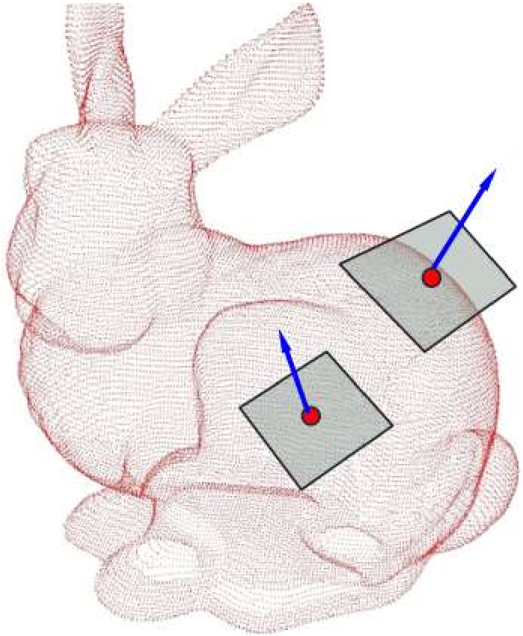
Procesamiento Geométrico y Análisis de Formas

Ivan Sipiran

Estimación de normales y eliminación de outliers

- Problemas fundamentales en procesamiento de geometría
- Pueden ser resueltas con el mismo enfoque



Estimación de normales

- Asumimos que tenemos un buen sampling de la superficie

- El objetivo es encontrar la mejor aproximación de la dirección tangente, y por lo tanto la normal a la línea tangente.

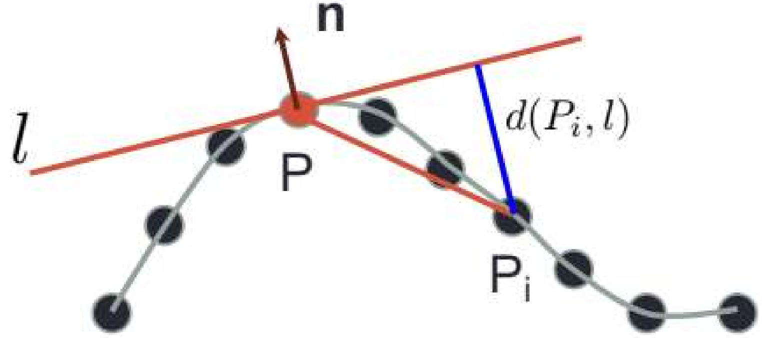
Estimación de normales

- Asumimos que tenemos un buen sampling de la superficie

- El objetivo es encontrar la mejor aproximación de la dirección tangente, y por lo tanto la normal a la línea tangente.

Estimación de normales

- Asumimos que tenemos un buen sampling de la superficie

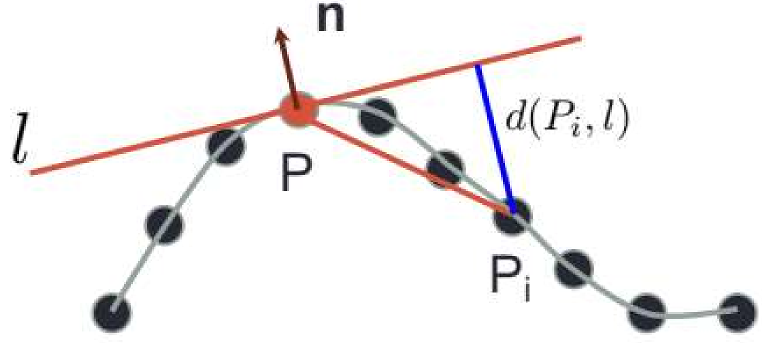


- Objetivo: encontrar la mejor aproximación de la normal en P
- Método: dada una línea que pasa por P con normal $\mathbf{n}$**,** para otro punto p_i

$$d(p_i, l)^2 = \frac{((p_i - P)^T \mathbf{n})^2}{\mathbf{n}^T \mathbf{n}} = ((p_i - P)^T \mathbf{n})^2 \text{ if } \|\mathbf{n}\| = 1$$

Estimación de normales

- Asumimos que tenemos un buen sampling de la superficie

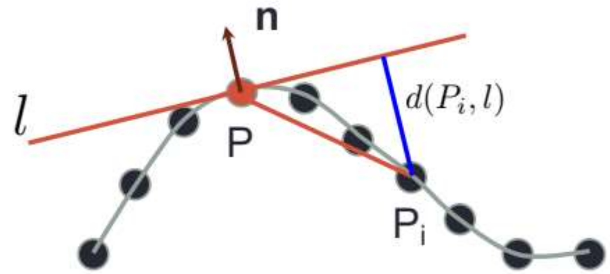


- Objetivo: encontrar la mejor aproximación de la normal en P
- Método: encontrar el $\mathbf{n}$ que minimiza $\sum_{i=1}^{k} d(p_i, l)$ para un conjunto de puntos vecinos de P

$$\mathbf{n}_{\text{opt}} = \arg \min_{\|\mathbf{n}\|=1} \sum_{i=1}^{k} ((p_i - P)^T \mathbf{n})^2$$

Estimación de normales

- Asumimos que tenemos un buen sampling de la superficie



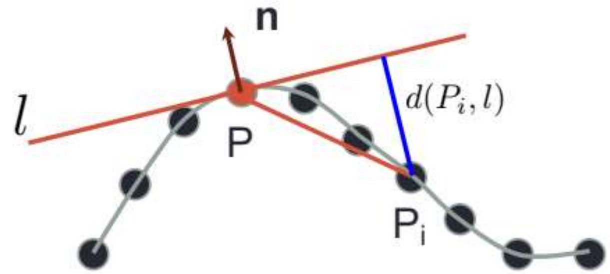
- Usando multiplicadores de Lagrange

$$\frac{\partial}{\partial \mathbf{n}}\left(\sum_{i=1}^{k}((p_i - P)^T\mathbf{n})^2\right) - \lambda\frac{\partial}{\partial \mathbf{n}}\left(\mathbf{n}^T\mathbf{n}\right) = 0$$

$$\left(\sum_{i=1}^{k}(p_i - P)(p_i - P)^T\right)\mathbf{n} = \lambda\mathbf{n} \quad \Rightarrow \quad C\mathbf{n} = \lambda\mathbf{n}$$

Estimación de normales

- Asumimos que tenemos un buen sampling de la superficie



- Normal $\mathbf{n}$ debe ser un vector propio de la matriz

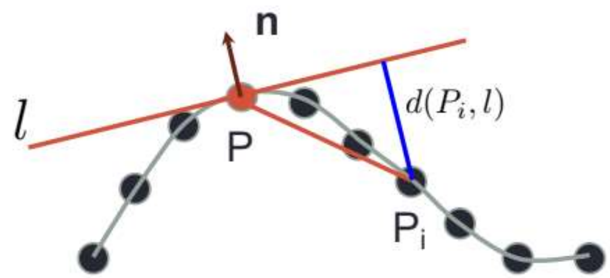
$$C\mathbf{n} = \lambda\mathbf{n} \qquad C = \sum_{i=1}^{k}(p_i - P)(p_i - P)^T$$

- Más aún

$$\mathbf{n}_{\text{opt}} = \arg\min_{\|\mathbf{n}\|=1} \sum_{i=1}^{k}((p_i - P)^T\mathbf{n})^2 = \arg\min_{\|\mathbf{n}\|=1} \mathbf{n}^T C\mathbf{n}$$

Estimación de normales

- Asumimos que tenemos un buen sampling de la superficie



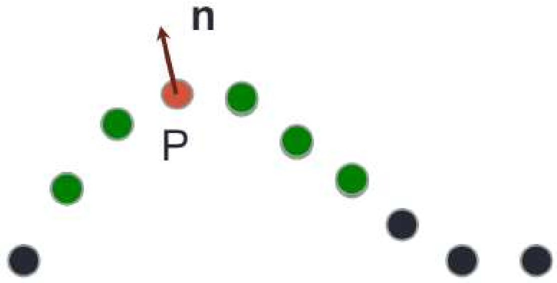
- Normal $\mathbf{n}$ debe ser un vector propio de la matriz

$$C\mathbf{n} = \lambda\mathbf{n} \qquad C = \sum_{i=1}^{k}(p_i - P)(p_i - P)^T$$

- La normal óptima debe ser el vector propio correspondiente al valor propio más pequeño en magnitud

