

# UNIVERSIDAD CATÓLICA SAN PABLO

## MAESTRÍA EN CIENCIA DE LA COMPUTACIÓN

### COMPUTACIÓN GRÁFICA

**Finding Surface Correspondences Using Symmetry Axis  
Curves**

**Tianqiang Liu, Vladimir G. Kim, and Thomas Funkhouser**

José Castro Ochante  
[Jose.castro.ochante@ucsp.edu.pe](mailto:Jose.castro.ochante@ucsp.edu.pe)

Arequipa, Marzo 2018



# Contenido

- Objetivo
- Introducción
- Métodos
- Conclusiones
- Limitaciones
- Referencias



# Objetivo

Implementación de un algoritmo para encontrar un mapeo de correspondencia entre dos superficies 3D.

$$f: S1 \rightarrow S2$$

Donde  $S1$  y  $S2$ , son superficies.



# Introducción

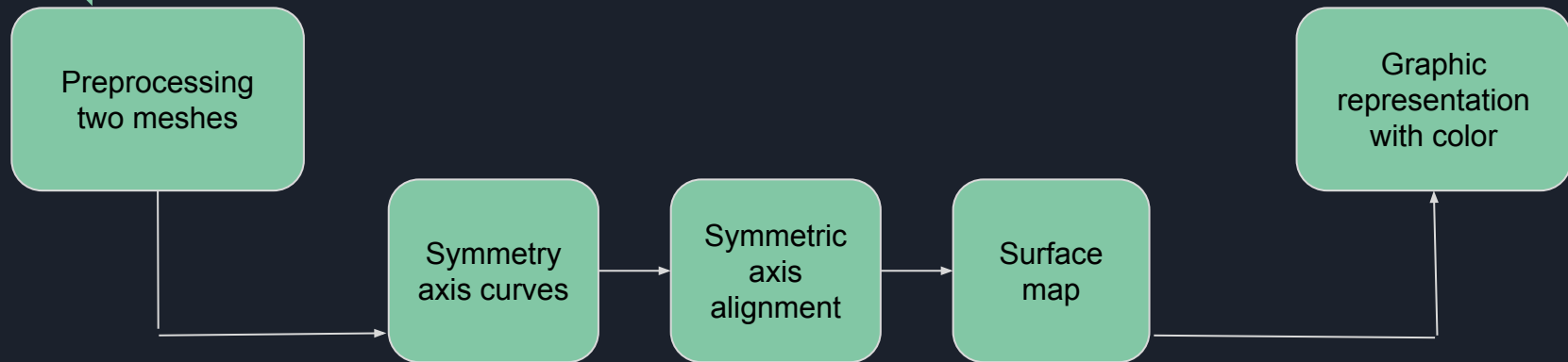
Encontrar correspondencias semánticamente significativas entre puntos en diferentes superficies es un problema fundamental en computación gráfica.

