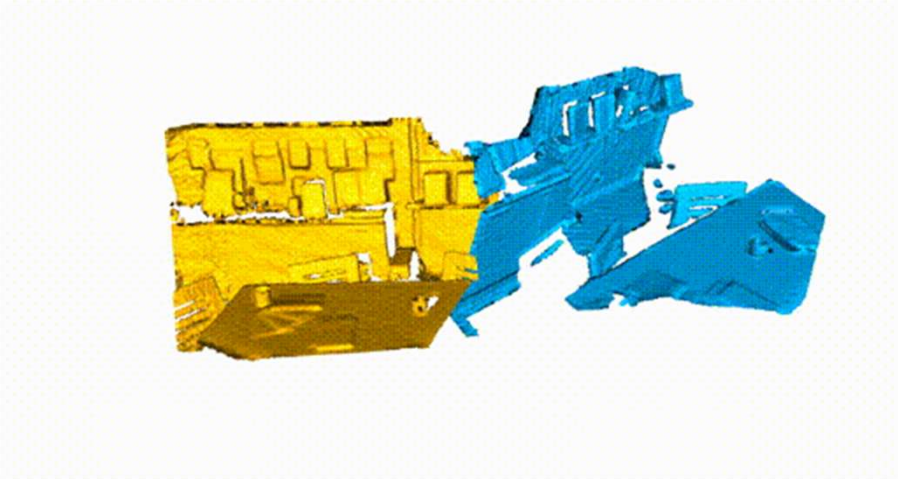
Procesamiento Geométrico y Análisis de Formas

Ivan Sipiran

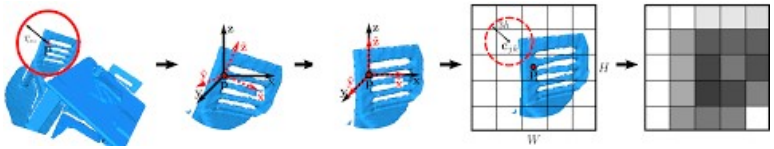
Global Registration

- ICP requiere que exista un alineamiento inicial bueno
- A veces esa condición no es fácil de obtener
- Necesitamos otro tipo de algoritmo para hacer “global registration”

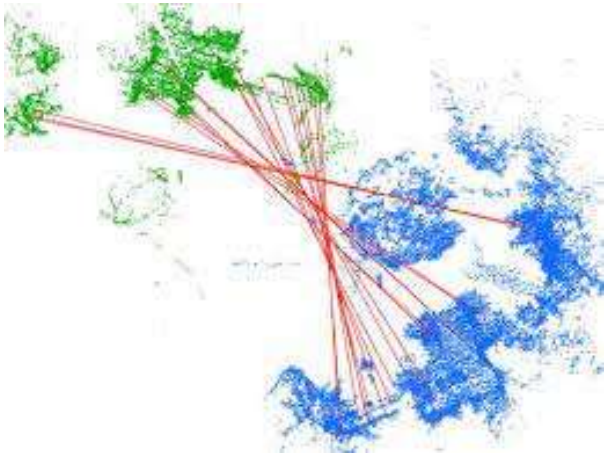


Global Registration Pipeline

- Es necesario contar con más información que solo la nube de puntos
- El problema específico es: cómo encontrar buenas correspondencias sin asumir cercanía de puntos.
- Solución: búsquedas en espacios característicos

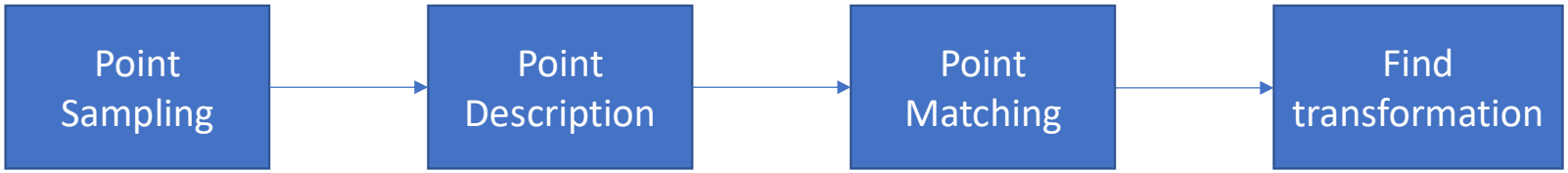


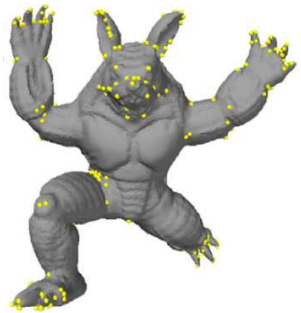
Point description

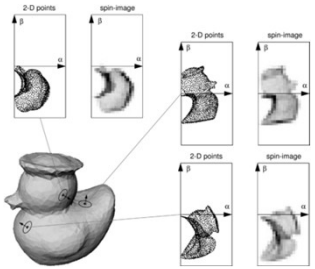


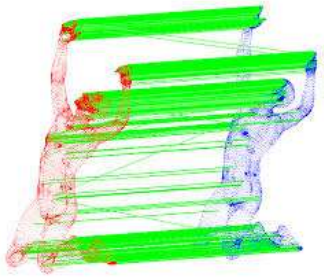
Point matching

Global Registration Pipeline





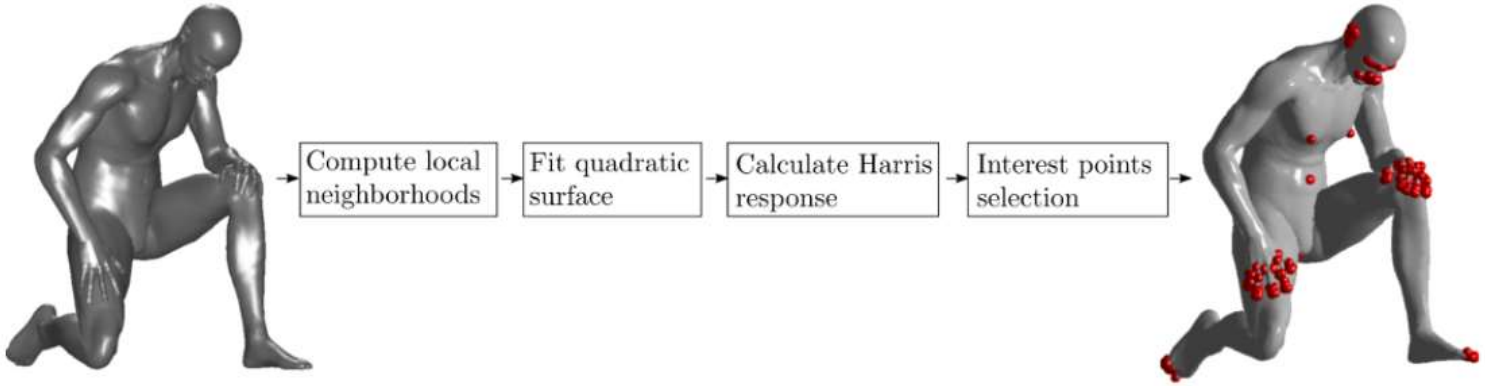




Point Sampling

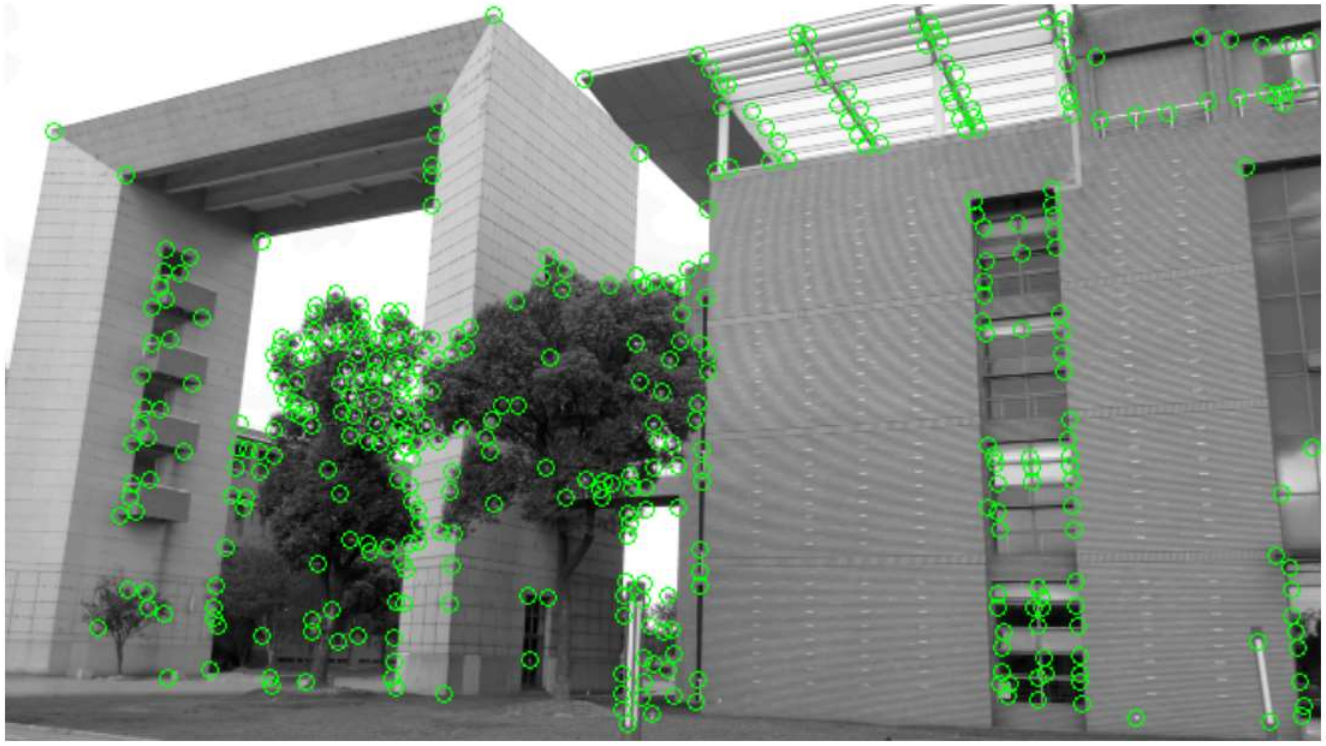
Algoritmo Harris 3D

- Detectar puntos de interés en geometría 3D
  - Funciona sobre mallas y sobre nubes de puntos
- Pipeline



Algoritmo Harris 3D

- Existe un algoritmo para detectar puntos de interés en imágenes



Algoritmo Harris 3D

- Definimos $I(x, y)$ una imagen como una función
- Para detectar una esquina, hay que evaluar los cambios de una vecindad consigo misma
- Función de correlación

$$e(x, y) = \sum_{x_i, y_i} W(x_i, y_i)[I(x_i + \triangle x, y_i + \triangle y) - I(x_i, y_i)]^2$$

- Se usa una función Gaussiana para ponderar la vecindad.

Algoritmo Harris 3D

- Usando una expansión de Taylor, la función de auto-correlación queda

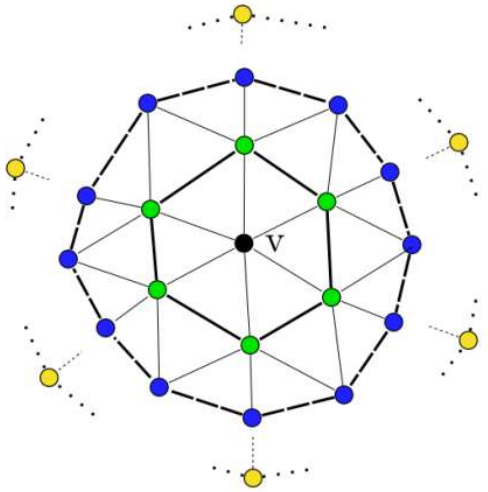
$$e(x,y) = \vec{S}\begin{bmatrix} \sum_{x_i,y_i} W.I_x^2 & \sum_{x_i,y_i} W.I_x.I_y \\ \sum_{x_i,y_i} W.I_x.I_y & \sum_{x_i,y_i} W.I_y^2 \end{bmatrix}\vec{S}^T$$
$$= \vec{S}E(x,y)\vec{S}^T$$

- Donde $\vec{S} = [\Delta x \ \ \Delta y]$ es un vector de desplazamiento. I_x, I_y son las derivadas parciales de la imagen en los ejes coordenados
- Estructura local de la imagen queda determinada por los valores propios de la matriz

$$h(x,y) = \det(E) - k\big(\mathrm{tr}(E)\big)^2$$

Algoritmo Harris 3D

- Extensión para geometría 3D no es trivial
  - Cómo se definen vecindades?
  - Cómo se calculan derivadas?
- Para mallas, vecindades adaptivas



$$ring_k(v) = \{w \in V' \text{ such that } |shortest\_path(v, w)| = k\}$$

$$d_{ring}(v, ring_k(v)) = max_{w \in ring_k(v)} \|v - w\|_2$$

$$radius_v = \{k \in \mathbb{N} \text{ such that } d_{ring}(v, ring_k(v)) \geq \delta \text{ and } d_{ring}(v, ring_{k-1}(v)) < \delta\}$$

Algoritmo Harris 3D

- Cómo lidiamos con las derivadas?
- Fitting de una superficie cuadrática en punto v
  - Trasladar la vecindad al origen
  - Normalizar orientación con análisis de componentes principales
  - Ajustar una superficie cuadrática

$$z = f(x, y) = \frac{p_1}{2}x^2 + p_2xy + \frac{p_3}{2}y^2 + p_4x + p_5y + p_6$$

  - Función $f(x, y)$ es similar a una imagen!