Estimación de normales

- Método



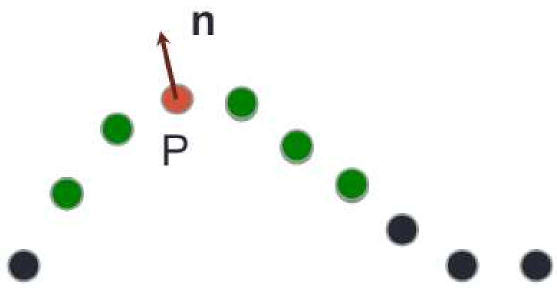
- Dado un punto P, encontrar sus k vecinos más cercanos
- Computar la matriz

$$C = \sum_{i=1}^{k} (p_i - P)(p_i - P)^T$$

- La normal $\mathbf{n}$ es el vector propio correspondiente al valor propio más pequeño

Estimación de normales

- Método

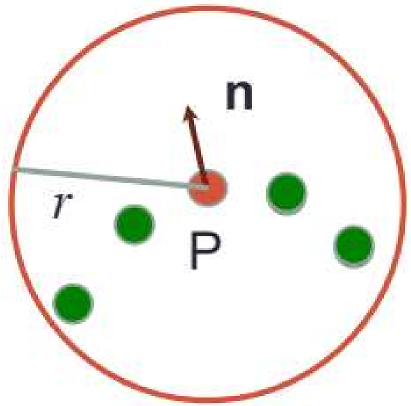


- Dado un punto P, encontrar sus k vecinos más cercanos
- Computar la matriz

$$C = \sum_{i=1}^{k} (p_i - \overline{P})(p_i - \overline{P})^T, \quad \overline{P} = \frac{1}{k} \sum_{i=1}^{k} p_i$$

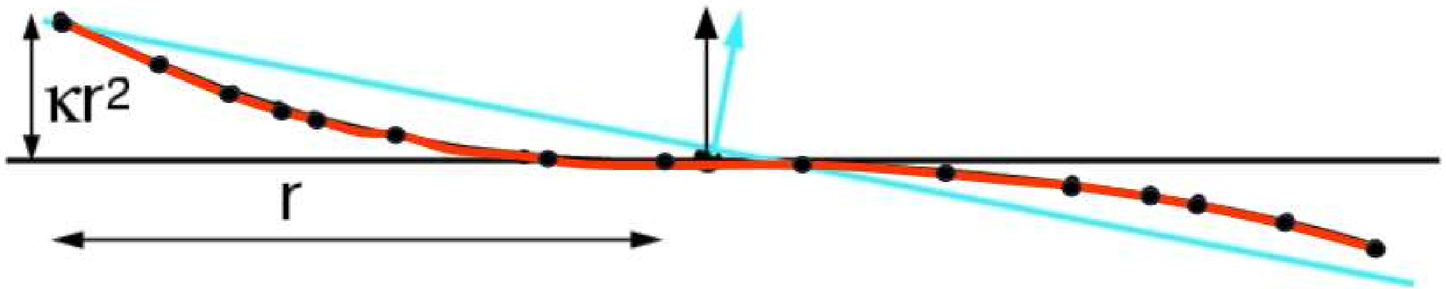
- La normal $\mathbf{n}$ es el vector propio correspondiente al valor propio más pequeño

Estimación de normales



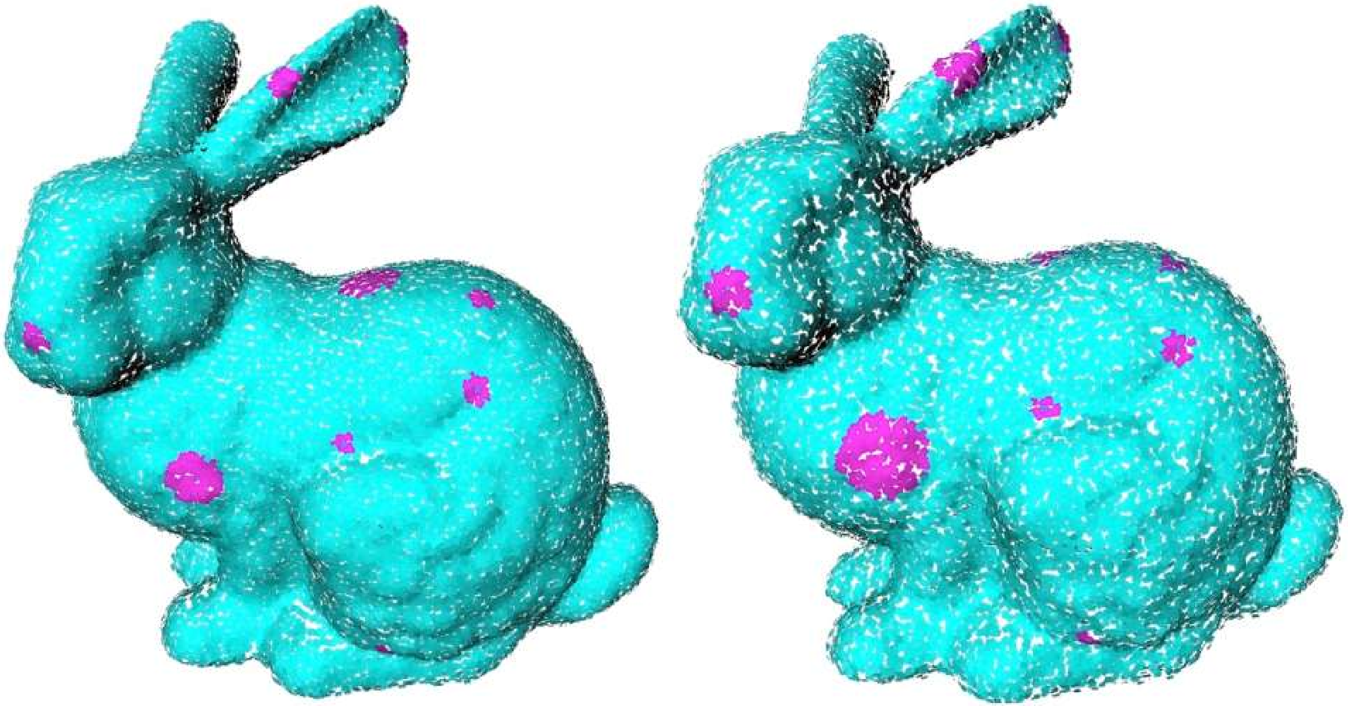
- Parámetro crítico k: Depende de la distribución de puntos
- Otra opción, usar un radio de búsqueda
- Qué radio usar?

Estimación de normales



- Efecto de la curvatura
  - Debido a la curvatura, valores de r grandes pueden llevarnos a un sesgo de estimación
  - Debido a ruido, valores pequeños de r pueden llevarnos a errores

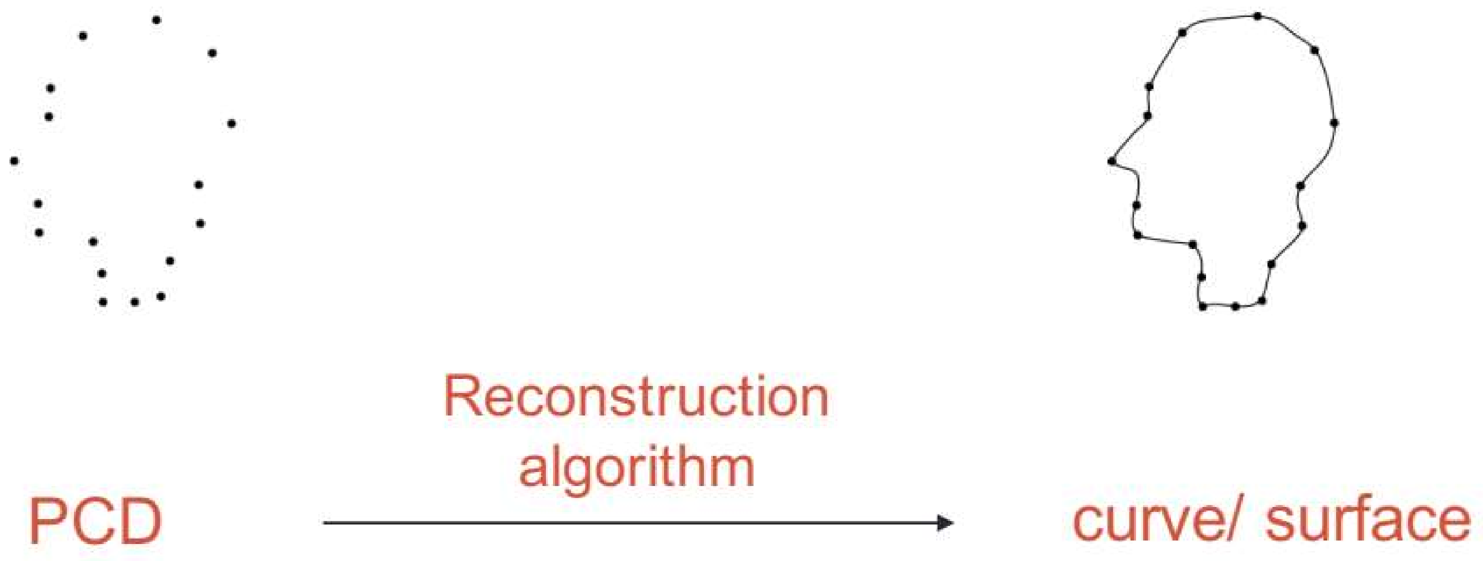
Estimación de normales



- Desafortunadamente, la curvatura no es conocida en la práctica. Difícil escoger tamaño óptimo.

