Procesamiento Geométrico y Análisis de Formas

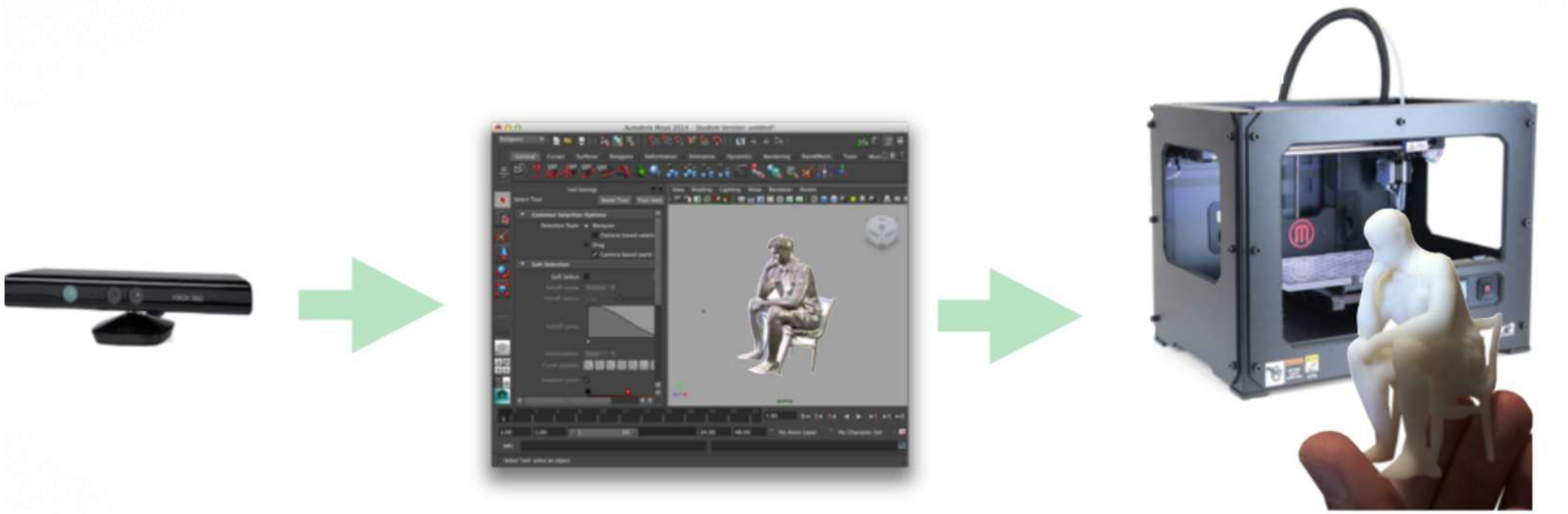
Ivan Sipiran

Adquisición de datos

- **Scanning, Procesamiento y Reconstrucción**
- Adquisición de formas 3D (escaneo)
- Alineamiento de formas: Iterative Closest Point (ICP)
- Procesamiento de nubes de puntos: estimación de normales y outliers
- Reconstrucción: obtención de superficie

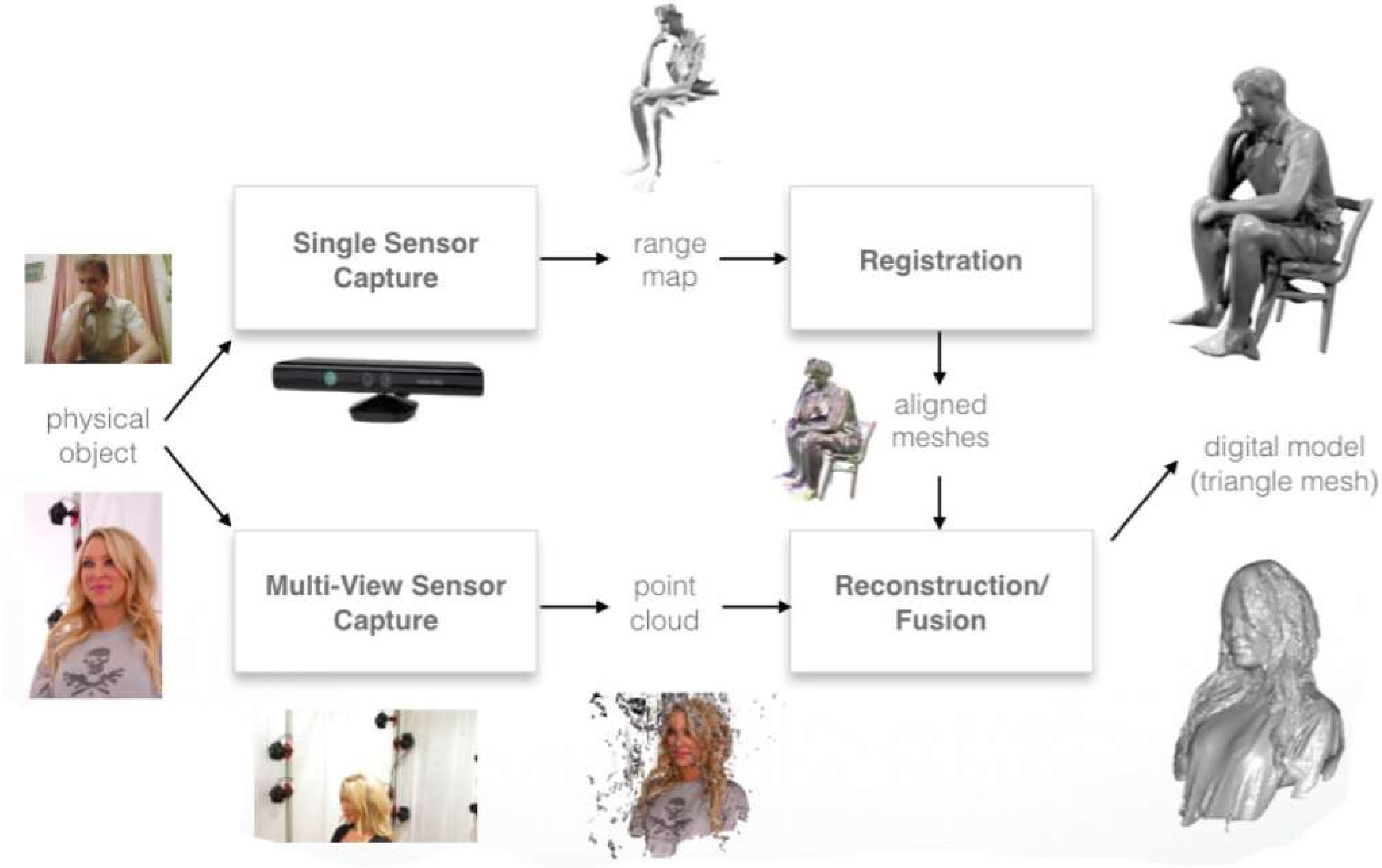
Pipeline de imágenes

Pipeline 3D



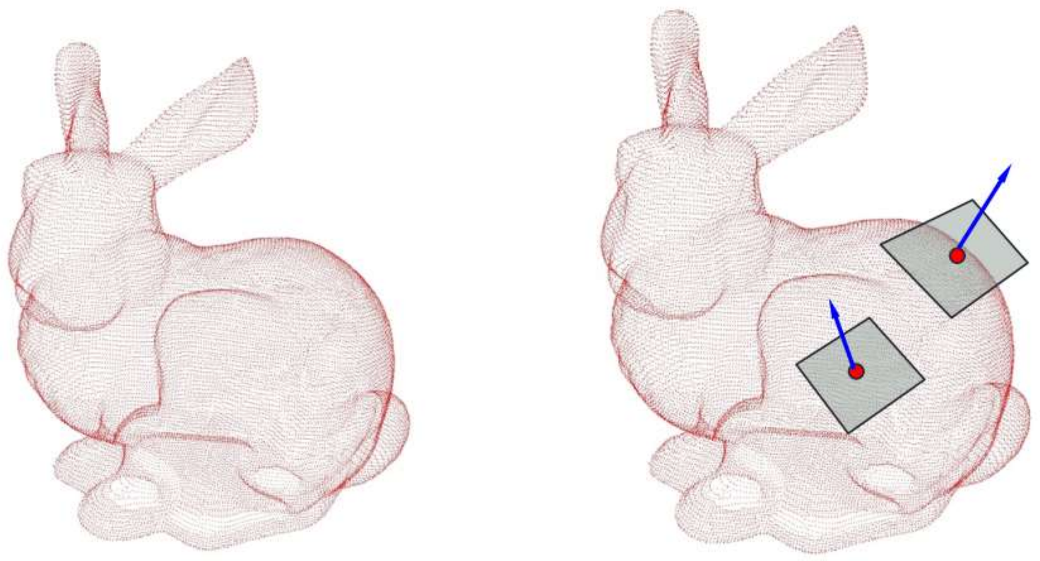
3D scanning — 3D processing/editing — 3D printing

