

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ**

**DESARROLLO DE HERRAMIENTA DE VISUALIZACIÓN PARA
LA REPARACIÓN DE PIEZAS ARQUEOLÓGICAS BASADO
EN SU SIMETRÍA**

Tesis para optar por el Título de **Ingeniero Informático**, que presenta el
bachiller:

Gerson Ricardo Toribio Ossio

ASESOR: PhD. Ivan Anselmo Sipiran Mendoza

Lima, Diciembre de 2018

Resumen

La simetría es la correspondencia exacta en la disposición regular de las partes o puntos de un cuerpo con relación a un centro, un eje o un plano. Esta característica está presente en la naturaleza y en objetos fabricados por el hombre. Asimismo, en el caso de objetos simétricos realizados por el hombre, existen vasijas de la cultura inca conocidos como keros que representan una simetría rotacional. Sin embargo, muchos de estos cerámicos no se encuentran en óptimas condiciones, debido a que fueron deteriorados por la tierra y por las piedras. Además, muchos de estos estuvieron enterrados o fueron parcialmente dañados en guerras.

En años recientes, el estudio de la simetría ha captado la atención de las comunidades de computación gráfica. La simetría puede facilitar el entendimiento computacional de algunas representaciones de manera visual, por lo tanto esta característica ubicua puede brindar una gran ayuda en la reconstrucción de objetos.

Uno de los problemas está relacionado con la verificación y detección de la simetría en un objeto. Algunos métodos encontrados en la literatura, tales como Momentos generalizados y Curvas del eje de simetría, para poder realizar la verificación de la simetría en un objeto necesita que este posea una estructura completa, dejando de esta manera inaplicable estos métodos para la reconstrucción de restos arqueológicos.

Otro problema encontrado es el recurso computacional que se necesita para realizar los cálculos, debido a que se componen de pasos muy complejos computacionalmente. Además, al ser algoritmos complejos, estos pueden poseer un tiempo de ejecución mayor a 15 minutos, siendo esto un factor negativo debido a que se espera que la reconstrucción de los objetos arqueológicos sea interactiva y rápida.

Desde el punto de vista de los arqueólogos, el proceso de reconstrucción de objetos arqueológicos de manera manual puede tomar varios días. Sumado a esto, en la mayoría de los casos, la reconstrucción no es precisa, porque se realizan muchas maquetas hasta obtener un modelo final con la posible reconstrucción del objeto. Existen distintos métodos como alternativa para el tratamiento de reconstrucción de objetos, claro que la precisión dependerá de qué tan completo se pueda tener a estos. En el presente proyecto de fin de carrera se pretende plantear una interfaz gráfica en la cual el usuario pueda interactuar con el método extraído de aproximación de detección de simetrías en Mallas 3D (Sipiran, Gregor, & Schreck, 2014).

Como objetivo se busca reducir el campo de análisis del método anteriormente mencionado, por lo cual se plantea el uso de una interfaz gráfica para la interacción entre el usuario con el uso de una herramienta OpenGL, para de esta manera reducir el tiempo de respuesta que presenta este método.

Esta reducción se realizó con la interacción que brindó el usuario usando la interfaz gráfica, donde el usuario podrá brindar información indicando dónde es posible que se encuentren los ejes de simetría o planos de simetría que servirán como entrada para el método a utilizar para la reconstrucción de los objetos arqueológicos.



Dedicatoria

A mis padres Gladys Lourdes Ossio Quispe y Julio Ricardo Toribio Aguilar, A mis hermanos Israel y Christian por el gran apoyo que me brindaron para lograr mis metas trazadas.

A mis abuelos por confiar en mí y apoyarme constantemente.

A mí enamorada Mayra por el apoyo moral durante y después de mi vida universitaria.

Y mis amigos más cercanos por la gran amistad brindada.

Contenido

1. Problemática	12
2. Estado del arte	17
2.1. Métodos basados en correspondencias	17
2.1.1. Curvas de eje de simetría (<i>Symmetry Axis Curves</i>)	18
2.1.2. Geometrías simétricas y regulares (<i>Geometric Symmetries and Regularities</i>)	19
2.2. Métodos basados en transformación	20
2.2.1. Momentos generalizados (<i>Generalized Moments</i>)	21
2.2.2. Vista basada en la detección de simetría (<i>View-based Symmetry Detection</i>)	21
2.3. Métodos basados en votos	22
2.3.1. Incorporación de simetría factorizada (<i>Symmetry Factored Embedding</i>)	23
2.3.2. Esqueleto basado en simetría intrínseca (<i>Skeleton-based Intrinsic Symmetry</i>)	24
2.3.3. Optimización K-sparse (<i>K-sparse Optimization</i>)	26
2.3.4. La simetría aproximada de mallas 3D (<i>Approximate Symmetry in Partial 3D Meshes</i>)	26
2.4. Conclusiones del estado del arte	27
3. Marco conceptual	28
3.1. Simetrías	28
3.2. Mapas Mezclados (<i>Blended Maps</i>)	29
3.3. Métodos de puntos característicos y simetrías rotacionales	29
3.4. Método de difusión de calor y simetrías rotacionales	31
3.5. Método de características locales en imágenes para detección de simetrías	33
3.6. Rotación alrededor de un eje arbitrario	35
3.7. Rotación alrededor de un plano arbitrario	38
4. Objetivo general	40
4.1. Objetivos específicos	40
4.2. Resultados esperados	40
4.3. Herramientas, métodos, metodologías y procedimientos	41
4.3.1. Herramientas	42