Algoritmo Harris 3D

- Calcular la matriz de Harris
- Para manejar cambios locales en la geometría: suavizamiento

$$A = \frac{1}{2\sigma^4\pi} \int_{\mathbb{R}^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \cdot \left(\frac{\partial f(x,y)}{\partial x} \right)^2 dxdy$$

$$B = \frac{1}{2\sigma^4\pi} \int_{\mathbb{R}^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \cdot \left(\frac{\partial f(x,y)}{\partial y} \right)^2 dxdy$$

$$C = \frac{1}{2\sigma^4\pi} \int_{\mathbb{R}^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \cdot \left(\frac{\partial f(x,y)}{\partial x} \right) \left(\frac{\partial f(x,y)}{\partial y} \right) dxdy$$

Algoritmo Harris 3D

- Al evaluar las integrales en el continuo, obtenemos
- Se forma matriz de auto-correlación

$$E = \begin{pmatrix} A & C \\ C & B \end{pmatrix}$$

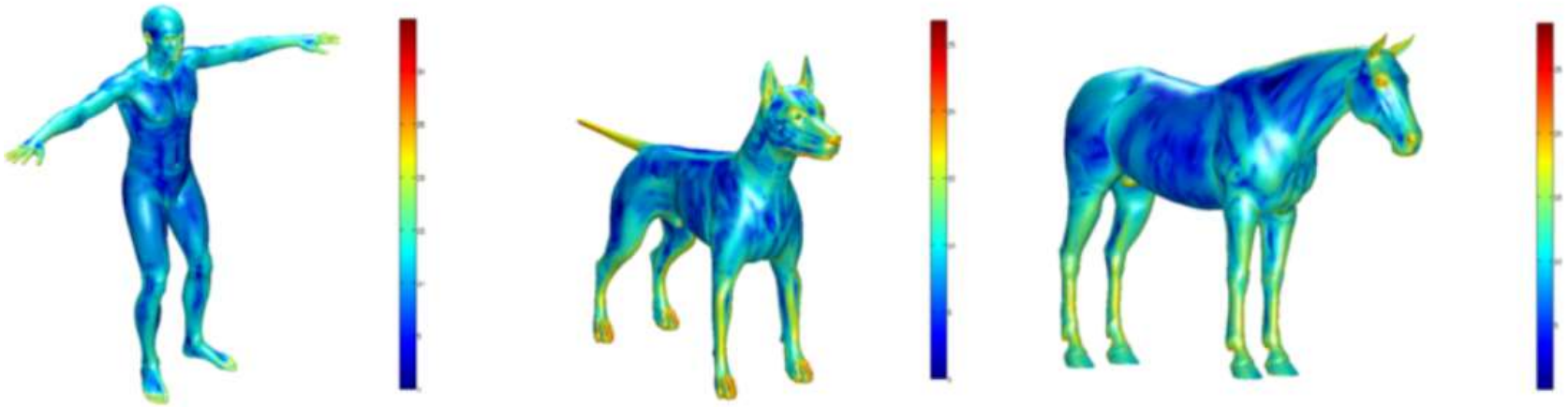
$$A = \frac{p_4^2}{\sigma^2} + p_1^2 + p_2^2$$

$$B = \frac{p_5^2}{\sigma^2} + p_2^2 + p_3^2$$

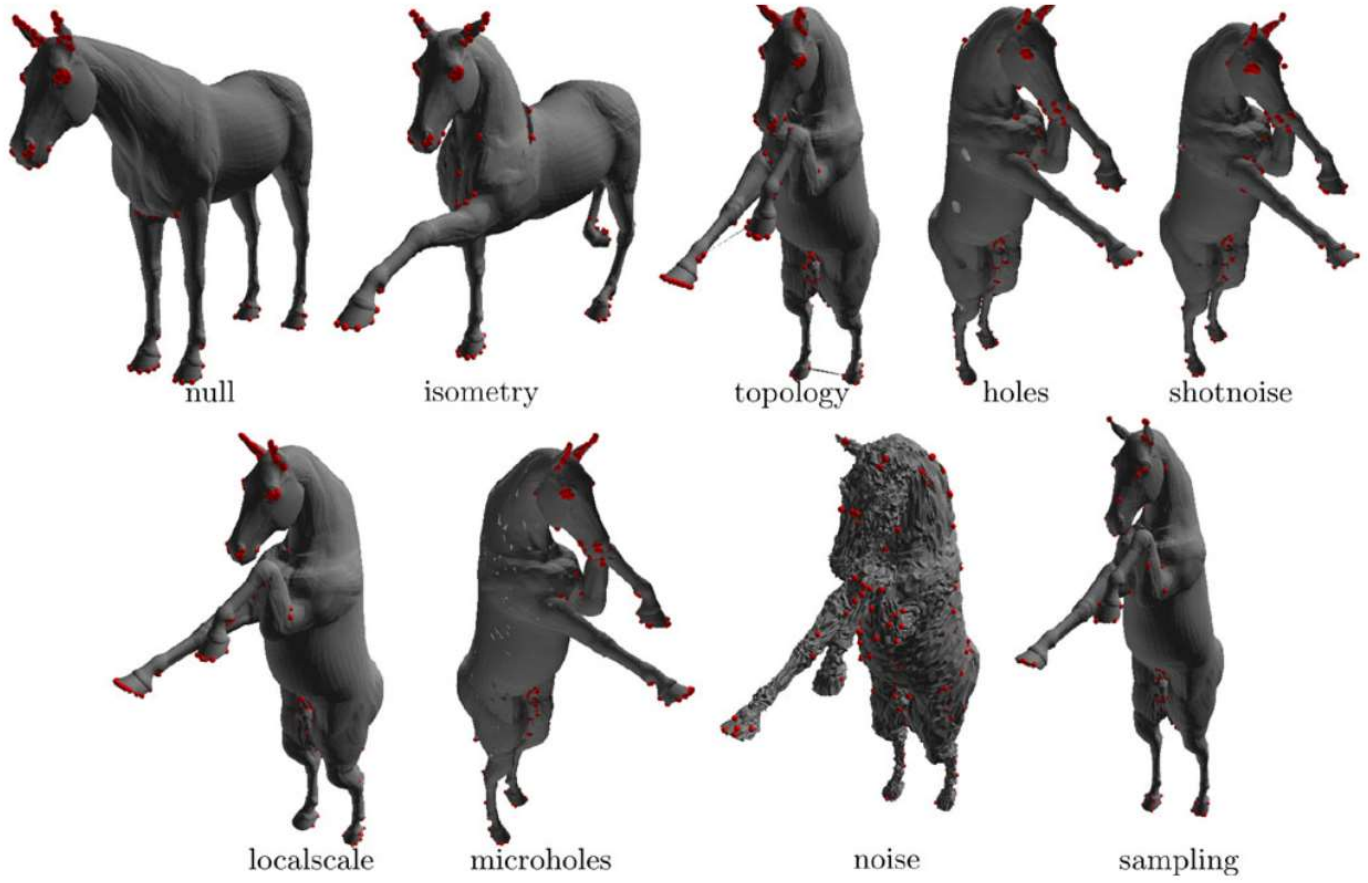
$$C = \frac{p_4 p_5}{\sigma^2} + p_1 p_2 + p_2 p_3$$

- Evaluamos el operador Harris y seleccionamos los puntos con la respuesta más alta.

Algoritmo Harris 3D



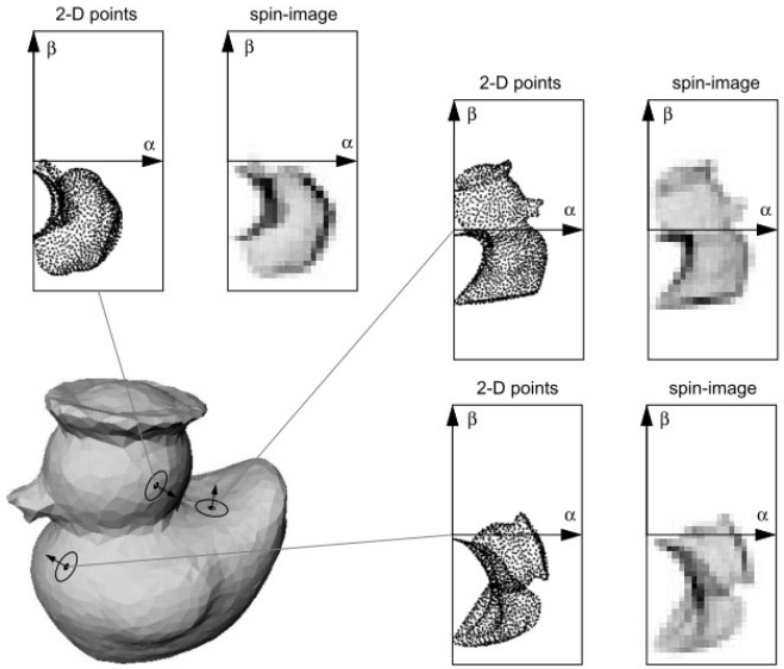
Algoritmo Harris 3D



Point Descriptor

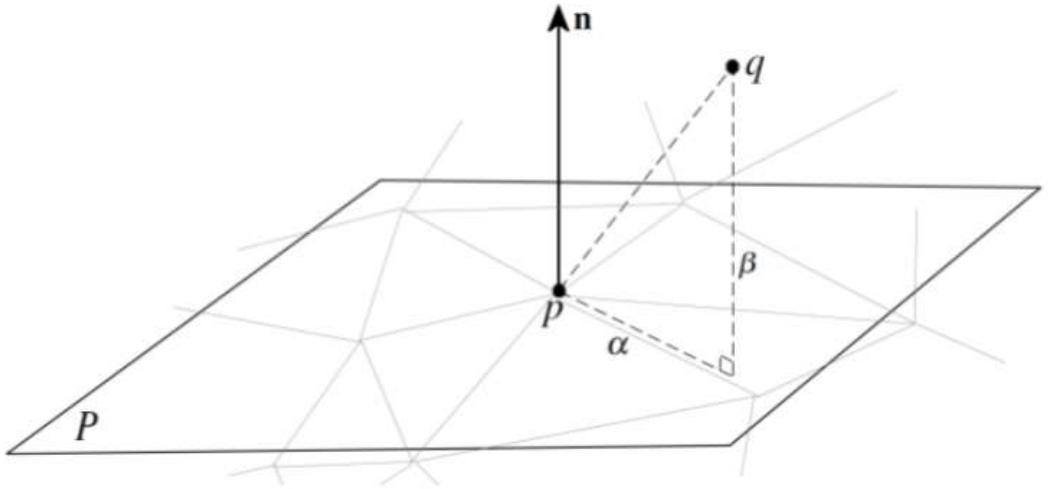
Spin Images

- Se basa en la distribución de puntos sobre la superficie de un objeto



Spin Images

- Se construye una base local
  - Punto p
  - Normal $\mathbf{n}$
  - Plano tangente P perpendicular a $\mathbf{n}$



Spin Images

- Cualquier punto q puede ser representado en esta base

$$S_O : \mathbb{R}^3 \to \mathbb{R}^2$$

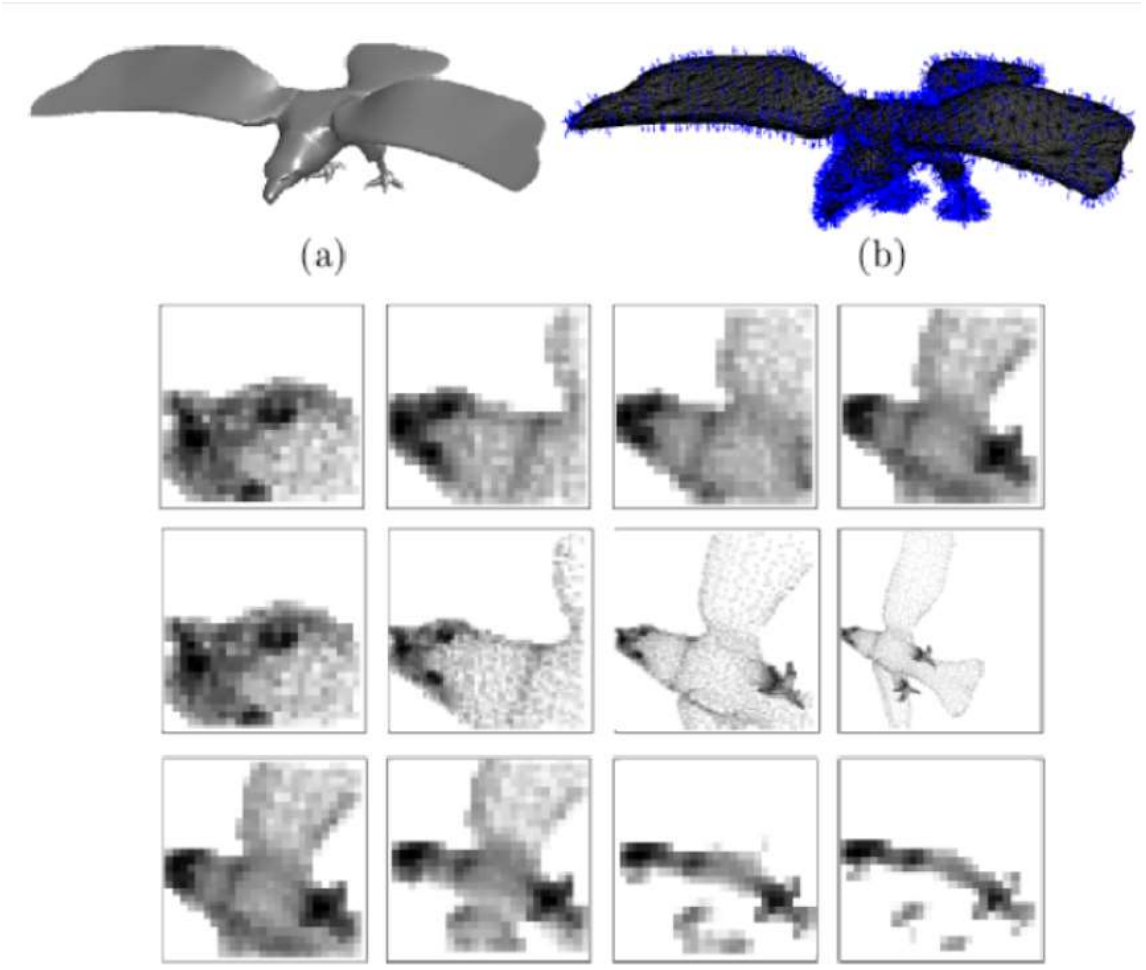
$$S_O(q) \to (\alpha, \beta) = (\sqrt{\|q-p\|^2 - (\vec{n} \cdot (q-p))^2}, \vec{n} \cdot (q-p))$$

- El punto q cae en la coordenada (α, β) de la imagen formada por la base local

$$i = \left\lfloor \frac{\frac{W * bin}{2} - \beta}{bin} \right\rfloor$$

$$j = \left\lfloor \frac{\alpha}{bin} \right\rfloor$$

Spin Images



Point Matching

Spin Images - Matching

- Dadas dos imágenes spin con N bins, computamos la correlación cruzada

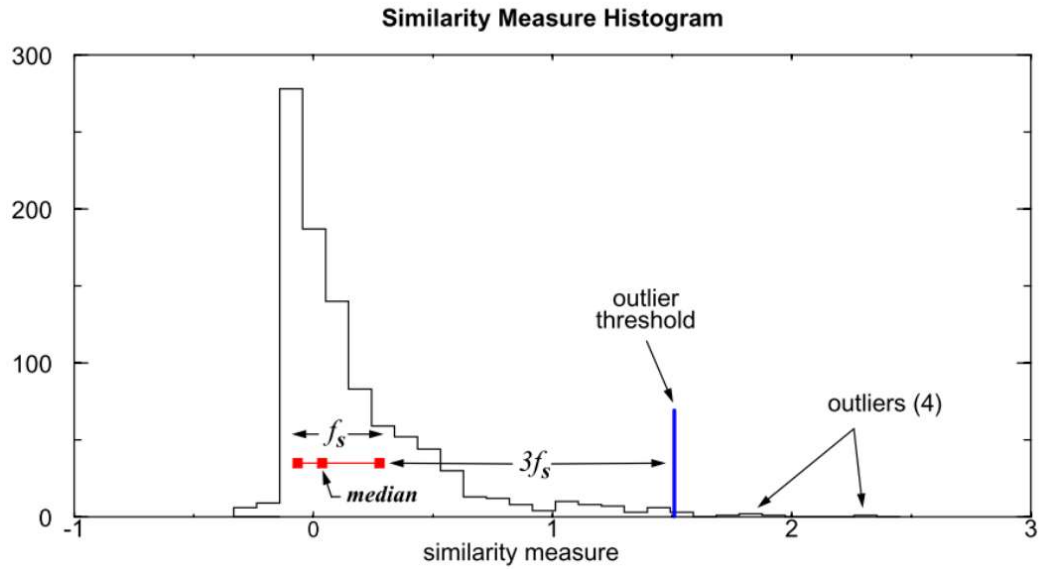
$$R(P,Q) = \frac{N\sum p_i q_i - \sum p_i \sum q_i}{\sqrt{(N\sum p_i^2 - (\sum p_i)^2)(N\sum q_i^2 - (\sum q_i)^2)}}$$

- La similitud toma en cuenta la varianza para evitar la dependencia a la densidad

$$C(P,Q) = (atanh(R(P,Q)))^2 - \lambda\left(\frac{1}{N-3}\right)$$

Spin Images - Matching

- Computar similitud entre spin images de puntos sampleados
- Solo es necesario un conjunto de correspondencias con los valores más altos de correlación



Spin Images - Matching

- Filtrado de correspondencias
  - Correspondencias con similitud menor que la mitad de la máxima similitud
  - Dadas dos correspondencias $C_1 = (s_1, m_1)$ y $C_2 = (s_2, m_2)$, la consistencia geométrica se define como

$$d_{gc}(C_1, C_2) = 2\frac{\|S_{m_2}(m_1) - S_{s_2}(s_1)\|}{\|S_{m_2}(m_1) + S_{s_2}(s_1)\|}$$

$$D_{gc}(C_1, C_2) = max(d_{gc}(C_1, C_2), d_{gc}(C_2, C_1))$$

- Consistencia involucra posición y normales
- D_{gc} es pequeño cuando hay consistencia
- Descartar correspondencias que no son consistentes con al menos un cuarto de la lista de correspondencias.

