

## T4. Reconstrucción 3D

# Visión 3D y Movimiento

*Master en “Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital”*



## Índice

- Introducción.
- Obtención de imágenes de rango:
  - Estereoscopia.
  - Luz estructurada.
  - Tiempo de vuelo.
- Técnicas de mado:
  - Mado triangular inicial.
  - Mado multiresolución.
  - Algoritmo de “desfile de cubos”.
- Registrado de imágenes de rango:
  - Algoritmo iterativo del punto más próximo.
  - Fusión de mados.
- Segmentación de superficies.

### ■ Objetivo:

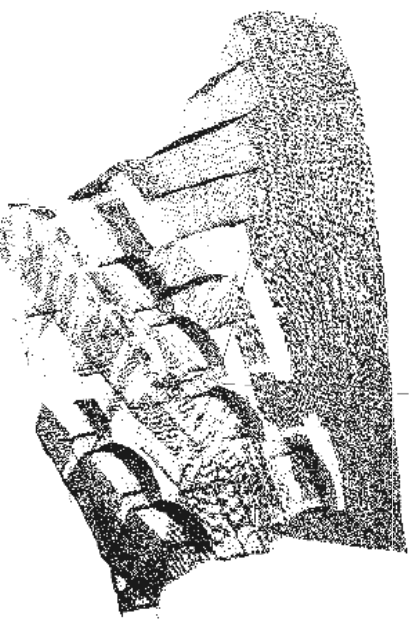
- Obtención de modelos 3D a partir de medidas.

### ■ Aplicaciones:

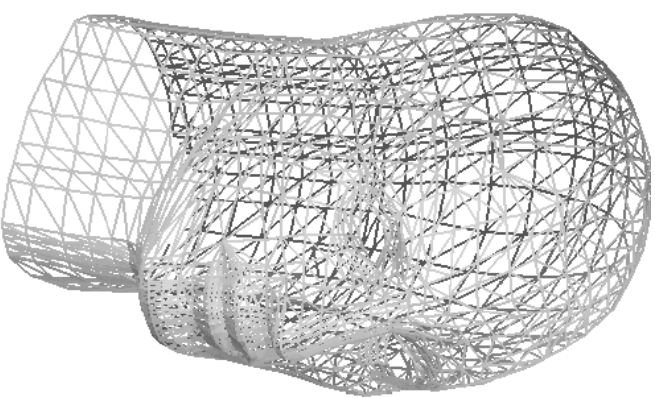
- Catálogos 3D de objetos.
- Arquitectura.
- Arqueología.
- Realidad virtual.
- Modelado del cuerpo humano.
- Fotos 3D.
- ...

### ■ Información sensorial:

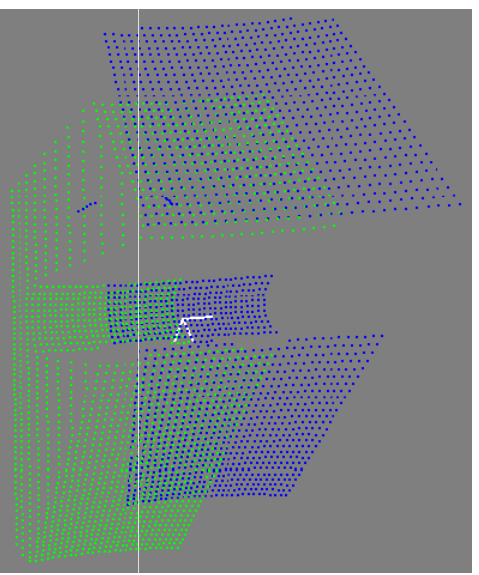
- Mapas densos de rango.



- **Representación:**
  - Mallado triangular:
    - ▶ Se puede añadir “textura”.
  - ▶ Representación de superficies curvas.
- **Modelos completos:**
  - Extensión de modelo:
    - ▶ Registrado de imágenes de rango.
    - ▶ Registrado de mallados.



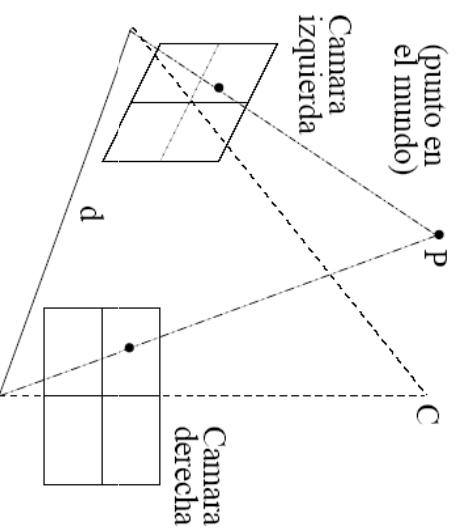
- **Segmentación:**
  - Clasificar imagen de rango en primitivas geométricas (planas, cilíndricas, etc.).
- **Razonamiento:**
  - Sobre una representación del modelo 3D:
    - ▶ Identificación de partes de un modelo.
    - ▶ Mejora de la calidad del modelo.
    - ▶ Extensión del modelo: dónde tomar la siguiente imagen.