# Introducción

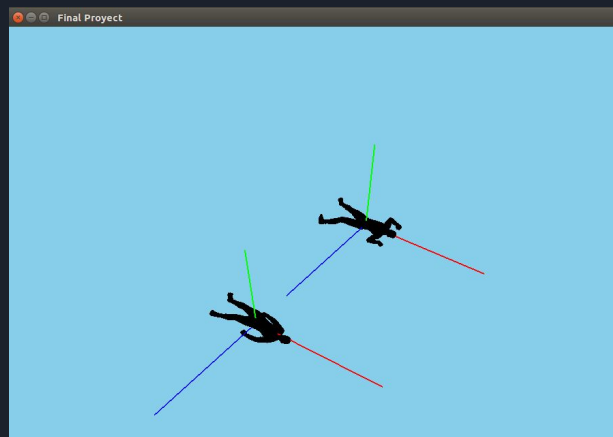
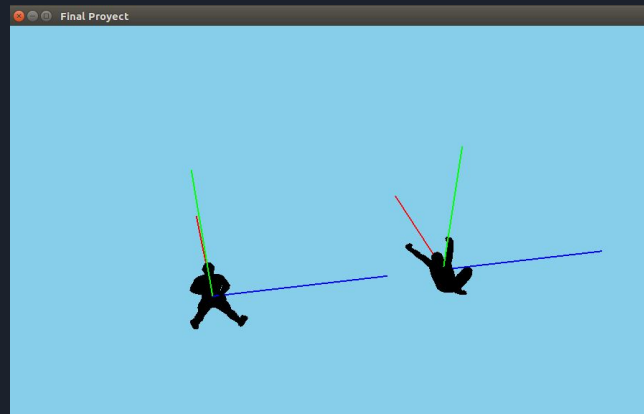
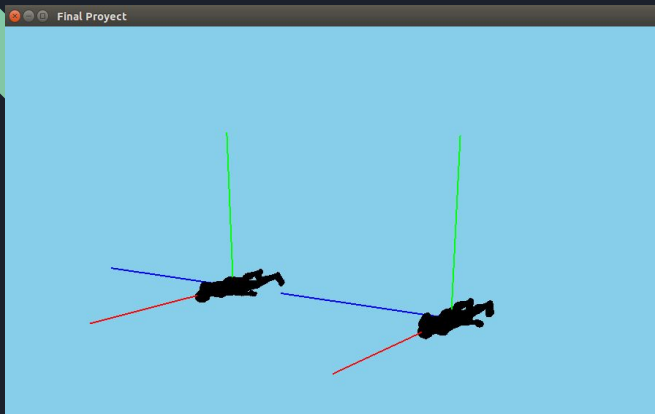
Idea principal: Aprovechar las simetrías reflexivas globales para simplificar el problema de correspondencia entre superficies.



# Métodos



# Preprocessing



## Symmetric Axis Curves

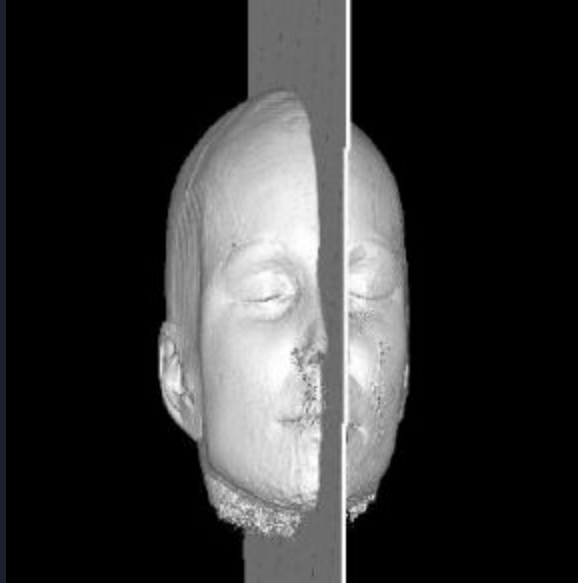
Pasos requeridos para el algoritmo y el orden de complejidad en cada paso:

<i>Step</i>	<i>Complexity</i>
(1) Read the image into memory and store as an array of intensity values.	$O(n)$
(2) Calculate the centroid of the object.	$O(n)$
(3) Calculate the covariance matrix for the object.	$O(n)$
(4) Calculate the eigenvectors of the covariance matrix for the object to determine the principal axes.	$k_1$
(5) For each principal axis, measure the degree of bilateral symmetry in the image about the hyperplane that is normal to the axis and contains the centroid.	$O(n)$
(6) Select the hyperplane with the highest measure of symmetry as the dominant hyperplane of bilateral symmetry.	$k_2$