Digitalización 3D



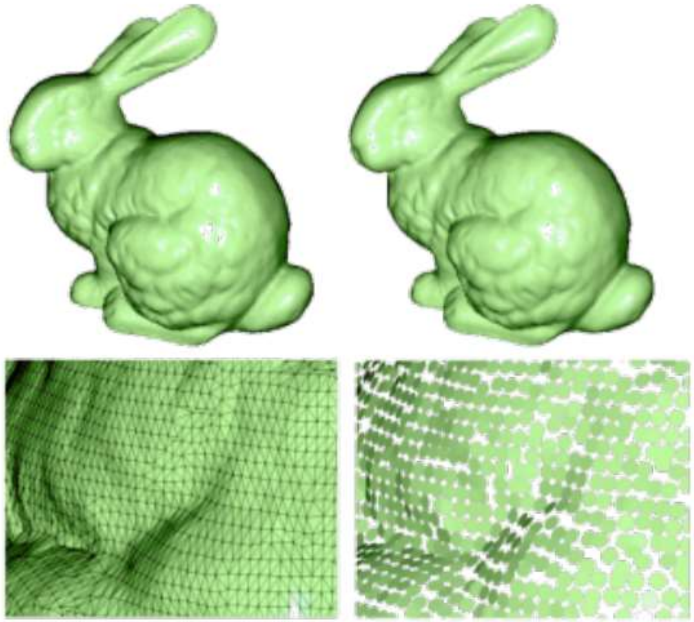
Point clouds

- Es la representación más simple: solo puntos, no conectividad
- Colección de coordenadas (x,y,z), posiblemente con normales



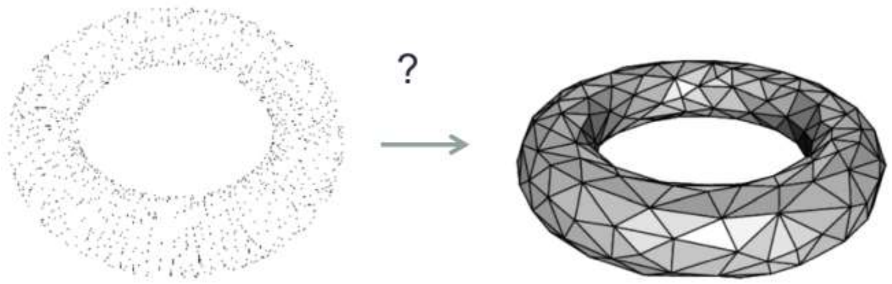
Point clouds

- Es la representación más simple: solo puntos, no conectividad
- Colección de coordenadas (x,y,z), posiblemente con normales
- Puntos con orientación son llamados surfels



Point clouds

- Es la representación más simple: solo puntos, no conectividad
- Colección de coordenadas (x,y,z), posiblemente con normales
- Puntos con orientación son llamados surfels
- Muchas limitaciones
  - No se simplifican o subdividen
  - No rendering suave directo
  - No información topológica

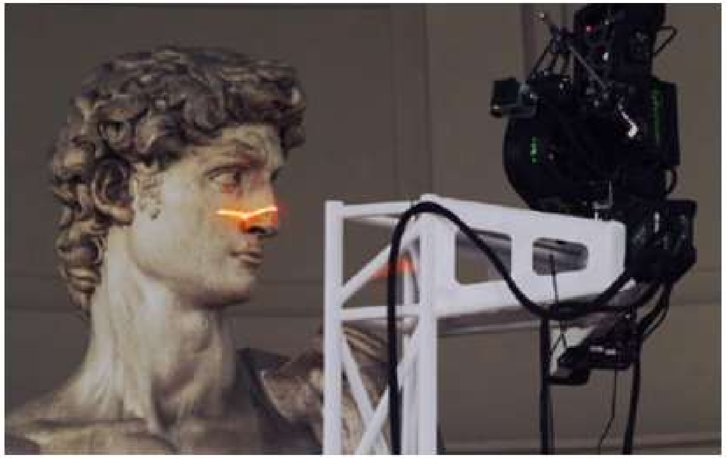


Point clouds

- Es la representación más simple: solo puntos, no conectividad
- Colección de coordenadas (x,y,z), posiblemente con normales
- Puntos con orientación son llamados surfels
- Muchas limitaciones
  - No se simplifican o subdividen
  - No rendering suave directo
  - No información topológica
  - Ruido

Porqué nubes de puntos?

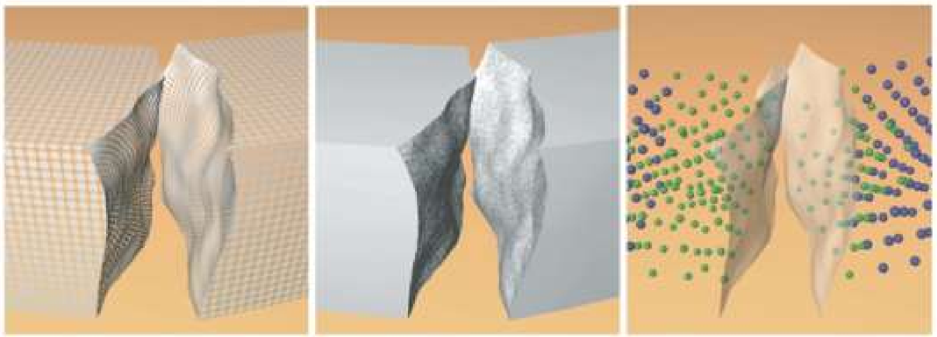
- Típicamente, es lo único disponible
  - Todos los dispositivos de captura generan nubes de puntos



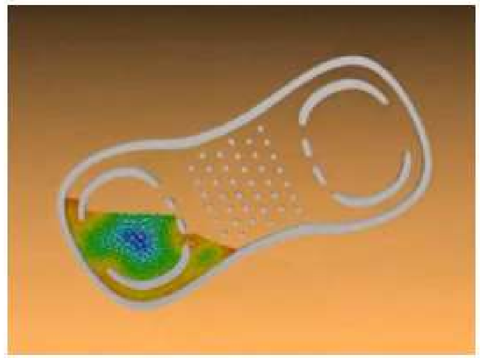
Porqué nubes de puntos?

- Típicamente, es lo único disponible
  - Todos los dispositivos de captura generan nubes de puntos
- Localidad: algunas veces, es lo más fácil de manejar

Fracturing Solids



Fluid Simulation

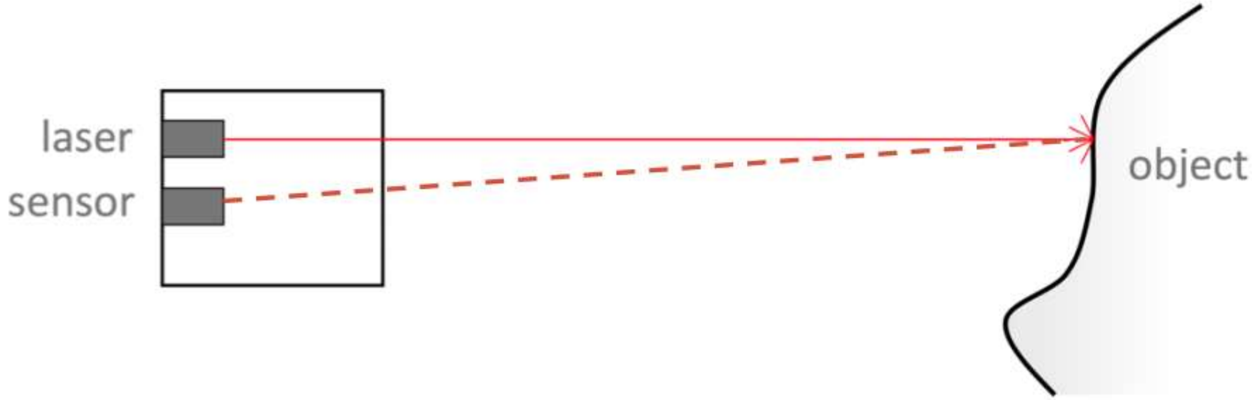


Modelo de captura típico

Escáners de una vista

- Escáners de rango
  - Escáner láser time-of-flight
  - Escáner láser basado en fase
- Triangulación
  - Escáner láser de barrido
  - Luz estructurada
- Estéreo / visión computacional
  - Estéreo pasivo
  - Estéreo activo

Escáners Time-of-Flight



- Emiten un pulso láser
- Capturan la reflexión
- Miden el tiempo de retorno

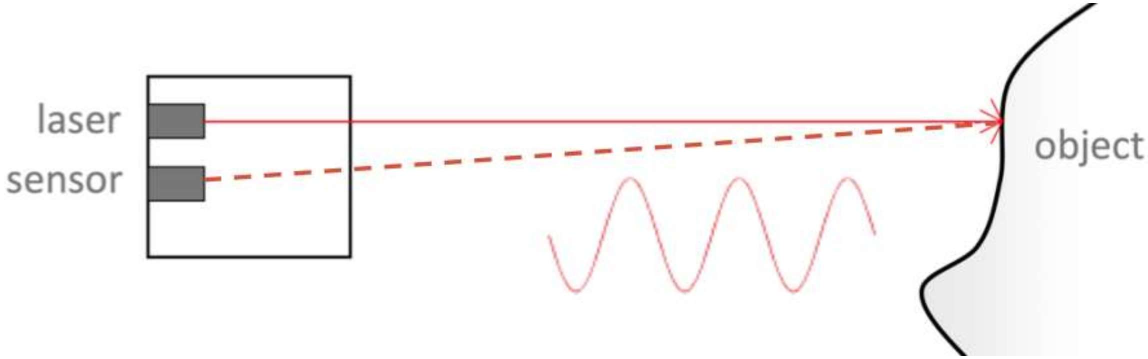
$$D = \frac{cT}{2}$$

c Velocidad de la luz

- Necesitan un reloj muy rápido: 1GHz logra 15 cm de precisión

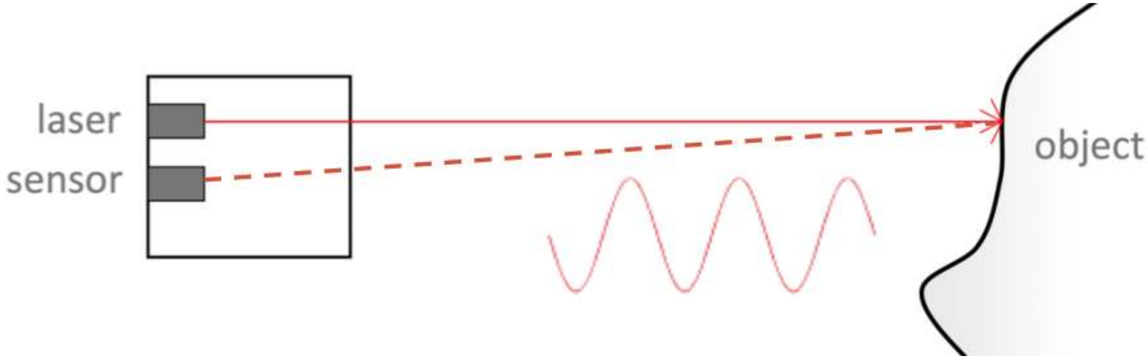
Escáners Time-of-Flight

Escáners de Rango basado en Fase



- Emiten un rayo modulado en fase continuo
- Capturan la reflexión
- Miden el shift de la fase entre la entrada y la salida