- Introducción.
- Obtención de imágenes de rango:
  - Estereoscopia.
  - Luz estructurada.
  - Tiempo de vuelo.
- Técnicas de mado:
  - Mado triangular inicial.
  - Mado multiresolución.
- Algoritmo de los “cubos marchantes”.
- Registrado de imágenes de rango:
  - Algoritmo iterativo del punto más próximo.
  - Fusión de mados.
- Segmentación de superficies.

## Obtención de imágenes de rango

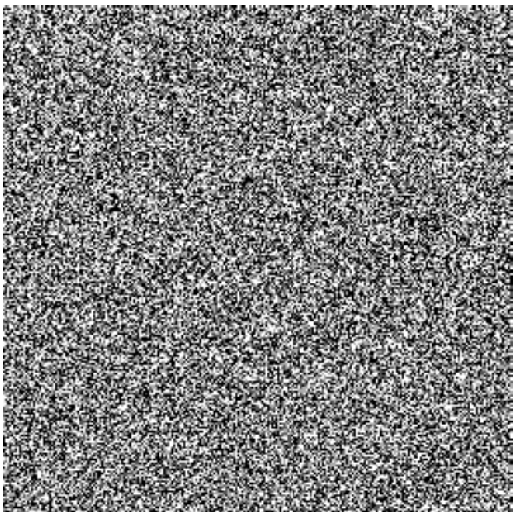
- **Estereoscopia:**
- **Información 3D:**
  - Puntos correspondientes.
  - Parámetros de calibración
- **Correspondencia** basada en correlación:
  - Problemas en objetos textura uniforme.



- **Proyectar patrón** que añadida textura a la escena:
  - Patrones aleatorios de puntos.
  - Patrones de color.

## Estereoscopia

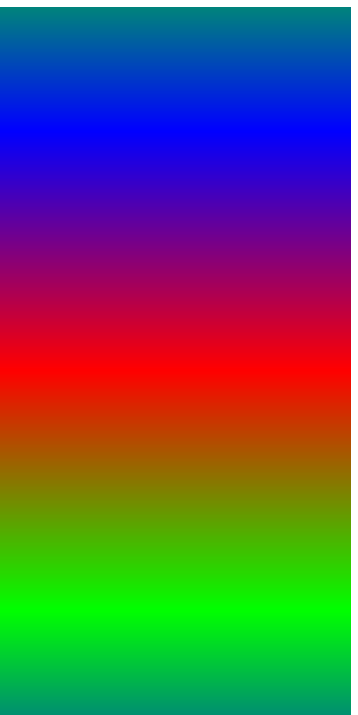
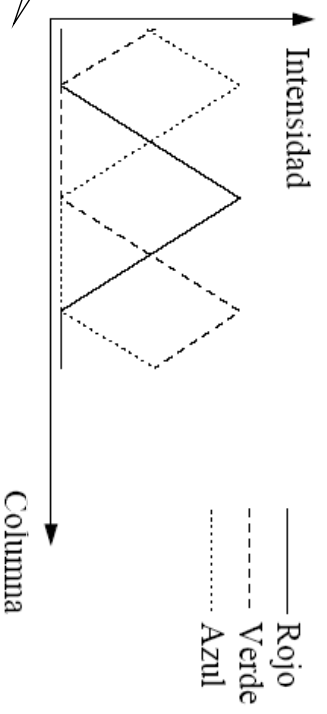
- Patrón aleatorio:
  - Par estéreo:
    - Imágenes de grises.
  - Información “textura” (color):
    - Cámara color adicional.
  - Procedimiento:
    - Se toma imagen en color.
    - Se proyecta el patrón.
    - Se toma par estéreo.



## Estereoscopia

### ■ Patrones de color

Una componente siempre es cero, y la intensidad del patrón es siempre uniforme.



- **Patrones de color:**
  - **Calibración patrón una cámara:**
    - ▶ Para cada punto:
      - Poner a cero componente más baja.
      - Escalar las otras dos de forma que la suma de el valor máximo.
  - **Calibración entre dos cámaras:**
    - ▶ Al comparar dos puntos en ambas imágenes:
      - La componente mayor que valga lo mismo, escalando el punto derecho:

$$\begin{pmatrix} R_l \\ G_l \\ B_l \end{pmatrix} = s \begin{pmatrix} R_r \\ G_r \\ B_r \end{pmatrix}$$

- **Correlación:**

El patrón es uniforme en dirección vertical (entre líneas).