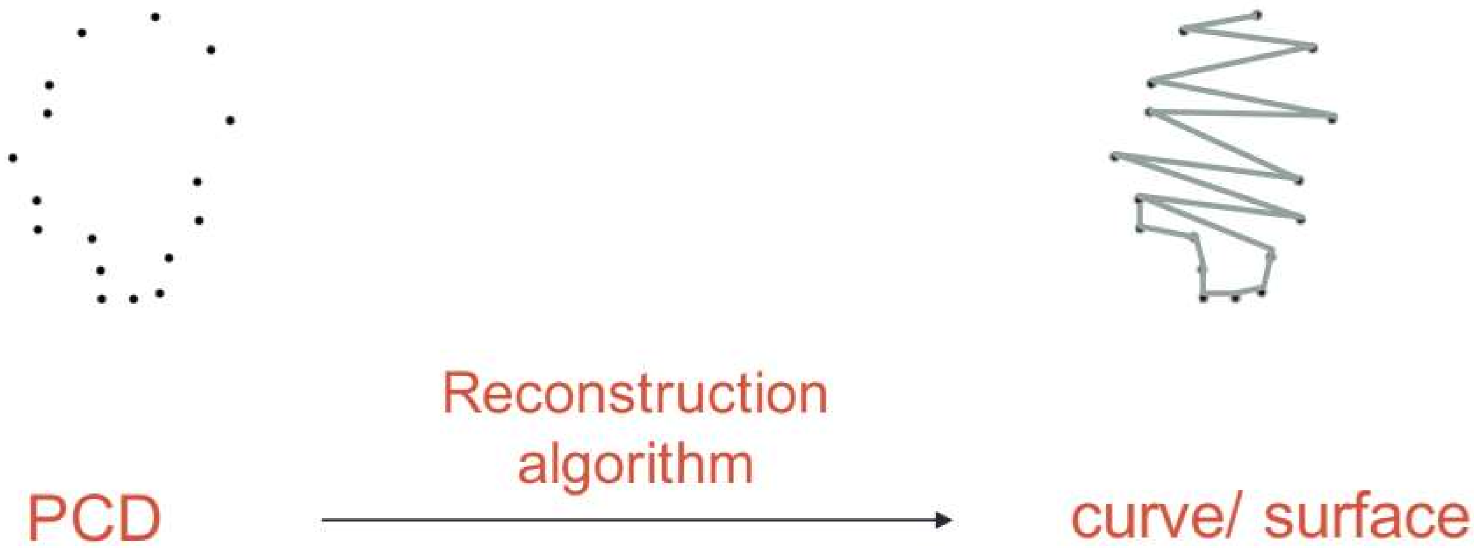
Reconstrucción de superficie

- Objetivo principal
  - Construir una representación poligonal de la nube de puntos



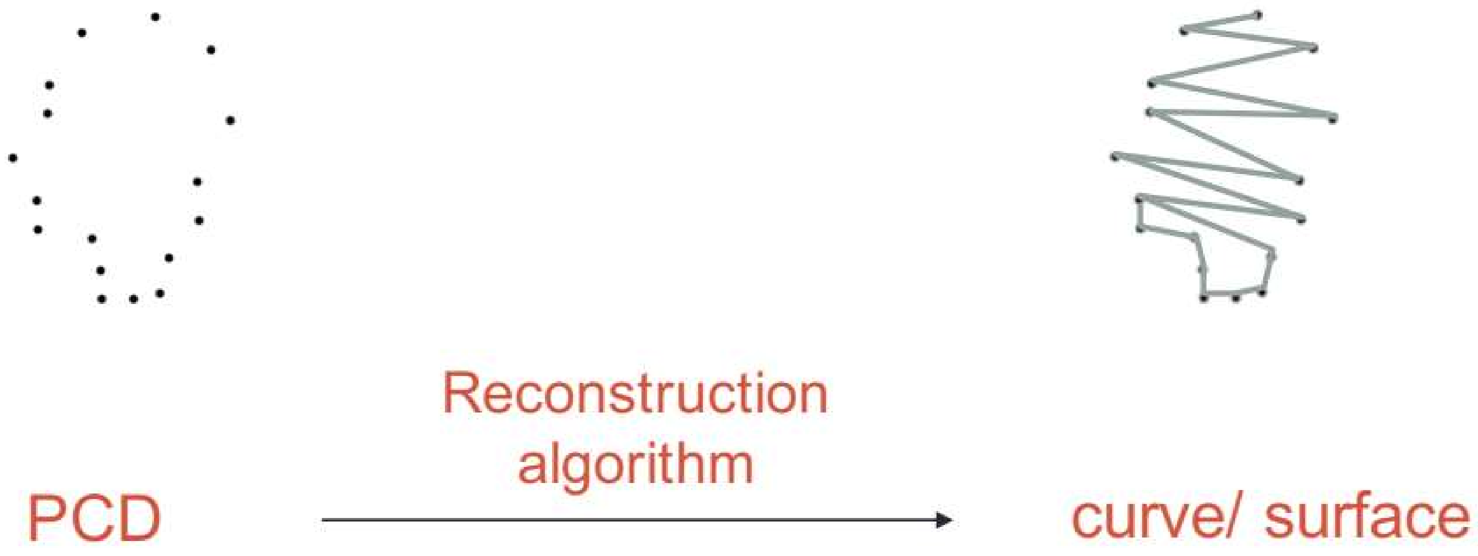
Reconstrucción de superficie

- Problema principal
  - Data es no estructurada. Los puntos no están ordenados.



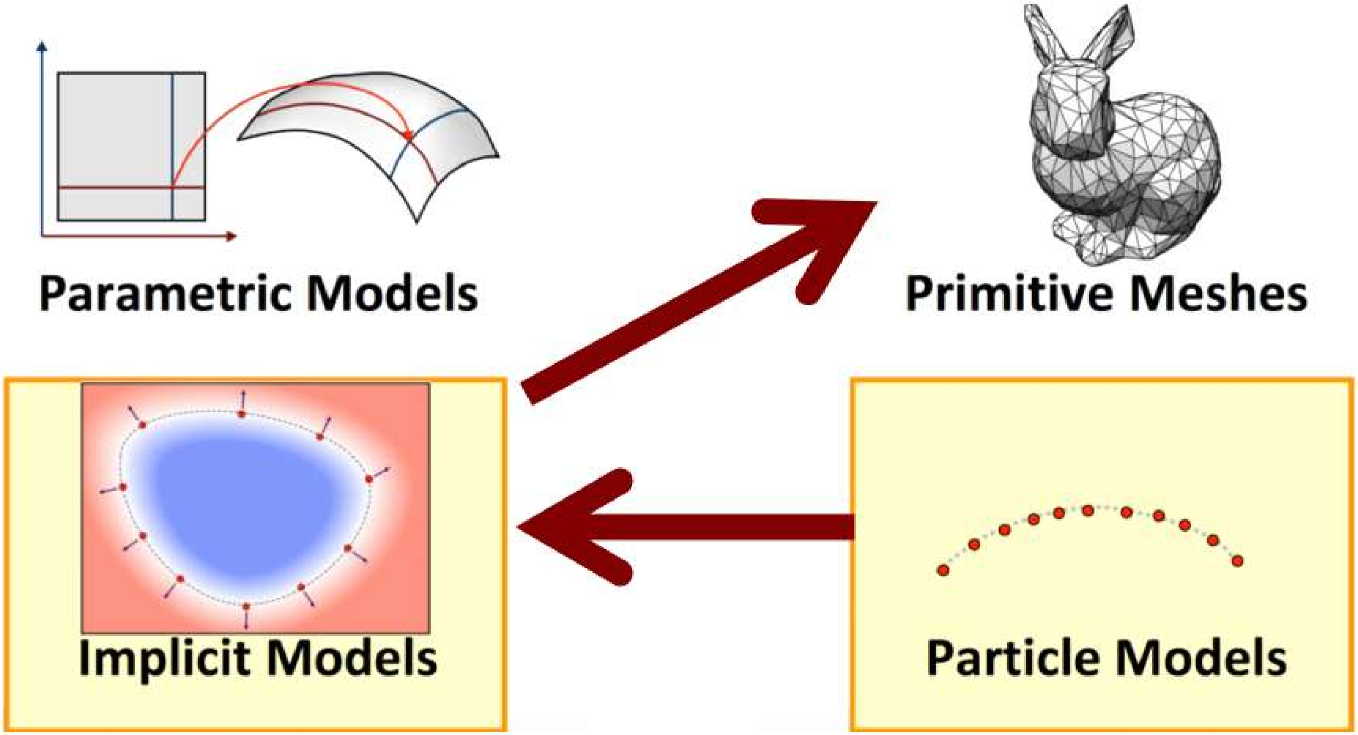
Reconstrucción de superficie

- Problema principal
  - Data es no estructurada. Los puntos no están ordenados.
  - Inherentemente es un problema mal condicionado.



