Universidad de Valparaíso CHILE

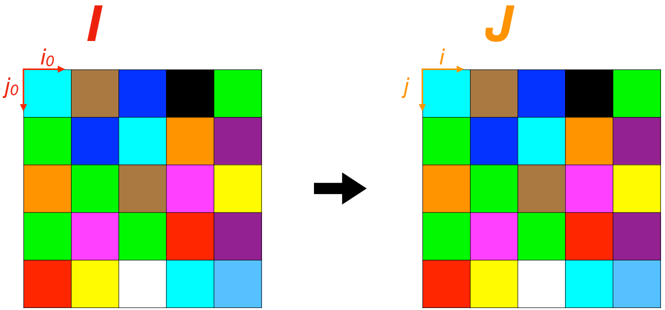
Transformaciones geométricas y registro

Unidad 3

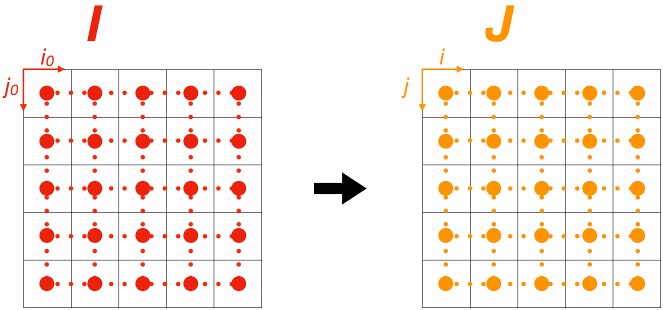
BME423 · Procesamiento de imágenes médicas

Alejandro Veloz
Ingeniería Biomédica

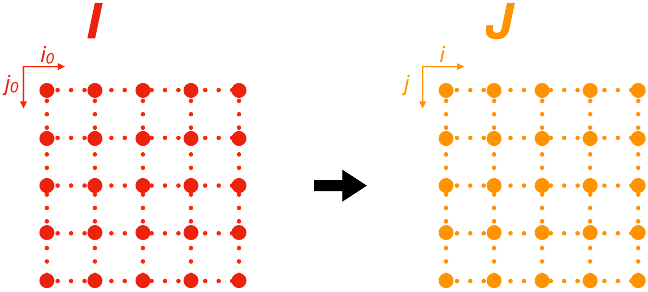
Transformaciones geométricas



Transformaciones geométricas



Transformaciones geométricas

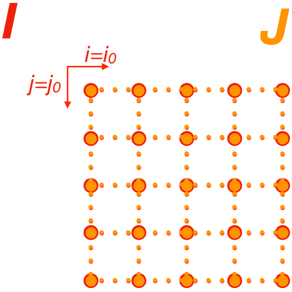


Transformación
de coordenadas

$$i = f_i(i_0, j_0)$$

$$j = f_j(i_0, j_0)$$

Transformaciones geométricas



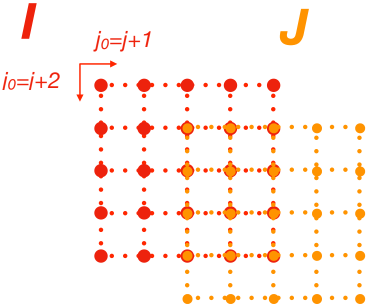
$$i = i_0$$

$$j = j_0$$

Transformación
de coordenadas

(Ejemplo)

Transformaciones geométricas



$$i = i_0 - 1$$

$$j = j_0 - 2$$

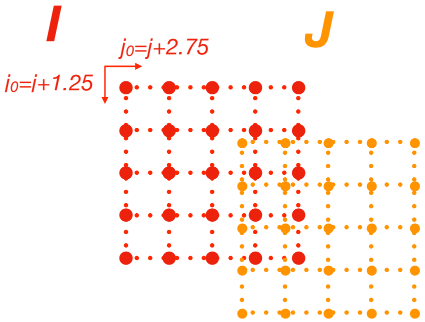
Transformación de coordenadas

(Ejemplo)

Algorithm:

1) For each (i, j) of **J** compute (i_0, j_0)
2) **J**(i, j) = **I**(i_0, j_0)