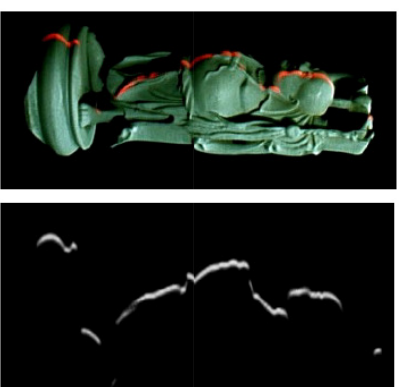
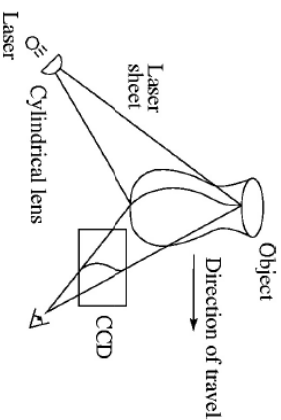
$$Q(x_l, x_r) = \sum_{i=-N}^N \left[ \left( R_l(x_l + i) - s_i R_r(x_r + i) \right)^2 + \left( G_l(x_l + i) - s_i G_r(x_r + i) \right)^2 + \left( B_l(x_l + i) - s_i B_r(x_r + i) \right)^2 \right]$$

- **Ventajas métodos estéreo:**
  - Material de bajo coste.
  - Captura de datos rápida.
  - Fácil de calibrar.
  - No hay partes móviles.
- **Inconvenientes:**
  - Poca precisión.
  - Campos visuales muy limitados.



## Luz estructurada

- Una cámara + un proyector.
- Fuente de luz y cámara calibradas.

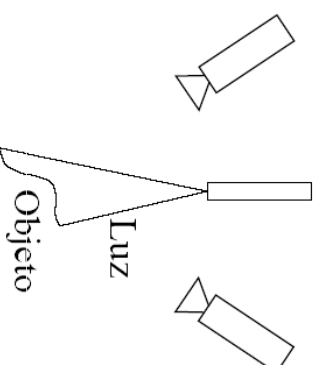


### ■ Procedimiento:

- Se explora la imagen línea a línea.
- Se detecta el centro del haz láser.
- Se consultan tablas de coordenadas 3D del punto.

## Luz estructurada

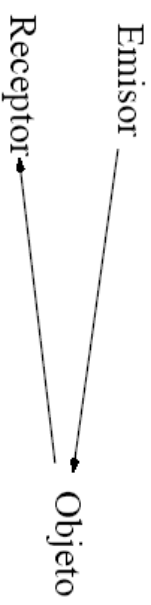
- **Uso de dos cámaras:**
  - Promedio de medidas.
  - Detectar perfiles ocultos.



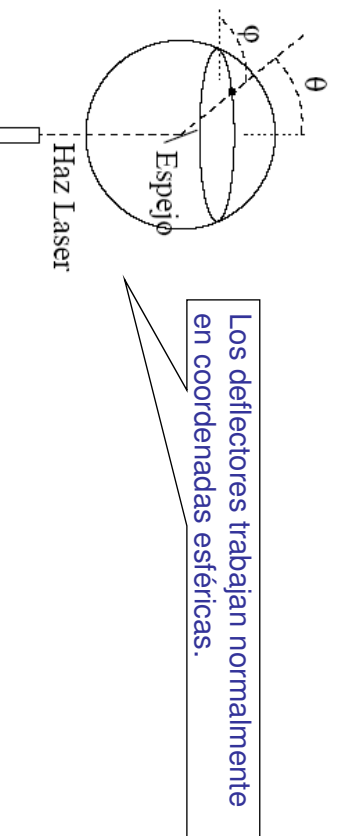
- Una línea de rango.
- Dispositivo para **mover** objeto o cámara-proyector.
- Parar referir todos los puntos al mismo sistema de coordenadas hay que **conocer la transformación 3D** entre cada posición.

## Tiempo de vuelo

- **Tiempo de vuelo de la luz:**
  - Sistema de medición láser.
  - Sistema de deflexión.



- **Deflectar el rayo láser por la escena:**



## Tiempo de vuelo

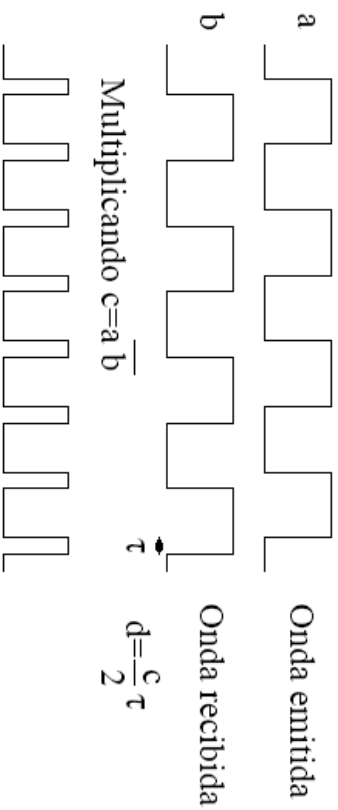
- **Características** sensores de rango láser:
  - Buena precisión.
  - Toma muchas medidas de la misma distancia:
    - ▶ Menor ruido, mayor precisión.
  - Buena resolución espacial:
    - ▶ Baja divergencia angular del haz.
  - Proporciona medida de reflectancia del objeto.