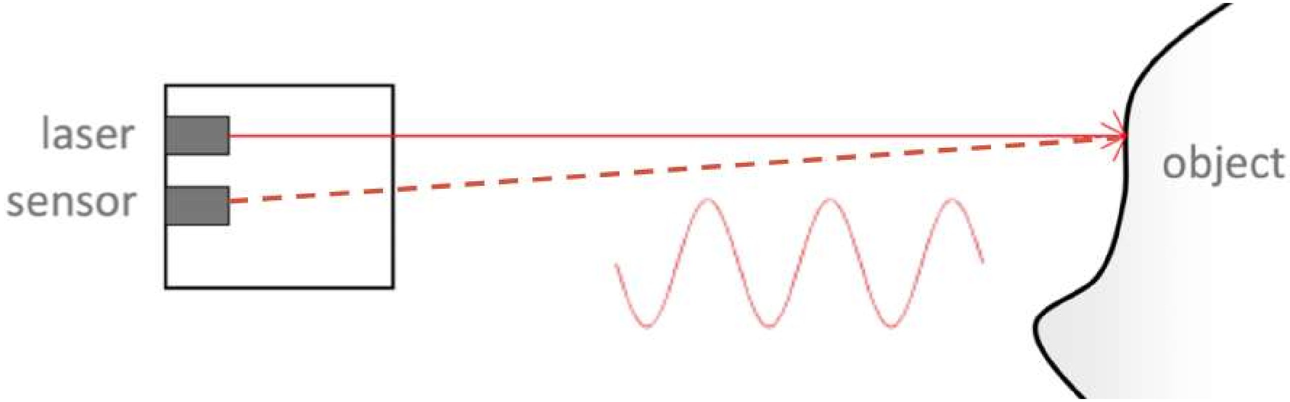
Escáners de Rango basado en Fase



- Señal emitida: $s(t) = a\cos(2\pi f t)$
- Señal recibida: $r(t) = A\cos(2\pi f(t-\tau)) + B$
- Correlación cruzada entre señales:

$$C(x) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} r(t)s(t+x)dt$$

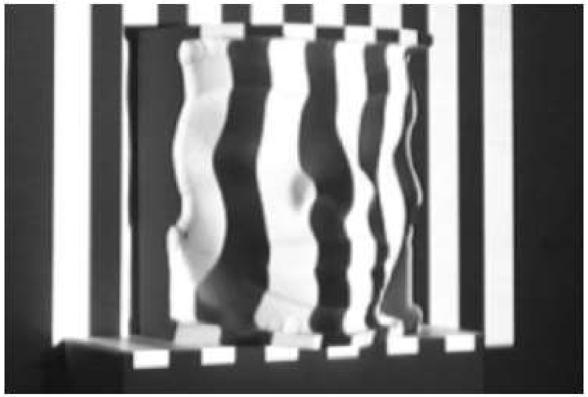
Escáneres de rango



- Típicamente, escáneres de rango proveen precisión limitada
- Requiere mucho post-procesamiento para tener una buena representación

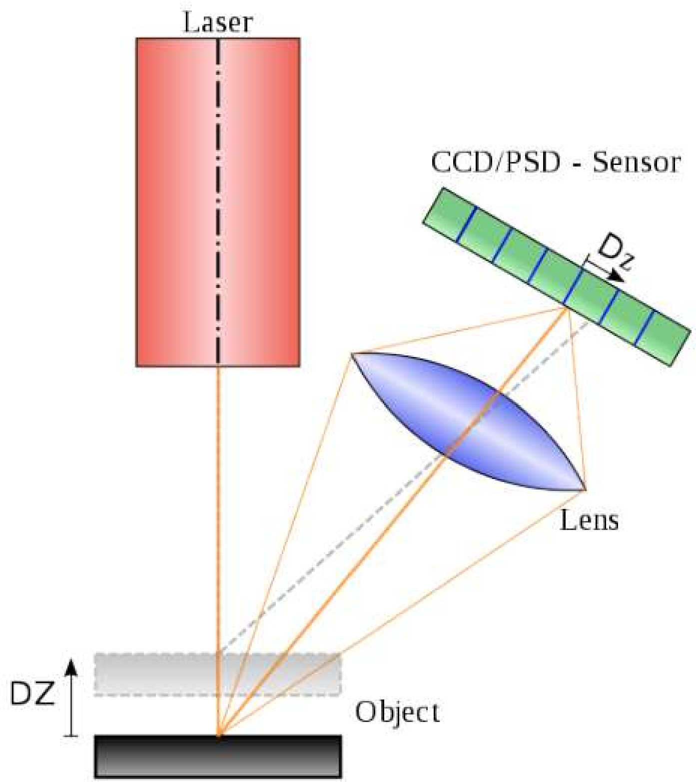
Enfoques basados en Triangulación

- Agregar un sensor fotométrico (cámara)
- Registrar la posición para un plano de referencia
- Cambio en la posición de registro puede ser usado para recuperar la profundidad



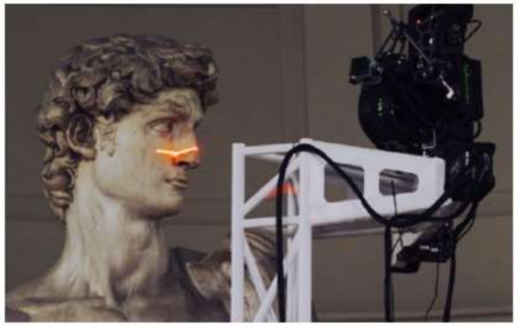

Enfoques basados en Triangulación

- Agregar un sensor fotométrico (cámara)
- Registrar la posición para un plano de referencia
- Cambio en la posición de registro puede ser usado para recuperar la profundidad

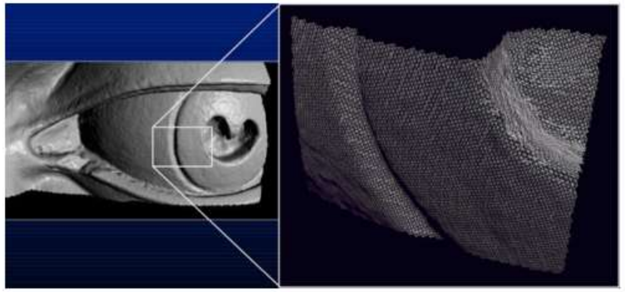


Enfoques basados en Triangulación

- Agregar un sensor fotométrico (cámara)
- Registrar la posición para un plano de referencia
- Si el sistema está bien calibrado, se puede obtener precisión extrema
- Problema: es lento y caro



David statue scan: over 1 billion points.

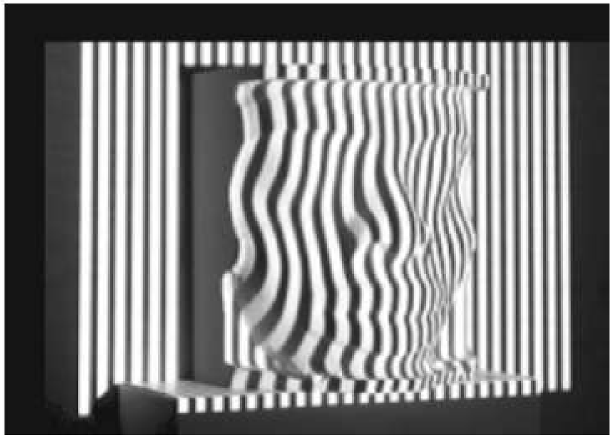


David's left eye:
source Levoy et al.