Transformaciones geométricas



$$i = i_0 - 1.25$$

$$j = j_0 - 2.75$$

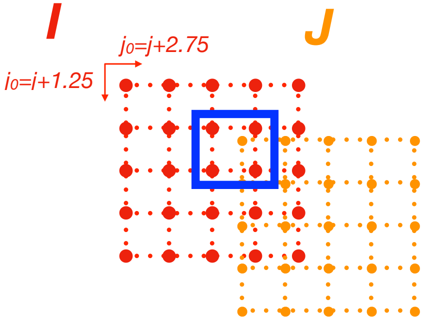
Transformación de coordenadas

(Ejemplo)

Algorithm:

1) For each (i, j) of **J** compute (i_0, j_0)
2) **J**(i, j) = Interpolation[**I**(i_0, j_0)]

Transformaciones geométricas



$$i = i_0 - 1.25$$

$$j = j_0 - 2.75$$

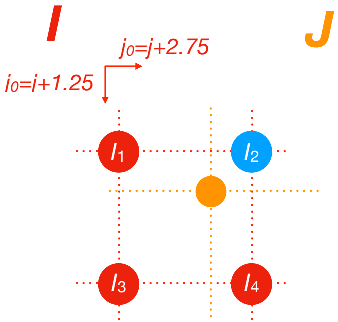
Transformación
de coordenadas

(Ejemplo)

Algorithm:

1) For each (i, j) of ***J*** compute (i_0, j_0)
2) ***J***(i, j) = Interpolation[***I***(i_0, j_0)]

Transformaciones geométricas



$$i = i_0 - 1.25$$

$$j = j_0 - 2.75$$

Algorithm:

1) For each (i, j) of ***J*** compute (i_0, j_0)
2) ***J***(i, j) = **Interpolation**[***I***(i_0, j_0)]

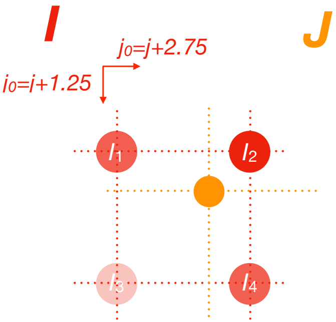
Nearest

J(i, j) = **Interpolation**[***I***(i_0, j_0)] = I_2

Transformación
de coordenadas

(Ejemplo)

Transformaciones geométricas



Transformación
de coordenadas

(Ejemplo)

$$i = i_0 - 1.25$$

$$j = j_0 - 2.75$$

Algorithm:

1) For each (i, j) of J compute (i_0, j_0)
2) $J(i, j)$ = **Interpolation**[$I(i_0, j_0)$]

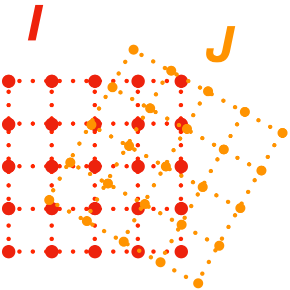
Bilinear interpolation

$J(i, j)$ = **Interpolation**[$I(i_0, j_0)$]

$J(i, j) = aI_1 + bI_2 + cI_3 + dI_4$

$a+b+c+d = 1$

Transformaciones geométricas



Algorithm:

1) For each (i, j) of ***J*** compute (i_0, j_0)
2) ***J***(i, j) = Interpolation[***I***(i_0, j_0)]

$$i_0 = i\ \cos\theta + j\ \sin\theta + a$$
$$j_0 = -i\ \sin\theta + j\ \cos\theta + b$$

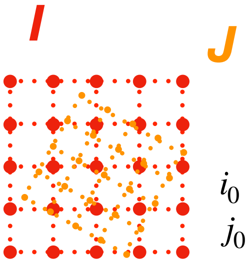
Rotación

Traslación

Transformación
de coordenadas

(Ejemplo)

Transformaciones geométricas



Algorithm:
1) For each (i, j) of ***J*** compute (i_0, j_0)
2) ***J***(i, j) = Interpolation[***I***(i_0, j_0)]

$$i_0 = i\, s\, \cos\theta + j\, s\, \sin\theta + a$$
$$j_0 = -\, i\, s\, \sin\theta + j\, s\, \cos\theta + b$$

Rotación

Traslación

Escalamiento

Transformación
de coordenadas

(Ejemplo)