## Tiempo de vuelo

### ■ Técnica:

- **Onda pulsante** para medida del tiempo de retardo entre señal emitida y recibida.



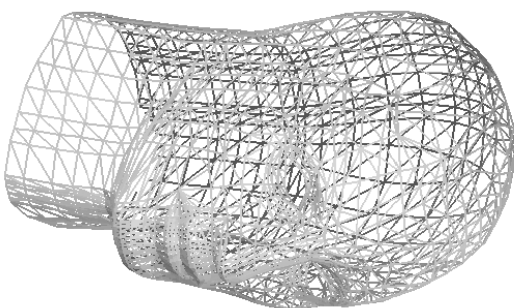
## Índice

- Introducción.
- Obtención de imágenes de rango:
  - Estereoscopia.
  - Luz estructurada.
  - Tiempo de vuelo.
- Técnicas de mallado:
  - Mallado triangular inicial.
  - Mallado multiresolución.
  - Algoritmo de “desfile cubos”.
- Registrado de imágenes de rango:
  - Algoritmo iterativo del punto más próximo.
  - Fusión de mallados.
- Segmentación de superficies.

## Técnicas de mallado

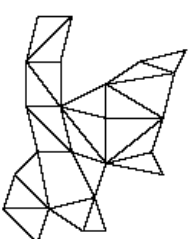
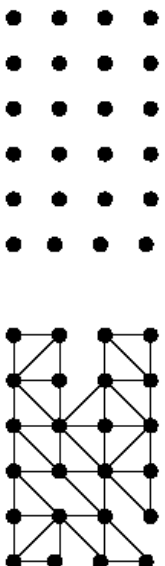
### ■ Mallado triangular:

- De nube de puntos a superficies por triángulos.
- Es conveniente para visualizar o razonar sobre las superficies.

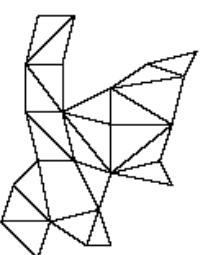
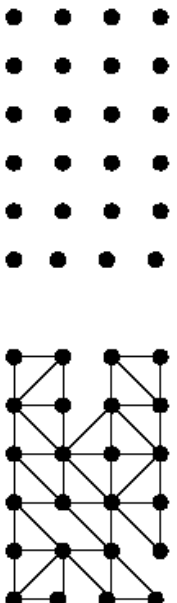


### ■ Mallado triangular inicial:

- Enlazan puntos vecinos en la imagen de rango.



## Mallado triangular



### ■ Triangulación con restricción de continuidad:

- Por cada 4 puntos vecinos:

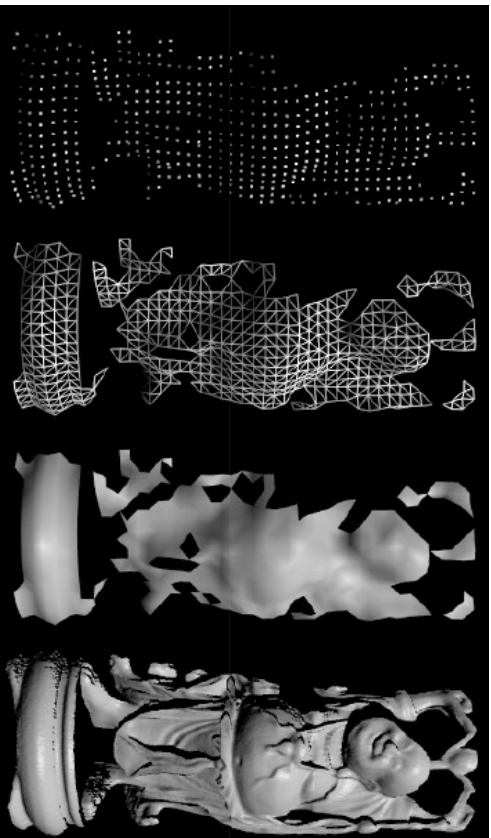
- ▶ Se unen los puntos vecinos cuya distancia es menor que un umbral.
- ▶ Se obtienen 0, 1 o 2 triángulos.

- ▶ Se consideran todos los posibles triángulos:

- Si se pueden obtener dos pares diferentes, se elige el par de suma de longitudes de aristas más pequeño.

Umbral preserva discontinuidades en la superficie.

## Mallado triangular



Rendering

Rendering a partir de  
imagen de rango de  
alta resolución.