4.3.2. Métodos	45
5. Alcance y limitaciones	47
5.1. Alcance	47
5.2. Limitaciones	47
6. Justificación y viabilidad	49
6.1. Justificación del proyecto de fin de carrera	49
6.2. Análisis de viabilidad del proyecto de fin de carrera	49
6.2.1. Viabilidad técnica	49
6.2.2. Viabilidad temporal	50
6.2.3. Viabilidad económica	50
6.3. Conclusión	50
7. Implementación de la interfaz gráfica y visualización del objeto	51
7.1. Programa que permita la visualización del objeto arqueológico analizado	51
7.1.1. Desarrollo	51
7.1.2. Resultado obtenido	51
7.2. Visualización de las guías de simetría que engloban al objeto, el paralelepípedo que encierra al objeto.	55
7.2.1. Desarrollo	55
7.2.2. Resultado	55
7.3. Barras de color que indica si un objeto es simétrico en un plano de corte	56
7.3.1. Desarrollo	57
7.3.2. Resultado	57
8. Interacción de entre el objeto analizado y usuario	64
8.1. Software integrado con la visualización e interacción con el objeto.	64
8.1.1. Desarrollo	64
8.1.2. Resultado	64
9. Implementación del algoritmo de reconstrucción y análisis del resultado de la reconstrucción del objeto	70
9.1. Implementación del algoritmo de reconstrucción	70
9.2. Análisis del resultado de la reconstrucción del objeto	76
9.3. Evaluación de la reconstrucción	78

10.	Observaciones, conclusiones y trabajos a futuro	84
11.	Referencias Bibliográficas	88



Índice de ilustraciones

Figura 1.1: Representación de las vasijas de la cultura Tiwanaku y Tumulaca. Extraído de (Knobloch, 2012)	12
Figura 1.2: Clasificación de los cerámicos incas. Extraído de (Bray).....	13
Figura 2.1: Resultado de aplicar el método de. Extraído de (Liu, Kim, & Funkhouser, 2012)	18
Figura 2.2: Selección de una lámpara (Azul) y la similitud de sus partes con otros objetos (Naranja). Extraído de (Tevs, Huang, Wand, Seidel, & Guibas, 2014)	19
Figura 2.3: El método detecta simetría de objetos escaneados. La detección de estas simetrías requiere comprobar simetrías candidatas. La imagen fue extraída de (Martinet, Soler, Holzschuch, & François, 2006).....	21
Figura 2.4: Primero se observa al objeto analizado. Luego, se ve la entropía del objeto en una esfera. Finalmente se muestra como resultado el plano de simetría. Extraído de (Li, Johan, Ye, & Lu, 2014)	22
Figura 2.5: La distancia de simetría factorizada (Symmetry Factored Distance) visualizado en un grupo de objetos con simetría parcial extrínseca. Extraído de (Lipman, Chen, Daubechies, & Funkhouser, 2010)	24
Figura 2.6: Detección de la simetría de las articulaciones de una muñeca de madera. Extraído de (Jiang, Xu, Cheng, & Zhang, 2013).....	25
Figura 2.7: Reconstrucción de un objeto mediante el uso de simetrías. Extraído de (Mavridis, Sipiran, Andreadis, & Papaioannou, 2015)	26
Figura 2.8: Proceso que sigue el método para la detección del plano de simetría en una malla que representa al objeto. Extraído de (Sipiran, Gregor, & Schreck, 2014)	27
Figura 3.1: Se muestran las distorsiones de los pesos, la mezcla de ambos y resultado de aplicar Mapas Mezclados. Extraído de (Kim, Lipman, & Funkhouser, 2011)	29
Figura 3.2: Resultado del método de difusión de calor utilizando el eje de rotación encontrado.	33
Figura 3.3: Líneas de simetría en los mapas de profundidad.....	34
Figura 3.4: Rotación alrededor de un eje arbitrario	35
Figura 3.5: Dirección de cosenos	36
Figura 3.6: Rotación alrededor del eje X.....	36
Figura 3.7: Rotación alrededor del eje Y.....	37
Figura 4.1: Visualización de un objeto 3D. Extraído de (OpenGL Programming, s.f.).....	43
Figura 4.2: Visualización del marco de referencia de CGAL. Extraído de (CGAL course, s.f.)	43
Figura 4.3: Visualización del programa de Qt. Extraído de (Qt Framework, s.f.).....	44
Figura 7.1: Interfaz inicial	52
Figura 7.2: Módulo de lectura que ofrece la interfaz.	53

Figura 7.3: Lectura del objeto 3D.	53
Figura 7.4: Visualización del objeto 3D.	54
Figura 7.5: rotación del objeto 3D	54
Figura 7.6: Visualización de las guías de simetría que encierran al objeto..	55
Figura 7.7: Objeto 3D rotado y asociado con la guías de simetría	56
Figura 7.8: Representación de la barra de color.....	57
Figura 7.9: Representación del cilindro en un espacio de tres dimensiones	58
Figura 7.10: Representación de los centros en las tapa superior e inferior.	59
Figura 7.11: Representación de los triángulos que conforman la estructura del cilindro.....	59
Figura 7.12: La normal del plano que representa al triángulo indica que cara de esta figura será pintada. Siguiendo la regla de la orientación de la mano derecha se indica la dirección en qué orden deben ser colocado los puntos que representan el triángulo.....	60
Figura 7.13: Representación de las tres barras de color. Las 3 barras están alineadas a ejes cartesianos.	60
Figura 7.14: Corte del objeto utilizando planos, explicando cuando el plano es óptimo.	61
Figura 7.15: Resultado luego de aplicar el método de cálculo de la simetría.....	62
Figura 7.16: Ruptura del objeto analizado	62
Figura 7.17: Barras de colores donde se muestra la simetría del objeto. Los colores en las barras de simetría indican la simetría del objeto desde cada plano cartesiano.	63
Figura 8.1: Opción de edición de guías de simetría.....	65
Figura 8.2: Opciones para seleccionar la cara del paralelepípedo que se desea modificar	65
Figura 8.3: Guías de simetría cambiadas por el usuario. Se observa que el usuario disminuyó la cara izquierda del paralelepípedo	66
Figura 8.4: Opción que permite la selección de los planos de simetría.....	67
Figura 8.5: Mensaje donde se indica si el usuario desea realizar la selección de los planos.	67
Figura 8.6: Visualización los planos seleccionados hasta el momento. Por defecto estos planos en un inicio son los mejores planos que pueden representar la simetría del objeto en los distintos ejes de simetría.	68
Figura 8.7: Cambio del plano Z, donde P2 era su posición inicial y con la interacción del usuario este plano cambio de posición.....	69
Figura 9.1: Seudocódigo del algoritmo de reconstrucción	70
Figura 9.2: Seudocódigo de la función de homogenización de vértices.....	72
Figura 9.3: Seudocódigo de la función de inserción de vértices homogéneos	72
Figura 9.4: La matriz de la izquierda es la matriz de translación T y la matriz de la derecha es la matriz de translación inversa T_{inv}	73
Figura 9.5: Seudocódigo de la función para la obtención de valores previos	