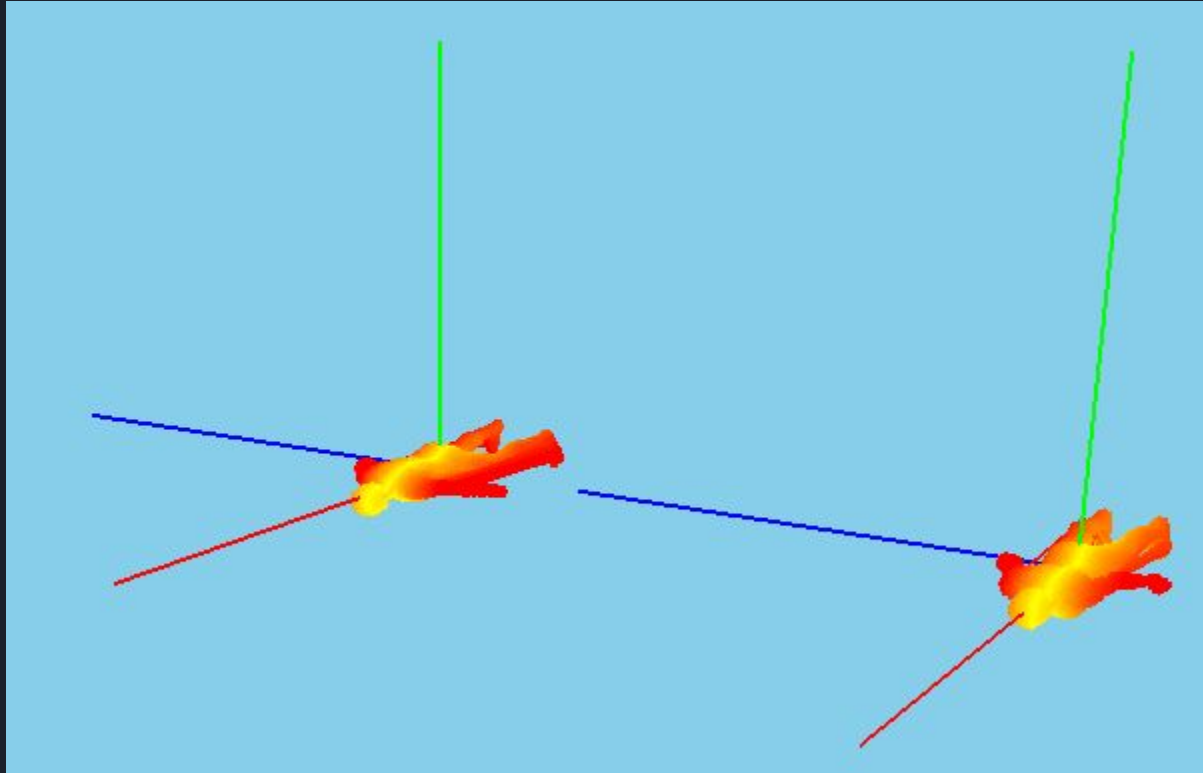




## Symmetric Axis Curves



## Symmetric Axis Curves





## Symmetric Axis Alignment

Encontrando los argumentos que maximice la calidad de la curva

$$(C_1^*, C_2^*, c^*) = \arg \max_{C_1, C_2^j, c} \{Q(C_1^i, C_2^j, c)\}$$

Definimos la calidad de alineación como:

$$Q(C_1^i, C_2^j, c) = Q_{Axis}(C_1^i) \cdot Q_{Axis}(C_2^j) \cdot Q_{Align}(C_1^i, C_2^j, c)$$

Donde:

$$Q_{Align}(A, B, c) = 1 / (D(A, B, c) + \epsilon)$$
$$D(A, B, c) = \sum_{0 \leq k < K} \gamma(\mathbf{a}_k, \mathbf{b}_{jk})$$