Transformaciones geométricas

En forma matricial:

$$M = \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & a_{14} \\ b_{11} & b_{12} & b_{13} & b_{14} \\ c_{11} & c_{12} & c_{13} & c_{14} \\ 0 & 0 & 0 & 1 \end{array}\right]$$

Transformaciones geométricas

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Traslación

$$x' = x + t_x$$
$$y' = y + t_y$$
$$z' = z + t_z$$

En forma matricial:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix}$$

Escalamiento

$$x' = s_x x$$
$$y' = s_y y$$
$$z' = s_z z$$

En forma matricial:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ s_z z \\ 1 \end{bmatrix}$$

Matrices de rotación

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

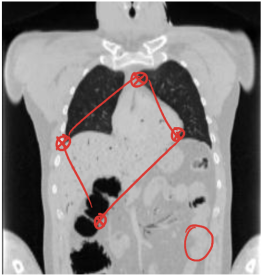
Combinación de rotaciones

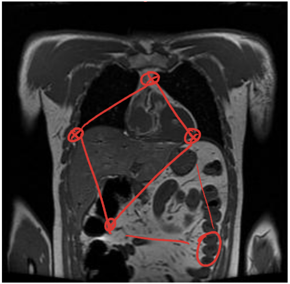
$$R_x(\alpha)R_y(\beta)R_z(\gamma)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Point-based registration

- Implica identificar puntos correspondientes en las dos imágenes para derivar una transformación que las alinee. Ej. CT-MRI

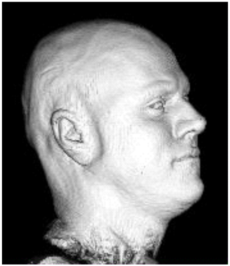


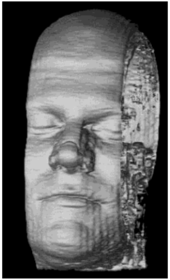


- Requiere conocimiento experto para identificar puntos y puede resultar en errores grandes si los puntos se marcan de manera inexacta o si hay una distancia significativa entre ellos.

Surface-based registration

- Implica extraer iso-superficies de las imágenes y alinear estas superficies.





- Extraer automáticamente la misma superficie de ambas imágenes puede ser difícil, y el método puede no asegurar un buen ajuste lejos de la superficie.

Intensity-based registration

- Esta categoría incluye varios métodos que utilizan valores de intensidad para medir la calidad de la alineación.

- Suma de Diferencias Absolutas (SAD)
  $\text{SAD}(f,g) = \sum_{m=1}^{M} \sum_{n=1}^{N} |f_{mn} - g_{mn}|$

- Cross-correlation (CC) $\quad \text{CC}(f,g) = \frac{\sum_{m,n} f_{mn} g_{mn}}{\sqrt{\sum_{m,n} f_{mn}^2 \cdot \sum_{m,n} g_{mn}^2}}$

- Suma de Diferencias al Cuadrado (SSD)
  $d_{\text{SSD}}(f,g) = \sum_{m,n} (f_{mn} - g_{mn})^2$

- Mutual information $\text{MI}(f,g) = H(f) + H(g) - H(f,g)$, donde

  $H(f)$ and $H(g)$ son las entropías marginales,

  $H(f) = -\sum_i p(i) \log p(i)$, $p(i)$ es la probabilidad del valor de intensidad i, y

  $H(f,g) = -\sum_{i,j} p(i,j) \log p(i,j)$ es la entropía conjunta.

El proceso de registro

Modelo de Transformación: Define cómo se transforma una imagen para alinearse con otra (por ejemplo, traslación, rotación).

Función de Costo: Cuantifica la calidad de la alineación (por ejemplo, SAD, CC, SSD).

Optimización: Ajusta los parámetros de transformación para optimizar la función de costo, logrando así la mejor alineación posible.

$$\mathbf{m} = \arg\min_{\mathbf{m}} \{d(T(\mathbf{m})f, g)\} \quad \text{o} \quad \mathbf{m} = \arg\max_{\mathbf{m}} \{d(T(\mathbf{m})f, g)\}$$

Ver Ejemplo

`syn_registration_3d.ipynb`

(Classroom - Librería Dipy)