a la función de Maple.....	74
Figura 9.6: Matriz Q que representa la forma final de la fórmula de Rodrigues aplicando Maple	75
Figura 9.7: Matriz de reflexión.....	76
Figura 9.8: Reconstrucción del objeto utilizando interacción del usuario.	77
Figura 9.9: Opción de exportar el nuevo objeto reconstruido	77
Figura 9.10: Fórmula para la evaluación de la efectividad de los algoritmos de reconstrucción.	79
Figura 9.11: Reconstrucción de una vasija, objeto número 3. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 789390 vértices y 1578768 aristas.	80
Figura 9.12: Reconstrucción de una vasija, objeto número 2 La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 585878 vértices y 1171744 aristas.	80
Figura 9.13: Reconstrucción de una vasija, objeto número 1. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 481730 vértices y 963436 aristas	80
Figura 9.14: Reconstrucción de una vasija, objeto número 6. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 541240 vértices y 1082452 aristas.	81
Figura 9.15: Reconstrucción de una vasija, objeto número 5. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 608834 vértices y 1217644 aristas.	81
Figura 9.16: Reconstrucción de una vasija, objeto número 4. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 710246 vértices y 1420468 aristas.	81
Figura 9.17: Reconstrucción de una vasija, objeto número 8. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 665496 vértices y 1330960 aristas.	82
Figura 9.18: Reconstrucción de una vasija, objeto número 7. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 664068 vértices y 1328132 aristas.	82
Figura 9.19: Reconstrucción de una vasija, objeto número 9. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 727042 vértices y 1454060 aristas.	82

Figura 9.20: Construcción de cerámico utilizando las manos y técnicas de base giratoria. 85

Figura 9.21: Kero con diseños geométricos de líneas horizontales, triángulos y cuadrados concéntricos. Imagen extraída de (Museo Larco, 2010)..... 86



Índice de tablas

Tabla 2.1: Lista de métodos de correspondencias	17
Tabla 2.2: Lista de métodos basados en transformación	20
Tabla 2.3: Lista de métodos basados en votos.	23
Tabla 4.1: Mapeo de herramientas y métodos versus resultados esperados.	41
Tabla 9.1: Tabla con los resultados al aplicar ha evaluación a los tres algoritmos de reconstrucción.....	79



1. Problemática

Según la (Real Academia Española, 2014), la simetría es la correspondencia exacta en la disposición regular de las partes o puntos de un cuerpo o figura con relación a un centro, un eje o un plano. Esta característica se encuentra presente en la naturaleza y en objetos fabricados por el hombre. Asimismo, en el caso de objetos simétricos realizados por el hombre, existen vasijas del estilo Tiwanaku y Tumulaca, como se observa en la Figura 1.1, conocidos en la literatura como keros incas que poseen una simetría rotacional (Knobloch, 2012). Al igual que objetos simétricos respecto a un eje, existen otras estructuras de la alfarería inca que son simétricos respecto a un plano, como cántaros (a), vasijas (b, c y d), ollas (d), recipientes (e), platos (f) y vasos (g) contenidos en la Figura 1.2 extraídos de (Bray).

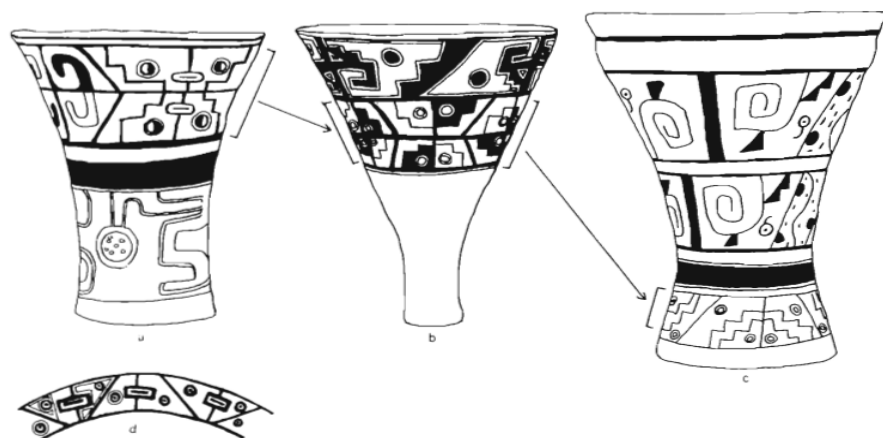


Figura 1.1: Representación de las vasijas de la cultura Tiwanaku y Tumulaca. Extraído de (Knobloch, 2012)

Sin embargo, muchas de estas esculturas y cerámicos no se encuentran completos, debido a la antigüedad de la época de su fabricación y a los distintos erosiones en estos. Adicionalmente, muchos de estos estuvieron enterrados y fueron deteriorados por la tierra y por las piedras, o fueron destruidos en guerras.

