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

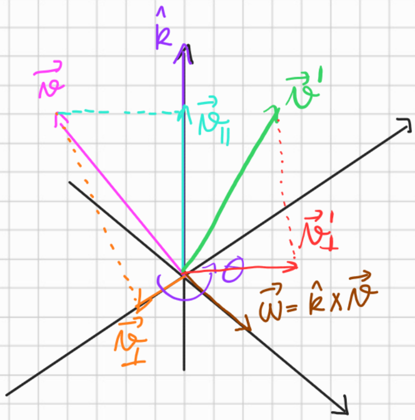
---

## Sumário



---

# 1 A




* 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:

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

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

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

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

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

Pela fórmula do produto interno triplo:

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

* Ao rotacionar $\vec{v}$ de forma arbitrária, obtém-se $\vec{v}'$, 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 à $\hat{k}$ terá sua posição alterada.

Considerando um sistema de coordenadas ortogonal em $\vec{v}_{\perp}$ e $\hat{k} \times \vec{v}_{\perp}$, pode escrever sua rotação de acordo com as coordenadas polares.

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

* Retornando $\vec{v}'$:

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

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

/ /

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

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

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

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

$$\hat{k}\times\vec{v} = \begin{vmatrix} \hat{\imath} & \hat{\jmath} & \hat{k} \\ k_1 & k_2 & k_3 \\ v_1 & v_2 & v_3 \end{vmatrix} = \hat{\imath}(k_2 v_3 - k_3 v_2) + \hat{\jmath}(k_3 v_1 - k_1 v_3) + \hat{k}(k_1 v_2 - k_2 v_1)$$

$$\hat{k}\times\hat{v} = \begin{bmatrix} k_2 v_3 - k_3 v_2 \\ k_3 v_1 - k_1 v_3 \\ k_1 v_2 - k_2 v_1 \end{bmatrix}\begin{bmatrix} \hat{\imath} \\ \hat{\jmath} \\ \hat{k} \end{bmatrix} = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}\begin{bmatrix} \hat{\imath} \\ \hat{\jmath} \\ \hat{k} \end{bmatrix}$$

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

Como $\hat{k}\times\vec{v} = K\vec{v}$

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

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

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

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

/ /

* Porém, como se relaciona a rotação de Rodrigues com a rotação nos ângulos de Euler $(\psi, \theta, \varphi)$?

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

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

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

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

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

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

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

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

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

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

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

$$b = n_1 \frac{\sin(\theta)}{2}$$

$$c = n_2 \frac{\sin(\theta)}{2}$$

$$d = n_3 \frac{\sin(\theta)}{2}$$

* Sendo $\hat{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}$

Em termos de Quaternios, um Quaternio $q$ é dado como:

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

A relação entre as rotações em x, y e z e os Parametros 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 inverso pode ser obtido por:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}(2(a.b + c.d),\ 1 - 2(b^2 + c^2)) \\ -\pi/2 + 2\operatorname{atan2}\left(\sqrt{1 + 2(a.c - b.d)},\ \sqrt{1 - 2(a.c - bd)}\right) \\ \operatorname{atan2}(2(ad + bc),\ 1 - 2(c^2 + d^2)) \end{bmatrix}$$

Onde $\operatorname{atan2}(x,y)$ é a $\arctan(\frac{x}{y})$, conservando 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 $\{\phi, \theta, \psi\} = \{2°, 1°, 2°\}$, pode-se escrever o ponto no *frame* $\{2\}$ conforme (1).

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

Onde:

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

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

```
class tf:
  def __init__(self, x, y, z, phi, theta, psi):
    self.x = x
    self.y = y
    self.z = z
    self.phi = np.deg2rad(phi)
    self.theta = np.deg2rad(theta)
    self.psi = np.deg2rad(psi)

    self.t = np.array([[x, y ,z]]).T

  def getPoint(self,p1):
    Rx = np.array([[1, 0, 0], [0, np.cos(self.psi), np.sin(self.psi)], [0, -np.sin(self.psi), np.cos(self.psi)]])
    Ry = np.array([[np.cos(self.theta), 0, -np.sin(self.theta)], [0, 1, 0], [np.sin(self.theta), 0, np.cos(self.theta)]])
    Rz = np.array([[np.cos(self.phi), np.sin(self.phi), 0], [-np.sin(self.phi), np.cos(self.phi), 0], [0, 0, 1]])
    R = np.matmul(Rz,np.matmul(Ry,Rx))

    p2 = np.matmul(R,p1) + self.t
    return p2
```

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

$$\mathbf{p}_2 = \begin{bmatrix} 2.02157298 \\ 4.06908819 \\ 17.94537989 \end{bmatrix} \tag{3}$$

# 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).

$$\mathbf{q} = \mathbf{p}\frac{f}{p_z} \tag{4}$$

Sendo $f = 0,05$ m, têm-se que:

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

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

# 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).

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

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

$$\mathbf{A} = \begin{bmatrix} fs_x & 0 & u_0 \\ 0 & fs_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Onde tem-se que:

$$s_x = \frac{M}{L_x} \tag{9}$$

$$s_y = \frac{N}{L_y} \tag{10}$$

$$u_0 = 0 \tag{11}$$

$$v_0 = 0 \tag{12}$$

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

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

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

# 5 E

$$\mathbf{q} = \frac{f}{p_z}\mathbf{p} \tag{15}$$

$$\hat{\mathbf{q}} = \frac{1}{f}\mathbf{A}\mathbf{q} = \frac{1}{f}\mathbf{A}\frac{f}{p_z}\mathbf{p} = \frac{1}{p_z}\mathbf{A}\mathbf{p} \tag{16}$$

$$\mathbf{p} = \mathbf{A}^{-1}\hat{\mathbf{q}}p_z \tag{17}$$

# 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.