



UNICAMP

Universidade Estadual de Campinas

Faculdade de Engenharia Mecânica

IM563 – Processamento de Imagens
Aplicado à Automação e Robótica

10 de abril de 2024

Docentes: Paulo R. G. Kurka

Discente:

– Gabriel Toffanetto França da Rocha – 289320

Trabalho 2

Sumário

1	A	2
2	B	6
3	C	7
4	D	7
5	E	8
	Anexos	9
	Referências	9

1 A

* Dado o plano \mathbb{R}^3 ao lado, sendo \vec{v} um vetor qualquer, e \hat{k} um vetor unitário, pode-se decompor \vec{v} em:

$$\vec{v} = \vec{v}_{\parallel} + \vec{v}_{\perp}, \text{ onde:}$$

\vec{v}_{\parallel} é a projeção de \vec{v} em \hat{k} :

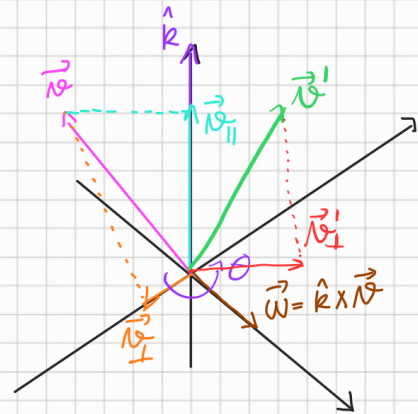
$$\vec{v}_{\parallel} = \langle \vec{v}, \hat{k} \rangle \hat{k}, \text{ já que } \|\hat{k}\|_2 = 1.$$

\vec{v}_{\perp} pode ser reescrito como:

$$\vec{v} = \vec{v}_{\perp} + \vec{v}_{\parallel} : \vec{v}_{\perp} = \vec{v} - \vec{v}_{\parallel} = \vec{v} - \langle \vec{v}, \hat{k} \rangle \hat{k} = \langle -\hat{k}, \vec{v} \rangle \hat{k} - \langle -\hat{k}, \hat{k} \rangle \vec{v}$$

Pela fórmula do produto interno triplo:

$$\vec{v}_{\perp} = -\hat{k} \times (\hat{k} \times \vec{v})$$



* Ao rotacionar \vec{v} de forma arbitrária, obtemos \vec{v}' , que pode ser decomposto como feito em \vec{v} . Como \hat{k} é um vetor perpendicular ao eixo de rotação, $\vec{v}_{\parallel} = \vec{v}'_{\parallel}$.

Porém, a componente ortogonal a \hat{k} tem sua posição alterada.

Considerando um sistema de coordenadas ortogonal em \vec{v}_{\perp} , $\hat{k} \times \vec{v}_{\perp}$, pode-se fazer uma rotação de acordo com as coordenadas polares.

$$\vec{v}'_{\perp} = \vec{v}_{\perp} \cos(\theta) + (\hat{k} \times \vec{v}_{\perp}) \sin(\theta)$$

* Retornando \vec{v}' :

$$\vec{v}' = \vec{v}'_{\parallel} + \vec{v}'_{\perp} = \vec{v}_{\parallel} + \vec{v}_{\perp} \cos(\theta) + (\hat{k} \times \vec{v}_{\perp}) \sin(\theta)$$

$$\text{Como } \hat{k} \times \vec{v}_{\perp} = \hat{k} \times \vec{v}:$$

$$\vec{r}' = \vec{r}_{\parallel} + \vec{r}_{\perp} \cos(\theta) + (\hat{k} \times \vec{r}) \sin(\theta), \text{ como } \vec{r}_{\parallel} = \vec{r} \cdot \hat{k} \hat{k} :$$

$$\begin{aligned} \vec{r}' &= \vec{r} - \vec{r}_{\perp} + \vec{r}_{\perp} \cos(\theta) + (\hat{k} \times \vec{r}) \sin(\theta) \\ &= \vec{r} - (1 - \cos(\theta)) \vec{r}_{\perp} + (\hat{k} \times \vec{r}) \sin(\theta) \end{aligned}$$

$$\vec{r}' = \vec{r} + (1 - \cos(\theta)) \hat{k} \times (\hat{k} \times \vec{r}) + (\hat{k} \times \vec{r}) \sin(\theta)$$

