# Geometria da transformação de imagens

**Lucas Pires Cobucci, Matheus Rambo da Roza**

[1]Universidade do Estado de Santa Catarina

## 1. Questão 1

Dada uma peça retangular, com regiões escuras e claras e escuras, paralela ao plano de imagem. O vetor normal ao plano da peça é colinear ao eixo OZc (eixo focal). O foco (centro de projeção) encontra-se na origem [0,0,0]T e o eixo OZc intercepta a peça (Figura 1). O centro de projeção encontra-se na origem [0,0,0]T , a distância focal d entre o centro de projeção f e o plano de imagem é igual a 5 mm, cada pixel do sensor é um quadrado de lado 7,5x10-6 m=0,0075 mm.

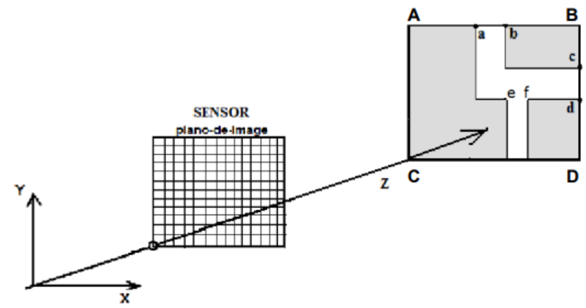


*Figura 1: Projeção da peça recortada.*

**Figura 1.**

Além disso, a distância entre o centro de projeção e a folha é igual a 500 mm. O sensor é binário: cada pixel ou está aceso (preto) ou apagado (branco). Um pixel está aceso se um ou mais raios projetores incidirem sobre o mesmo, caso contrário o pixel é apagado (branco).

Pontos: a=(150,75; 1500,0; 500,0; 1,0) mm b=(153,5; 1500,0; 500,0; 1,0) mm c=(1500,0; 155,0; 500,0; 1,0) mm d=(1500,0; 150,75; 500,0; 1,0) mm e=(145,0; 153,0; 500,0; 1,0) mm f=(144,3; 153,0; 500,0; 1,0) mm

### 1.1. Questão 1 - Letra A

Qual a área projetada em perspectiva do quadrilátero ABCD em mm2?

Para a resolução dessa questão foi considerado os a distância dos pontos a e d em relação ao centro, assim pegando essas distâncias e atribuindo a CA e CD, com isso levamos em consideração A= L * L para calcular a área de um quadrado, e então fazemos CA * CD e obtemos a área de ABCD em mm2, o qual foi 2250000 mm2.

```python
#Coordenadas
# A = [0, 1500, 500]
# B = [1500, 1500, 500]
# C = [0, 0, 500]
# D = [1500, 0, 500]
CA = 1500 #Distancia do centro até A
CD = 1500 #Distancia do centro até D
print("Questao A :")
print(f'CA: {CA} | CD: {CD} | ABCD em mm²: {CA*CD}')
```

**Figura 2. Código para descobrir a área de ABCD**

### 1.2. Questão 1 - Letra B

Quais as coordenadas dos pontos a, b, c, d, e, f projetados em perspectiva (pontos dados)?

A’: [[1.5075], [15], [1]]

B’: [[1.535], [15], [1]]

C’: [[15], [1.55], [1]]

D’: [[15], [1.5075], [1]]

E’: [[1.45], [1.53], [1]]

F’: [[1.443], [1.53], [1]]

Para a realização desse exercício obtemos a matriz de projeção:

M = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1 / 5, 0]]) Obtemos os pontos:

```python
pa = np.array([[150.75], [1500.0], [500.0], [1.0]])
pb = np.array([[153.5], [1500], [500.0], [1.0]])
pc = np.array([[1500.0], [155.00], [500.0], [1.0]])
pd = np.array([[1500.0], [150.75], [500.0], [1.0]])
pe = np.array([[145.0], [153.0], [500.0], [1.0]])
pf = np.array([[144.3], [153.0], [500.0], [1.0]])
```