Finding Transformation

Computar transformación

- Se define una medida de grupo de correspondencias

$$w_{gc}(C_1, C_2) = \frac{d_{gc}(C_1, C_2)}{1 - \exp(-(\|S_{m_2}(m_1)\| + \|S_{s_2}(s_1)\|)/2)}$$

$$W_{gc}(C_1, C_2) = max(w_{gc}(C_1, C_2), w_{gc}(C_2, C_1))$$

- Y una medida entre una correspondencia y un grupo

$$W_{gc}(C, \{C_1, C_2, \ldots, C_n\}) = max_i(W_{gc}(C, C_i))$$

Computar transformación

- Para cada correspondencia, se inicializa un grupo $G_i = \{C_i\}$
- Encontrar una correspondencia que minimice $W_{gc}(C_j, G_i)$. Si esta métrica está por debajo de un umbral, agrandar el grupo $G_i = G_i \cup \{C_j\}$
- Continuar hasta que no hayan más correspondencias para añadir

Computar transformación

- Para cada grupo de correspondencias, computar una transformación rígida T que minimice el error

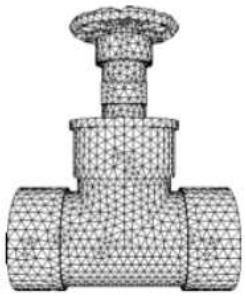
$$E_T = \min_T \sum \|s_i - T(m_i)\|^2$$

- T consiste de una matriz R y un vector de traslación t

Computar transformación

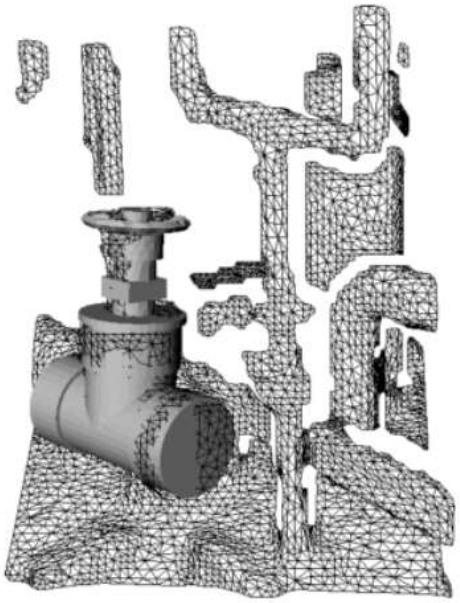
- Refinar la solución con ICP

Model



Scene Intensity Image

Recognition Result

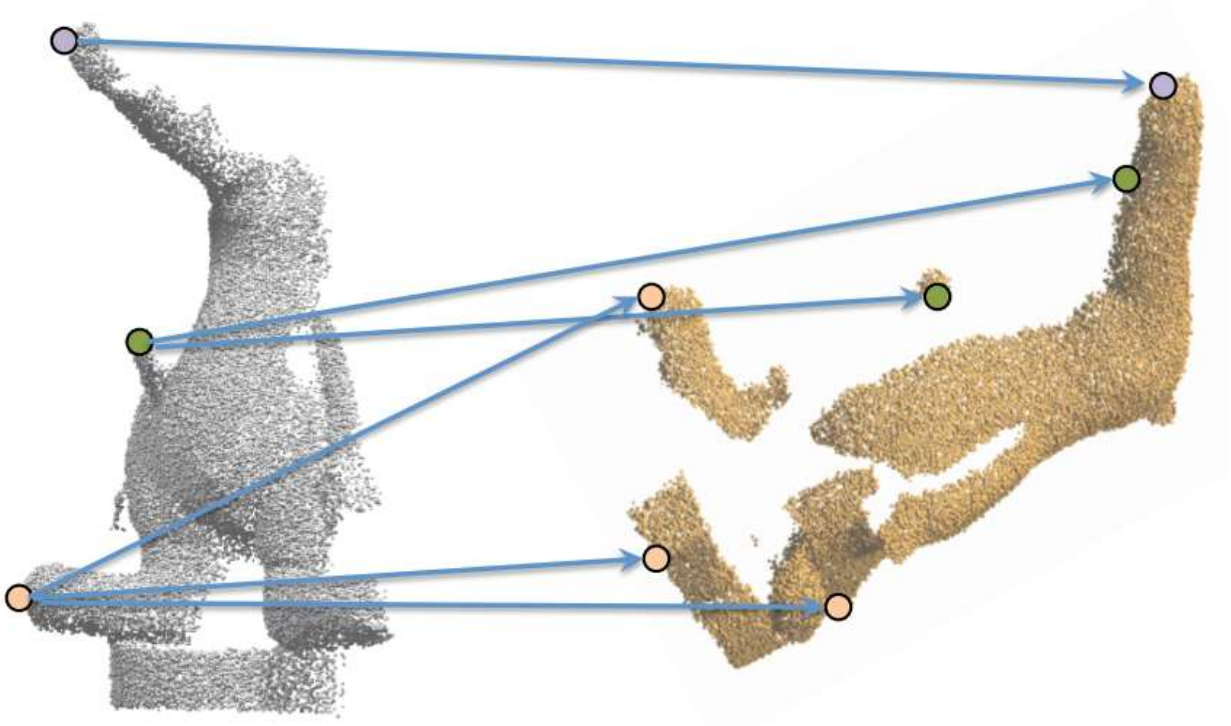


Métodos Sin Características

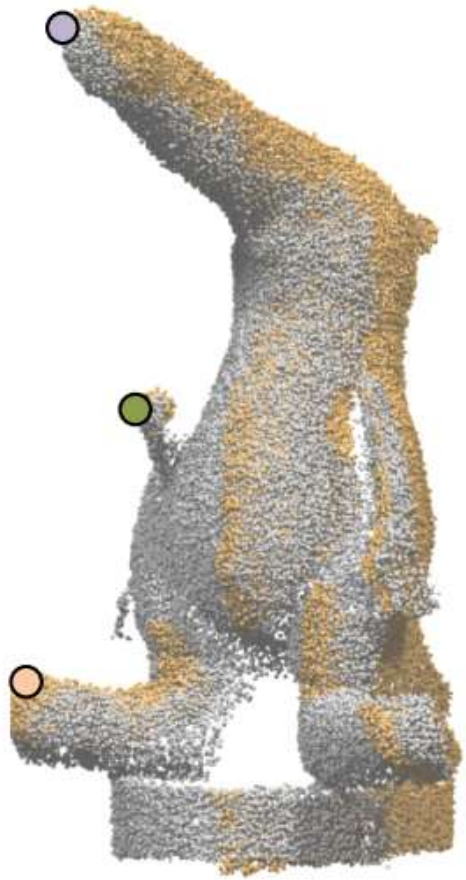
Featureless registration

Alineamiento con puntos de interés

Alineamiento con puntos de interés



Alineamiento con puntos de interés



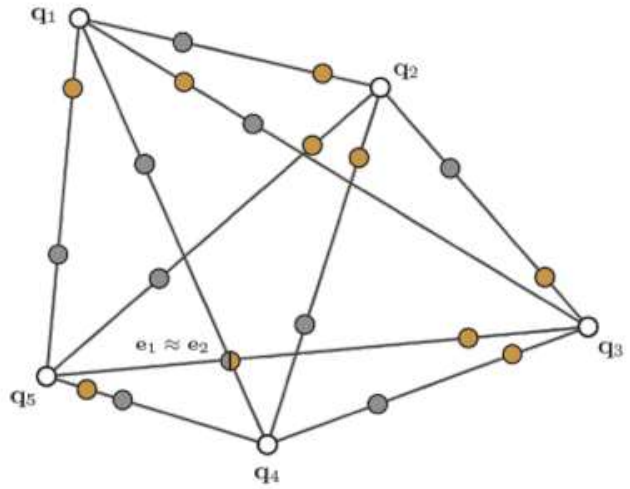
Ruido, Outliers y Agujeros

- Puntos característicos son generalmente entidades diferenciales
  - Ruido y outliers generan malas correspondencias
  - Una solución son descriptores basados en integrales (no diferenciales)
  - Invariantes integrales
    - Robustos a datos faltantes

- En general, ruido + outliers + agujeros → puntos característicos fallan

Observación importante

- Un par de tripletas es suficiente para definir una única transformación rígida → $O(n^3)$
- Un conjunto especial de conjuntos congruentes de 4 puntos, se puede resolver en menor complejidad → $O(n^2)$