* O produto vetorial de $\hat{k} \times \vec{r}$ pode ser reescrito de forma matricial, onde:

$$\hat{k} \times \vec{r} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ k_1 & k_2 & k_3 \\ r_1 & r_2 & r_3 \end{vmatrix} = \hat{i}(k_2 r_3 - k_3 r_2) + \hat{j}(k_3 r_1 - k_1 r_3) + \hat{k}(k_1 r_2 - k_2 r_1)$$

$$\hat{k} \times \vec{r} = \begin{bmatrix} k_2 r_3 - k_3 r_2 \\ k_3 r_1 - k_1 r_3 \\ k_1 r_2 - k_2 r_1 \end{bmatrix} = \begin{bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{bmatrix} = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{bmatrix}$$

$$\hat{k} \times = K = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \text{ é a matriz produto vetorial de } \hat{k}.$$

$$\text{Como } \hat{k} \times \vec{r} = K \vec{r}$$

$$\hat{k} \times (\hat{k} \times \vec{r}) = \hat{k} \times K \vec{r} = K^2 \vec{r}$$

$$\vec{r}' = \vec{r} + (1 - \cos(\theta)) K^2 \vec{r} + \sin(\theta) K \vec{r}$$

De $R = I + \sin(\theta) K + (1 - \cos(\theta)) K^2$, pode-se dizer que:

$$\vec{r}' = R \vec{r}, \text{ onde } R \text{ é a matriz de rotação de } \vec{r} \text{ para } \vec{r}'.$$

* Porém, como se relaciona a rotação de Rodrigues com a rotação nos ângulos de Euler (ψ, θ, ϕ) ?

Como $\vec{B}' = R\vec{B}$, $R = R(\theta)$ é a matriz da rotação de Rodrigues de um ângulo θ em torno de um vetor unitário \hat{n} .

A matriz R é dada como:

$$R = \begin{bmatrix} -\cos(\theta)(n_2^2 - n_3^2) - n_2^2 - n_3^2 + 1 & -\cos(\theta)n_1n_2 + n_1n_2 - \sin(\theta)n_3 & -\cos(\theta)n_1n_3 + n_1n_3 + \sin(\theta)n_2 \\ -\cos(\theta)n_1n_2 + n_1n_2 + \sin(\theta)n_3 & -\cos(\theta)(n_1^2 - n_3^2) - n_1^2 - n_3^2 + 1 & -\cos(\theta)n_2n_3 + n_2n_3 + \sin(\theta)n_1 \\ -\cos(\theta)n_1n_3 + n_1n_3 - \sin(\theta)n_2 & -\cos(\theta)n_2n_3 + n_2n_3 + \sin(\theta)n_1 & -\cos(\theta)(n_1^2 - n_2^2) - n_1^2 - n_2^2 + 1 \end{bmatrix}$$

As rotações em x, y e z podem ser expressas em $R(\theta)$, onde:

$$R_x = R(\theta) : \hat{n} = [1 \ 0 \ 0]^T$$

$$R_y = R(\theta) : \hat{n} = [0 \ 1 \ 0]^T$$

$$R_z = R(\theta) : \hat{n} = [0 \ 0 \ 1]^T$$

$$\text{Logo: } R_{z,y,x} = R_z R_y R_x = R(\theta) : \hat{n} = n_1 [1 \ 0 \ 0]^T + n_2 [0 \ 1 \ 0]^T + n_3 [0 \ 0 \ 1]^T$$

$$\forall n_1, n_2, n_3 \in \sqrt{n_1^2 + n_2^2 + n_3^2} = 1$$

Porém, a generalização da rotação de Rodrigues e a rotação nos eixos x, y, z é feita por meio dos Parâmetros de Euler, que estão diretamente aos Quaterniões.