Según estudios realizados en conjunto por Sipiran, Gregor y Schreck (Sipiran, Gregor, & Schreck, 2014) se puede postular lo siguiente: (i) en años recientes, el estudio de la simetría ha captado la atención de las comunidades de

computación gráfica. Además, (ii) la simetría puede facilitar el entendimiento computacional de algunas representaciones de manera visual, por lo tanto esta característica ubicua puede brindar una gran ayuda en la reconstrucción de objetos. (iii) Cabe recalcar que existen muchas propuestas que poseen como objetivo atacar el problema de detección de simetría suponiendo que el objeto está completo para este análisis; por otra parte, esta suposición no puede ser válida para aplicaciones en las que cierto grado de geometría perdida puede afectar sus resultados.

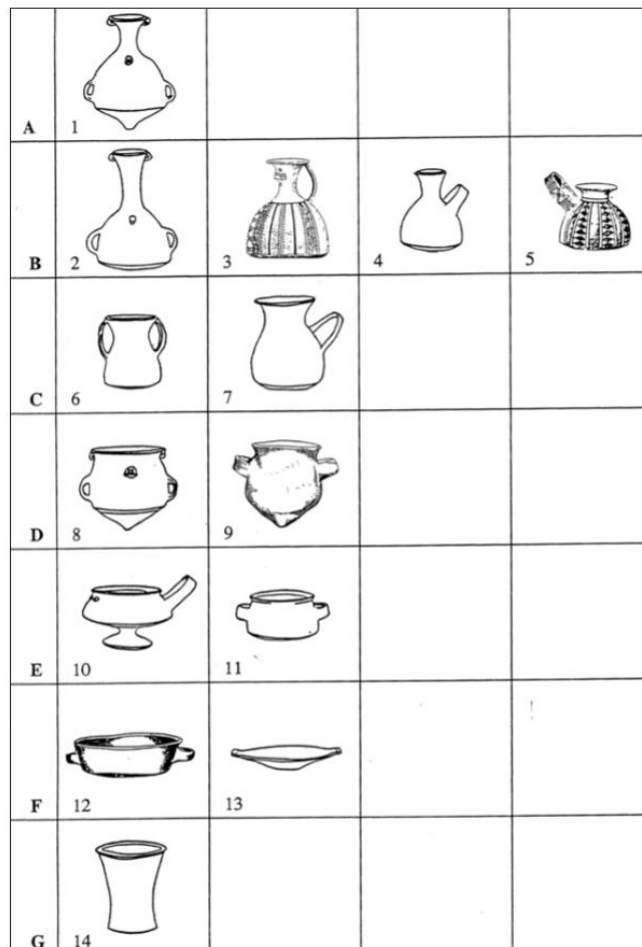


Figura 1.2: Clasificación de los cerámicos incas. Extraído de (Bray)

Un primer problema que se ha encontrado para el análisis de simetrías es en la verificación y detección de esta característica en diversos objetos, debido a que muchos están incompletos (Sipiran, Gregor, & Schreck, 2014). Se han encontrado distintos métodos para el tratamiento de la simetría de objetos 3D.

Estos se han clasificado en tres categorías, de las cuales solo dos presentan este problema: métodos basados en transformación y métodos basados en correspondencias. De los métodos basados en transformación, el más conocido y citado por la literatura es el de *momentos generalizados* (conocido como *Generalized moments*) (Martinet, Soler, Holzschuch, & François, 2006): Este método propone representar al objeto en análisis mediante el uso de nuevas clases de funciones, demostrando que las simetrías de estas funciones representan simetrías del objeto original. Sin embargo, el cuerpo a analizar necesita poseer una estructura completa para que la simetría pueda ser verificada, dejando al método de manera inaplicable para el problema de reconstrucción de restos arqueológicos. De los métodos basados en correspondencias, el más conocido es el método de curvas del eje de simetría (conocido como *Symmetry Axis Curves*). Este tiene como objetivo la detección de ejes de simetría de un objeto 3D; no obstante, este método asume que el objeto debe estar completo, en otras palabras, presenta el mismo problema mencionado anteriormente.

Otro problema encontrado es el recurso computacional que se necesita para realizar los cálculos, debido a que se componen de pasos muy complejos computacionalmente. Estos toman un tiempo de ejecución de 15 minutos en promedio, que para el campo de visión computacional no es adecuado según la entrevista, que se encuentra en el **Anexo 1: Entrevista realizada al experto PhD. Ivan Anselmo Sipiran Mendoza sobre tiempo de ejecución de los algoritmos de reconstrucción objetos**. Un ejemplo de este problema es el método de votos llamado incorporación de simetría factorizada (*Symmetry Factored Embedding*) (Lipman, Chen, Daubechies, & Funkhouser, 2010), el cual posee un tiempo de ejecución mayor a 15 minutos, pero el resultado es más preciso, debido a que la relación precisión-tiempo es directamente proporcional. Desde el punto de vista de los arqueólogos, el proceso de reconstrucción de objetos arqueológicos de manera manual puede tomar varios días, lo que significa un retraso en las labores de comprensión a través de estos monumentos. Sumado a esto, en la mayoría de los casos, la reconstrucción no es precisa, porque se realizan muchas maquetas hasta

obtener un modelo final con la posible reconstrucción del objeto, lo cual demuestra la importancia del tiempo de respuesta de los métodos de reconstrucción. La arqueología peruana es un claro ejemplo de esto, ya que para realizar una reconstrucción de piezas arqueológicas se necesitan maquetas, las cuales toman muchos días en realizarse. En algunos casos, las maquetas no son precisas, y el arqueólogo debe poseer una gran experiencia para realizar la reconstrucción adecuada del objeto. Es importante mencionar que no todos los métodos toman importancia al tiempo de ejecución. Por ejemplo, uno de los métodos dentro de la categoría de los métodos de votos que cumple esta afirmación es el llamado incorporación de simetría factorizada (*Symmetry Factored Embedding*) (Lipman, Chen, Daubechies, & Funkhouser, 2010). Este método propone el uso de una matriz que representa un grafo de relaciones simétricas, de esta manera es posible encontrar correspondencias simétricas del objeto a analizar. Dentro de la matriz se tienen todos los errores mínimos de una transformación en el objeto. Sin embargo, si se desea obtener una respuesta con mayor precisión, se necesitarán mayores recursos computacionales, los cuales suelen representar grafos muy densos. La consecuencia de esto es que se requiere un mayor tiempo de ejecución.