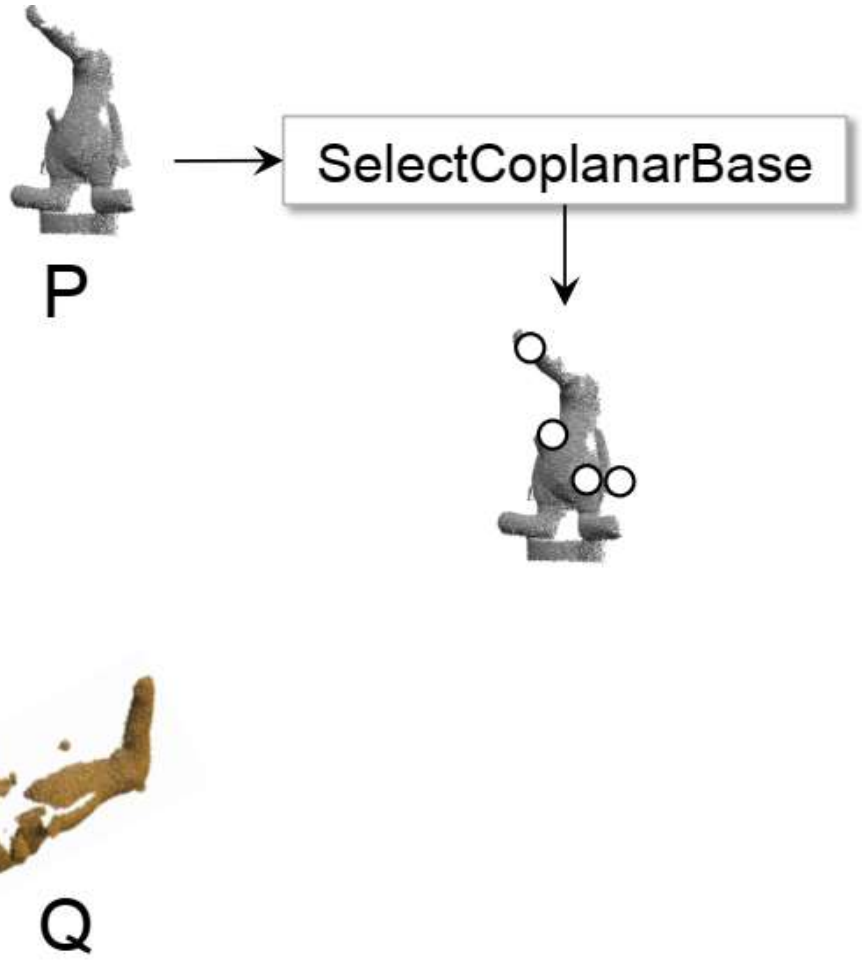


Algoritmo 4PCS

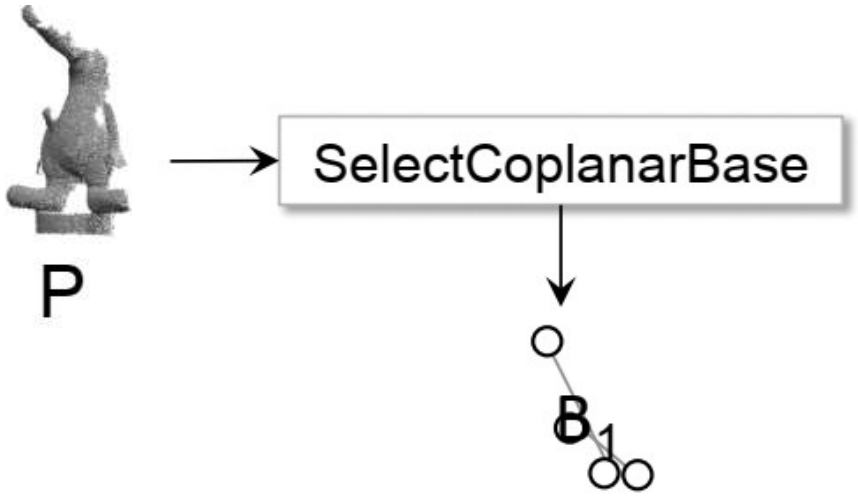
P

Q

Algoritmo 4PCS



Algoritmo 4PCS



Algoritmo 4PCS

P

SelectCoplanarBase

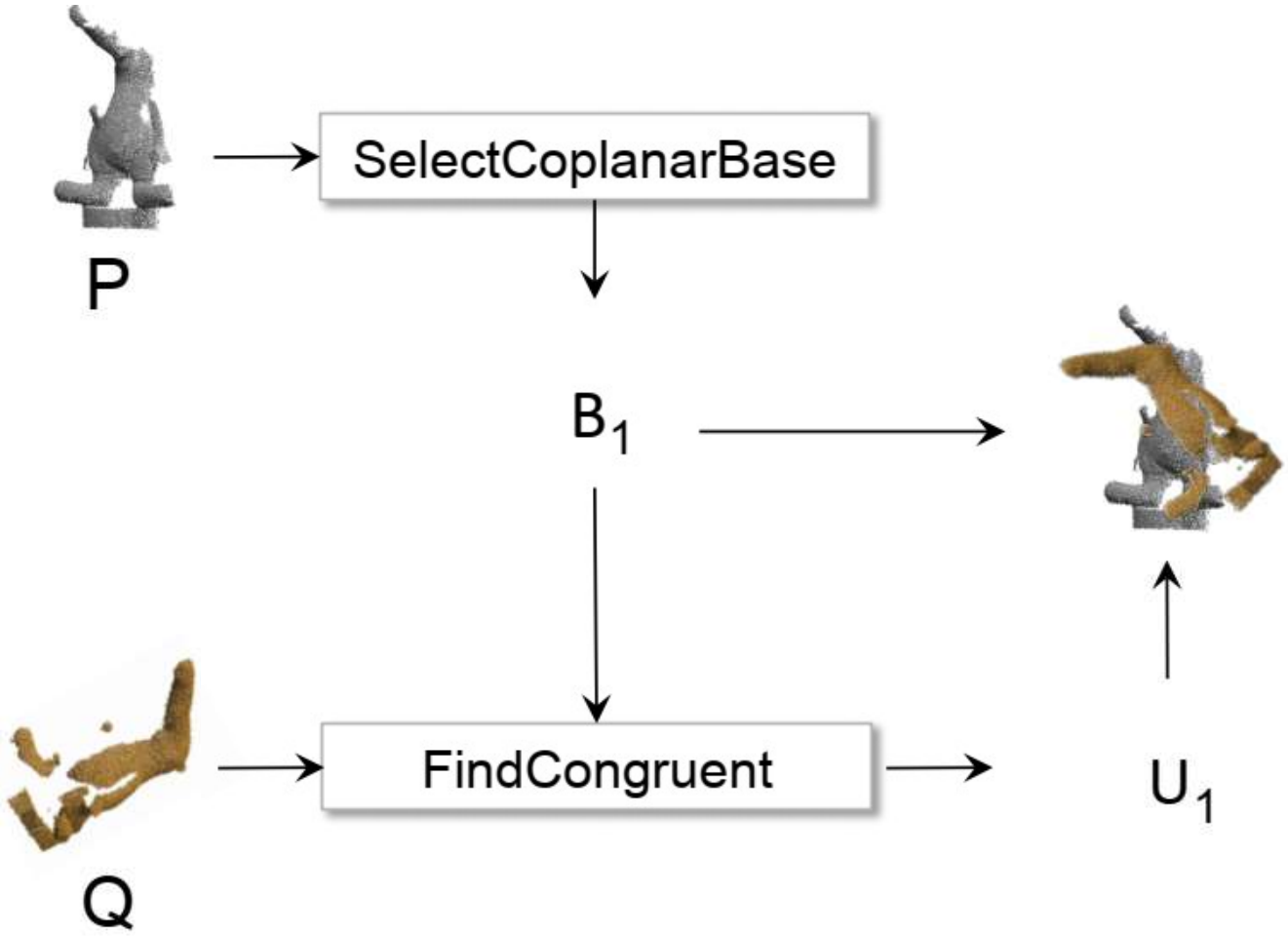
B_1

Q

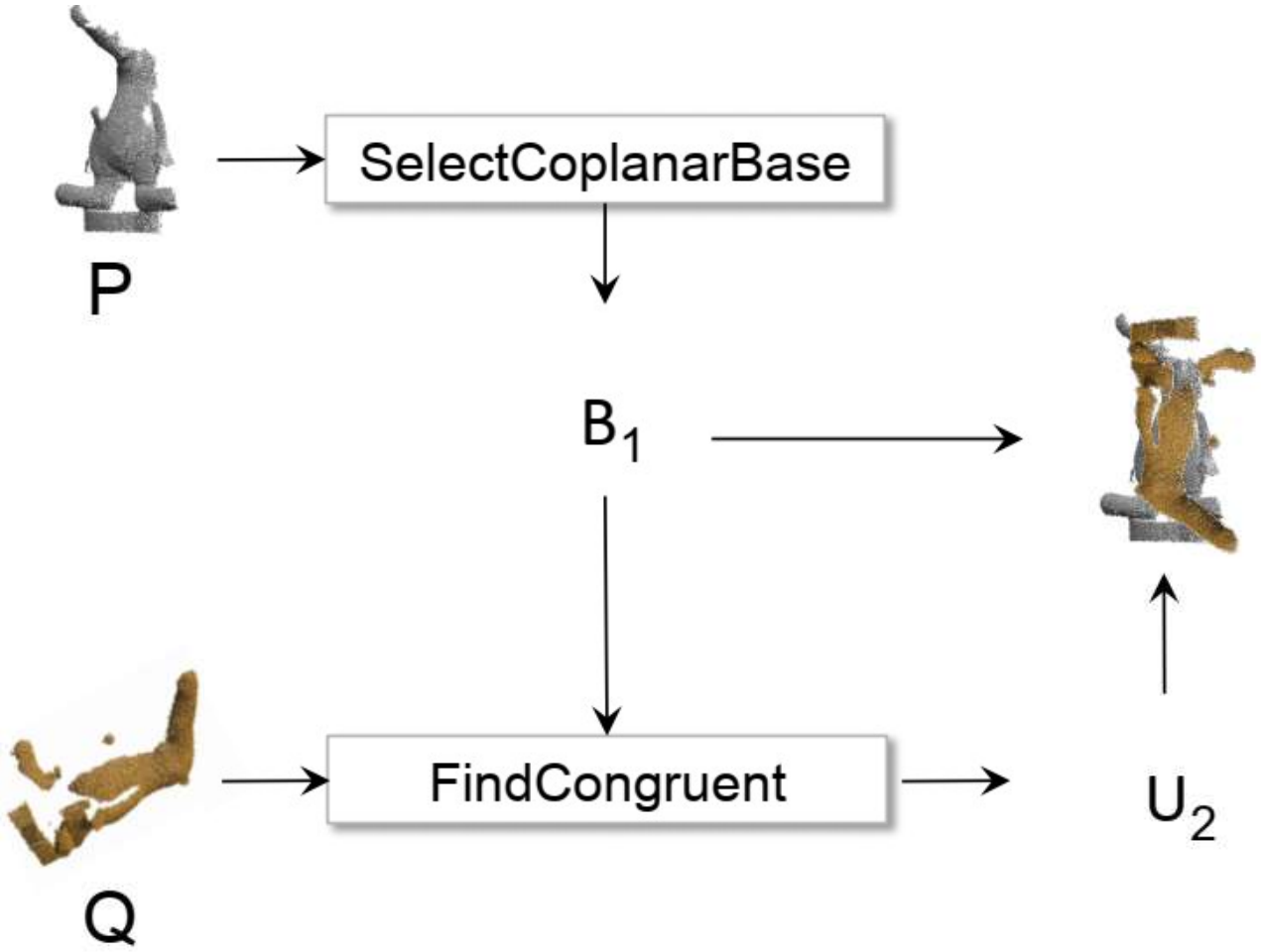
FindCongruent

$U = \{U_1, U_2, \ldots\}$

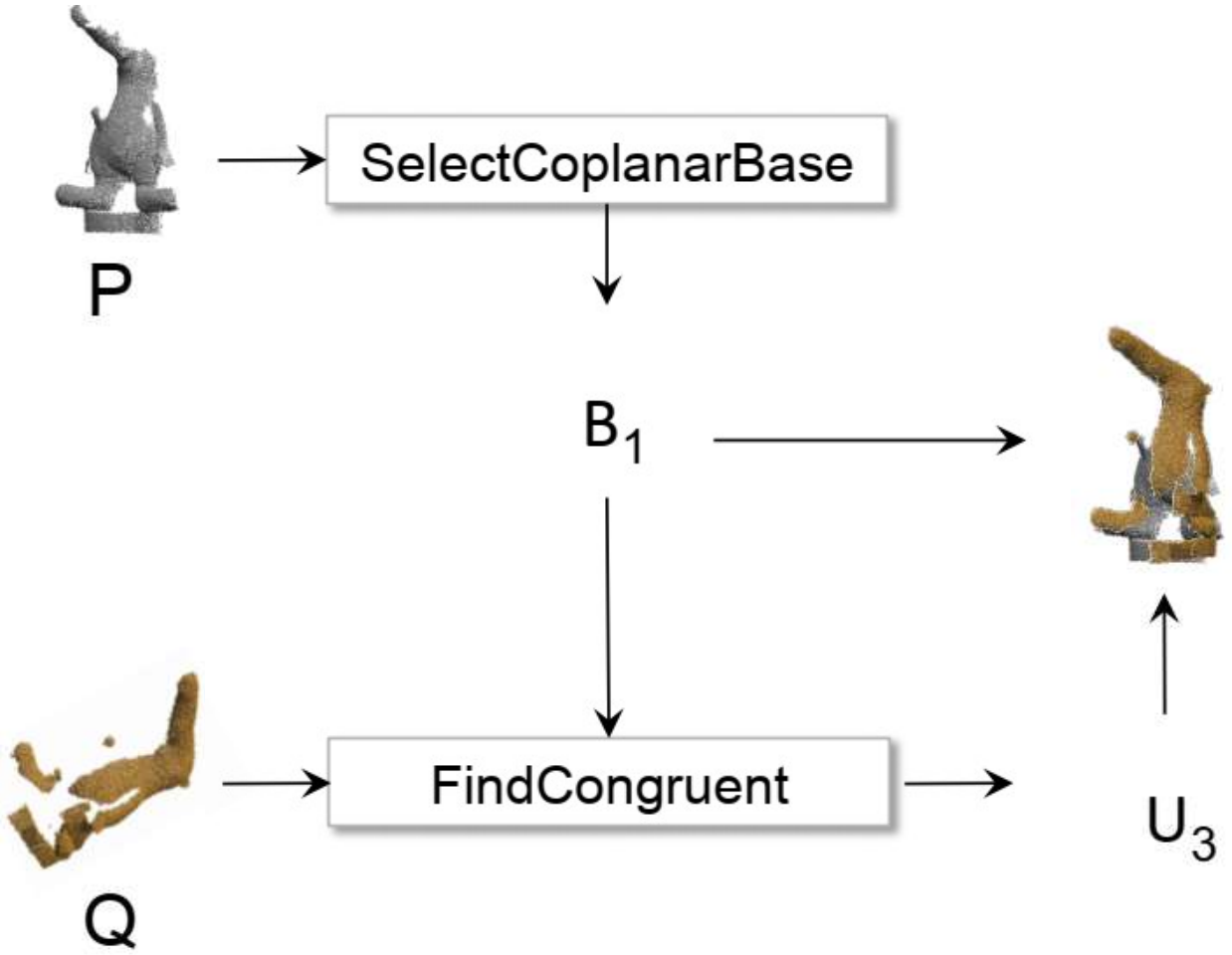
Algoritmo 4PCS



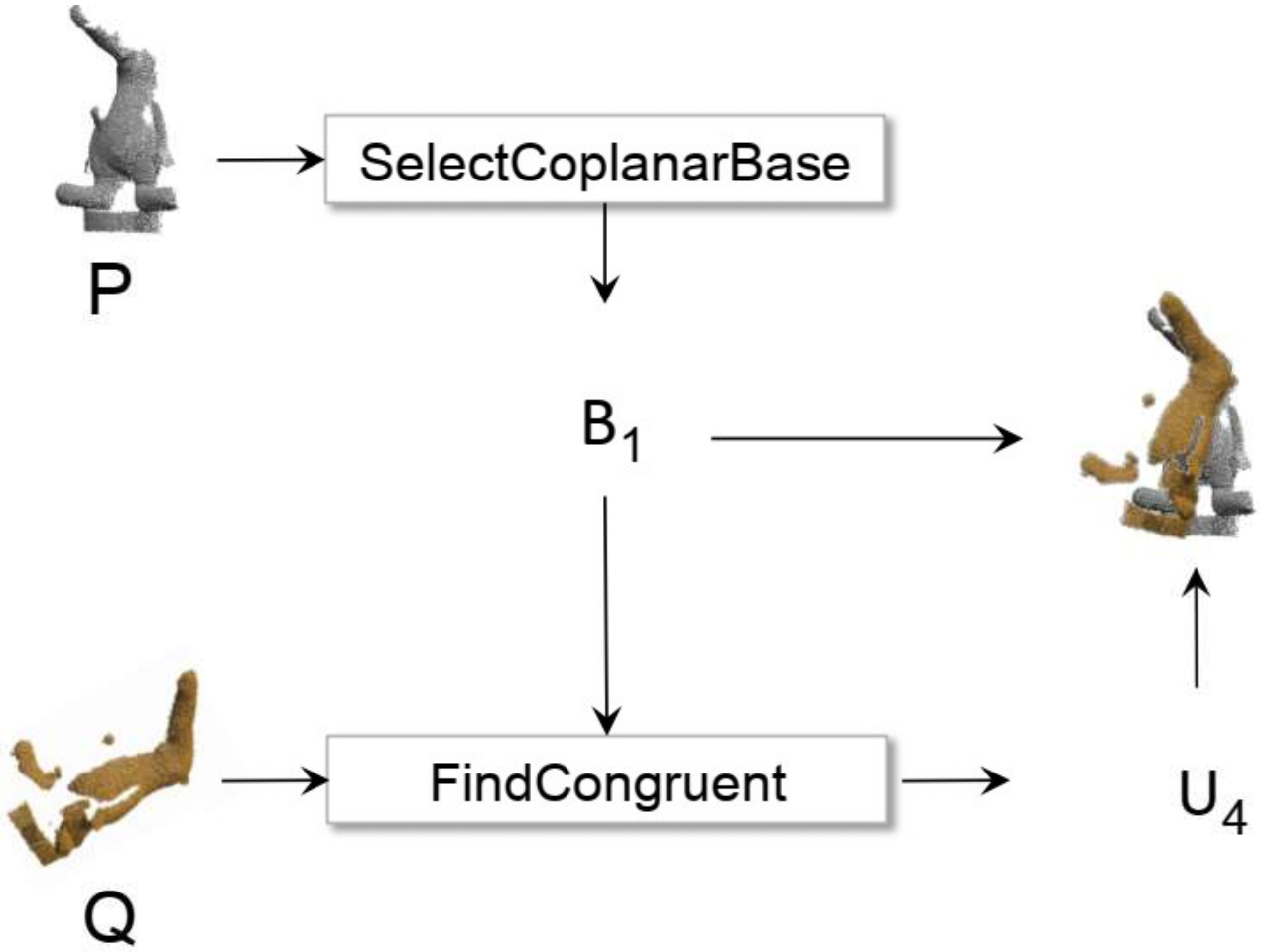
Algoritmo 4PCS



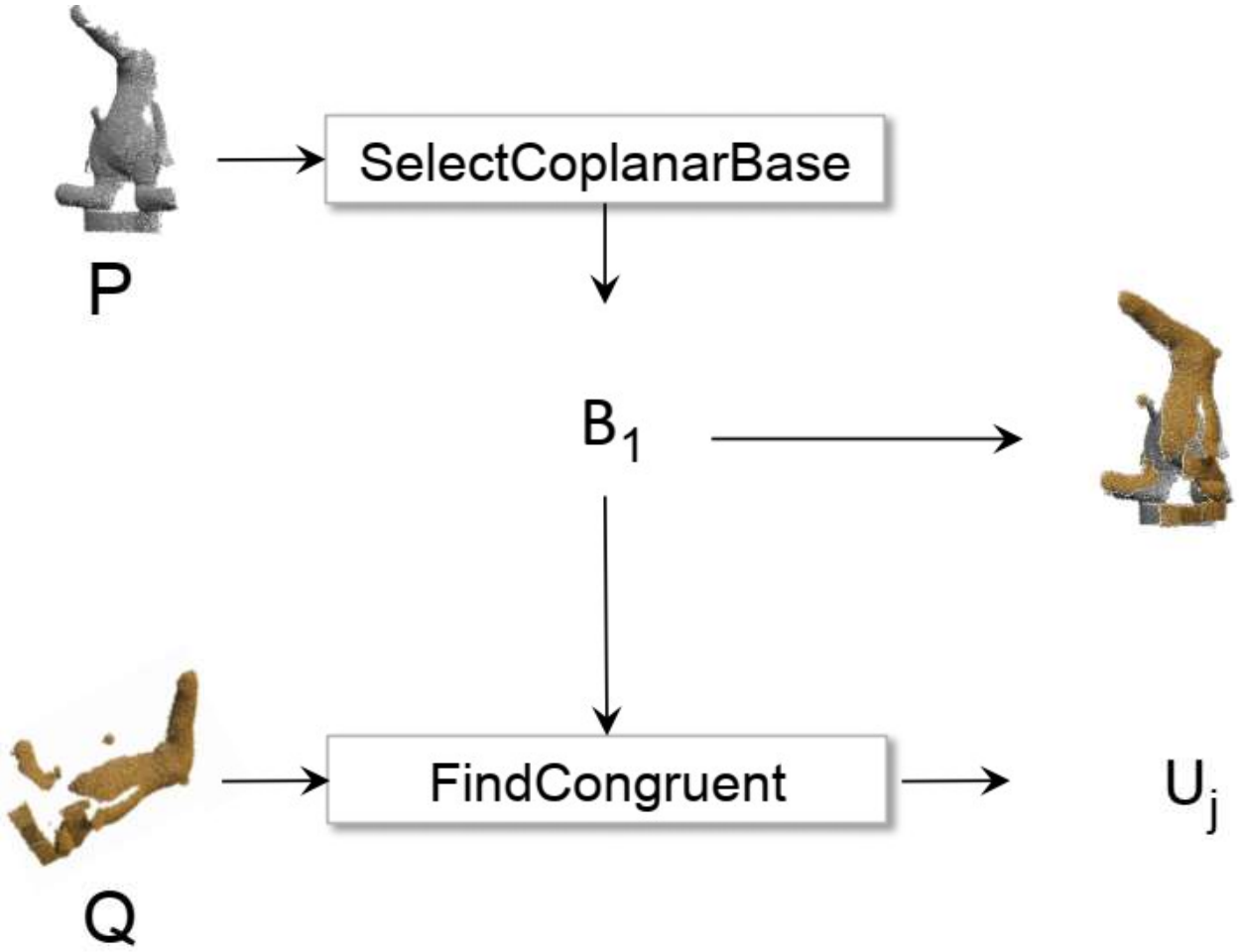
Algoritmo 4PCS