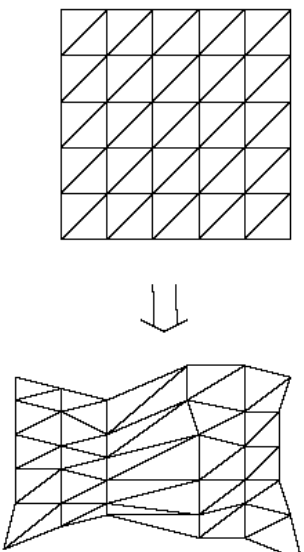
## Mallado multiresolución

- Preservar características importantes del modelo con **un menor número** de triángulos.
- Reducción del número de datos.
- **Conservar discontinuidades.**
- **Técnica:**
  - Partir de triángulos grandes.
  - Comprobar si:
    - ▶ Son válidos.
    - ▶ Se subdividen.
    - ▶ Se rechazan.

## Mallado multiresolución

### ■ Algoritmo:

- Dividir la imagen en bloques  $N \times N$  (ej.  $64 \times 64$ ).
- Dividir cada cuadro en dos triángulos.
- Identificar triángulos correspondientes a discontinuidades (alargados) y rechazarlos:

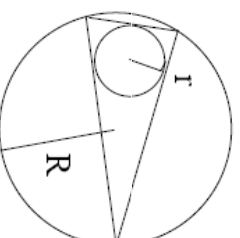


## Mallado multiresolución

### ■ Algoritmo (continuación):

- ▶ Triángulos válidos deben cumplir:
  - Realizar test de relación de aspecto

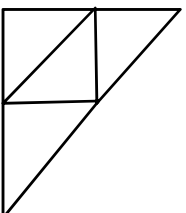
$$t = r/R > \tau$$



- Distancia media entre los puntos 3D y el plano del triángulo sea menor que un valor.

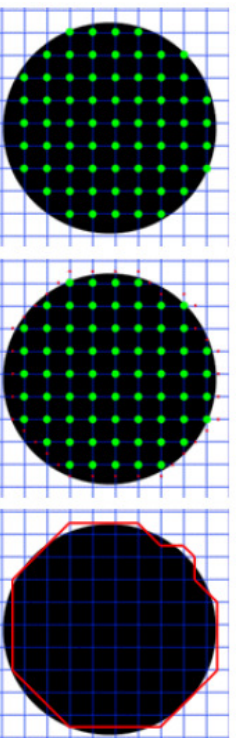
## Mallado multiresolución

- Algoritmo (continuación):
  - ▶ Si se cumplen **ambas condiciones**, el triángulo es **válido**.
  - ▶ Sino, el proceso se repite dividiendo cada triángulo en **cuatro triángulos**, y el proceso se repite.



## “Desfile de cubos”

- Algoritmo “**desfile de cubos**”:
  - Información de superficie a partir de **volumen**.
  - Utiliza campo escalar tridimensional
  - Mallado solo puede sacar información 2D de las superficies.
- **Principio**:
  - Dividir el espacio en cubos (**vóxeles**).
  - **Identificar cubos** del y **sustituirlos por triángulos**.



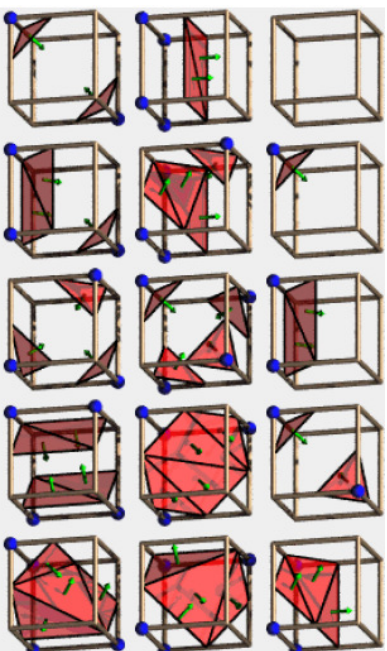
## “Desfile de cubos”

### ■ Caso 3D:

- Cubos: 8 vértices.
- Cada vértice está dentro o fuera del objeto.
- Codificar vértices en los bits de un byte.
- Asignar a cada combinación una configuración posible de triángulos que aproximen la superficie:

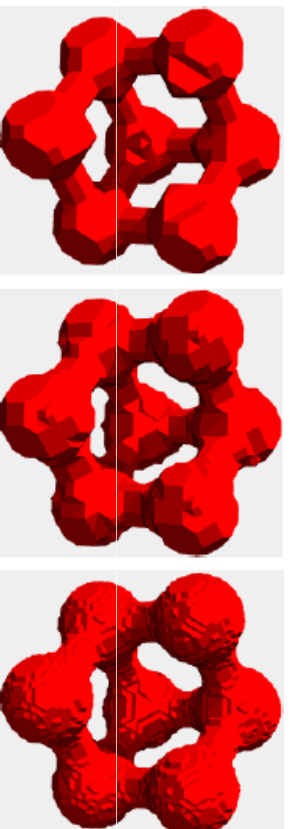
Solo hay 15 posibilidades originales.