Reconstrucción de superficie

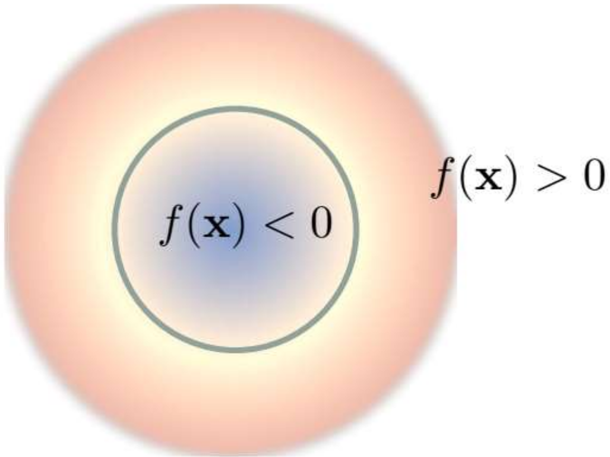
- Reconstrucción a través de modelos implícitos



Superficies implícitas

- Dada una función $f(x)$, la superficie es definida como

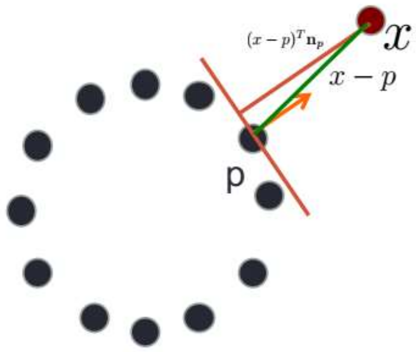
$$\{\mathbf{x}, \text{s.t.} f(\mathbf{x}) = 0\}$$



$$f(x, y) = x^2 + y^2 - r^2$$

Superficies Implícitas

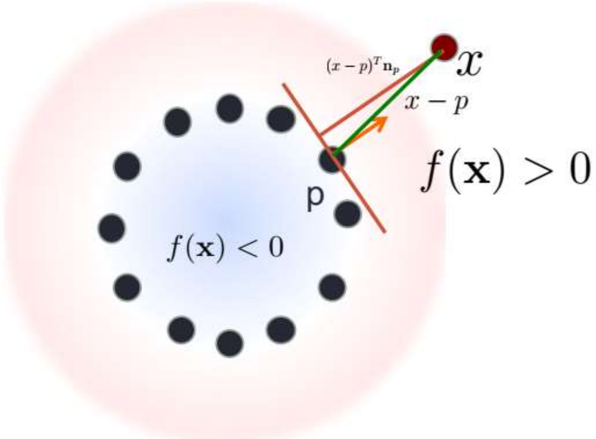
- Convertir desde una nube de puntos a una superficie implícita



- El método más simple:
  - Dado un punto x en el espacio, encontrar el punto más cercano
  - Hacer $f(x) = (x-p)^T\mathbf{n}_p$ es la distancia al plano tangente

Superficies Implícitas

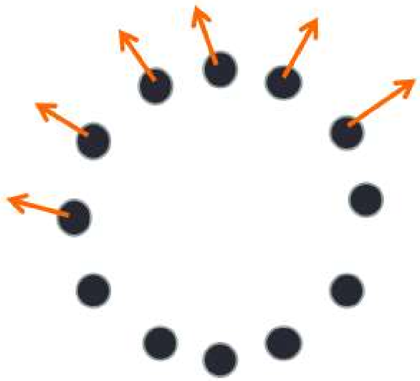
- Convertir desde una nube de puntos a una superficie implícita



- El método más simple:
  - Dado un punto x en el espacio, encontrar el punto más cercano
  - Hacer $f(x) = (x-p)^T\mathbf{n}_p$ es la distancia al plano tangente

Superficies Implícitas

- Convertir desde una nube de puntos a una superficie implícita

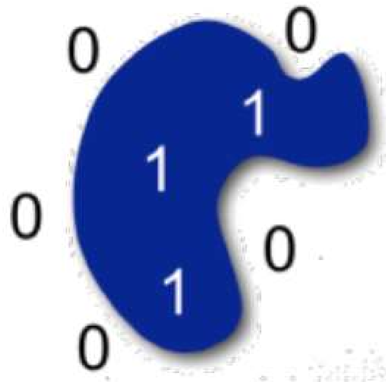


- El método más simple:
  - Dado un punto x en el espacio, encontrar el punto más cercano
  - Hacer $f(x) = (x - p)^T \mathbf{n}_p$ es la distancia al plano tangente
  - Normales tienen que ser consistentes

Reconstrucción de Poisson

- Objetivo principal: construir una función indicadora de la superficie

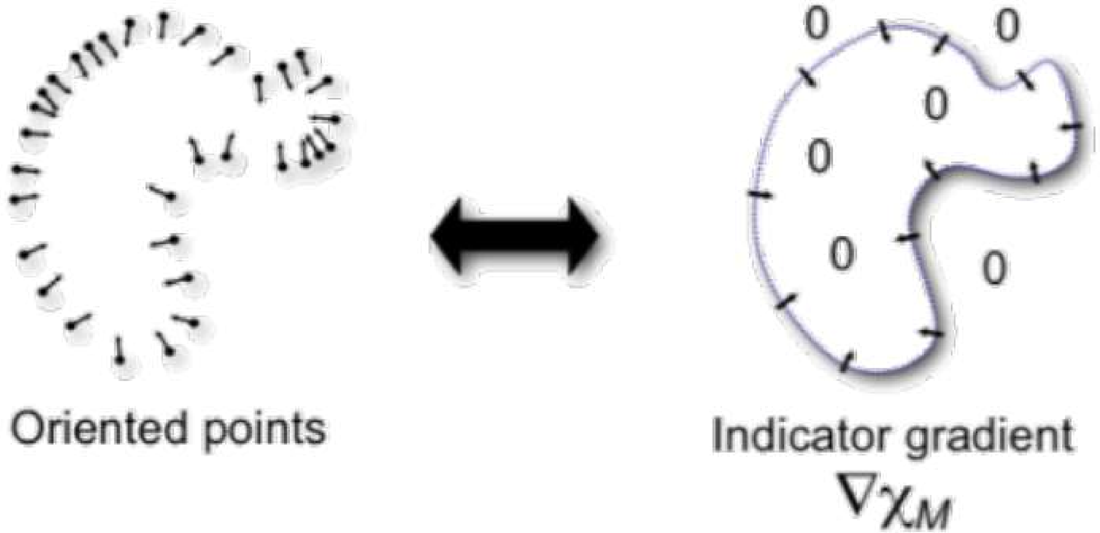
$$\chi_M(p)=\begin{cases}1 & \text{if } p\in M\\ 0 & \text{if } p\notin M\end{cases}$$



Indicator function
χ_M

Reconstrucción de Poisson

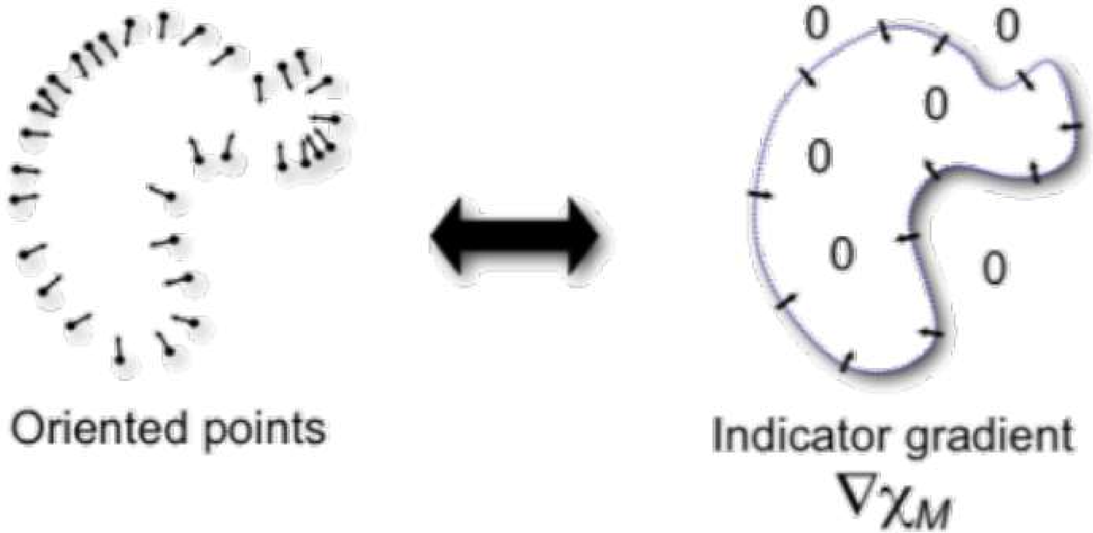
- Observación



- El gradiente de la función indicadora debería concordar con el campo vectorial de los puntos orientados