Se ha expuesto la existencia de problemas relacionados a la reconstrucción de objetos arqueológicos debido al uso de simetrías y el tiempo de respuesta de estos métodos; además del problema de los arqueólogos de seguir utilizando el método manual, a pesar de ser menos efectivo y demandar más tiempo. Detectar las simetrías de un objeto es una labor complicada, a causa de la limitación de no tener un objeto completo en la mayoría de casos. Existen distintos métodos como alternativa para el tratamiento de reconstrucción de objetos, claro que la precisión dependerá de qué tan completo se pueda tener a estos. En el presente proyecto de fin de carrera se pretende plantear una interfaz gráfica en la cual el usuario pueda interactuar con el algoritmo de reconstrucción basado en un plano de simetría.

Lo que se busca es reducir el campo de evaluación que realizará el método extraído de aproximación de detección de simetrías en Mallas 3D (Sipiran,

Gregor, & Schreck, 2014), el cual se explicará en este documento. Se usa una interfaz gráfica para la interacción entre el usuario con el uso de una herramienta OpenGL, para de esta manera reducir el tiempo de respuesta por parte del método de aproximación de detección de simetrías en Mallas 3D. Se ingresarán los datos de entrada al método, convirtiéndolo de esta forma en un método semiautomático, que es considerado así cuando los datos de entrada pueden ser brindados por el usuario. Al ser un método semiautomático, no analizará todas las regiones de un objeto, sino que un usuario podrá brindar información indicando dónde es posible que se encuentren los ejes de simetría o planos de simetría que servirán como entrada para el método a utilizar para la reconstrucción de los objetos arqueológicos.



2. Estado del arte

El detectar y reconstruir objetos mediante el uso de la simetría ha sido un problema recurrente en la comunidad científica y ha sido atacado de diversas maneras. Al 2017 existen métodos muy variados para la detección de simetrías en objetos 3D. Los métodos han podido ser clasificados en cuatro grupos: métodos basados en correspondencias, métodos basados en transformación, métodos basados en votos y algunos métodos alternativos a estos tres últimos ya mencionados.

2.1. Métodos basados en correspondencias

Los métodos basados en simetría analizan correspondencias entre puntos o regiones de un objeto 3D que pueden ser correspondencias simétricas, basándose en el emparejamiento, dadas ciertas condiciones específicas para la detección de simetrías. A continuación se presentan los métodos contenidos en esta categoría en la Tabla 2.1.

Tabla 2.1: Lista de métodos de correspondencias

Referencia	Nombre
(Bokeloh, Berner, Wand, Seidel, & Schilling, 2009)	Línea de características (<i>Feature Lines</i>)
(Berner, Wand, Mitra, Mewes, & Seidel, 2011)	Simetrías Subespaciales (<i>Subspace Symmetries</i>)
(Grushko, Raviv, & Kimmel, 2012)	Detección parcial de simetría (<i>Partial Symmetry Detection</i>)
(Wang, Xu, Li, Zhang, & Shamir, 2011)	Jerarquía de simetría (<i>Symmetry Hierarchy</i>)
(Liu, Kim, & Funkhouser, 2012)	Curvas de eje de simetría (<i>Symmetry Axis Curves</i>)
(Tevs, Huang, Wand, Seidel, & Guibas, 2014)	Geometrías simétricas y regulares (<i>Geometric Symmetries and Regularities</i>)

De la tabla anterior se seleccionaron los siguientes métodos. A continuación se presentan:

2.1.1. Curvas de eje de simetría (*Symmetry Axis Curves*)

Según lo propuesto por Liu, Kim y Funkhouser (Liu, Kim, & Funkhouser, 2012): (i) el objetivo del método de curvas de eje de simetría es ubicar el eje de simetría de un objeto 3D; y (ii) si este objeto analizado es cerrado, el eje de simetría es como una curva cerrada y el problema de detectar simetrías se convierte en detectar los posibles ejes de simetría.

El procedimiento de reconstrucción consiste en el siguiente: (i) se ubica un conjunto de curvas que como característica principal contengan puntos estacionarios de la simetría reflectiva del objeto; (ii) luego se realiza un análisis de las funciones de correspondencia, encontrando los puntos que pueden ser simétricos. (iii) Se detectan los puntos característicos en los extremos de una función y para cada punto se calculan las distancias geodésicas a todos los puntos del objeto, se promedian las distancias obtenidas; (iv) después de tener todas las distancias promedios, estos valores son asignados a cada punto; (v) los puntos extremos son característicos. Luego, se realiza los mapas mezclados (*Blended Maps*) con tripletas de puntos característicos; y (vi) finalmente, para cada conjunto de correspondencia se extrae una curva simétrica. El resultado de este método se puede observar en la Figura 2.1.

