.



A partir dos dados, foi calculada a matriz de projeção de câmera **P** utilizando o método dlt normalizado. Sua implementação está disponível no código do projeto no python notebook. Após a execução do dlt foi encontrada a seguinte matriz de projeção (**Imagem 3**):

Imagem 3 – Matriz de projeção arredondada em 2 casas decimais.

78.61	11.55	32.10	300.00
0.00	83.48	0.03	249.94
-0.00	0.04	0.11	1.00

Para a verificação da matriz de projeção, foi aplicado aos pontos do cubo espacial, e observado que os resultados de projeção estão realmente muito próximos dos pontos fornecidos pela lista (a menos do fator de erro). Este teste está sinalizado no python notebook, logo após a obtenção de **P**.

A partir da matriz de projeção encontrada, utilizou-se a fatoração RQ para se encontrar a matriz triangular superior **K** e a matriz de parâmetros extrínsecos (**R**, **t**). Para a obtenção do centro ótico e da distância focal, utiliza-se a forma teórica de **K** exemplificada na **Imagem 4** e se compara com **K** obtido pela fatoração RQ (**Imagem 5**).

Imagem 4 – K teórico.

fmx	s	px
0	fmy	py
0	0	1

Imagem 5 – K encontrado a partir dos dados.

78.61	-0.00	-301.93
0.00	78.54	-251.53
0.00	0.00	-1.01

Comparando as 2 matrizes, obtemos que a posição do centro ótico da câmera do robô em relação ao centro do cubo é aproximadamente: **px** = **-301.93** e **py** = **-251.53**. Lembrando que o eixo do problema se encontra no canto esquerdo superior devido à biblioteca do Python. E sua distância focal é aproximadamente: **f** = **78.6**. Este resultado está a menos de um fator de escala m que não é dado pelo problema que representa o número de pixels por unidade de distância nas coordenadas da imagem.