**Figura 3. Pontos**

As coordenadas Homogenias:

```python
a_linha = M.dot(pa)
b_linha = M.dot(pb)
c_linha = M.dot(pc)
d_linha = M.dot(pd)
e_linha = M.dot(pe)
f_linha = M.dot(pf)
```

**Figura 4. Coordenadas Homogenias**

E por fim temos de fato as coordenadas em perspectiva:

```
a = a_linha / a_linha[2, 0]
b = b_linha / b_linha[2, 0]
c = c_linha / c_linha[2, 0]
d = d_linha / d_linha[2, 0]
e = e_linha / e_linha[2, 0]
f = f_linha / f_linha[2, 0]
```

**Figura 5. Coordenadas**

### 1.3. Questão 1 - Letra C

As três regiões cinzas exibidas no desenho são distinguíveis na imagem projetada? Formam-se quantos objetos?

Não, apenas 2 regiões (ACD e B) são distinguíveis, formando assim 2 objetos isso pelo fato de que a distancia entre 'e' e 'f' é muito pequena.

## 2. Questão 2

A matriz de projeção transforma pontos P(X,Y,Z) nos respectivos pontos bidimensionais p(x,y). O Algoritmo dos Seis Pontos permite o cálculo da matriz de projeção de uma câmera por meio da solução de um sistema de equações (conforme discutido em sala e apresentado no capítulo 2 do Gonzalez e Woods). Considere o padrão de calibração formado por um cubo virtual de 1 metro de lado e a sua imagem gerada por uma câmera sintética.

### 2.1. Questão 2 - Letra A

Obtenha a matriz M desejada utilizando seis pontos cujas coordenadas encontram-se no Moodle.

RESPOSTA: Primeiramente foi necessário ler os pontos de um arquivo texto. A lógica por traz seguiu da seguinte forma: para cada linha do arquivo, utilizar a função split() do Python para separar cada valor. Os 3 primeiros valores (que representam x, y, z) de cada linha foram introduzidos em um vetor chamado matriz e o 2 últimos valores (valores da projeção) foram introduzidos em um vetor denominado projeção. A Figura 1 demonstra como foi feita a leitura e a inserção dos valores em seus vetores correspondentes. O mesmo foi feito para os valores que foram utilizados no algoritmo dos seis pontos, assim como mostra a Figura 7.

Para calcular a matriz M, foi necessário colocar os valores dos pontos em todos os Xn, Yn, Zn, xn e yn, onde n foi de 1 ate 6. A Figura 8 demonstra essa inserção.

Apos esse passo, foram introduzidos os valores na matriz denominada C, seguindo a metodologia da figura 9.

```python
#Lendo a base de dados
arquivo = open("baseDados.txt","r")
texto = arquivo.readlines()
matriz = [];
projecao = [];
aux = [];
for linha in texto :
    valores = linha.split()
    aux = [valores[0], valores[1], valores[2]]
    matriz.append(aux)
    aux = [valores[3], valores[4]]
    projecao.append(aux)
arquivo.close()
```

**Figura 6. Leitura do arquivo com todos os pontos**

```python
#Lendo pontos para o algoritmo de seis pontos
arquivo = open("seisPontos.txt","r")
texto = arquivo.readlines()
matrizSeisPontos = [];
projecaoSeisPontos = [];
for linha in texto :
    valores = linha.split()
    aux = [valores[0], valores[1], valores[2]]
    matrizSeisPontos.append(aux)
    aux = [valores[3], valores[4]]
    projecaoSeisPontos.append(aux)
arquivo.close()
```

**Figura 7. Leitura do arquivo com os pontos para serem utilizados pelo algoritmo dos seis pontos**

A Figura 10 mostra como foi feita em Python.