Os Parâmetros de Euler, $\{a, b, c, d\}$, são dados em função de \hat{n} e θ , como segue:

$$a = \frac{\cos(\theta)}{2}$$

$$b = m_1 \frac{\sin(\theta)}{2}$$

$$c = m_2 \frac{\sin(\theta)}{2}$$

$$d = m_3 \frac{\sin(\theta)}{2}$$

$$\text{sendo } \hat{n} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}$$

Em termos de Quaterniões, um Quaternião q é dado como:

$$q = a + b\hat{i} + c\hat{j} + d\hat{k}, \text{ sendo } \hat{i}, \hat{j}, \hat{k} \text{ números imaginários}$$

A relação entre as rotações em x, y, z e os Parâmetros de Euler é dada por:

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \end{bmatrix}$$

O ângulo pode ser obtido por:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan2(2(a.b + c.d), 1 - 2(b^2 + c^2)) \\ -\pi/2 + 2\arctan2(\sqrt{1 + 2(a.c - b.d)}, \sqrt{1 - 2(a.c - b.d)}) \\ \arctan2(2(ad + bc), 1 - 2(c^2 + d^2)) \end{bmatrix}$$

Onde $\arctan2(x, y)$ é a arctangente $\left(\frac{x}{y}\right)$, considerando os sinais de x e y .

2 B

Sendo um ponto descrito no sistema de coordenadas $\{1\}$, dado por $\mathbf{p}_1 = [1 \ 2 \ 3]^T$ (m), o mesmo pode ser escrito em outro sistema de referências desde que se saiba qual é a translação e a rotação entre esses dois sistemas de referência.

Uma vez que a translação entre o sistema de referência $\{1\}$ e $\{2\}$ é dado por $\mathbf{t} = [1 \ 2 \ 15]^T$, e a rotação $\{\phi, \theta, \psi\} = \{2^\circ, 1^\circ, 2^\circ\}$, pode-se escrever o ponto no *frame* $\{2\}$ conforme (1).

$$\mathbf{p}_2 = \mathbf{R}(\{\phi, \theta, \psi\})\mathbf{p}_1 + \mathbf{t} \quad (1)$$

Onde:

$$\mathbf{R}(\{\phi, \theta, \psi\}) = \mathbf{R}(\psi)\mathbf{R}(\theta)\mathbf{R}(\phi) \quad (2)$$

A transformação de referencial foi implementada por meio da classe `tf`, descrita abaixo.

```

1 class tf:
2     def __init__(self, x, y, z, phi, theta, psi):
3         self.x = x
4         self.y = y
5         self.z = z
6         self.phi = np.deg2rad(phi)
7         self.theta = np.deg2rad(theta)
8         self.psi = np.deg2rad(psi)
9
10        self.t = np.array([[x, y, z]]).T
11
12    def getPoint(self, p1):
13        Rx = np.array([[1, 0, 0], [0, np.cos(self.psi), np.sin(self.psi)], [0, -
14            np.sin(self.psi), np.cos(self.psi)]])
15        Ry = np.array([[np.cos(self.theta), 0, -np.sin(self.theta)], [0, 1, 0],
16            [np.sin(self.theta), 0, np.cos(self.theta)]])
17        Rz = np.array([[np.cos(self.phi), np.sin(self.phi), 0], [-np.sin(self.
18            phi), np.cos(self.phi), 0], [0, 0, 1]])
19        R = np.matmul(Rz, np.matmul(Ry, Rx))
20
21        p2 = np.matmul(R, p1) + self.t
22        return p2

```

Executando a transformação, obteve-se a representação (3) do ponto \mathbf{p}_1 no referencial $\{2\}$.

$$\mathbf{p}_2 = \begin{bmatrix} 2.02157298 \\ 4.06908819 \\ 17.94537989 \end{bmatrix} \quad (3)$$

3 C

Dados as representações do ponto \mathbf{p}_1 e \mathbf{p}_2 , ambos podem ser representados nas coordenadas da projeção desses pontos no plano da imagem, por meio dos pontos \mathbf{q}_1 e \mathbf{q}_2 . A transformação é dada por (4).