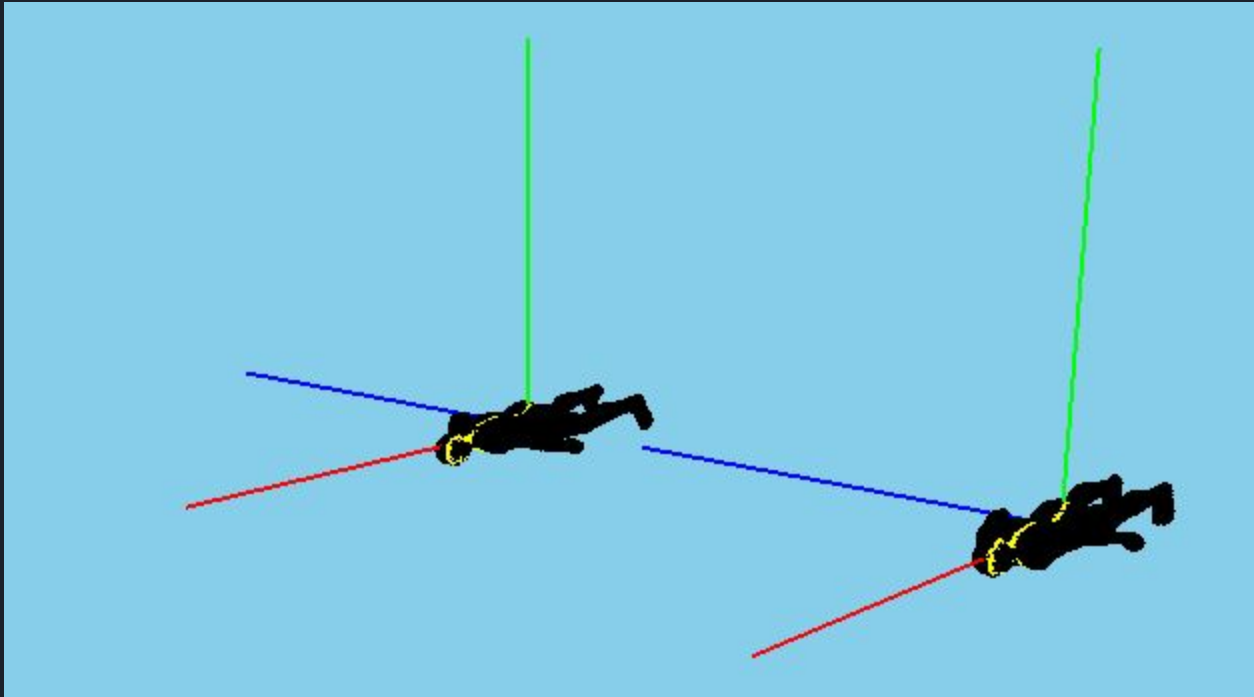


# Symmetric Axis Alignment

## Edit Distance Algorithm

$$D(A_{1:i}, B_{1:j}) = \min \begin{cases} D(A_{1:i-1}, B_{1:j-1}) + \gamma(a_i, b_j) \\ D(A_{1:i-1}, B_{1:j}) + \min(\gamma(a_i, b_j), \gamma(a_i, \lambda)) \\ D(A_{1:i}, B_{1:j-1}) + \min(\gamma(a_i, b_j), \gamma(\lambda, b_j)) \end{cases}$$