Vale ressaltar que a ideia foi a mesma, portanto a programação se deu em volta de atribuir os valores corretos nas celular corretas. Foi utilizado a ideia de array, disponibilizado pela biblioteca numpy. Posteriormente foi utilizado a formula :

$$A = C^{-1}P \tag{1}$$

onde A = matriz que desejo encontrar, C inverso é a matriz C da figura 9 inversa e P é a matriz com os valores da projeção em x e y. Por fim a Figura 11 representa a saída do programa em relação a questão 2A. Para esse calculo pelo Python foi necessário utilizar a biblioteca linalg para inversão da matriz e a biblioteca numpy para multiplicação de matrizes.

### 2.2. Questão 2 - Letra B

Utilize a base de dados completa e avalie o erro médio das projeções da matriz M.

RESPOSTA: Para essa questão foi necessário somar todos os valores encontrados da matrizes A e da matriz projeção e comparar os valores. A figura 12 representa os

```python
X1 = float(matrizSeisPontos[0][0])
X2 = float(matrizSeisPontos[1][0])
X3 = float(matrizSeisPontos[2][0])
X4 = float(matrizSeisPontos[3][0])
X5 = float(matrizSeisPontos[4][0])
X6 = float(matrizSeisPontos[5][0])

Y1 = float(matrizSeisPontos[0][1])
Y2 = float(matrizSeisPontos[1][1])
Y3 = float(matrizSeisPontos[2][1])
Y4 = float(matrizSeisPontos[3][1])
Y5 = float(matrizSeisPontos[4][1])
Y6 = float(matrizSeisPontos[5][1])

Z1 = float(matrizSeisPontos[0][2])
Z2 = float(matrizSeisPontos[1][2])
Z3 = float(matrizSeisPontos[2][2])
Z4 = float(matrizSeisPontos[3][2])
Z5 = float(matrizSeisPontos[4][2])
Z6 = float(matrizSeisPontos[5][2])

x1 = float(projecaoSeisPontos[0][0])
x2 = float(projecaoSeisPontos[1][0])
x3 = float(projecaoSeisPontos[2][0])
x4 = float(projecaoSeisPontos[3][0])
x5 = float(projecaoSeisPontos[4][0])
x6 = float(projecaoSeisPontos[5][0])

y1 = float(projecaoSeisPontos[0][1])
y2 = float(projecaoSeisPontos[1][1])
y3 = float(projecaoSeisPontos[2][1])
y4 = float(projecaoSeisPontos[3][1])
y5 = float(projecaoSeisPontos[4][1])
y6 = float(projecaoSeisPontos[5][1])
```

**Figura 8. Inserção dos pontos para utilizar o algoritmo**

valores obtidos por ambas. O somatório da matriz A foi igual A(ou M) foi igual a 5.05569 e da matriz de projeção foi 30.6785 e o erro girou em torno de 25.62285.

A Figura 13 mostra os valores dos pontos das matrizes A e projeção, respectivamente.

### 2.3. Questão 2 - Letra C

A partir da base completa, escolha aleatoriamente outros três conjuntos de seis pontos e determine as três respectivas matrizes M. Faça o passo b para cada uma delas e compare os erros médios entre todas as quatro matrizes encontradas. Há alguma variação?

O primeiro conjunto de pontos esta representados pela Figura 14, o segundo pela Figura 15 e o terceiro pela figura 16

A matriz encontrada pela primeira iteração esta representada na Figura 17. O seu somatório é igual a 7.00934 e somatório da matriz projeção para este caso foi igual a 20.04118 e o erro médio foi 13.03184.

A matriz encontrada pela segunda iteração esta representada na Figura 18. O seu somatório é igual a 8.05928 e somatório da matriz projeção para este caso foi igual a 31.41925 e o erro médio foi 23.35997.