Reconstrucción de Poisson

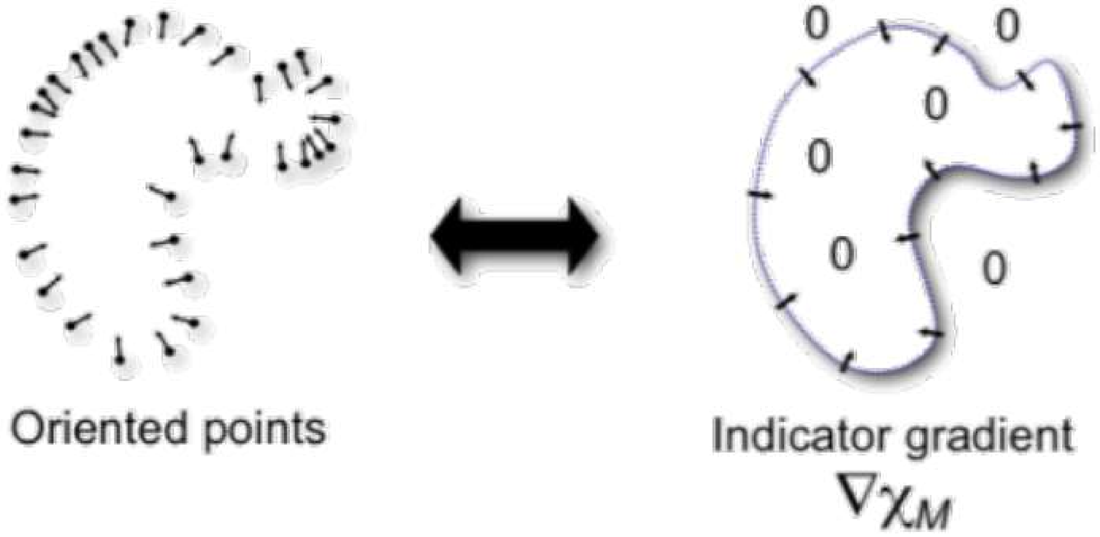
- Observación



$$\min_{\chi} \|\nabla\chi - V\|$$

Reconstrucción de Poisson

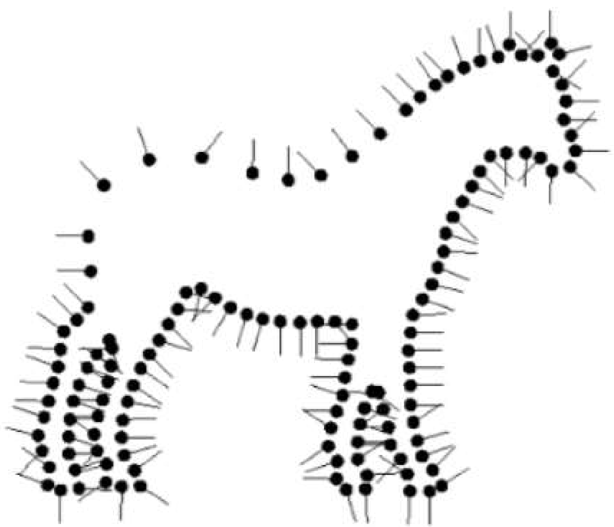
- Observación

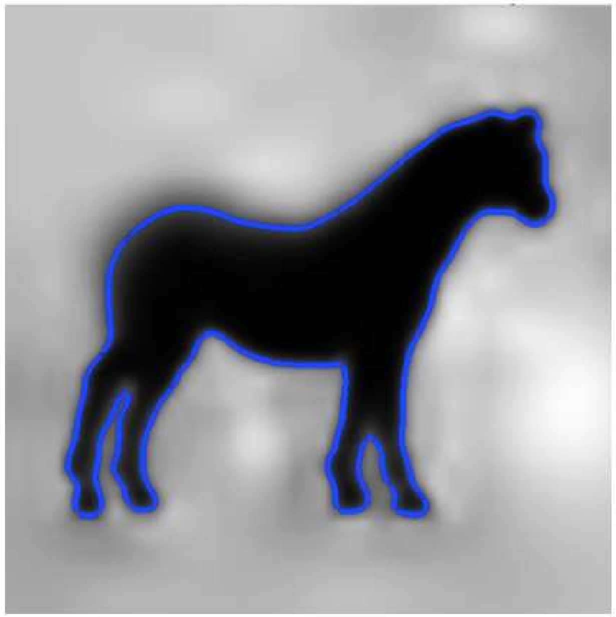


$$\min_{\chi} \|\Delta\chi - \mathrm{div}(V)\|$$

Reconstrucción de Poisson

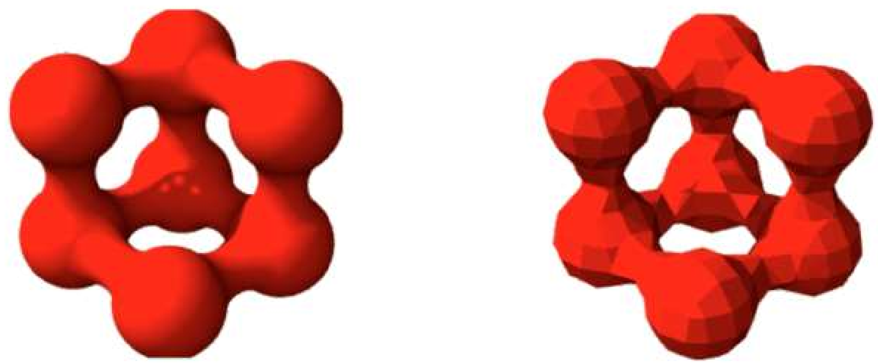
- Dada la función indicadora, se puede extraer la iso-superficie





Marching Cubes

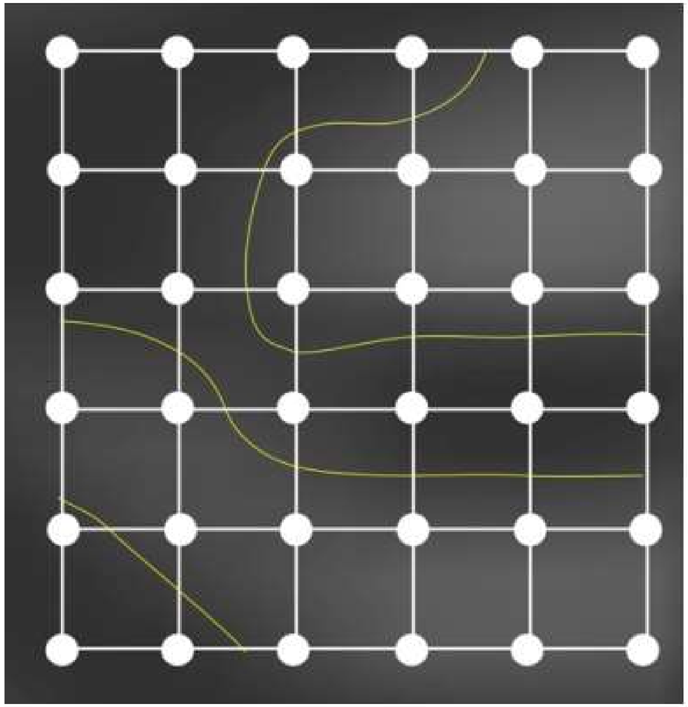
- Convertir una representación implícita en una explícita
- Objetivo: dada la representación implícita $\{\mathbf{x}, \text{s.t.} f(\mathbf{x}) = 0\}$
- Crear una malla triangular que aproxima la superficie