"The digital Michelangelo project: 3D scanning of large statues", Levoy et al., 2000

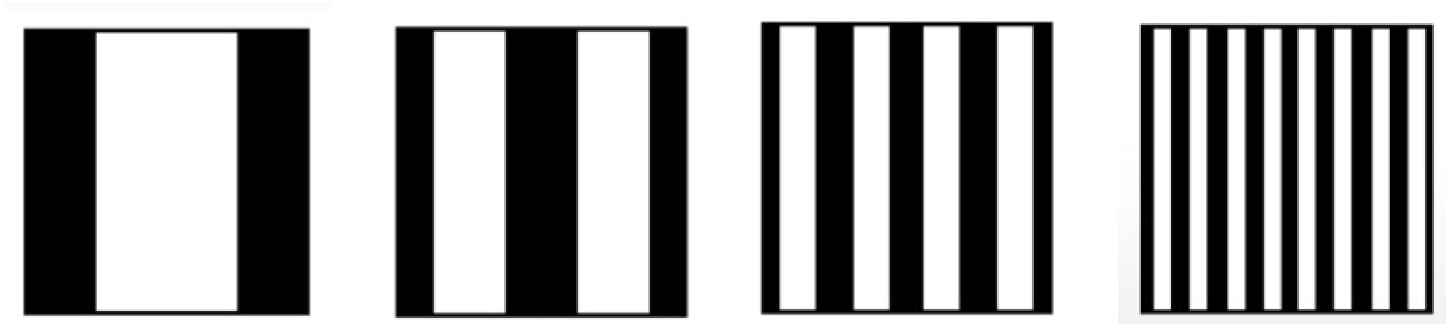
Escáners de Luz Estructurada

- Misma idea que enfoque de triangulación
- Reemplazar láser con un proyector que proyecta patrones de luz
- Reto: Identificar la correspondencia entre líneas de patrón de entrada con el patrón deformado



Escáners de Luz Estructurada

- Misma idea que enfoque de triangulación
- Proyectar múltiples patrones para identificar la posición de un punto



- Log(N) proyecciones son suficientes para identificar N patrones

Escáners de Luz Estructurada

- Misma idea que enfoque de triangulación
- Proyectar múltiples patrones para identificar la posición de un punto

Ventaja

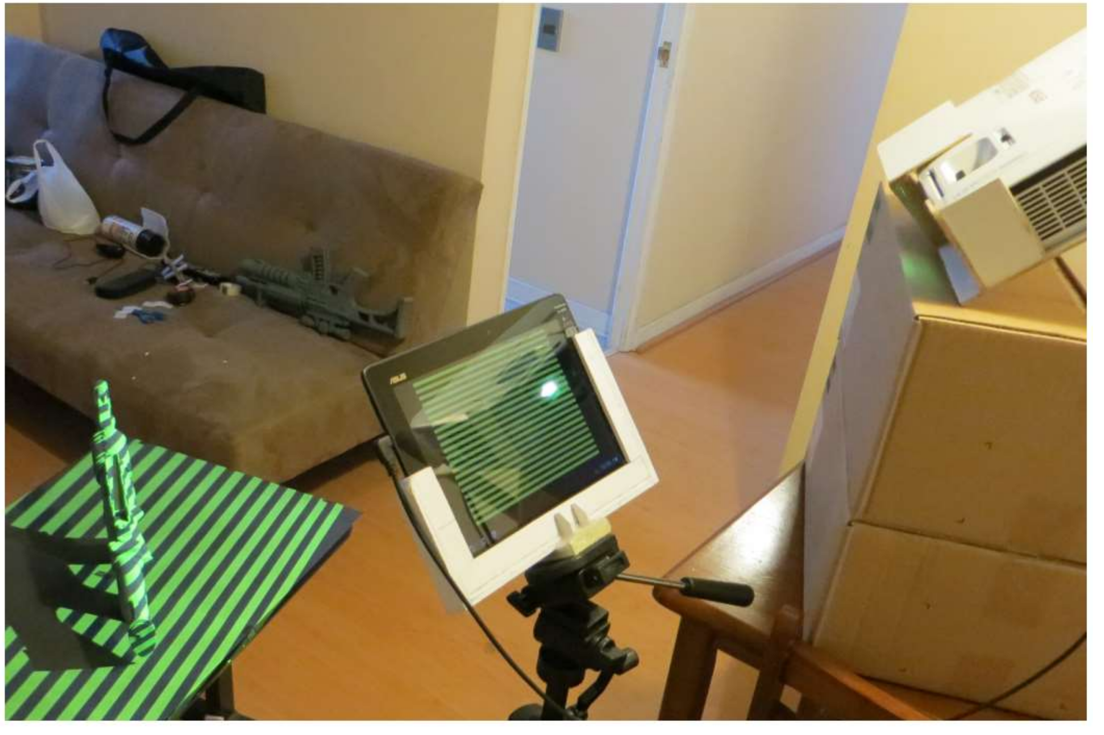
Costo y rapidez

Desventaja

Condiciones controladas y calibración de cámara / proyector

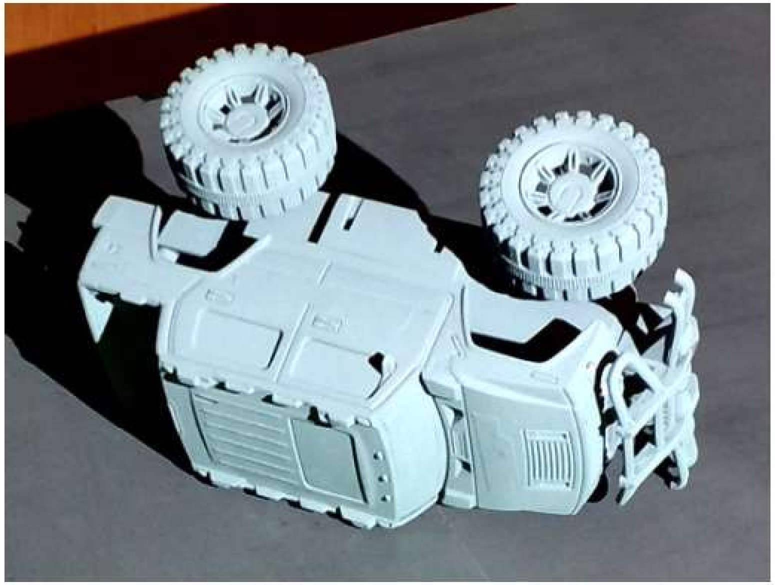
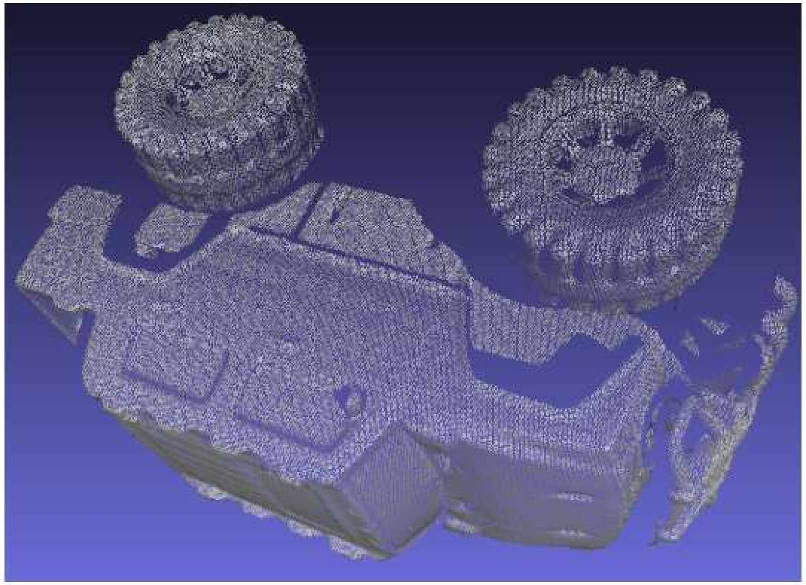
Escáners de Luz Estructurada

- Evaluación de algoritmos de matching parcial en objetos 3D utilizando un escáner 3D óptico – Rafael Meruane



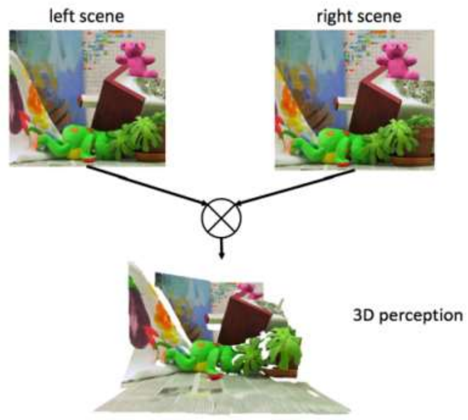
Escáners de Luz Estructurada

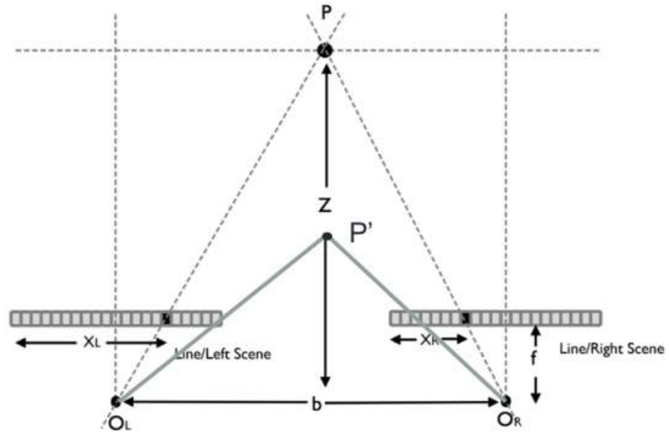
- Evaluación de algoritmos de matching parcial en objetos 3D utilizando un escáner 3D óptico – Rafael Meruane