Se identifican combinaciones a partir de tablas pre-calculadas.



## “Desfile de cubos”

### ■ Ejemplo:



12 subdivisiones    20 subdivisiones    50 subdivisiones



- Introducción.
- Obtención de imágenes de rango:
  - Estereoscopia.
  - Luz estructurada.
  - Tiempo de vuelo.
- Técnicas de mallado:
  - Mallado triangular inicial.
  - Mallado multiresolución.
  - Algoritmo de “desfile cubos”.
- Registrado de imágenes de rango:
  - Algoritmo iterativo del punto más próximo.
  - Fusión de mallados.
- Segmentación de superficies.

## Registrado de imágenes de rango

- **Problema:**
  - Alineamiento de una imagen de rango con un modelo 3D.
- **Objetivo:**
  - Obtener la **transformación 3D**, rotación y traslación (6 parámetros), que alinean mejor la imagen y el modelo.
- **Aplicaciones:**
  - Identificar parte del modelo que corresponde la imagen.
  - Conocer posición del sensor.
  - Incrementar precisión o extender el modelo.

## Registrado de imágenes de rango

- Algoritmo iterativo del punto más próximo:
  - Independiente de la representación del modelo.
  - No requiere preproceso de la imagen.
  - Coste cuadrático.
  - No descarta grandes errores en los datos.
  - Converge a un mínimo local.
  - Seis grados de libertad:
    - ▶ 3 parámetros de rotación + 3 de traslación.

## Registrado de imágenes de rango

- Conceptos iniciales:
  - Distancias:
  - ▶ Punto a punto:

$$d(\vec{r}_1, \vec{r}_2) = \|\vec{r}_1 - \vec{r}_2\|$$

- ▶ Punto a segmento de línea:

$$d(\vec{p}, l) = \min_{u+v=1} \|u\vec{r}_1 + v\vec{r}_2 - \vec{p}\|$$

$$u \in [0, 1], v \in [0, 1]$$

- ▶ Punto a triángulo:

$$d(\vec{p}, t) = \min_{u+v+w=1} \|u\vec{r}_1 + v\vec{r}_2 + w\vec{r}_3 - \vec{p}\|$$

$$u \in [0, 1], v \in [0, 1], w \in [0, 1]$$

## Registrado de imágenes de rango

- Conceptos iniciales:

- Transformación 3D:
  - Rotación + traslación

$$\vec{q}_R = [q_0, q_1, q_2, q_3]^t, \quad q_i \geq 0.$$

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

$$R(\vec{q}_R) \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{pmatrix}$$

$$\vec{q}_T = [q_4, q_5, q_6]^t$$

$$\vec{q} = [\vec{q}_R \mid \vec{q}^T]^t$$

## Registrado de imágenes de rango

- Conceptos iniciales:

- Función criterio:

$$P = \{\vec{p}_i\} \quad \text{Puntos de la imagen de rango}$$

$$X = \{\vec{x}_i\} \quad \text{Puntos del modelo}$$

$$N_p = N_x \quad \text{Puntos correspondientes en cada conjunto}$$

Error cuadrático medio de la transformación

$$d = f(\vec{q}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|\vec{x}_i - R(\vec{q}_R)\vec{p}_i - \vec{q}_T\|^2$$

$$\vec{\mu}_p = \frac{1}{N_p} \sum_{i=1}^{N_p} \vec{p}_i \quad \vec{\mu}_x = \frac{1}{N_x} \sum_{i=1}^{N_x} \vec{x}_i$$

## Registrado de imágenes de rango

- Cálculo parámetros:

Matriz de covarianza cruzada

$$\Sigma_{px} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left( (\vec{p}_i - \vec{\mu}_p)(\vec{x}_i - \vec{\mu}_x)^t \right)$$

$$A = (\Sigma_{px} - \Sigma_{px}^t) \quad \Delta = [A_{23}, A_{31}, A_{32}]^t$$

$$Q(\Sigma_{px}) = \begin{pmatrix} tr(\Sigma_{px}) & \Delta^t \\ \Delta & \Sigma_{px} + \Sigma_{px}^t - tr(\Sigma_{px})I_3 \end{pmatrix}$$

$\vec{q}_R$  Autovector de mayor autovalor de  $Q(\Sigma_{px})$

$$\vec{q}_T = \vec{\mu}_x - R(\vec{q}_R)\vec{\mu}_p$$

Error cuadrático del registrado  $d = f(\vec{q})$   $\vec{q} = [\vec{q}_R \mid \vec{q}_T]^t$   $\vec{q}(P)$  Transformación del conjunto P con los parámetros q