A matriz encontrada pela terceira iteração esta representada na Figura 19. O seu somatório é igual a 5.07349 e somatório da matriz projeção para este caso foi igual a 32.67266 e o erro médio foi 27.59917.

$$
\underbrace{\begin{pmatrix}
X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 \\
0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 \\
X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 & -x_2Z_2 \\
0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -y_2X_2 & -y_2Y_2 & -y_2Z_2 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
X_6 & Y_6 & Z_6 & 1 & 0 & 0 & 0 & 0 & -x_6X_6 & -x_6Y_6 & -x_6Z_6 \\
0 & 0 & 0 & 0 & X_6 & Y_6 & Z_6 & 1 & -y_6X_6 & -y_6Y_6 & -y_6Z_6
\end{pmatrix}}_{C}
\begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{41} \\ a_{42} \\ a_{43} \end{pmatrix}
=
\begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \cdots \\ x_5 \\ y_5 \\ x_6 \\ y_6 \end{pmatrix}
$$

**Figura 9. Calibração de câmera**

```
C = np.array([ [X1, Y1, Z1, 1, 0, 0, 0, 0, -x1*X1, -x1*Y1, -x1*Z1],
               [0, 0, 0, 0, X1, Y1, Z1, 1, -y1*X1, -y1*Y1, -y1*Z1],
               [X2, Y2, Z2, 1, 0, 0, 0, 0, -x2*X2, -x2*Y2, -x2*Z2],
               [0, 0, 0, 0, X2, Y2, Z2, 1, -y2*X2, -y2*Y2, -y2*Z2],
               [X3, Y3, Z3, 1, 0, 0, 0, 0, -x3*X3, -x3*Y3, -x3*Z3],
               [0, 0, 0, 0, X3, Y3, Z3, 1, -y3*X3, -y3*Y3, -y3*Z3],
               [X4, Y4, Z4, 1, 0, 0, 0, 0, -x4*X4, -x4*Y4, -x4*Z4],
               [0, 0, 0, 0, X4, Y4, Z4, 1, -y4*X4, -y4*Y4, -y4*Z4],
               [X5, Y5, Z5, 1, 0, 0, 0, 0, -x5*X5, -x5*Y5, -x4*Z5],
               [0, 0, 0, 0, X5, Y5, Z5, 1, -y5*X5, -y5*Y5, -y5*Z5],
               [X6, Y6, Z6, 1, 0, 0, 0, 0, -x6*X6, -x6*Y6, -x6*Z6]
])
```

**Figura 10. Calibração de câmera em Python**

. Os 4 erros médios foram: 25.62286, 13.03184, 23.35997, 27.59917. Estes estão representados na Figura 20.

.

```
^[[Alucas@lucas-Inspiron-15-7000-Gaming:~/Udesc/PIM$ python exer2.py
RESPOSTA DA 2A

MATRIZ M como pedido no exercicio ou MATRIZ A como demonstrado na formula
-0.00225
0.00113
-0.00011
2.54557
0.00053
0.00080
-0.00265
2.51303
-0.00018
-0.00017
-0.00000
```

**Figura 11. Resultado da questão 2A via console**

```
RESPOSTA DA 2B
Somatorio Matriz Obtida (M) = 5.05569
Somatorio Matriz dada = 30.67854
ERRO = 25.62285
```

**Figura 12. Resultado da questão 2B via console**

```
MATRIZ A:
[ -2.24505332e-03   1.12630270e-03  -1.08505526e-04   2.54556527e+00
   5.33216793e-04   8.03205152e-04  -2.65273526e-03   2.51302628e+00
  -1.78415212e-04  -1.71251229e-04  -4.09684901e-06]
MATRIZ PROJECAO:
[ 4.40879  3.36464  3.16839  3.57618  1.69582  3.84762  0.91381  2.36587
  2.09732  3.48494  1.75516]
```