Técnicas de Visión Computacional

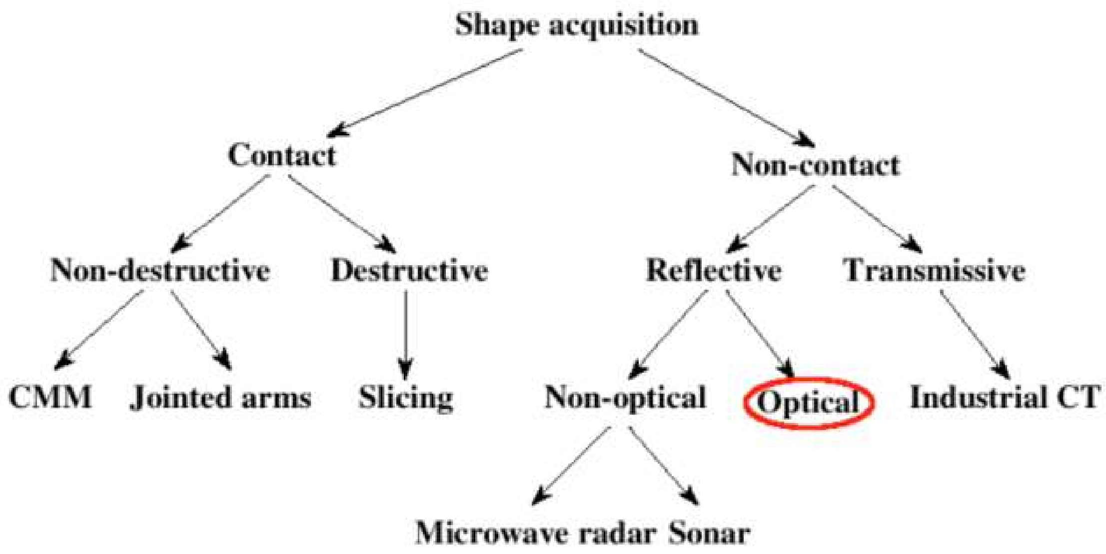
- Visión Estéreo

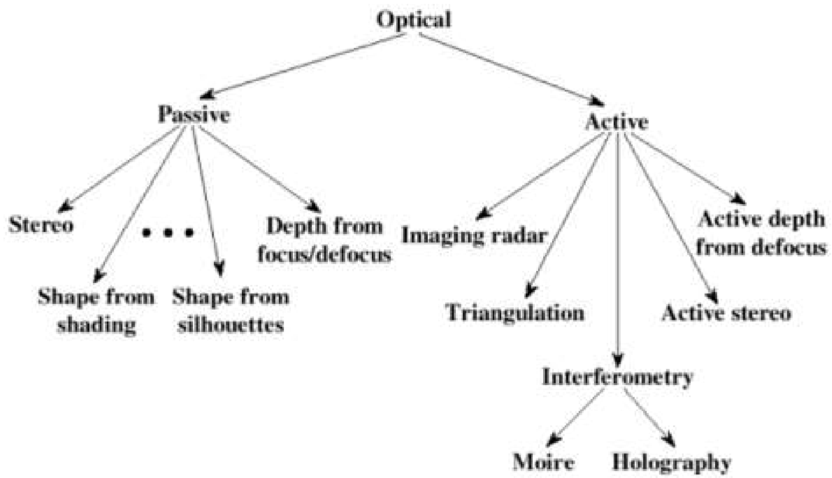




- Dadas dos imágenes, shift en el eje X está relacionado a la profundidad
- Problema complejo: determinar correspondencias en imágenes

Taxonomía





- 3D Imaging, Analysis and Applications – Capítulos 1-4. 2020

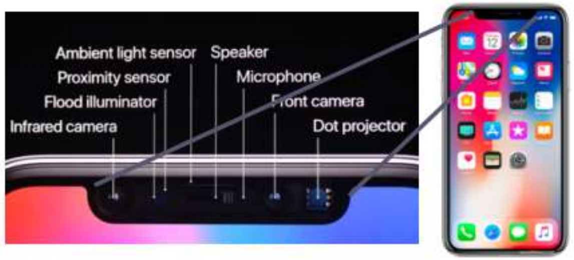
Microsoft Kinect (2009)

- Escáner 3D de bajo costo

- Imágenes de 640x480 y 300K puntos a 30FPS
- Usa iluminación activa infrarroja. Precisión de 1mm (a 0.5 m) a 4cm (a 2m)

Dispositivos Modernos

intel REALSENSE TECHNOLOGY



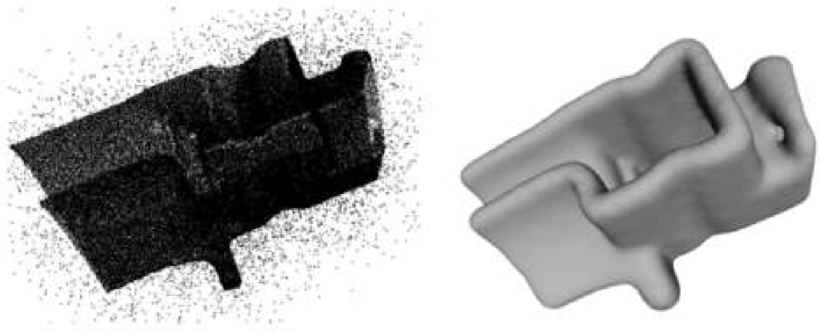
Apple iPhone X

Asus Zenfone AR

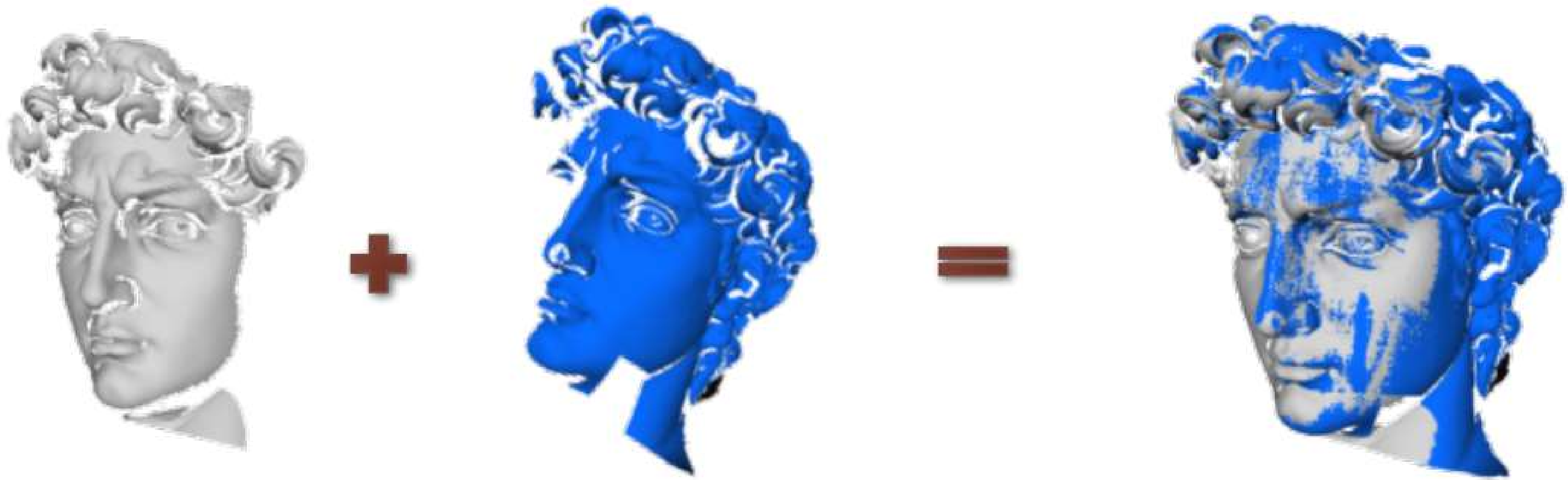
Sony Xperia XZ1

Procesamiento de Nubes de Puntos

- Típicamente, nubes de puntos son insuficientes para muchas aplicaciones. Pipeline de procesamiento:
  - Escaneo de objeto (adquisición)
  - Si hay múltiples escaneos, alinearlos
  - Suavizar – remover ruido
  - Estimar normales
  - Reconstruir superficie



Problema Fundamental de Registro



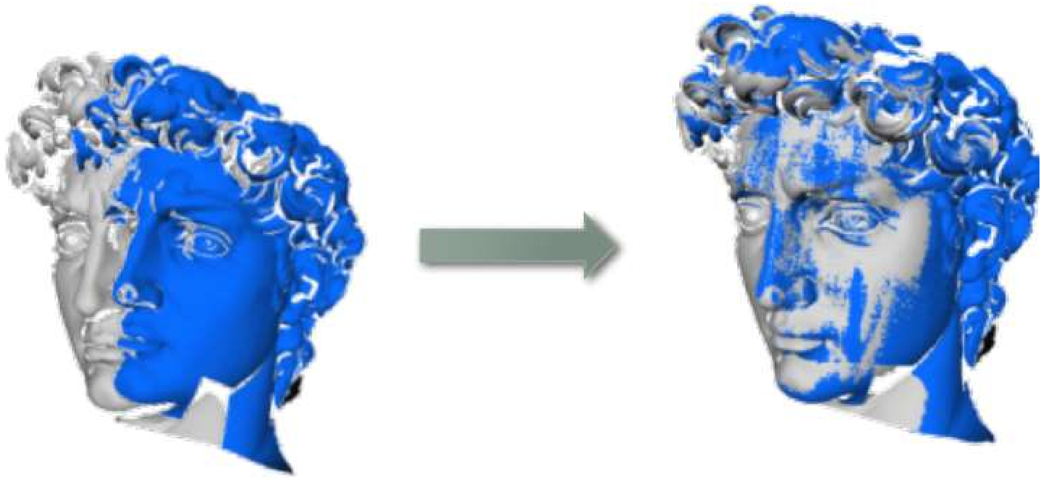
- Dadas (al menos) dos formas con geometría parcialmente sobrepuesta, encontrar un alineamiento entre ellas

Porqué registro?