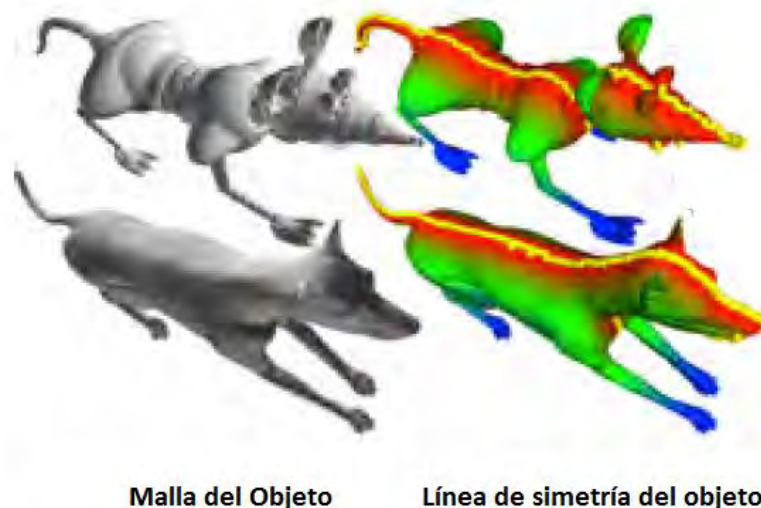


Figura 2.1: Resultado de aplicar el método de.
Extraído de (Liu, Kim, & Funkhouser, 2012)

2.1.2. Geometrías simétricas y regulares (*Geometric Symmetries and Regularities*)

Según lo propuesto por Tevs, Huang, Wand, Seidel y Guibas (Tevs, Huang, Wand, Seidel, & Guibas, 2014): (i) el objetivo de este método es detectar regiones para que estas puedan ser analizadas con las de otros objetos; (ii) luego, con estas regiones se define un grafo en donde los nodos son piezas geométricas y los arcos definen tipos de simetría que asocian a la región geométrica.

Este método busca encontrar los bordes en las superficies del objeto. Cada punto que se asocia con el centro del borde y un marco de referencia es calculado con las normales de las caras adyacentes y los vectores del borde. Con toda esta información se generan simetrías candidatas entre los objetos básicos; y finalmente, se realiza un algoritmo de comparación entre grafos para determinar relaciones simétricas entre objetos de la misma clase. El resultado de este método se puede observar en la Figura 2.2.

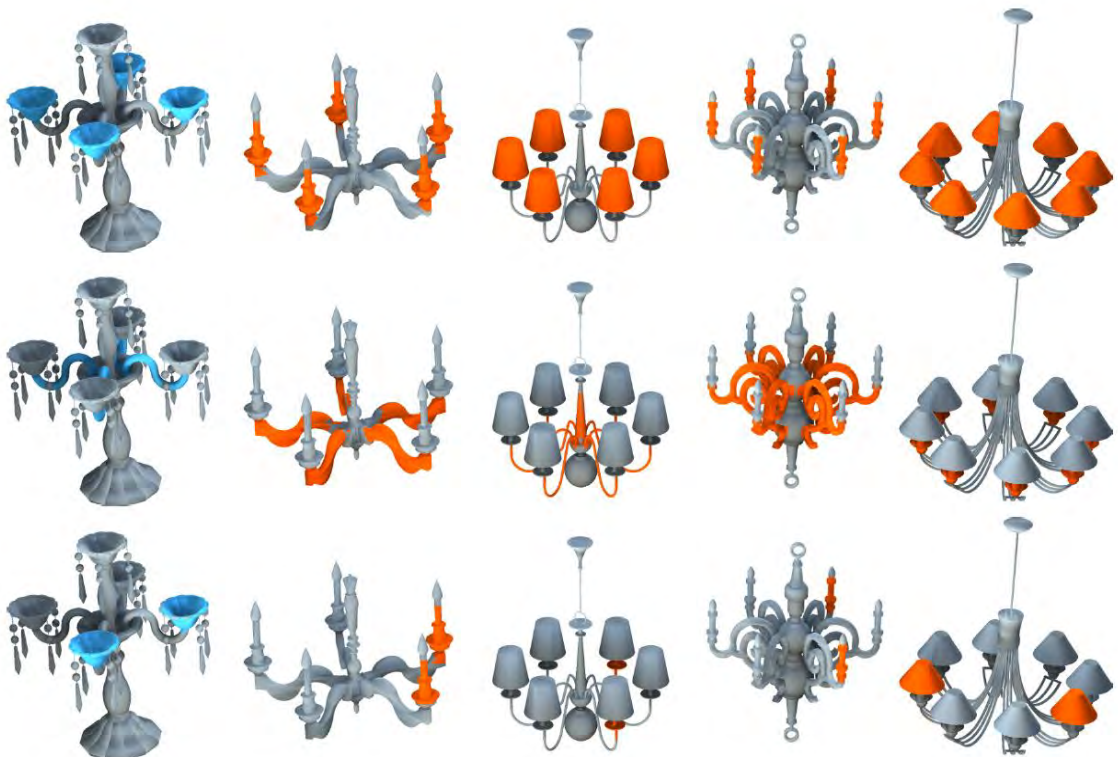


Figura 2.2: Selección de una lámpara (Azul) y la similitud de sus partes con otros objetos (Naranja). Extraído de (Tevs, Huang, Wand, Seidel, & Guibas, 2014)

2.2. Métodos basados en transformación

En este grupo, los métodos transforman el problema de encontrar simetrías a otro dominio distinto al original. Tiene como objetivo hallar la simetría dentro de la transformación del objeto analizado.

A continuación se presentan los métodos contenidos en esta categoría en la Tabla 2.2.