Algoritmo 4PCS

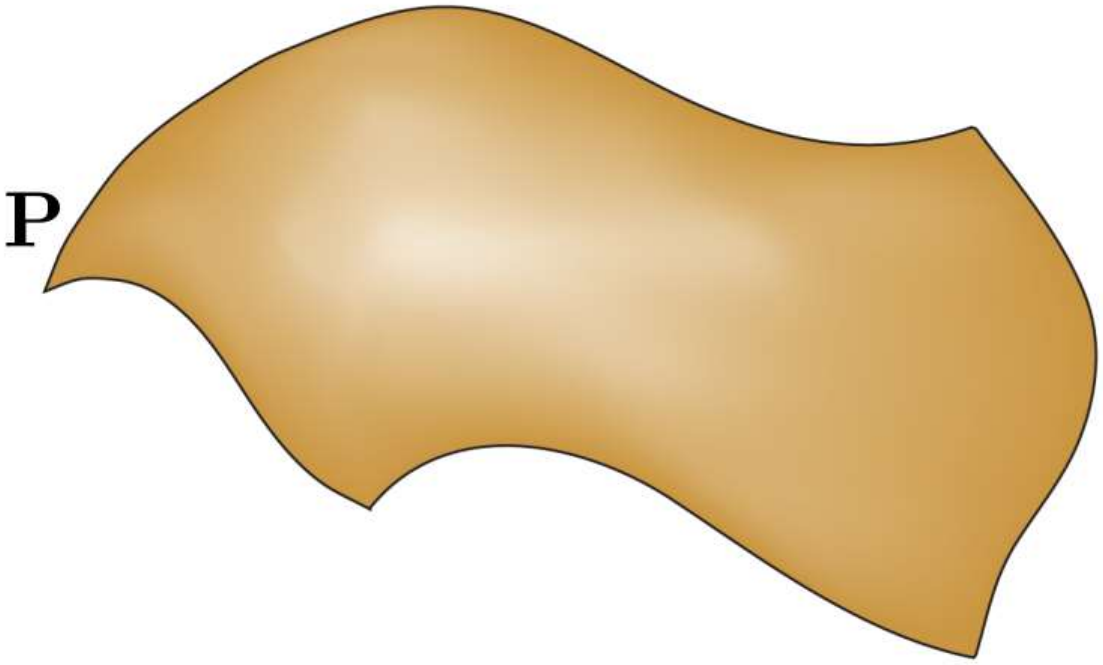


Algoritmo 4PCS

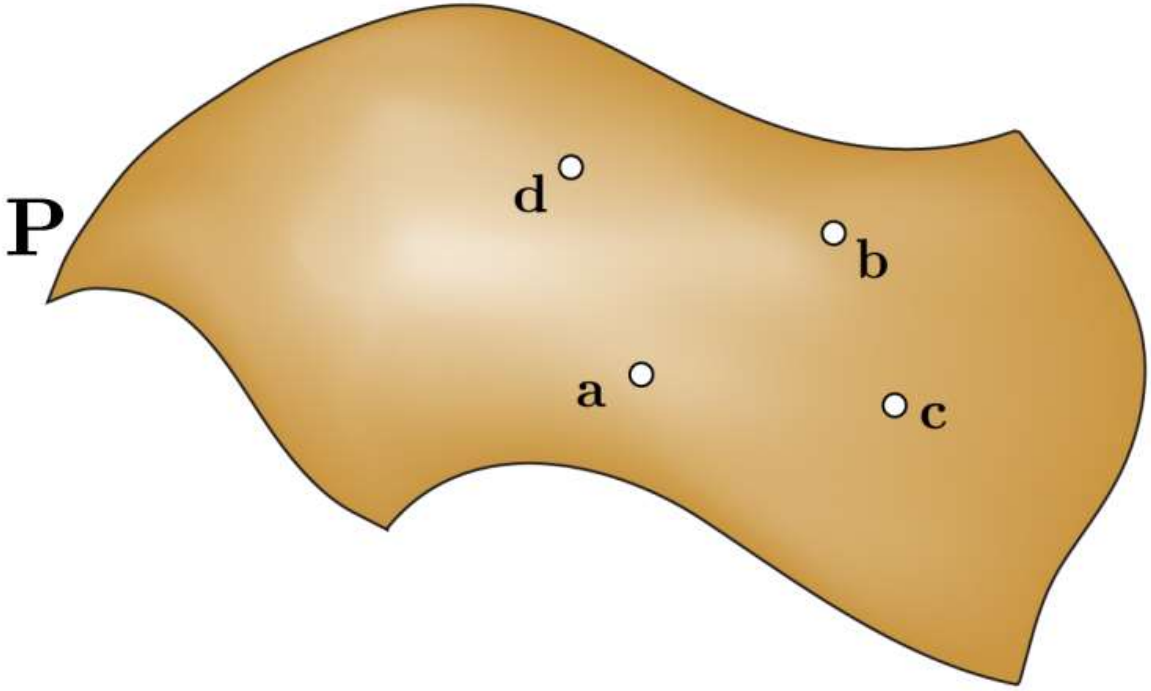


Nos quedamos con la mejor transformación

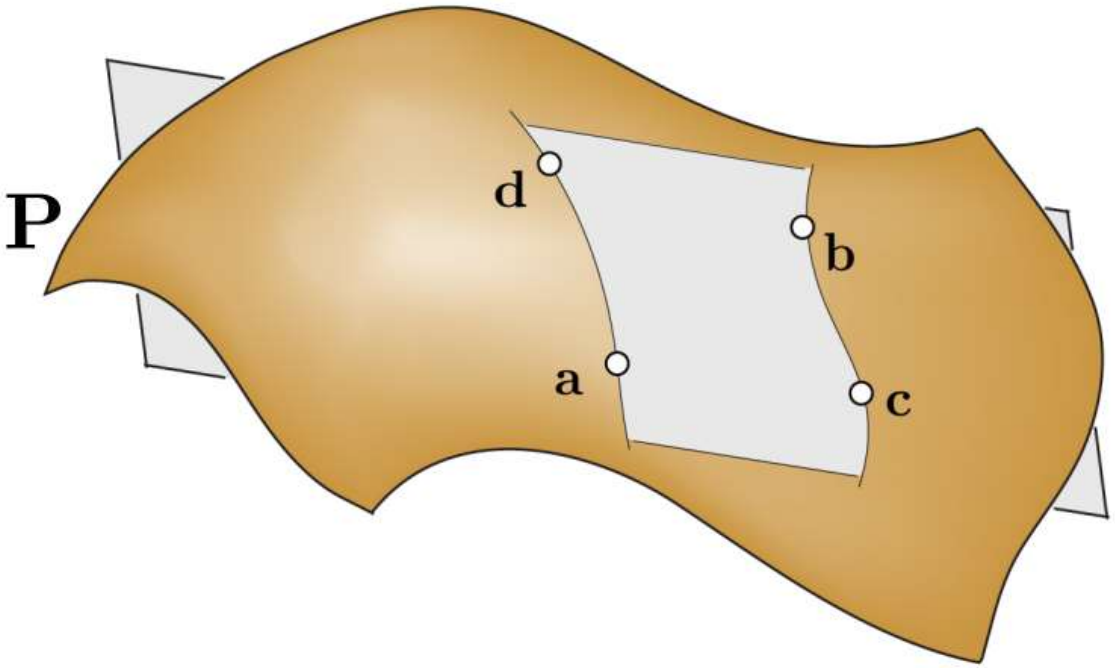
FindCongruent



FindCongruent

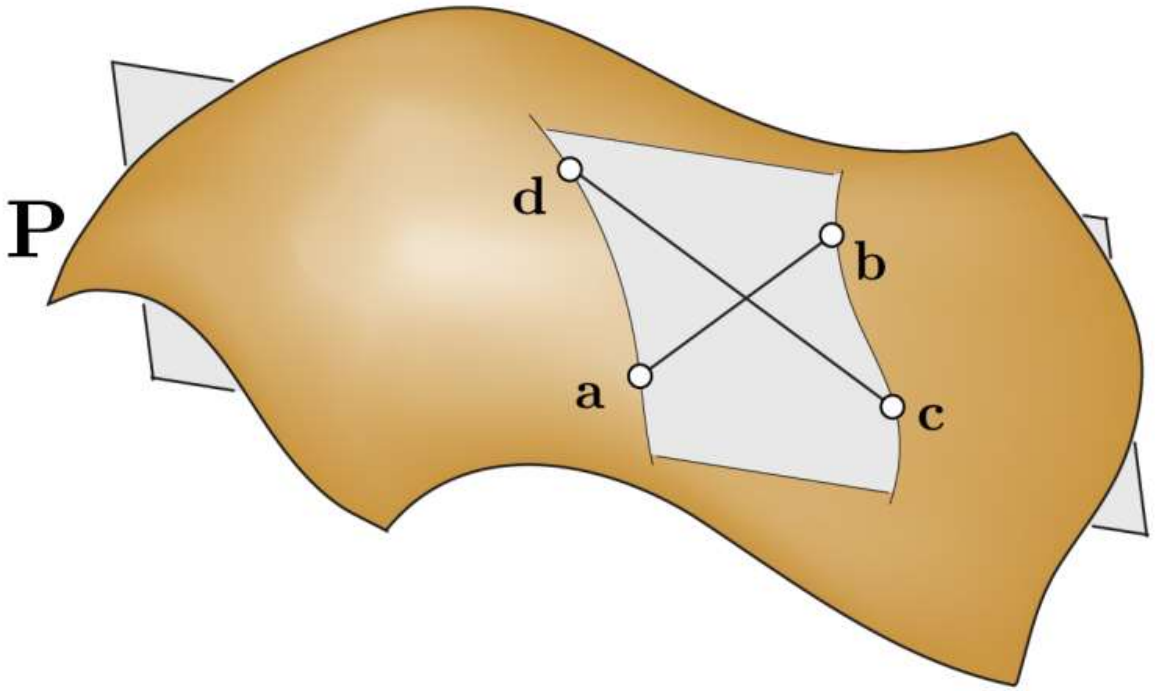


FindCongruent

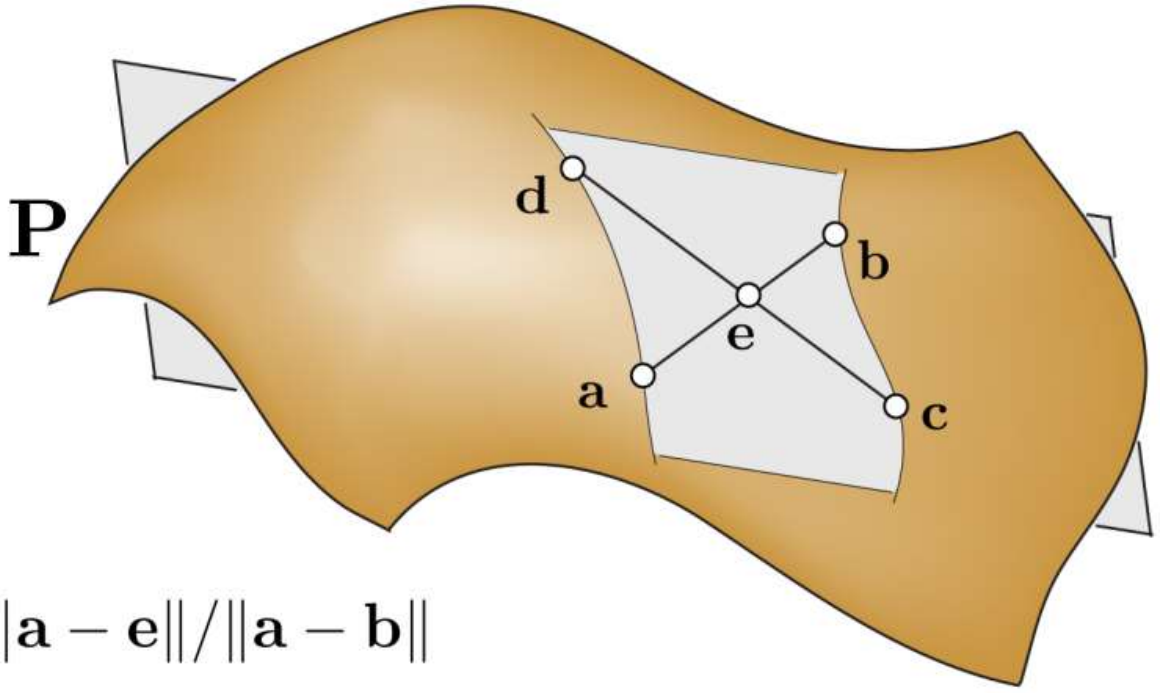


4 coplanar points

FindCongruent



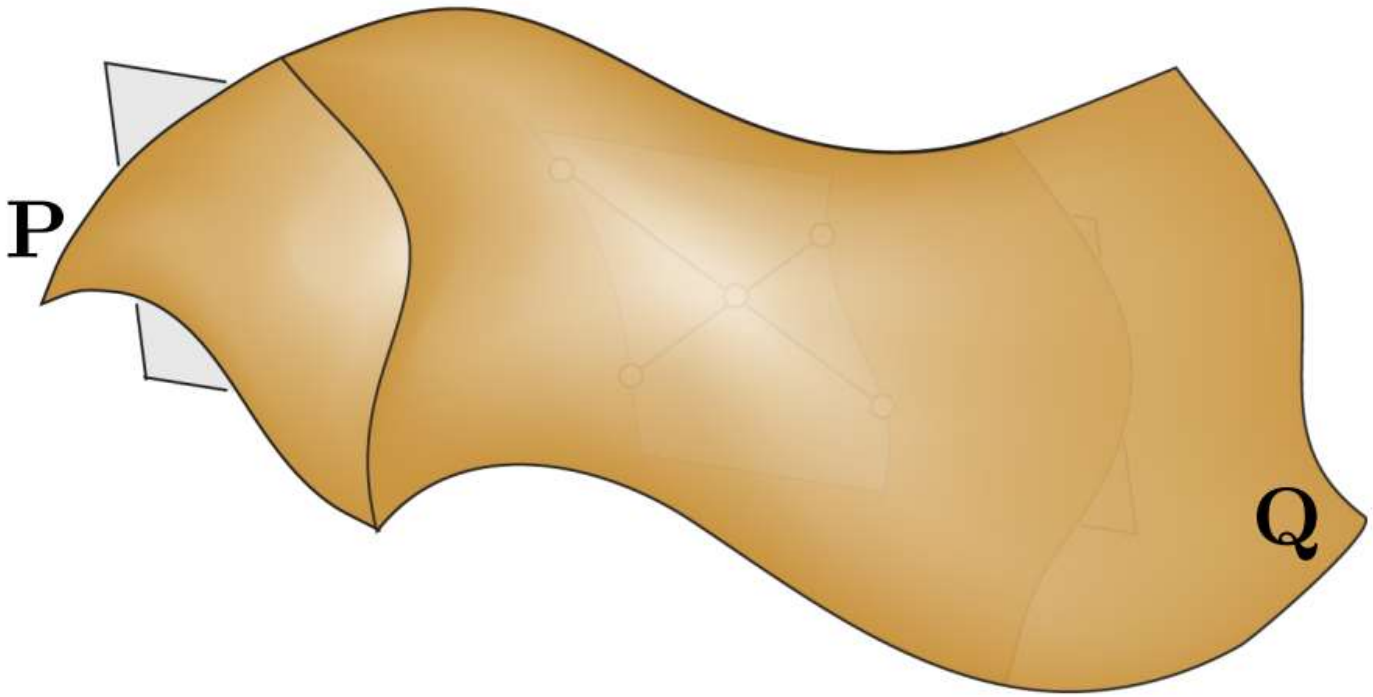
FindCongruent



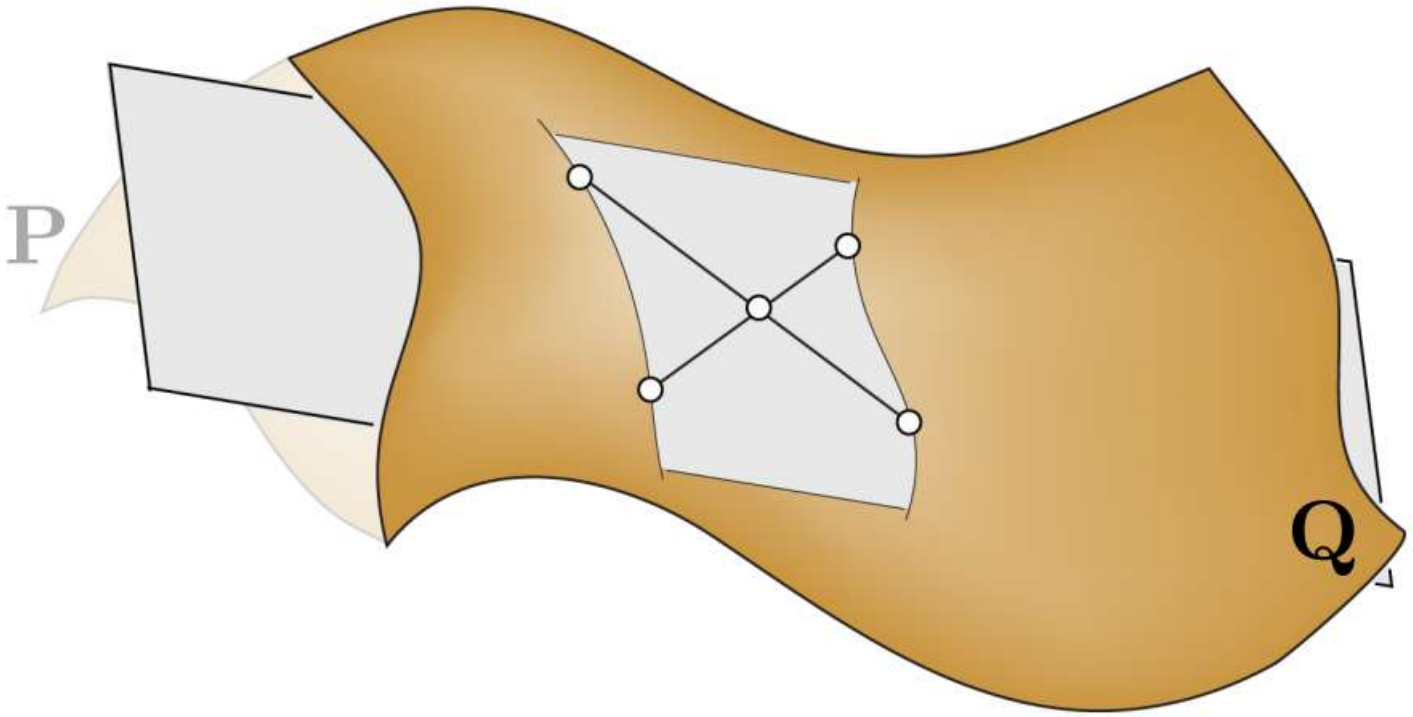