## Symmetric Axis Alignment



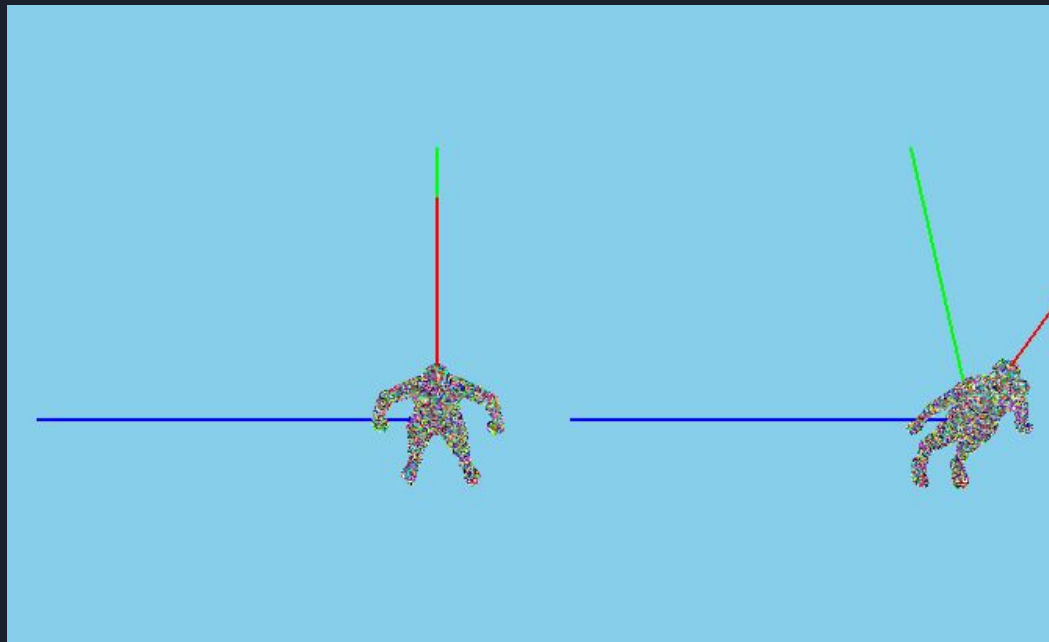


# Correspondence Extrapolation

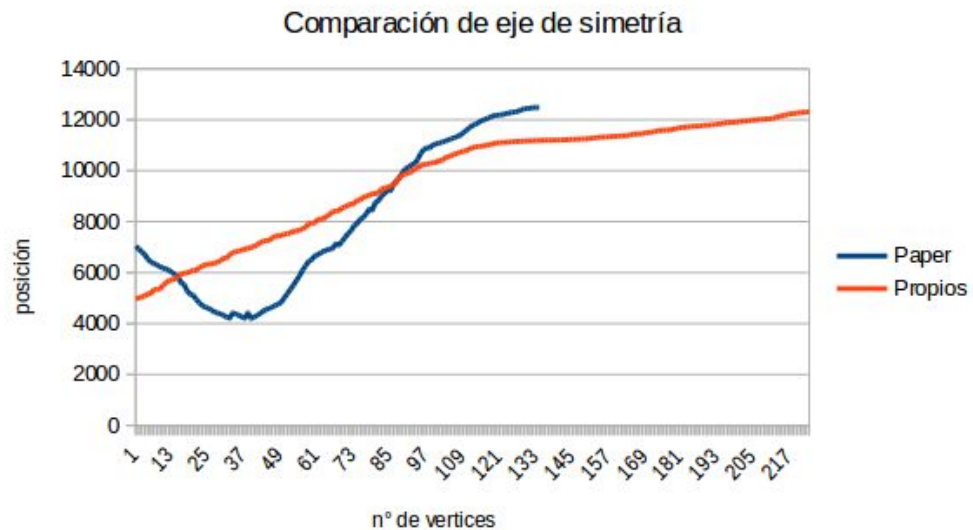
Generalized Multidimensional Scaling(GMDS)

$$\sigma(\mathbf{X}; \mathbf{D}_{\mathcal{S}_N}) = \frac{1}{N} \sum_{i>j} (d_{\mathbb{R}^m}(x_i, x_j) - d_{\mathcal{S}}(s_i, s_j))^2,$$

# Correspondence Extrapolation

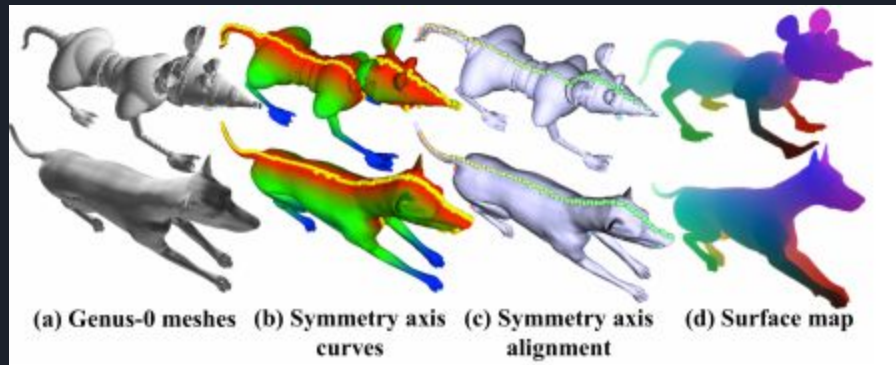


# Resultados.





# Resultados.





## Limitaciones.

- Métodos matemáticos muy poco detallado.
- DataSet incompletos y no actualizados.
- En lo personal me limitaba el tiempo en entender temas de topología y generación de mallas.



# Propuesta

- Replicar el paper usando la versión moderna de OpenGL.
- Uso de CUDA para el ejecución de métodos de alto costo computacional.