- Problema fundamental en análisis de geometría
- ICP: Uno de los algoritmos ampliamente conocido en computación gráfica. Ampliamente usado en la industria.
- Es un muy buen ejercicio de programación
- Área activa de investigación

Alineamiento Local

- Instancia más simple del problema



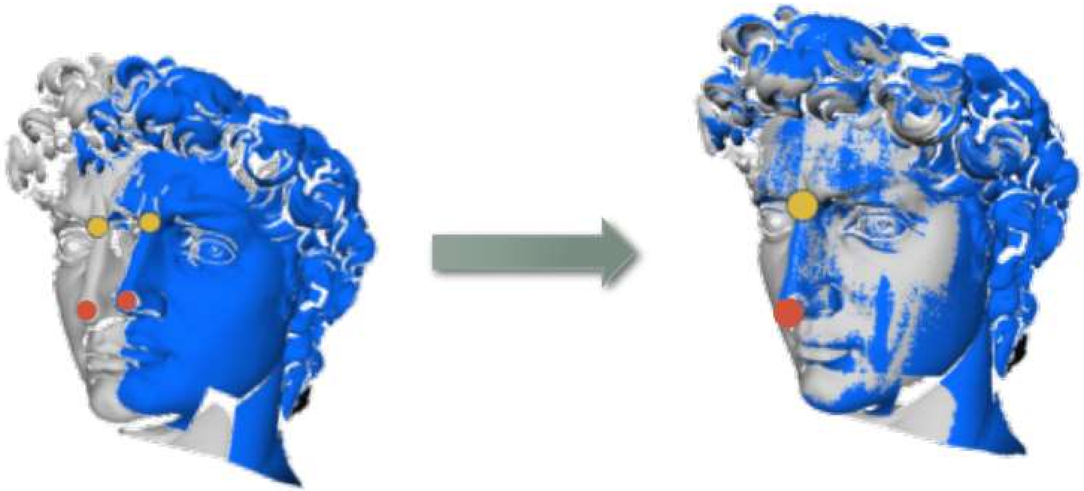
- Dadas dos formas aproximadamente alineadas, encontrar la transformación óptima

Otras aplicaciones

- Manufactura
  - Una forma es un modelo y la otra es un escaneo de un producto. Encontrar defectos
- Medicina
  - Encontrar correspondencias entre MRI de la misma persona
- Reconstrucción de animaciones
- Análisis Estadístico de Formas
  - Construir modelos desde una colección de formas

Alineamiento Local

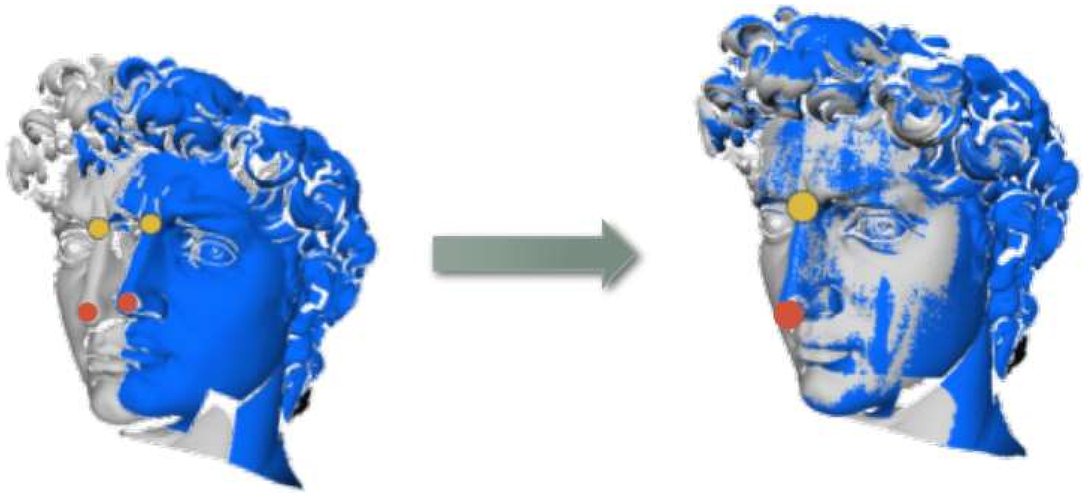
- Qué significa que un alineamiento sea bueno?



- Intuición: puntos correspondientes son cercanos después de transformar

Alineamiento Local

- Qué significa que un alineamiento sea bueno?
- Problemas
  - No sabemos qué puntos corresponden
  - No conocemos la transformación óptima



Iterative Closest Point(ICP)

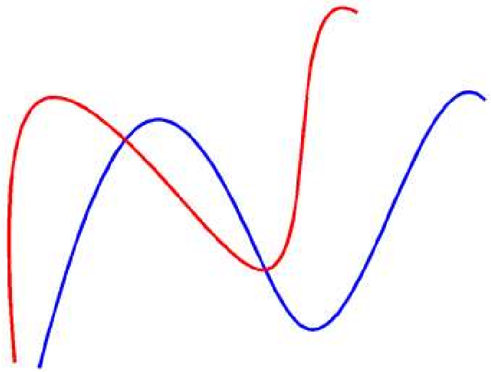
- Método: iterar entre dos pasos 1)encontrar correspondencias y 2) encontrar la transformación

- Dado un par de formas, Xe Y, iterar
  - Para cada $x_i \in X$ encontrar el vecino más cercano $y_i \in Y$
  - Encontrar la deformación $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point(ICP)

- Método: iterar entre dos pasos 1)encontrar correspondencias y 2) encontrar la transformación

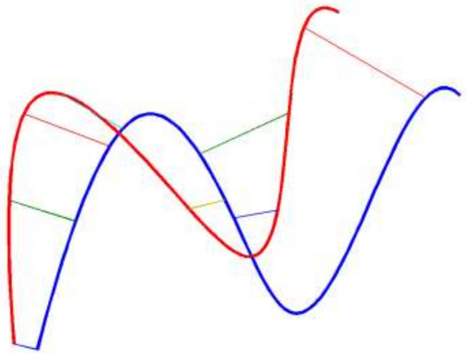


- Dado un par de formas, Xe Y, iterar
  - Para cada $x_i \in X$ encontrar el vecino más cercano $y_i \in Y$
  - Encontrar la deformación $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point(ICP)

- Método: iterar entre dos pasos 1)encontrar correspondencias y 2) encontrar la transformación

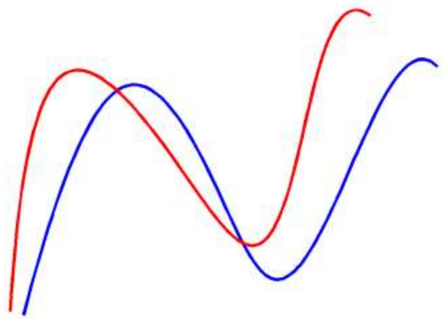


- Dado un par de formas, Xe Y, iterar
  - Para cada $x_i \in X$ encontrar el vecino más cercano $y_i \in Y$
  - Encontrar la deformación $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point(ICP)

- Método: iterar entre dos pasos 1)encontrar correspondencias y 2) encontrar la transformación

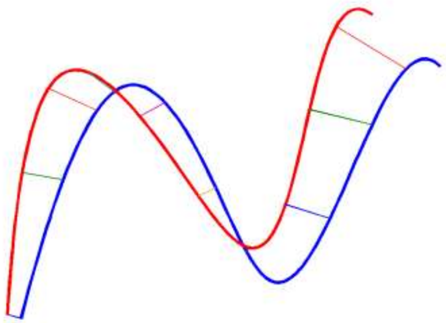


- Dado un par de formas, Xe Y, iterar
  - Para cada $x_i \in X$ encontrar el vecino más cercano $y_i \in Y$
  - Encontrar la deformación $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point(ICP)

- Método: iterar entre dos pasos 1)encontrar correspondencias y 2) encontrar la transformación

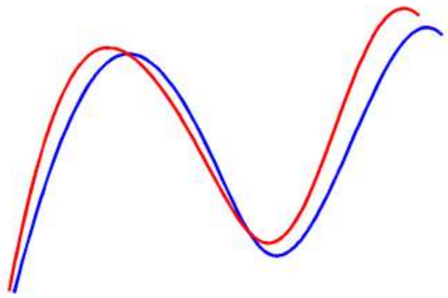


- Dado un par de formas, Xe Y, iterar
  - Para cada $x_i \in X$ encontrar el vecino más cercano $y_i \in Y$
  - Encontrar la deformación $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point(ICP)

- Método: iterar entre dos pasos 1)encontrar correspondencias y 2) encontrar la transformación

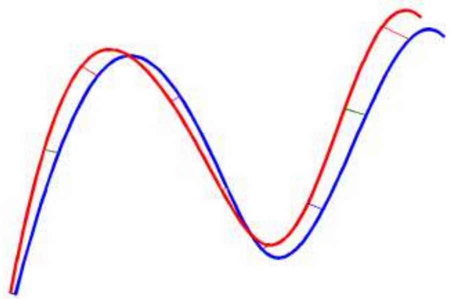


- Dado un par de formas, Xe Y, iterar
  - Para cada $x_i \in X$ encontrar el vecino más cercano $y_i \in Y$
  - Encontrar la deformación $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point(ICP)