$$r_1 = \|\mathbf{a} - \mathbf{e}\| / \|\mathbf{a} - \mathbf{b}\|$$

$$r_2 = \|\mathbf{c} - \mathbf{e}\| / \|\mathbf{c} - \mathbf{d}\|$$

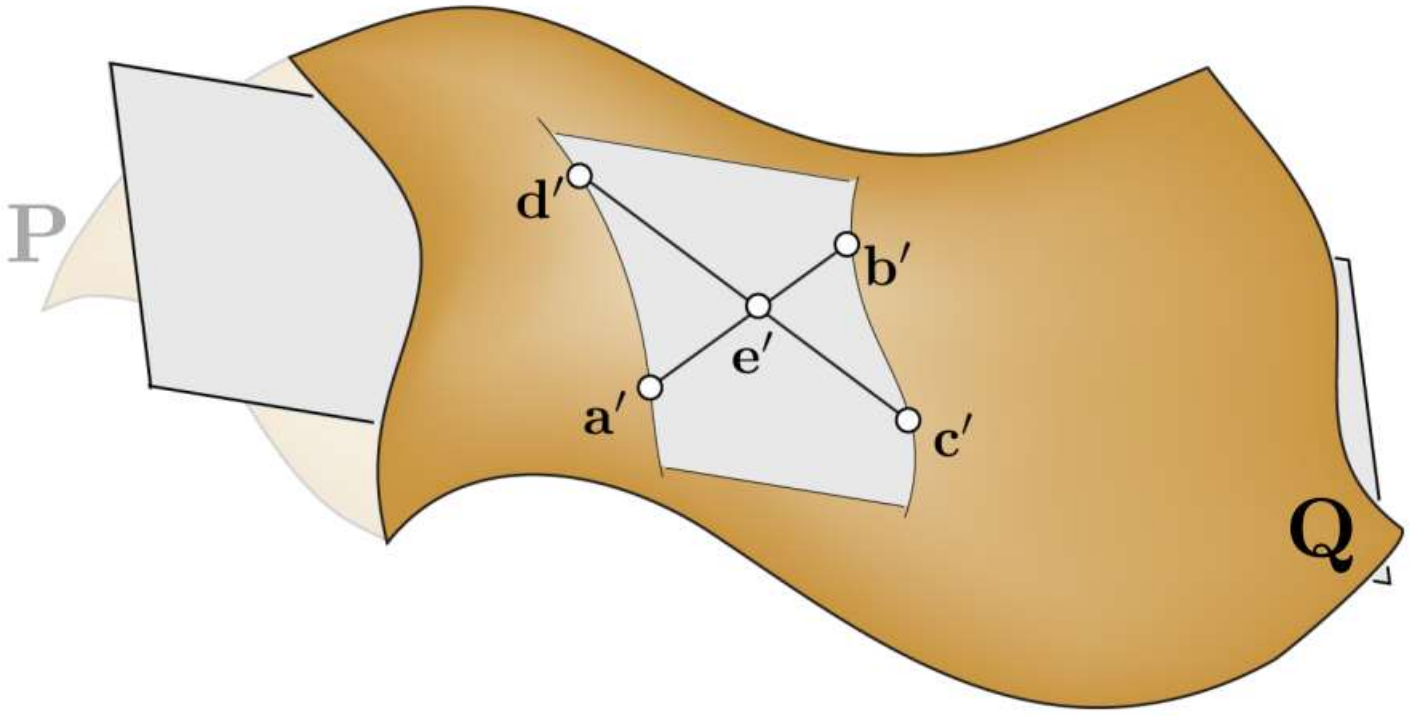
FindCongruent



FindCongruent



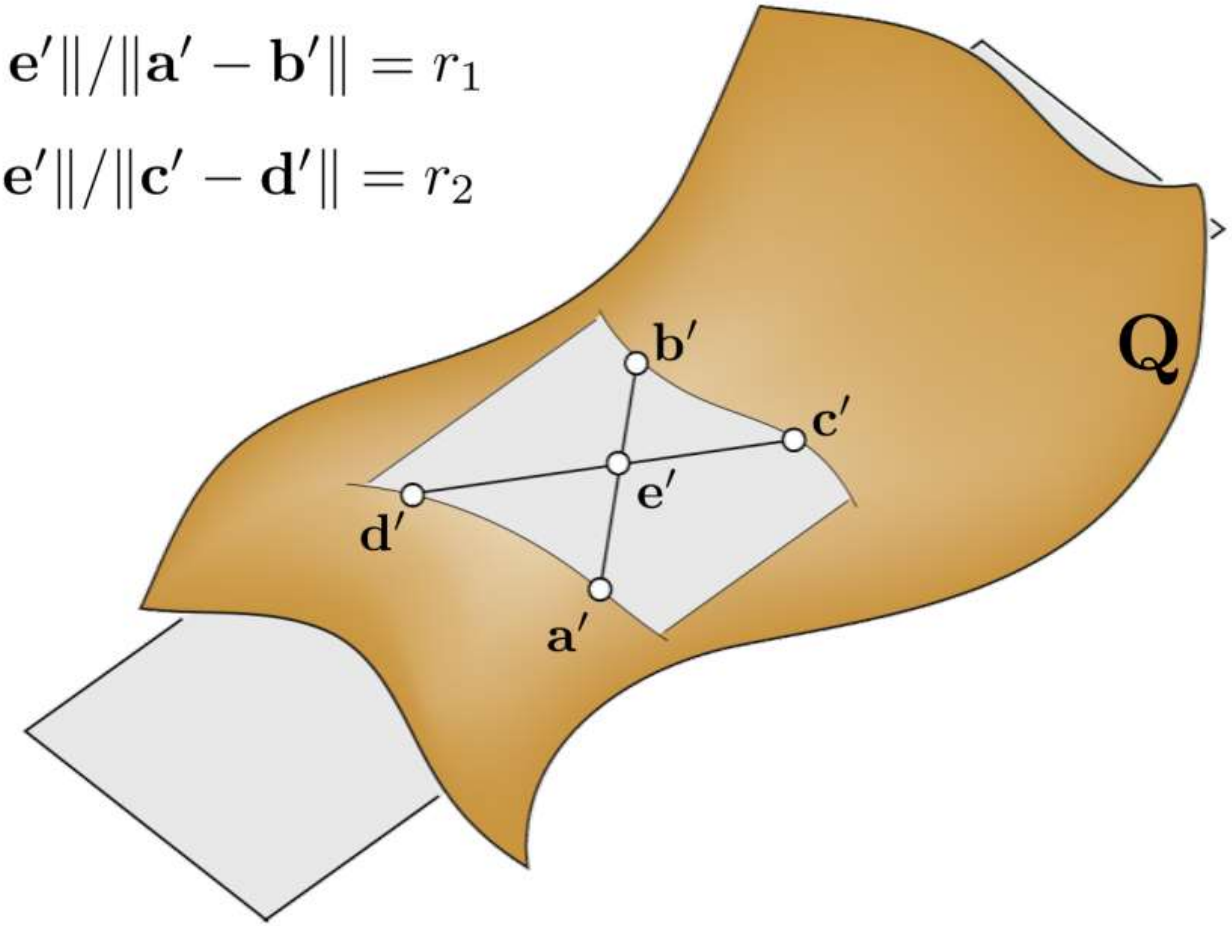
FindCongruent



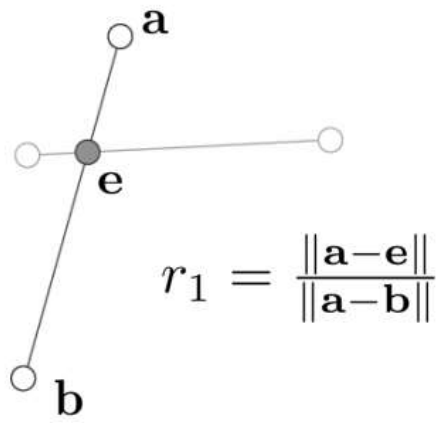
FindCongruent

$$\|\mathbf{a}' - \mathbf{e}'\| / \|\mathbf{a}' - \mathbf{b}'\| = r_1$$

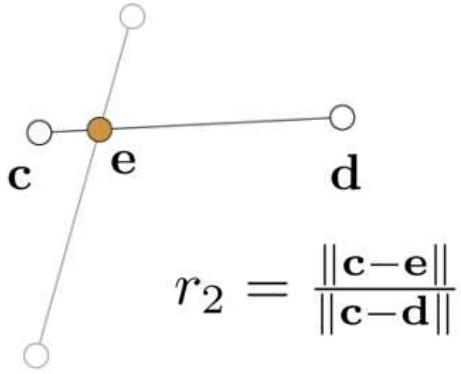
$$\|\mathbf{c}' - \mathbf{e}'\| / \|\mathbf{c}' - \mathbf{d}'\| = r_2$$