Marching squares

- Dada una función $f(x)$
  - $f(x) < 0$ dentro
  - $f(x) > 0$ fuera

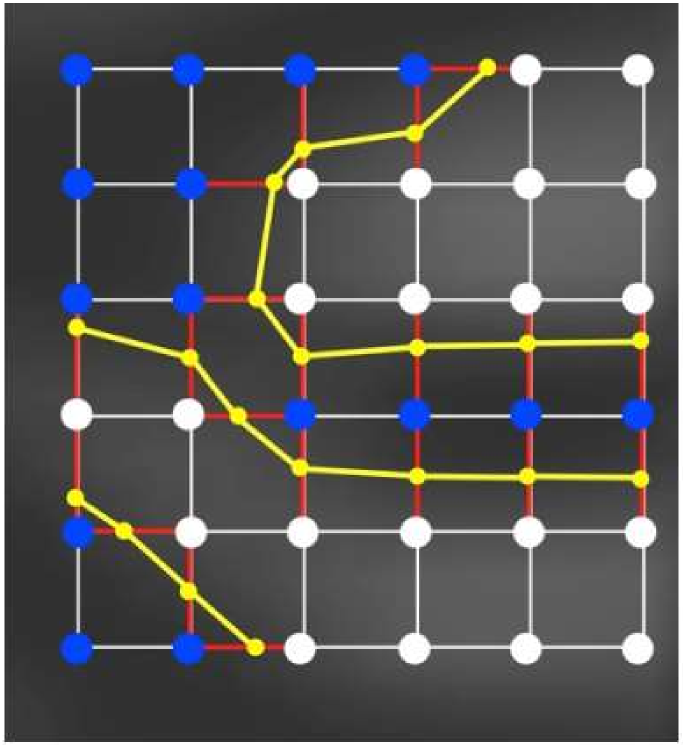
- Discretizar el espacio
- Evaluar $f(x)$ en la grilla



Marching squares

- Dada una función $f(x)$
  - $f(x) < 0$ dentro
  - $f(x) > 0$ fuera

- Discretizar el espacio
- Evaluar $f(x)$ en la grilla
- Clasificar puntos
- Clasificar aristas
- Computar intersecciones
- Conectar intersecciones



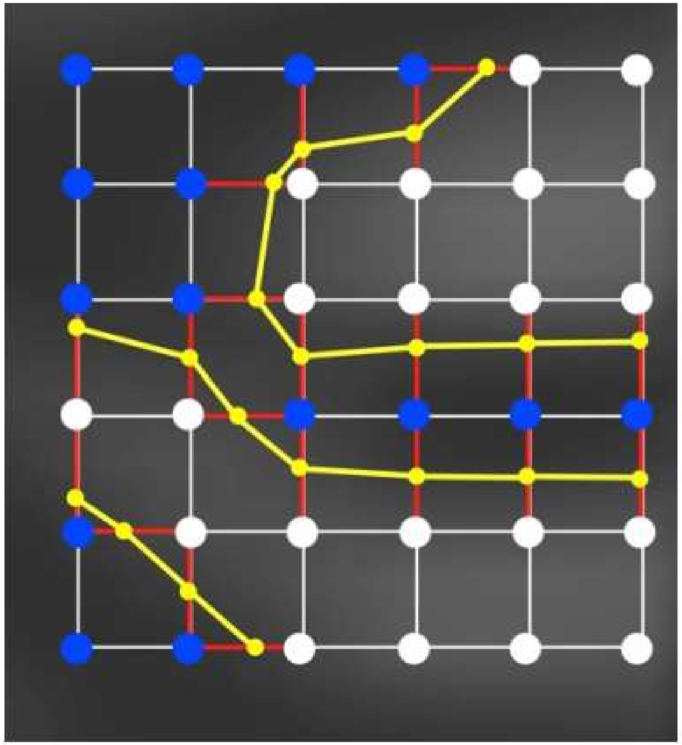
Marching squares

- Computar intersecciones
- Aristas con cambio de signo contienen
- Intersecciones

$$f(x_1) < 0, f(x_2) > 0 \Rightarrow$$
$$f(x_1 + t(x_2 - x_1)) = 0$$
$$\text{for some } 0 \le t \le 1$$

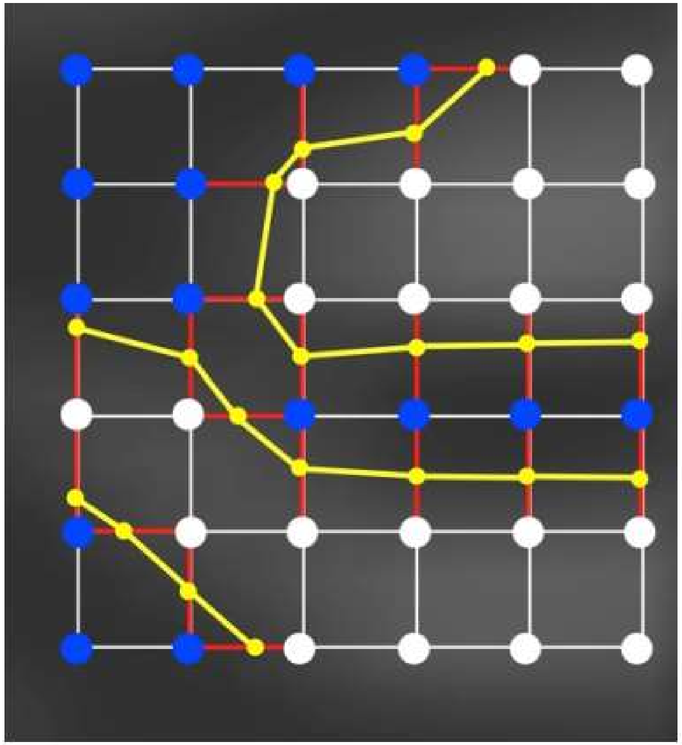
- Forma simple de computar t

$$t = \frac{f(x_1)}{f(x_2) - f(x_1)}$$



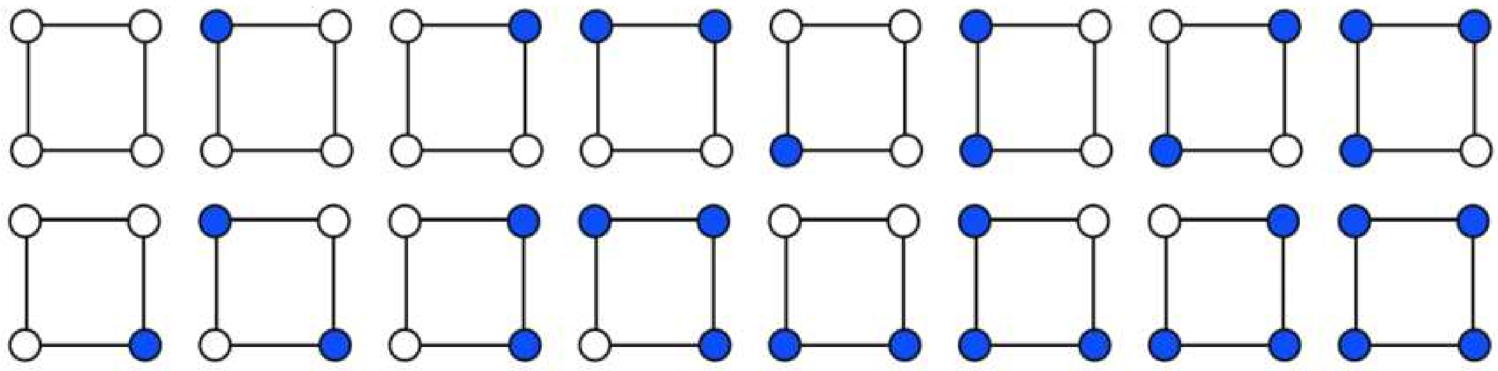
Marching squares

- Conectar intersecciones
- Tratar cada celda por separado
- Enumerar todas las posibles combinaciones entrada/salida



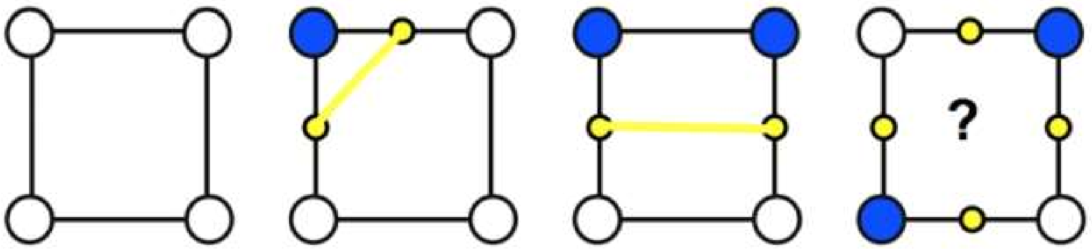
Marching squares

- Conectar intersecciones
- Tratar cada celda por separado
- Enumerar todas las posibles combinaciones entrada/salida
- Agrupar las combinaciones