- Método: iterar entre dos pasos 1)encontrar correspondencias y 2) encontrar la transformación

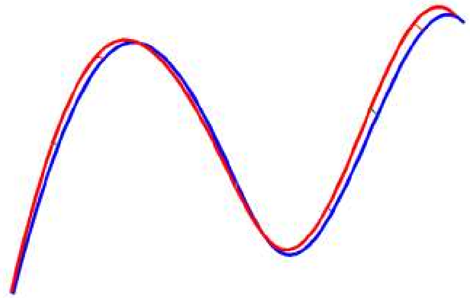


- Dado un par de formas, Xe Y, iterar
  - Para cada $x_i \in X$ encontrar el vecino más cercano $y_i \in Y$
  - Encontrar la deformación $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point(ICP)

- Método: iterar entre dos pasos 1)encontrar correspondencias y 2) encontrar la transformación

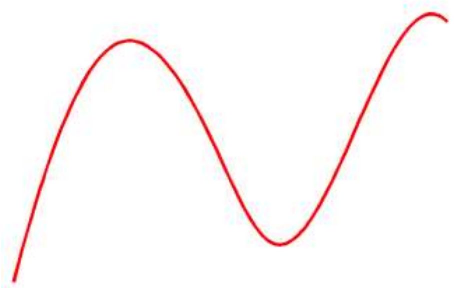


- Dado un par de formas, Xe Y, iterar
  - Para cada $x_i \in X$ encontrar el vecino más cercano $y_i \in Y$
  - Encontrar la deformación $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point(ICP)

- Método: iterar entre dos pasos 1)encontrar correspondencias y 2) encontrar la transformación



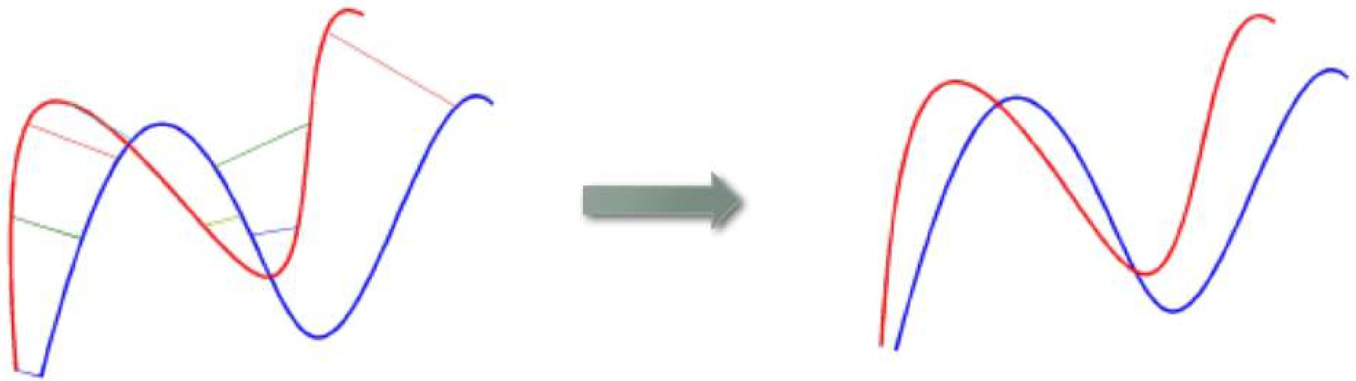
- Dado un par de formas, Xe Y, iterar
  - Para cada $x_i \in X$ encontrar el vecino más cercano $y_i \in Y$
  - Encontrar la deformación $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Iterative Closest Point(ICP)

- Se requieren dos computaciones básicas

- Computar vecinos más cercanos
- Computar la transformación óptima



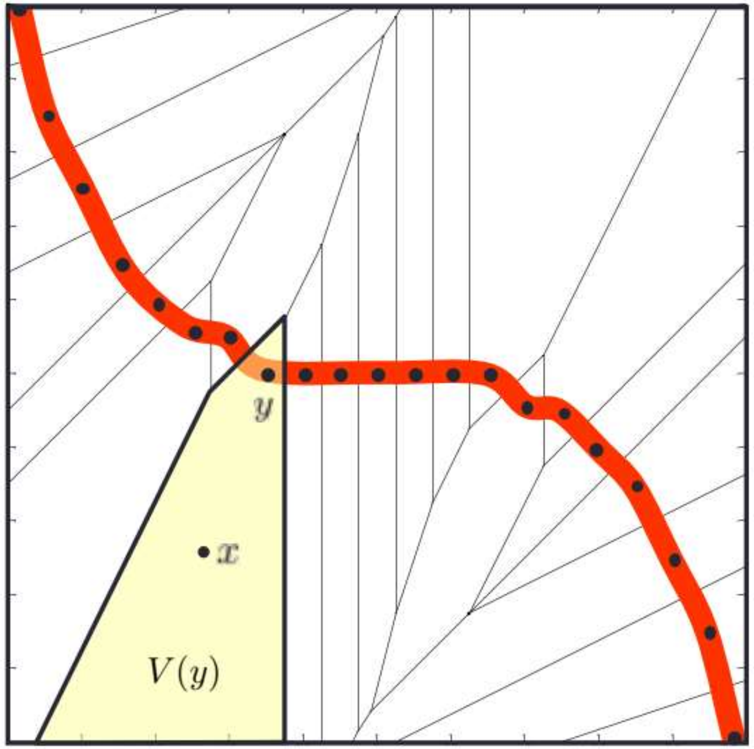
ICP: Vecinos más cercanos

- Puntos más cercanos

$$y_i = arg \min_{y \in Y} \|y - x_i\|$$

- Cómo hacerlo eficientemente?
- Complejidad $O(MN)$
- Y divide el espacio en celdas Voronoi
- Problema es encontrar la celda a la que pertenece un punto

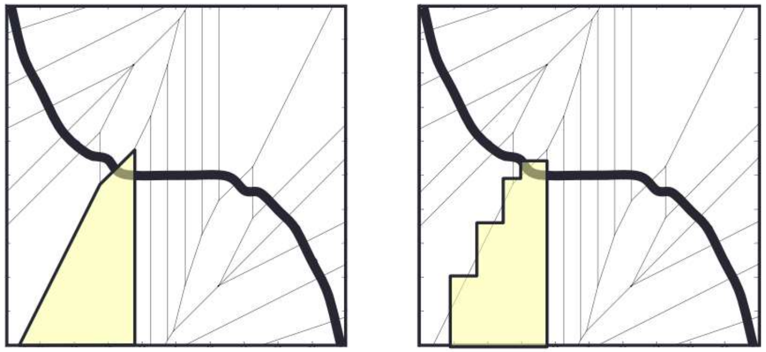
ICP: Vecinos más cercanos



$$V(y \in Y) = \{z \in \mathbb{R}^3 : \|y - z\| < \|y' - z\| \ \forall \ y' \in Y \neq y\}$$

ICP: Vecinos más cercanos

- Vecinos más cercanos aproximados

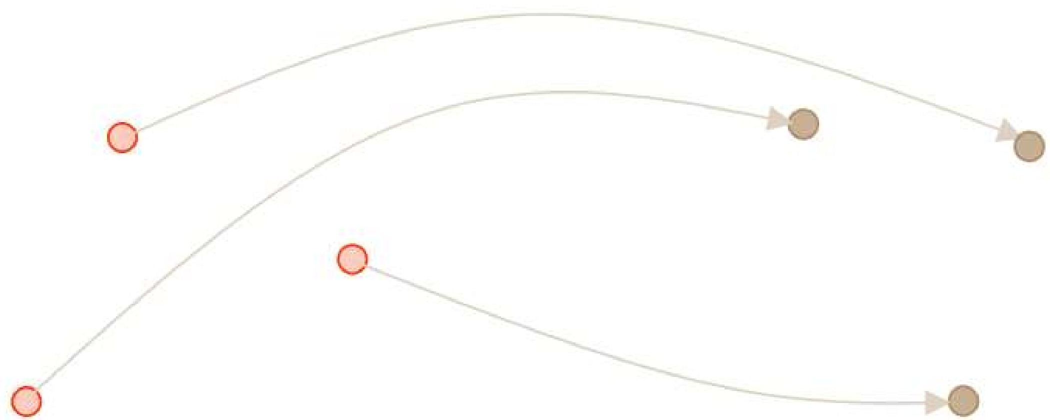


- Usar árboles de partición (kd-tree, por ejemplo)
- Complejidad de algoritmo aproximado $O(N \log M)$

ICP: Transformación óptima

- Formulación
  - Dados dos conjuntos de puntos: $\{x_i\}, \{y_i\} \in \mathbb{R}^3, i = 1..n$. Encontrar la transformación rígida $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$



ICP: Transformación óptima

- Formulación
  - Dados dos conjuntos de puntos: $\{x_i\}, \{y_i\} \in \mathbb{R}^3, i = 1..n$. Encontrar la transformación rígida $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