FindCongruent

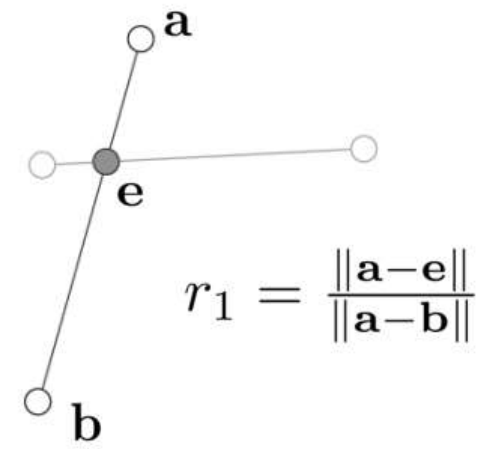


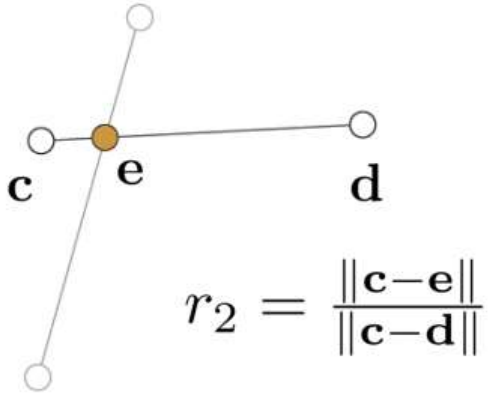
$$r_1 = \frac{\|\mathbf{a}-\mathbf{e}\|}{\|\mathbf{a}-\mathbf{b}\|}$$

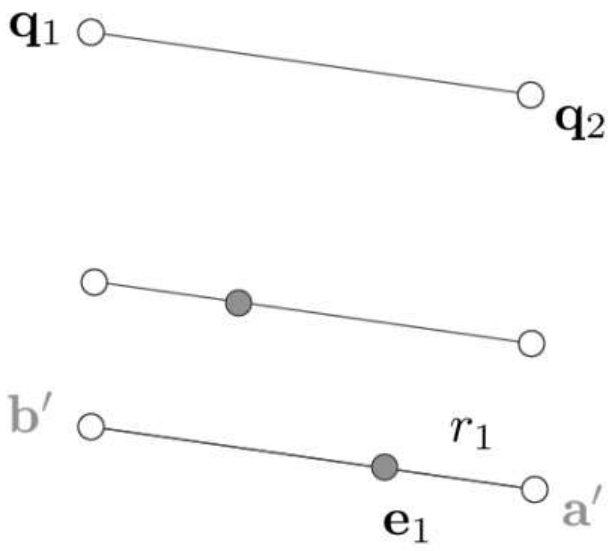


$$r_2 = \frac{\|\mathbf{c}-\mathbf{e}\|}{\|\mathbf{c}-\mathbf{d}\|}$$

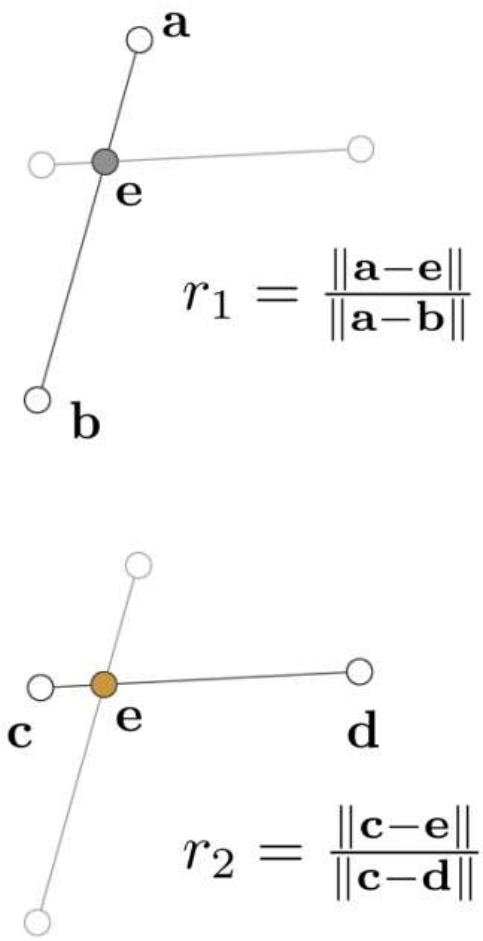
FindCongruent

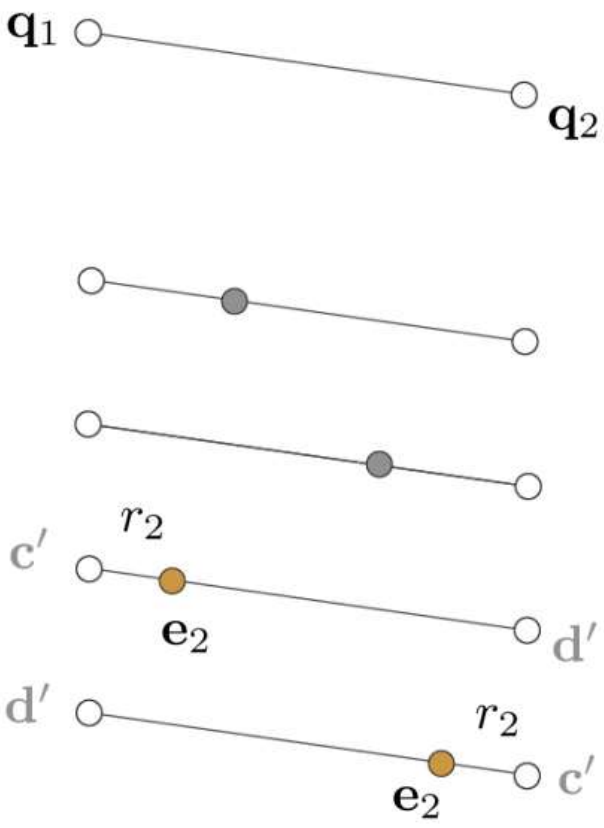




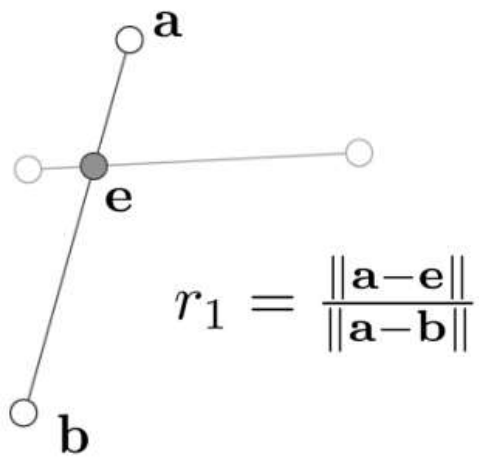


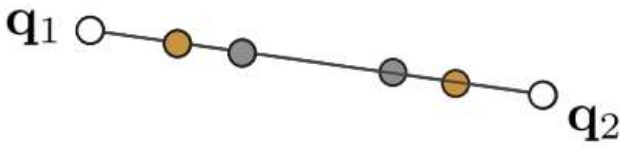
FindCongruent

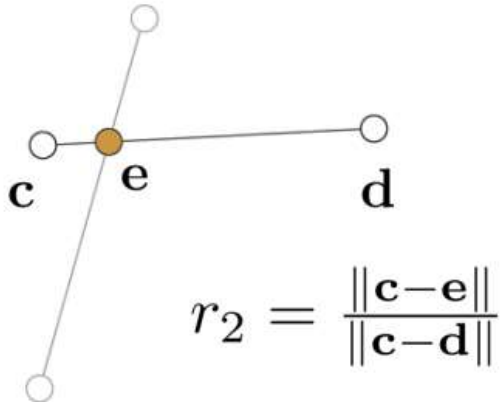




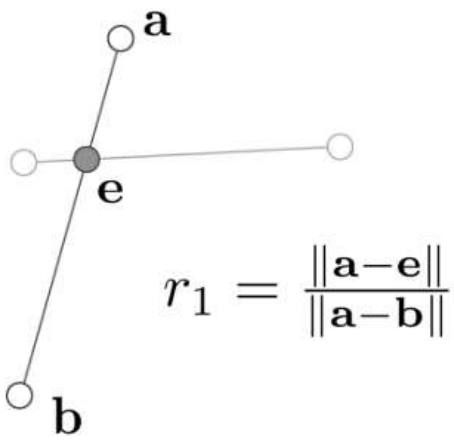
FindCongruent



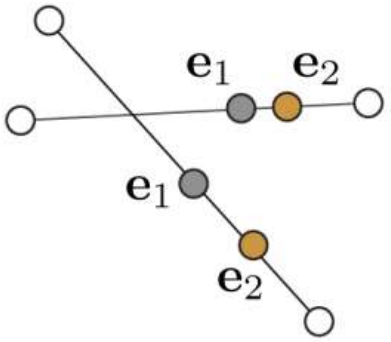




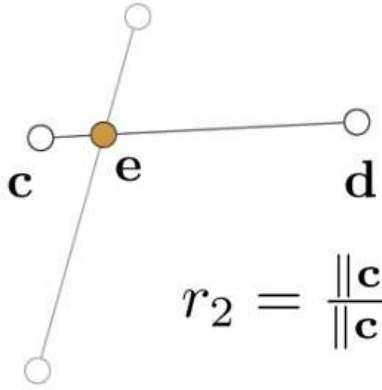
FindCongruent (cuando $e_1 \neq e_2$)



$$r_1 = \frac{\|\mathbf{a}-\mathbf{e}\|}{\|\mathbf{a}-\mathbf{b}\|}$$

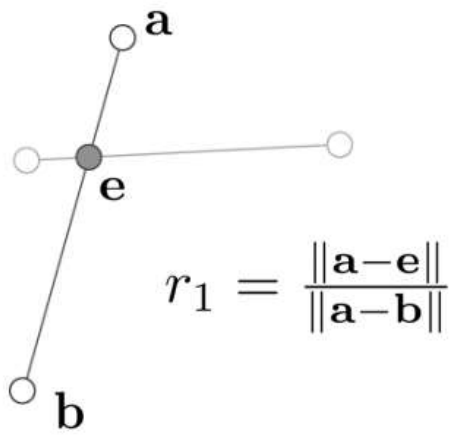


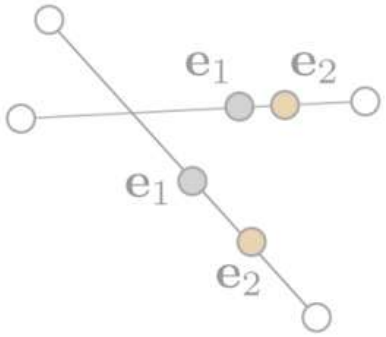
typical scenario



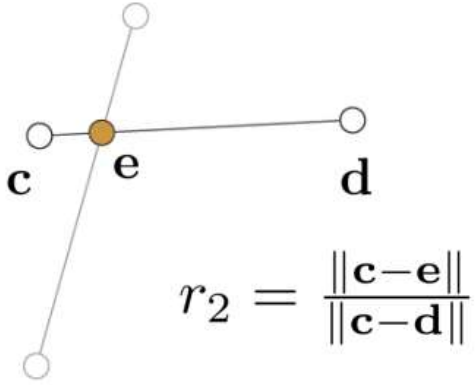
$$r_2 = \frac{\|\mathbf{c}-\mathbf{e}\|}{\|\mathbf{c}-\mathbf{d}\|}$$

FindCongruent (cuando $e_1 \neq e_2$)