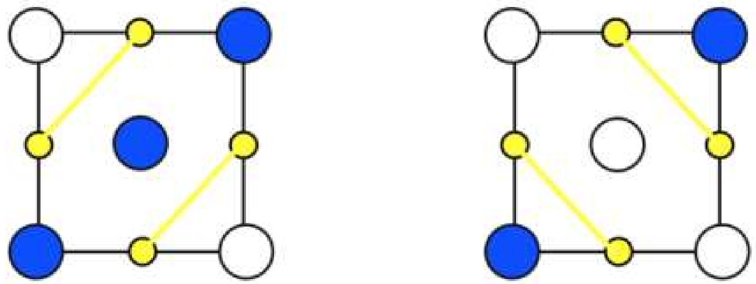
Marching squares

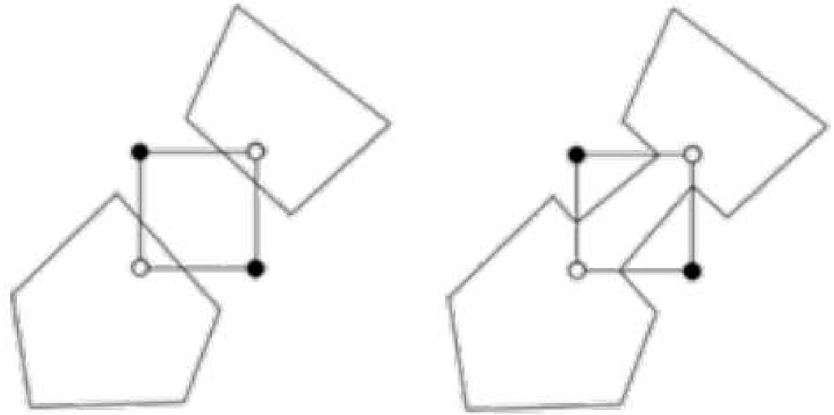
- Conectar intersecciones
- Tratar cada celda por separado
- Enumerar todas las posibles combinaciones entrada/salida
- Agrupar las combinaciones con equivalencias de rotación



Marching squares

- Conectar intersecciones
- Casos ambiguos

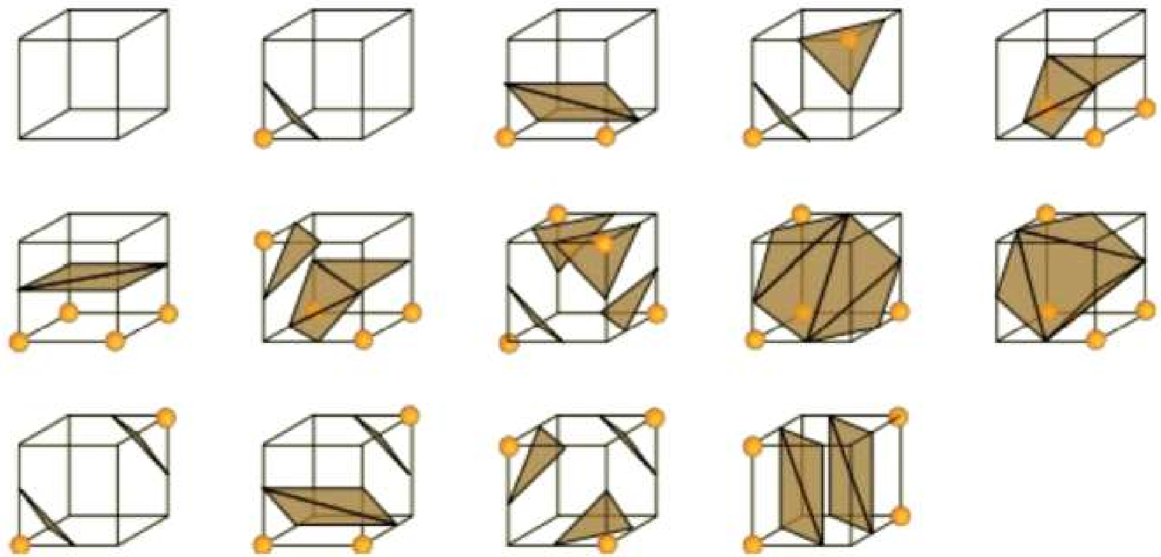




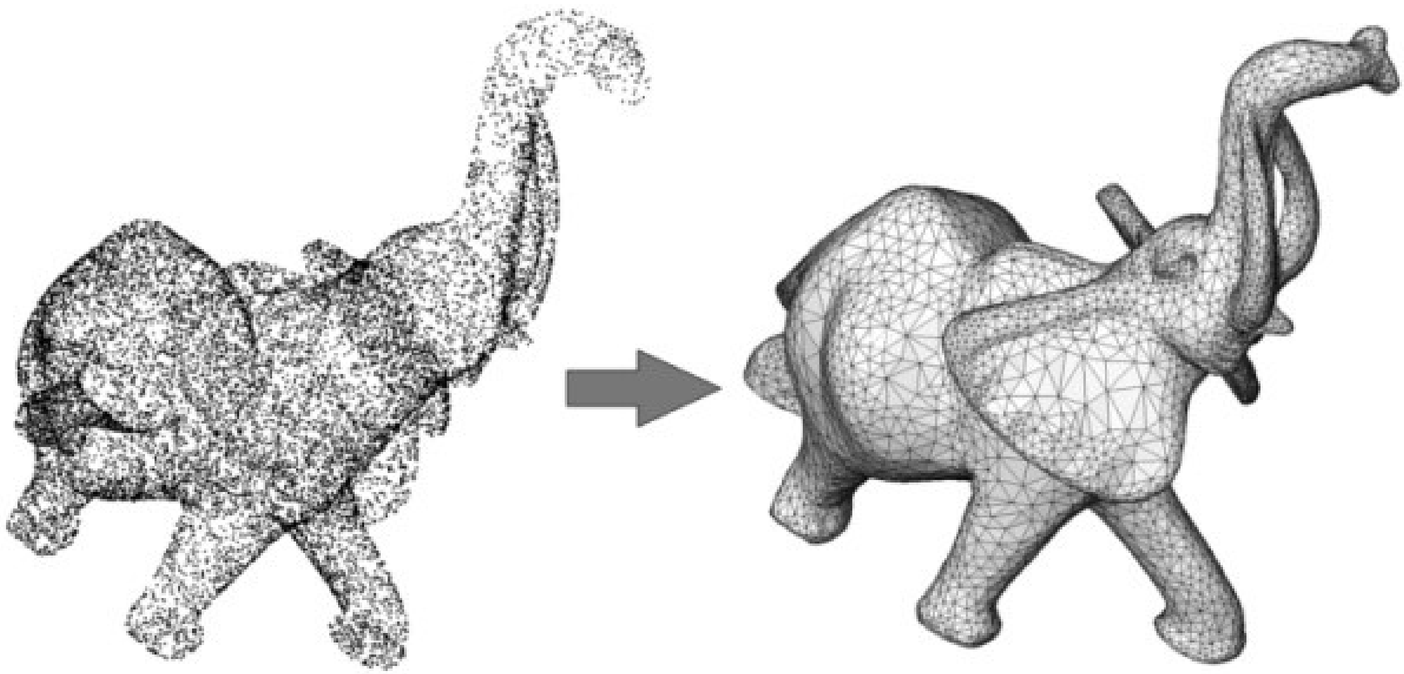
Break contour Join contour

Marching cubes (3D)

- Celdas son voxels
- Líneas de intersección se vuelven triángulos
- 256 casos diferentes
- 14 casos únicos (simetría)