Tabla 2.2: Lista de métodos basados en transformación

Referencia	Nombre
(Kazhdan, Chazelle, Dobkin, Funkhouser, & Rusinkiewicz, 2003)	Descriptor de Simetría Reflectiva (<i>Reflective Symmetry Descriptor</i>)
(Kazhdan, Funkhouser, & Rusinkiewicz, Symmetry Descriptors and 3D Shape Matching, 2004)	Descriptor de simetría (<i>Symmetry Descriptor</i>)
(Martinet, Soler, Holzschuch, & François, 2006)	Momentos generalizados (<i>Generalized moments</i>)
(Podolak, Shilane, Golovinskiy, Rusinkiewicz, & Funkhouser, 2006)	Transformada de simetría planar-reflectante (<i>Planar-Reflective Symmetry Transform</i>)
(Li, Johan, Ye, & Lu, 2014)	Vista basada en la detección de simetría (<i>View-based Symmetry Detection</i>)
(Shape Matching via Quotient Spaces, 2013)	Espacios de cociente (<i>Quotient Spaces</i>)
(Liu, y otros, 2015)	Mapas funcionales ortonormales (<i>Orthonormal Functional Maps</i>)

Se tomaron en cuenta dos de los métodos contenidos en la tabla anterior. A continuación se presentan:

2.2.1. Momentos generalizados (*Generalized Moments*)

Según lo propuesto por Martinet, Soler, Holzschuch y François (Martinet, Soler, Holzschuch, & François, 2006): (i) el objetivo de este método es representar el objeto analizado mediante el uso de clase de funciones; (ii) si se comprueba la simetría de estas funciones, entonces se logrará la simetría del objeto; (iii) estas funciones son llamadas momentos generalizados; (iv) primero, se deben escoger direcciones como los vectores formados los vértices de un icosaedro; (v) después de tener las direcciones, se examina la magnitud del gradiente del momento generalizado en esa dirección; sin embargo, las magnitudes pequeñas que resultan de realizarlo son examinadas nuevamente para tener un análisis detallado; y (vi) las direcciones candidatas se convierten a una matriz de transformaciones que es verificada. El resultado de este método se puede observar en la Figura 2.3.

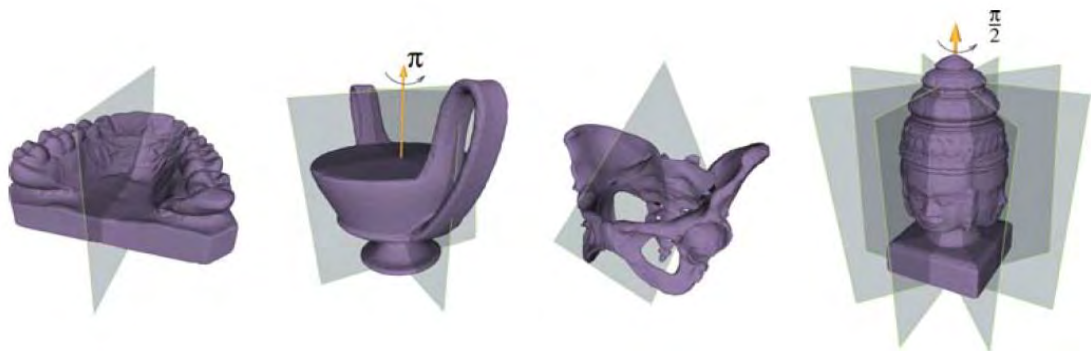


Figura 2.3: El método detecta simetría de objetos escaneados. La detección de estas simetrías requiere comprobar simetrías candidatas. La imagen fue extraída de (Martinet, Soler, Holzschuch, & François, 2006)

2.2.2. Vista basada en la detección de simetría (*View-based Symmetry Detection*)

Según lo propuesto por Li, Johan, Ye y Lu (Li, Johan, Ye, & Lu, 2014): (i) el objetivo del método es calcular las distribuciones de la vista del objeto usando su propia entropía, esta vista de la entropía del objeto se encuentra en la Figura 2.4; (ii) se busca obtener las simetrías del objeto original; sin embargo, se buscarán las simetrías en una esfera que represente la entropía de las vistas del objeto; (iii) las simetrías en la esfera representará la simetría en el objeto analizado; (iv) el método comienza con normalizar el tamaño, escala y rotación de un objeto en un sistema de coordenadas 3D, permitiendo que el

análisis se realice de manera controlada; (v) el algoritmo encierra el modelo normalizado en un icosaedro regular y este es subdividido usando iteraciones del algoritmo de bucle (*Loop*) (Loop, 1987); teniendo los poliedros con distintos número de vértices, se toman los vértices de los poliedros como puntos de vista para calcular los mapas de entropía; y finalmente, el algoritmo realiza un análisis completo en la esfera, buscando todos los planos posibles entre pares de punto de vista. El resultado de este método se puede observar en la Figura 2.4.

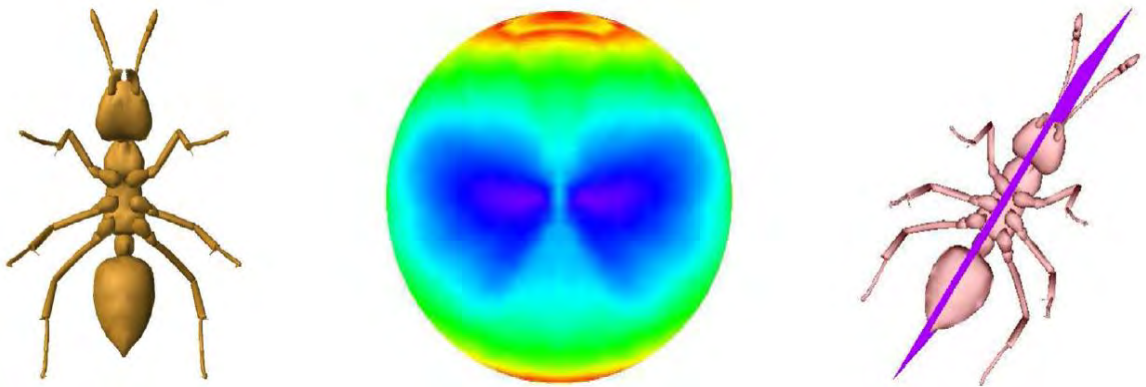


Figura 2.4: Primero se observa al objeto analizado. Luego, se ve la entropía del objeto en una esfera. Finalmente se muestra como resultado el plano de simetría. Extraído de (Li, Johan, Ye, & Lu, 2014)

2.3. Métodos basados en votos

Estos métodos se basan en la acumulación de evidencias de existencias de transformaciones que son simétricas. Los métodos de la Tabla 2.3 codifican las transformaciones. Luego, las transformaciones son analizadas para encontrar las transformaciones que corresponden a las simetrías.