## Registrado de imágenes de rango

- Algoritmo:

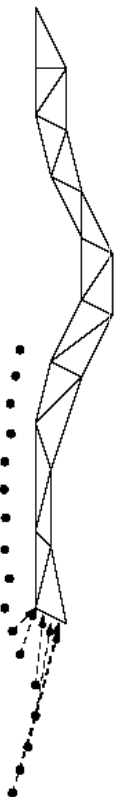
- Inicialización  $\vec{q}_0 = [1, 0, 0, 0, 0, 0]^t$ ,  $d_0 = f(\vec{q}_0)$  y  $k = 1$   
 $P_0 = P$
- Repetir
  - ▶ Para cada punto  $\vec{p}_i$  de  $P_{k-1}$ 
    - Calcular punto más próximo  $X$ ,  $\vec{y}_i$
    - Añadirlo a  $Y_{k-1} = \{\vec{y}_i\}$
  - ▶ Calcular registrado  $(\vec{q}_k, d_k)$  entre  $P_{k-1}$   $Y_{k-1}$
  - ▶ Aplicar transformación 3D encontrada para calcular  
 $P_k = \vec{q}_k(P_{k-1})$
- hasta  $d_{k-1} - d_k < \tau$

## Fusión de mallados

- **Geometría completa** de un objeto:
  - Varias imágenes de rango.
- **Fusión de varias imágenes:**
  - Registrado:
    - alinear imagen de rango y modelo.
  - Integración:
    - Crear una representación única a partir de dos o varias imágenes de rango.
- **Algoritmo iterativo del punto más próximo** modificado:
  - No se puede aplicar directamente el algoritmo del punto más próximo.
  - Puntos de un mallado no tienen porque tener correspondiente en el otro mallado.

## Fusión de mallados

- **Modificaciones:**
  - Descartar puntos correspondientes entre mallados que **estén lejos el uno del otro**.
  - Descartar **puntos correspondientes** que en uno de los dos mallados **sean borde del modelo**.

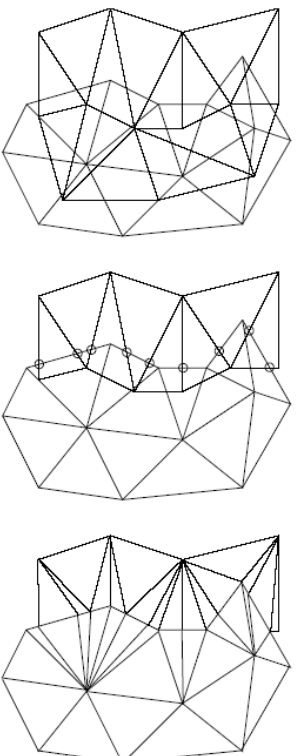


- También se puede aplicar **multirresolución**:
  - Utilizar transformación encontrada en un nivel de resolución como transformación inicial para el siguiente.

## Fusión de mallados

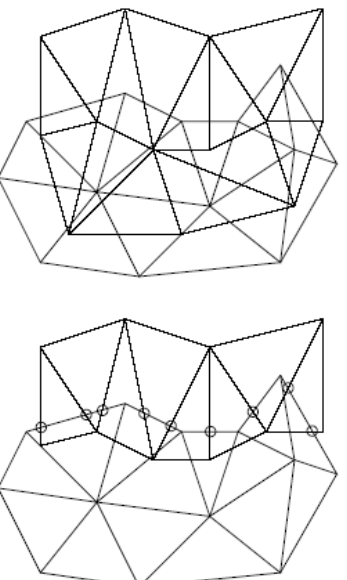
### ■ Procedimiento:

- Calcular el **registrado**, parámetros de transformación 3D, entre ambos mallados:
  - Algoritmo del punto más próximo modificado.
- Eliminar partes **superpuestas** en ambos mallados.
- Conectar ambos **mallados** con triángulos adicionales que rellenen el espacio entre ambos.



## Fusión de mallados

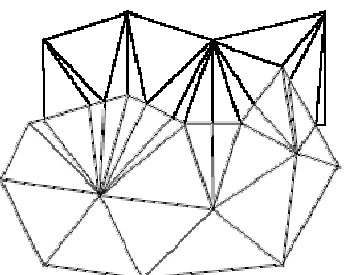
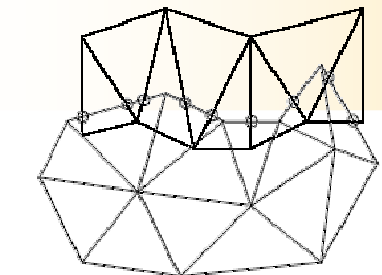
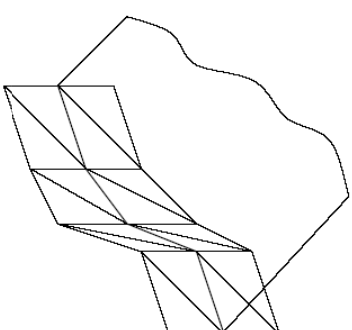
- Eliminar partes **superpuestas** (redundantes):
  - Para cada **triángulo T** del **mallado A**:
    - Calcula puntos más próximos de los vértices de T en **mallado B**.
    - Si las tres **distancias** entre puntos son **menores** que un **umbral**, entonces **T** es **redundante**.