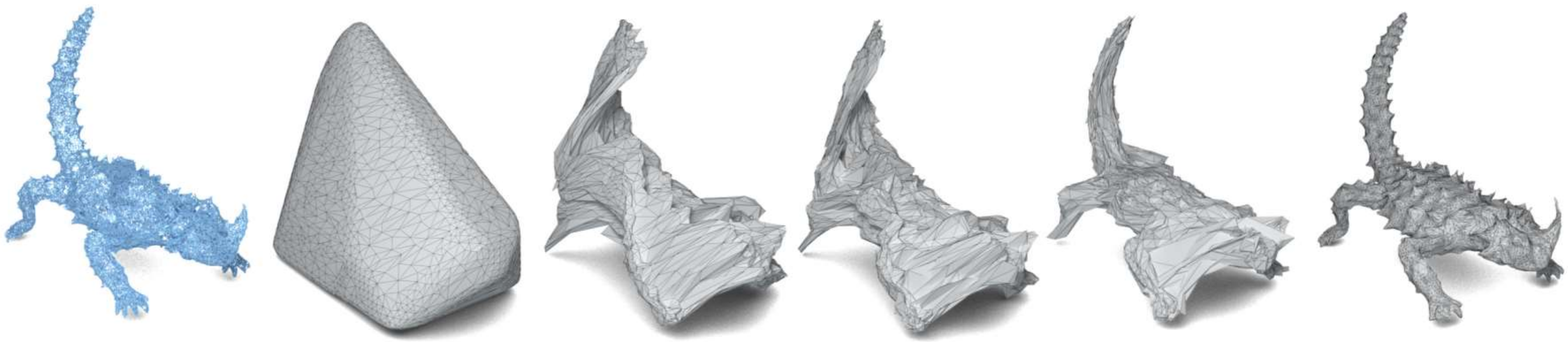


Resultados



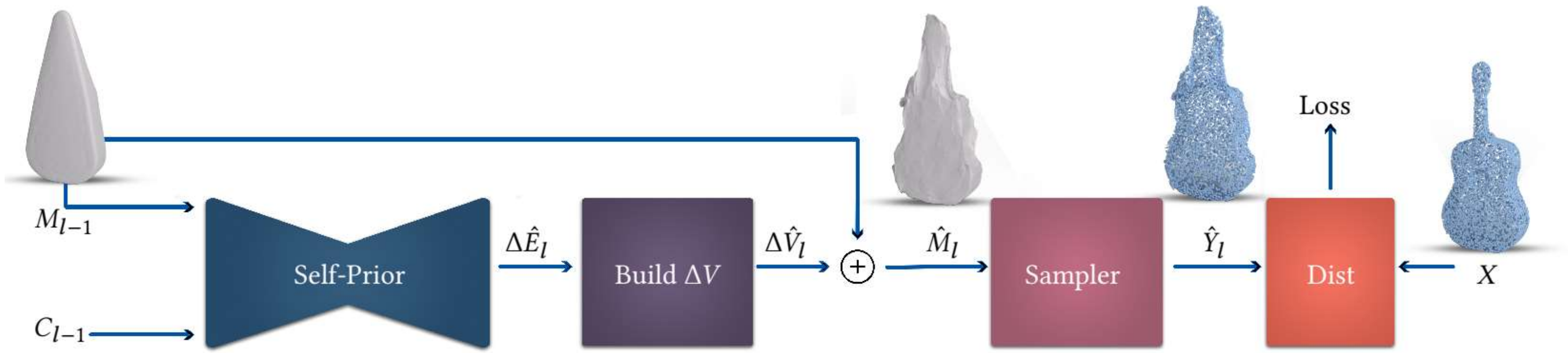
Point2Mesh

- Red neuronal para reconstrucción de superficies



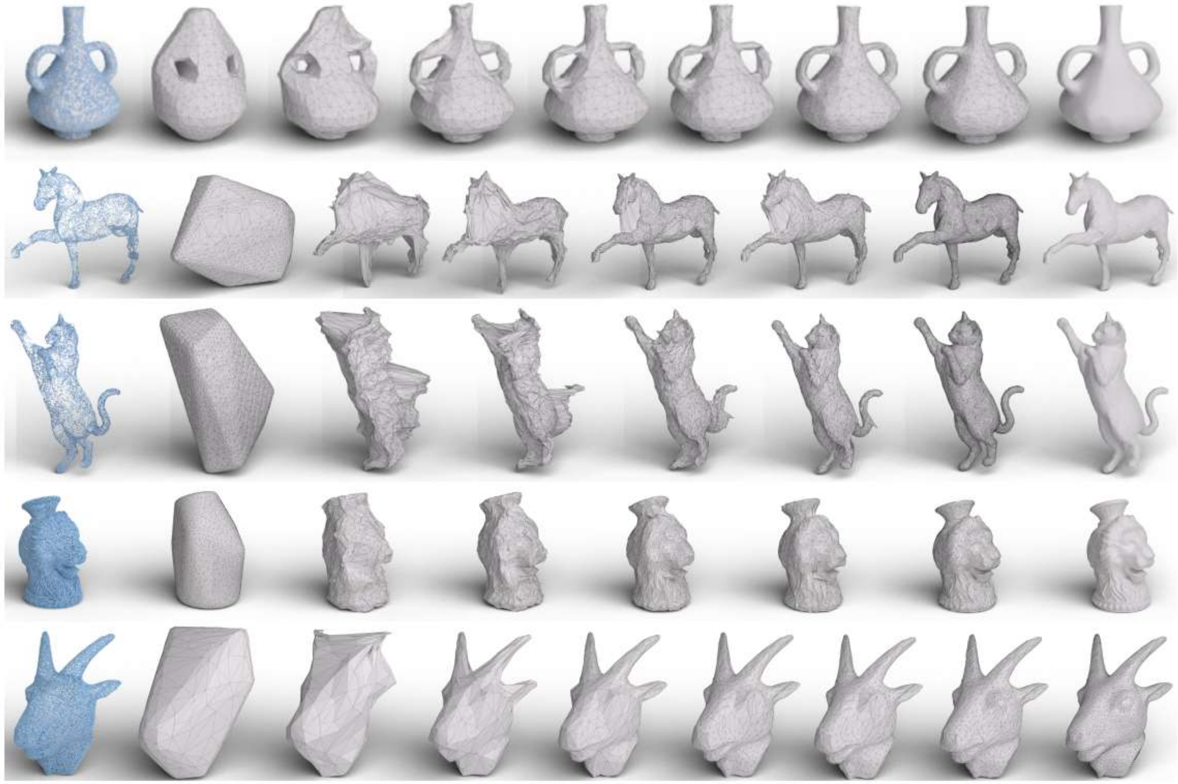
Point2Mesh

- Red neuronal para reconstrucción de superficies



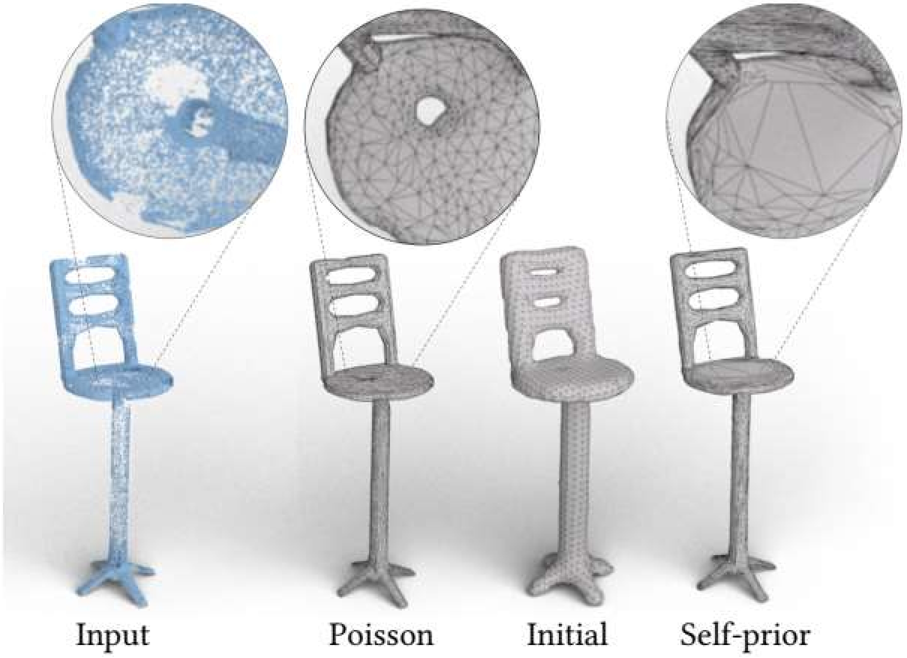
Point2Mesh

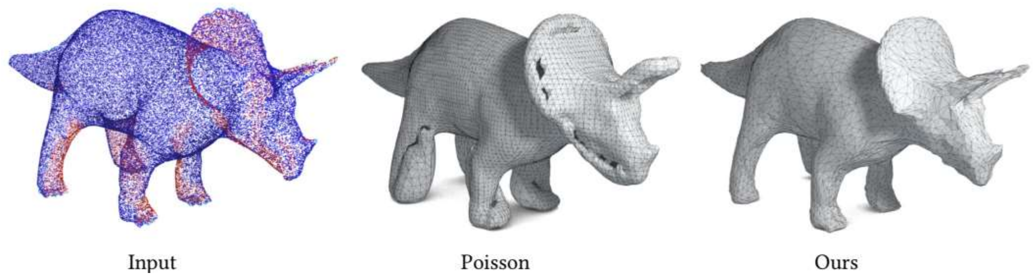
- Red neuronal para reconstrucción de superficies



Point2Mesh

- Red neuronal para reconstrucción de superficies





Point2Mesh

- Red neuronal para reconstrucción de superficies