**Figura 13. valores dos pontos das matriz A e matriz projeção**

```
RESPOSTA DA 2C - Iteracao 1

OS PONTOS SAO:
800.00000 1000.00000 600.00000 2.66760 3.13730
1000.00000 200.00000 600.00000 0.58885 2.05860
200.00000 1000.00000 0.00000 3.97308 3.13730
1000.00000 400.00000 1000.00000 0.79087 0.74579
1000.00000 0.00000 1000.00000 0.13717 0.26327
```

**Figura 14. Primeiro conjunto de pontos**

```
RESPOSTA DA 2C - Iteracao 2

OS PONTOS SAO:
1000.00000 400.00000 0.00000 1.06921 4.41365
1000.00000 600.00000 0.00000 1.39725 4.80664
1000.00000 400.00000 400.00000 0.96909 4.41365
1000.00000 1000.00000 200.00000 2.13185 5.02032
600.00000 1000.00000 0.00000 3.12786 4.94049
```

**Figura 15. Segundo conjunto de pontos**

```
RESPOSTA DA 2C - Iteracao 3

OS PONTOS SAO:
600.00000 1000.00000 200.00000 3.14743 4.28184
1000.00000 200.00000 400.00000 0.65222 2.76585
1000.00000 600.00000 200.00000 1.35777 4.28184
200.00000 1000.00000 0.00000 3.97308 4.28140
600.00000 1000.00000 400.00000 3.16839 3.57618
```

**Figura 16. Terceiro conjunto de pontos**

```
RESPOSTA DA 2C - Iteracao 1

OS PONTOS SAO:
800.00000 1000.00000 600.00000 2.66760 3.13730
1000.00000 200.00000 600.00000 0.58885 2.05860
200.00000 1000.00000 0.00000 3.97308 3.13730
1000.00000 400.00000 1000.00000 0.79087 0.74579
1000.00000 0.00000 1000.00000 0.13717 0.26327

MATRIZ M como pedido no exercicio ou MATRIZ A como demonstrado na formula
-0.00334
0.00058
-0.00004
3.48352
-0.00190
0.00050
-0.00152
3.53213
-0.00064
-0.00002
0.00007
Somatorio Matriz Obtida (M) = 7.00934
Somatorio Matriz dada = 20.04118
```

**Figura 17. Matriz M da primeira iteração**

```
RESPOSTA DA 2C - Iteracao 2

OS PONTOS SAO:
1000.00000 400.00000 0.00000 1.06921 4.41365
1000.00000 600.00000 0.00000 1.39725 4.80664
1000.00000 400.00000 400.00000 0.96909 4.41365
1000.00000 1000.00000 200.00000 2.13185 5.02032
600.00000 1000.00000 0.00000 3.12786 4.94049

MATRIZ M como pedido no exercicio ou MATRIZ A como demonstrado na formula
-0.00313
-0.00000
0.00000
3.12786
-0.00494
0.00000
0.00000
4.94049
-0.00100
0.00000
0.00000
Somatorio Matriz Obtida (M) = 8.05928
Somatorio Matriz dada = 31.41925
```

**Figura 18. Matriz M da segunda iteração**

```
RESPOSTA DA 2C - Iteracao 3

OS PONTOS SAO:
600.00000 1000.00000 200.00000 3.14743 4.28184
1000.00000 200.00000 400.00000 0.65222 2.76585
1000.00000 600.00000 200.00000 1.35777 4.28184
200.00000 1000.00000 0.00000 3.97308 4.28140
600.00000 1000.00000 400.00000 3.16839 3.57618

MATRIZ M como pedido no exercicio ou MATRIZ A como demonstrado na formula
-0.00213
0.00106
-0.00024
2.51990
0.00056
0.00073
-0.00277
2.55677
-0.00016
-0.00017
-0.00007
Somatorio Matriz Obtida (M) = 5.07349
Somatorio Matriz dada = 32.67266
```

**Figura 19. Matriz M da terceira iteração**

```
ERROS :
[25.622845780176284, 13.031838480198452, 23.359968350063006, 27.599169494534081]
```

**Figura 20. Erros médios de todas as iterações**