$$\mathbf{q} = \mathbf{p} \frac{f}{p_z} \quad (4)$$

Sendo $f = 0,05$ m, têm-se que:

$$\mathbf{q}_1 = \begin{bmatrix} 0.01667 \\ 0.03333 \\ 0.05 \end{bmatrix} \quad (\text{m}) \quad (5)$$

$$\mathbf{q}_2 = \begin{bmatrix} 0.00563 \\ 0.01134 \\ 0.05 \end{bmatrix} \quad (\text{m}) \quad (6)$$

4 D

A projeção no plano da imagem é representadas em unidades métricas, não sendo essa a representação final da imagem capturada em termos da imagem em si. A projeção da imagem pode ser transformada para coordenadas de imagem, por meio de (7).

$$\hat{\mathbf{q}} = \text{floor} \left(\frac{1}{f} \mathbf{A} \mathbf{q} \right) \quad (7)$$

Sendo a matriz de parâmetros intrínsecos (\mathbf{A}) dada por (8).

$$\mathbf{A} = \begin{bmatrix} f s_x & 0 & u_0 \\ 0 & f s_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Onde tem-se que:

$$s_x = \frac{M}{L_x} \quad (9)$$

$$s_y = \frac{N}{L_y} \quad (10)$$

$$u_0 = 0 \quad (11)$$

$$v_0 = 0 \quad (12)$$

Dessa forma, obtém-se $\hat{\mathbf{q}}_1$ e $\hat{\mathbf{q}}_2$:

$$\hat{\mathbf{q}}_1 = \text{floor} \left(\frac{1}{f} \mathbf{A} \mathbf{q}_1 \right) = \begin{bmatrix} 27 \\ 69 \\ 1 \end{bmatrix} \text{ (pixel)} \quad (13)$$

$$\hat{\mathbf{q}}_2 = \text{floor} \left(\frac{1}{f} \mathbf{A} \mathbf{q}_2 \right) = \begin{bmatrix} 9 \\ 23 \\ 1 \end{bmatrix} \text{ (pixel)} \quad (14)$$

5 E

Relacionando o ponto no mundo com sua projeção no plano da imagem, como feito em (15), pode-se substituir o ponto projetado para obter diretamente a coordenada do mundo em coordenadas de imagem, como em (16).

Isolando \mathbf{p} em (16), obtém-se em (17) a transformação de reprojeção, do ponto em coordenadas de imagem para coordenadas do mundo.

$$\mathbf{q} = \frac{f}{p_z} \mathbf{p} \quad (15)$$

$$\hat{\mathbf{q}} = \frac{1}{f} \mathbf{A} \mathbf{q} = \frac{1}{f} \mathbf{A} \frac{f}{p_z} \mathbf{p} = \frac{1}{p_z} \mathbf{A} \mathbf{p} \quad (16)$$

$$\hat{\mathbf{p}} = \mathbf{A}^{-1} \hat{\mathbf{q}} p_z \quad (17)$$

Sendo $\hat{\mathbf{p}}$ o ponto reprojetado.

Dessa forma, foi possível reprojetar os pontos \mathbf{p}_1 e \mathbf{p}_2 , obtendo os resultados de (18) e (19).

$$\hat{\mathbf{p}}_1 = \mathbf{A}^{-1} \hat{\mathbf{q}}_1 p_{1z} = \begin{bmatrix} 0.972 \\ 1.987 \\ 3 \end{bmatrix} \quad (18)$$

$$\hat{\mathbf{p}}_2 = \mathbf{A}^{-1} \hat{\mathbf{q}}_2 p_{2z} = \begin{bmatrix} 1.938 \\ 3.962 \\ 17.945 \end{bmatrix} \quad (19)$$

O erro de reprojeção foi calculado com base na média da distância euclidiana do vetor erro de reprojeção, dando dado por (20).

$$e = \frac{1}{N} \sum_{i=1}^N \|\mathbf{e}_i\|_2 = \frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_i - \hat{\mathbf{p}}_i\|_2 \quad (20)$$

Para os pontos recuperados, o erro médio foi de 0,083.

Anexos

Códigos fonte

Todos os códigos fonte e arquivos de dados utilizados para a elaboração deste documento podem ser encontrados no repositório do GitHub no link: github.com/toffanetto/im563.