- Solución cerrada
  - Construir $C = \sum_{i=1}^{N} (y_i - \mu^Y)(x_i - \mu^X)^T$
  - Computar la SVD de C

$$C = U\Sigma V^T$$

    - Si $\det(UV^T) = 1, R_{opt} = UV^T$
    - Sino $R_{opt} = U\tilde{\Sigma}V^T, \tilde{\Sigma} = diag(1,1, \dots, -1)$
  - $t_{opt} = \mu^Y - R_{opt}\mu^X$

$$\mu^X = \frac{1}{N}\sum_i x_i$$

$$\mu^Y = \frac{1}{N}\sum_i y_i$$

ICP: Transformación óptima

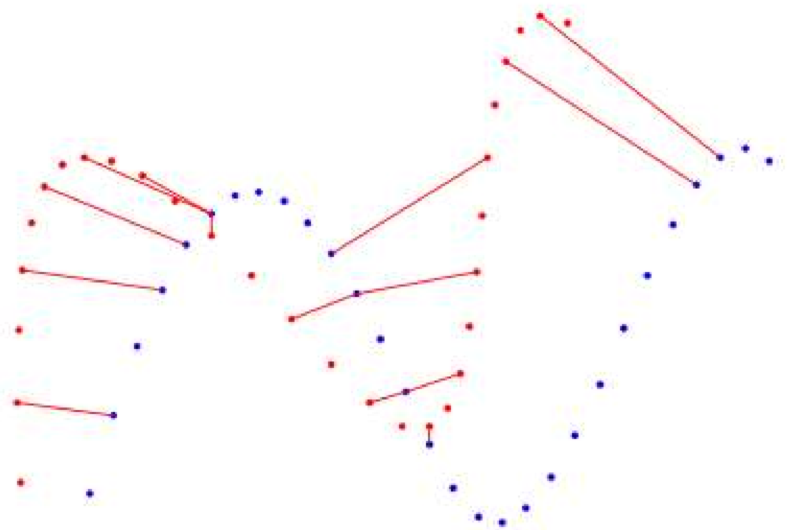
- Formulación
  - Dados dos conjuntos de puntos: $\{x_i\}, \{y_i\} \in \mathbb{R}^3, i = 1..n$. Encontrar la transformación rígida $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

  - Convergencia
    - En cada iteración $\sum_{i=1}^{N} d^2(x_i, Y)$ siempre decrece
    - Converge a un mínimo local
    - Buena inicialización: mínimo global

Variaciones de ICP

- Seleccionar puntos: sampling
- Correspondencias a puntos en la otra malla
- Ponderar las correspondencias
- Rechazar outliers
- Asignar una métrica a cada transformación
- Minimizar la métrica con respecto a la transformación

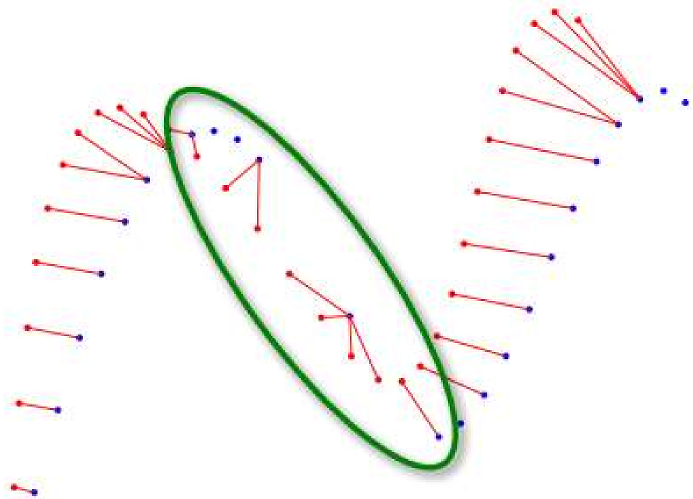


ICP: Transformación óptima

- Formulación
  - Dados dos conjuntos de puntos: $\{x_i\}, \{y_i\} \in \mathbb{R}^3, i = 1..n$. Encontrar la transformación rígida $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Problema: sampling disparejo

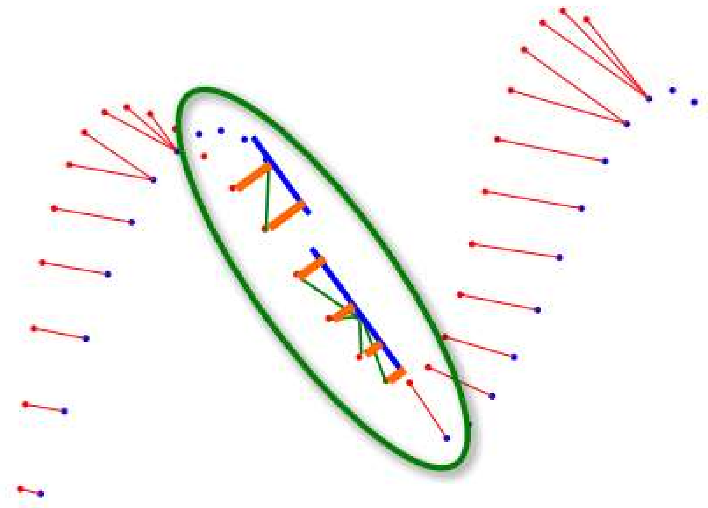


ICP: Transformación óptima

- Formulación
  - Dados dos conjuntos de puntos: $\{x_i\}, \{y_i\} \in \mathbb{R}^3, i = 1..n$. Encontrar la transformación rígida $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} \|\boldsymbol{R}x_i + t - y_i\|_2^2$$

Solución: minimizar distancia a plano tangente

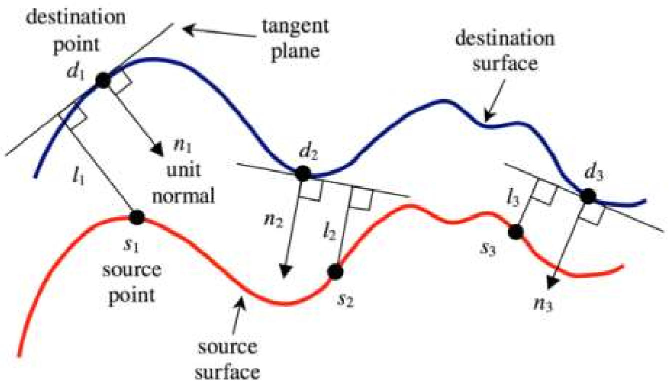


ICP: Transformación óptima

- Formulación
  - Dados dos conjuntos de puntos: $\{x_i\}, \{y_i\} \in \mathbb{R}^3, i = 1..n$. Encontrar la transformación rígida $\boldsymbol{R}, t$ que minimiza

$$\sum_{i=1}^{N} d(\boldsymbol{R}x_i + t, P(y_i))^2 = \sum_{i=1}^{N} ((\boldsymbol{R}x_i + t - y_i)^T \boldsymbol{n}_{y_i})^2$$

Solución: minimizar distancia a plano tangente



ICP: Transformación óptima

- Formulación
  - Dados dos conjuntos de puntos: $\{x_i\}, \{y_i\} \in \mathbb{R}^3, i = 1..n$. Encontrar la transformación rígida $\boldsymbol{R}, t$ que minimiza

$$\boldsymbol{R}_{opt}, t_{opt} = arg \min_{\boldsymbol{R}^T\boldsymbol{R}=Id,t} \sum_{i=1}^{N} ((\boldsymbol{R}x_i + t - y_i)^T \boldsymbol{n}_{y_i})^2$$

  - Cómo minimizar el error?
  - Aunque el error es cuadrático, el espacio de matrices de rotación no es lineal
  - No hay solución cerrada!

ICP: Transformación óptima

- Formulación
  - Dados dos conjuntos de puntos: $\{x_i\}, \{y_i\} \in \mathbb{R}^3, i = 1..n$. Encontrar la transformación rígida $\boldsymbol{R}, t$ que minimiza

$$\boldsymbol{R}_{opt}, t_{opt} = arg \min_{\boldsymbol{R}^T\boldsymbol{R}=Id,t} \sum_{i=1}^{N} ((\boldsymbol{R}x_i + t - y_i)^T \boldsymbol{n}_{y_i})^2$$

  - Solución:
    - Linearizar rotación. Asumir que el ángulo de rotación es pequeño.

$$\boldsymbol{R}x_i \approx x_i + rx_i$$

r eje

$\|r\|$ Ángulo de rotación

$$R(r,\alpha)x_i = x_i \cos\alpha + (r \times x_i)\sin\alpha + r(r^T x_i)(1 - \cos\alpha)$$

$\sin\alpha \approx \alpha$

$\cos\alpha \approx 1$

ICP: Transformación óptima

- Formulación
  - Dados dos conjuntos de puntos: $\{x_i\}, \{y_i\} \in \mathbb{R}^3, i = 1..n$. Encontrar la transformación rígida $\boldsymbol{R}, t$ que minimiza

$$E(r,t) = \sum_{i=1}^{N}((x_i \;\; + r \times x_i \;\; + t - y_i \;\;)^T \boldsymbol{n}_{y_i})^2$$

$$E(r,t) = \sum_{i=1}^{N}((x_i \;\; - y_i \;\;)^T n_i + r^T(x_i \times n_i) + t^T n_i \;\;)^2$$

  - Haciendo $\frac{\partial E(r,t)}{\partial r} = 0$ y $\frac{\partial E(r,t)}{\partial t} = 0$ nos lleva a un sistema lineal de 6x6

$$x = \begin{pmatrix} r \\ t \end{pmatrix} \quad A = \sum \begin{pmatrix} x_i \times \mathbf{n}_{y_i} \\ \mathbf{n}_{y_i} \end{pmatrix} \begin{pmatrix} x_i \times \mathbf{n}_{y_i} \\ \mathbf{n}_{y_i} \end{pmatrix}^T \quad b = \sum (y_i - x_i)^T \mathbf{n}_{y_i} \begin{pmatrix} x_i \times \mathbf{n}_{y_i} \\ \mathbf{n}_{y_i} \end{pmatrix}$$