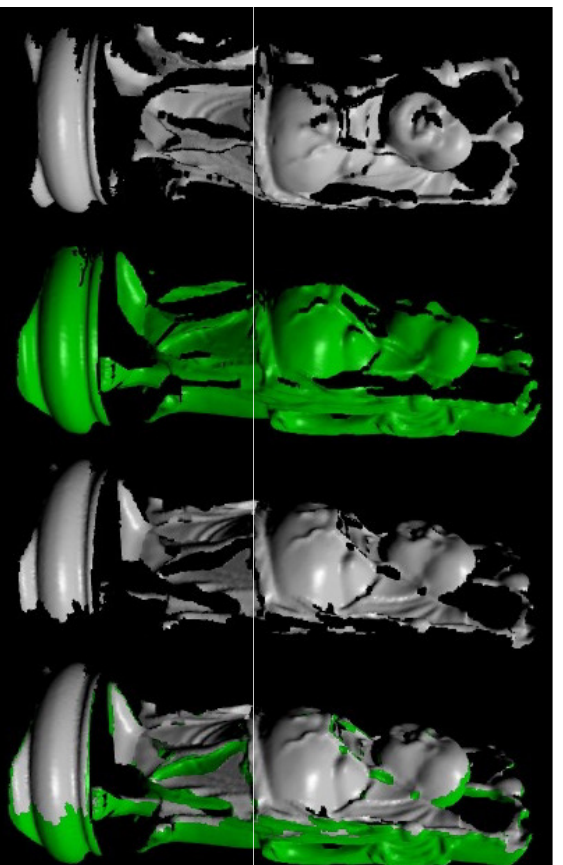
## Fusión de mallados

- **Conectar ambos mallados:**
  - **Añadir nuevos vértices** a los bordes de B, donde las aristas de triángulos de A lo intersectan.



- **Dividir cada triángulo borde de A** en dos partes, dentro y fuera del borde de B:
  - ▶ **Eliminar** la parte que queda dentro de B.
- **Re-triangularizar** la parte que está fuera de B.

## Fusión de mallados



Registrado de imágenes de rango y fusión de mallados. Representación en rendering.

- Introducción.
- Obtención de imágenes de rango:
  - Estereoscopia.
  - Luz estructurada.
  - Tiempo de vuelo.
- Técnicas de mado:
  - Mado triangular inicial.
  - Mado multiresolución.
  - Algoritmo de “desfile de cubos”.
- Registrado de imágenes de rango:
  - Algoritmo iterativo del punto más próximo.
  - Fusión de mados.
- **Segmentación de superficies.**

## Segmentación de superficies

- Determinar que **partes de la imagen** pertenecen a un **mismo modelo** de superficie:
  - Planos, cilindros, esferas, ...
  - **Calcular parámetros** de cada modelo.
  - Calcular **puntos que soportan** cada modelo.
  - Identificar puntos que no se ajustan a ninguna superficie (**errores**).
  - Estimar el **error del ajuste** de cada superficie.

## Segmentación de superficies

- **Algoritmo de clasificación simultánea:**
  - **Distintos modelos de superficie** simultáneamente.
  - Determina el conjunto de superficies que mejor describe los datos con una **codificación mínima**.
  - Estrategia de **minimización global**.
  - **Ajuste robusto** de los datos a las superficies.
- **Fases del algoritmo:**
  - Exploración.
  - Selección.
  - Ajuste.
  - Selección final.

## Segmentación de superficies

- **Exploración:**
  - Extrae un conjunto de **modelos candidatos** (hipótesis).
  - La generación de hipótesis se realiza mediante una búsqueda de forma aleatoria.
- **Selección:**
  - **Selecciona** aquellos que explican los datos de forma óptima de acuerdo a un criterio de error.
  - Se plantea como un problema de optimización discreta booleana, utilizando un **algoritmo de búsqueda tabú**.
- **Ajuste:**
  - Se realiza un ajuste por **mínimos cuadrados modificado**, para aumentar la robustez.
- **Selección final:**
  - El ajuste puede cambiar la clasificación de los datos soporte de cada modelo (superficie), realizando una **clasificación final de los datos**.

- Jain, R.C. and Jain, A.K. "Analysis and Interpretation of Range Images", Springer-Verlag, 1990.
- Curless, B. "From Range Scans to 3D Models", Computer Graphics, Vol 33, No 4, Nov 1999.