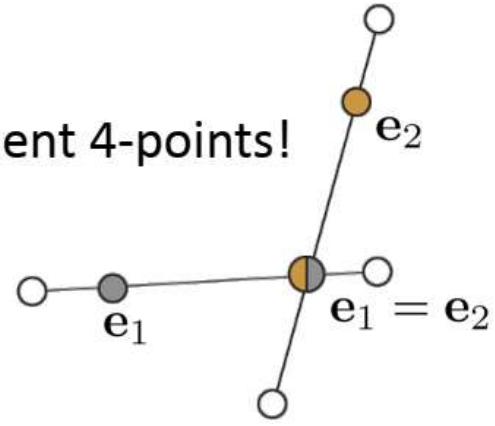




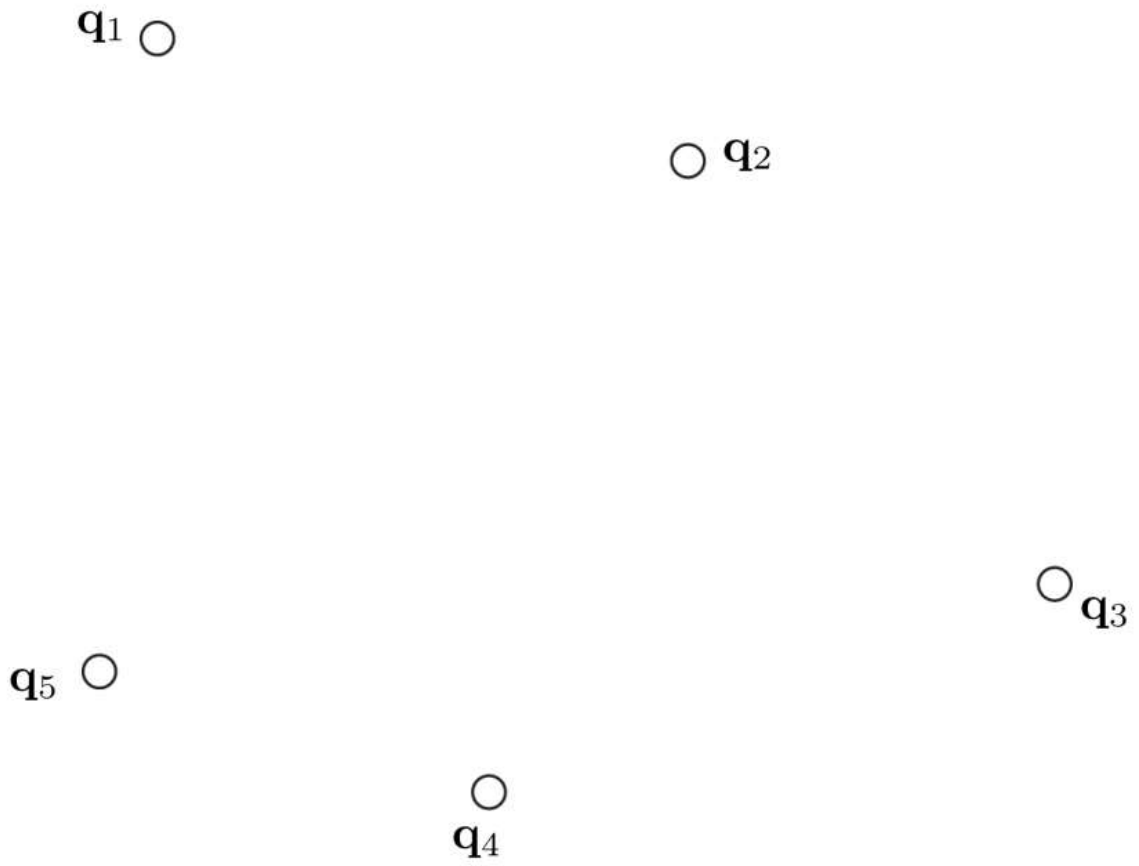
typical scenario



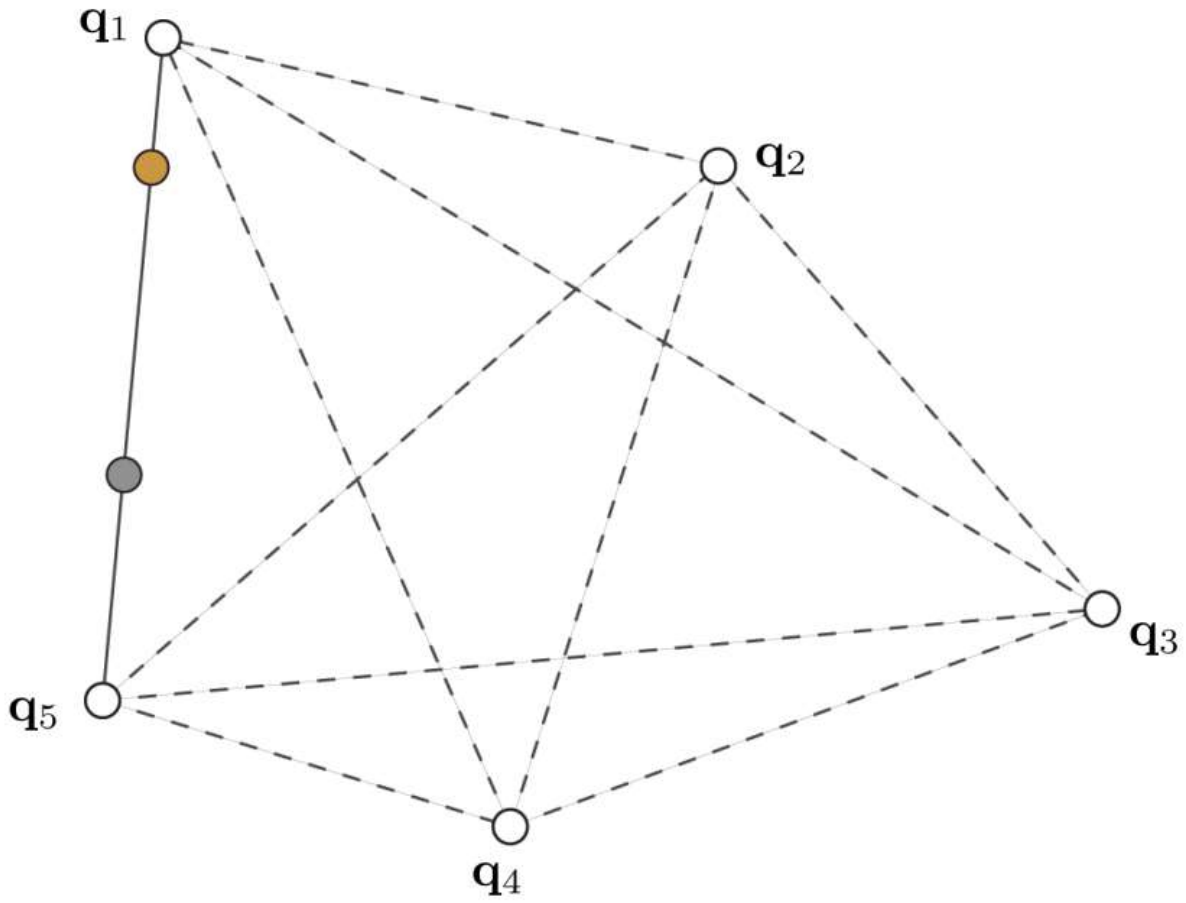
congruent 4-points!



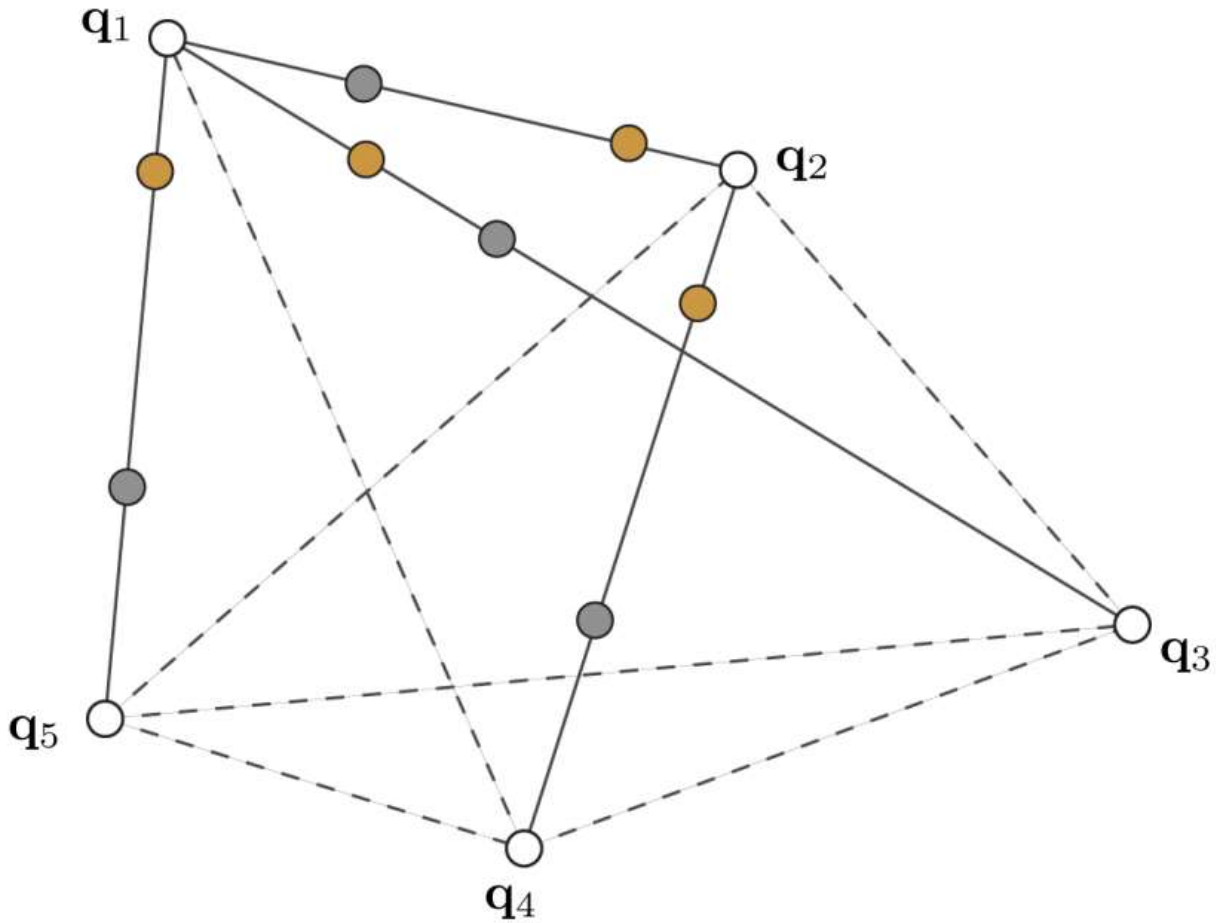
Extrayendo 4 puntos congruentes



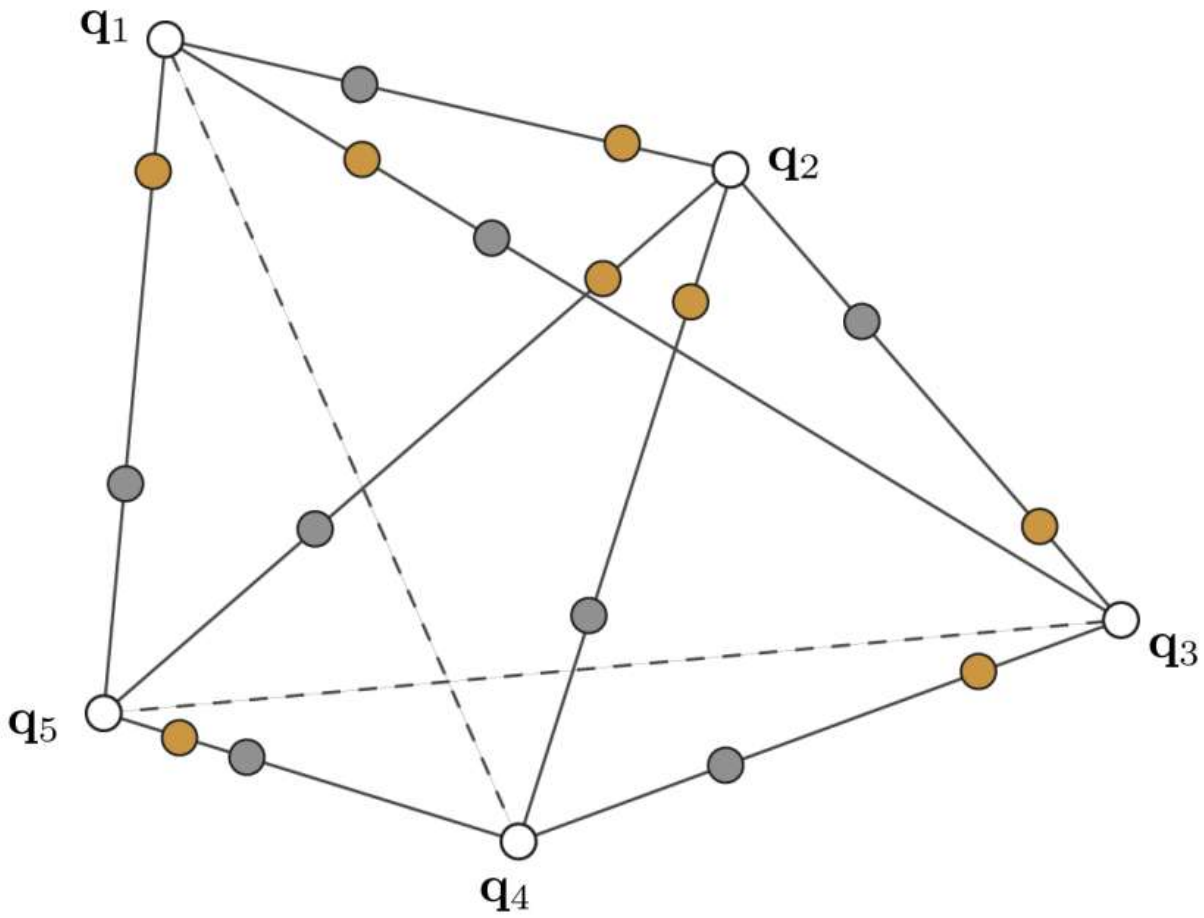
Extrayendo 4 puntos congruentes



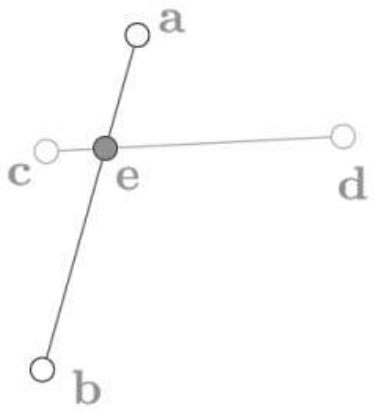
Extrayendo 4 puntos congruentes



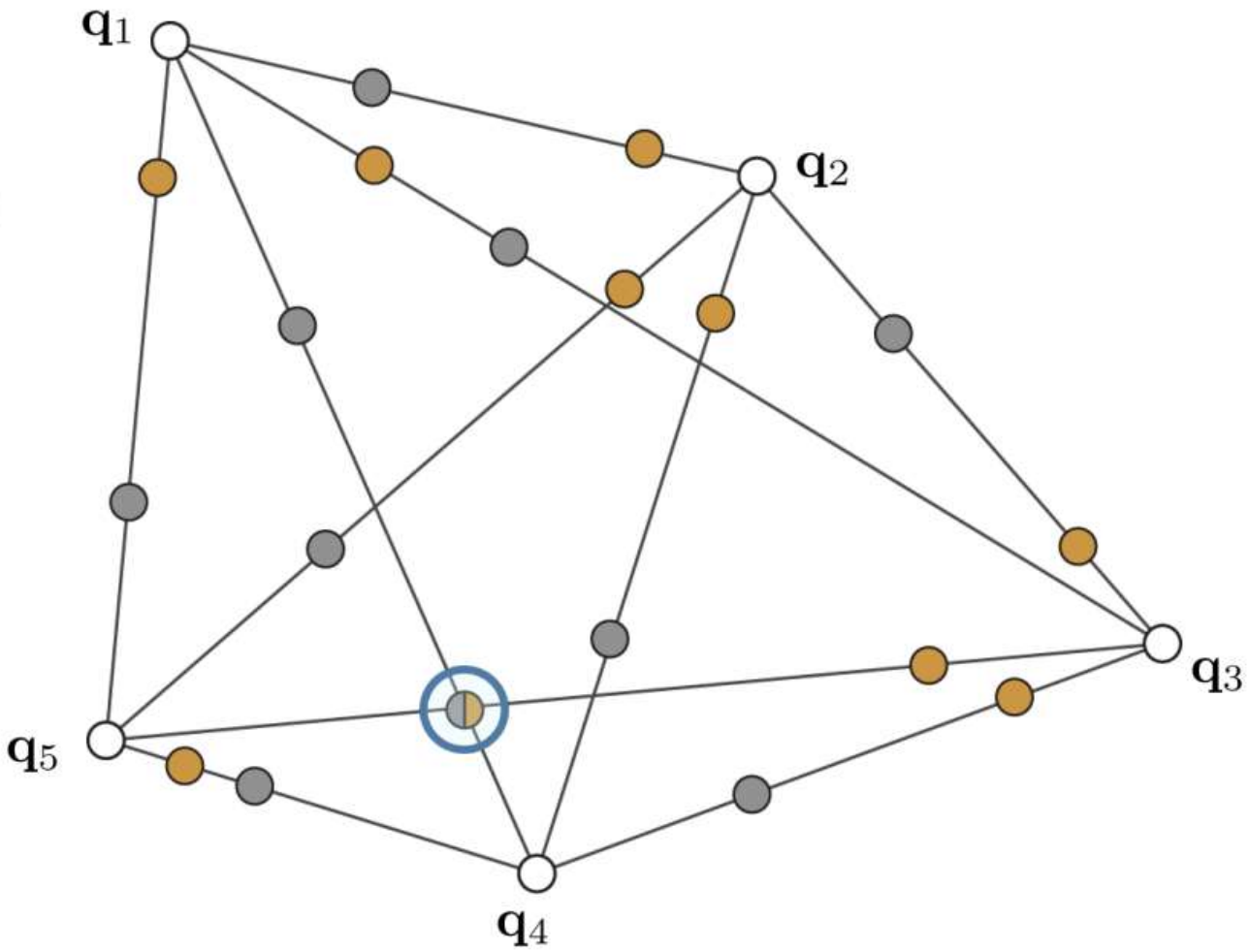
Extrayendo 4 puntos congruentes



Extrayendo 4 puntos congruentes



$\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\} \equiv$
$\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4\}$



Resultados

- Ante ruido