La Tabla 2.3 muestra una lista de los métodos basados en votos que se encuentran en la literatura.

Tabla 2.3: Lista de métodos basados en votos.

Referencia	Nombre
(Thrun & Wegbreit, 2005)	Forma de simetría (<i>Shape from Symmetry</i>)
(Lipman, Chen, Daubechies, & Funkhouser, 2010)	Incorporación por simetría factorizada (<i>Symmetry Factored Embedding</i>)
(Jiang, Xu, Cheng, & Zhang, 2013)	Esqueleto basado en simetría intrínseca (<i>Skeleton-based Intrinsic Symmetry</i>)
(Xu, y otros, 2012)	Simetría intrínseca parcial a varias escalas (<i>Multi-scale Partial Intrinsic Symmetry</i>)
(Mavridis, Sipiran, Andreadis, & Papaioannou, 2015)	Optimización K-sparse (<i>K-sparse Optimization</i>)
(Zheng, y otros, 2010)	Consolidación de escaneo no local (<i>Non-local Scan Consolidation</i>)
(Xu, y otros, 2009)	Simetrías Reflejadas Intrínsecas Parciales (<i>Partial Intrinsic Reflectional Symmetries</i>)
(Mitra, Guibas, & Pauly, Symmetrization., 2007)	Simetrización (<i>Symmetrization</i>)

Se tomaron en cuenta cuatro de los métodos más recientes encontrados en la literatura que pertenecen a este subconjunto. A continuación se presentan:

2.3.1. Incorporación de simetría factorizada (*Symmetry Factored Embedding*)

Según lo propuesto por Lipman, Chen, Daubechies y Funkhouser (Lipman, Chen, Daubechies, & Funkhouser, 2010): (i) el método está basado en la construcción de una matriz de correspondencias la cual representa un grado de relaciones geométricas; (ii) esto muestra que se puede encontrar una representación entre los vectores de la matriz de correspondencias y los valores; (iii) además, éste concluye que la distancia entre estos vectores representa simetrías del objeto original; (iv) primero se crea una matriz, en la que cada una de sus celdas es el error mínimo posible entre cualquier transformación rígida que convierta un punto del objeto en otro punto; luego, este error mínimo es calculado usando la desviación promedio de errores

mínimos cuadrados sobre los puntos de la transformación del objeto; y (vi) finalmente, la matriz con los errores mínimos es convertida a una matriz de correspondencia.

En la práctica, se escogen 4 puntos A, B, C y D. Los puntos A y B del objeto original. Se debe tomar la consideración de que la distancia entre los puntos A y B no debe de ser muy pequeña para evitar la presencia de una transformación de identidad; y luego, se procede a la transformación de la línea con puntos A y B en otra formada con puntos C y D. Después se calcula la congruencia entre el objeto original y el transformado, y el error se almacena en la matriz; y finalmente, se escoge un punto fijo y se calcula la distancia espectral a todos los puntos de dicho de objeto analizado, lo puntos con mínima distancia serán correspondencias simétricas a dicho punto previamente analizado. En la Figura 2.5 se observa las distancias del objeto.

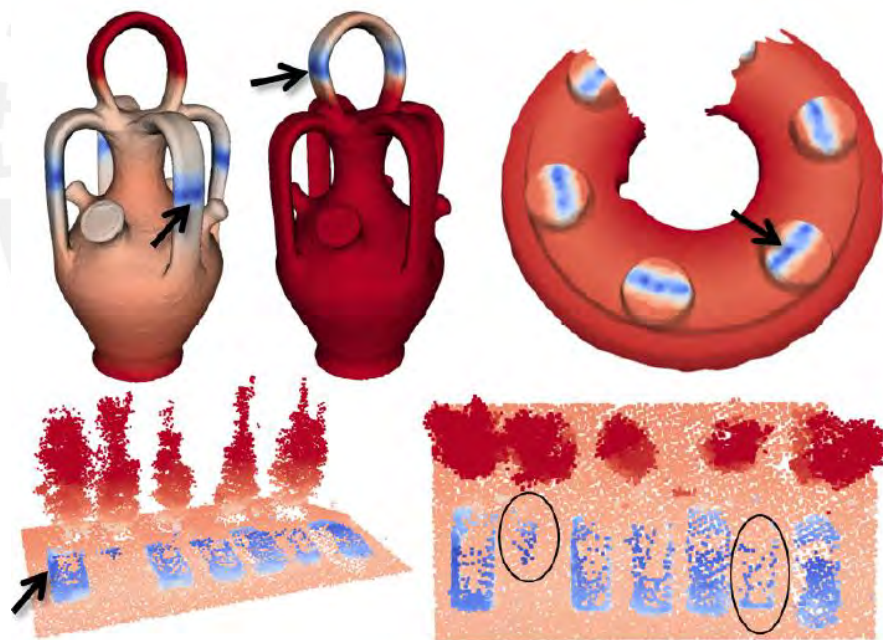


Figura 2.5: La distancia de simetría factorizada (Symmetry Factored Distance) visualizado en un grupo de objetos con simetría parcial extrínseca. Extraído de (Lipman, Chen, Daubechies, & Funkhouser, 2010)

2.3.2. Esqueleto basado en simetría intrínseca (*Skeleton-based Intrinsic Symmetry*)

Según lo propuesto por Jiang, Xu, Cheng y Zhang (Jiang, Xu, Cheng, & Zhang, 2013): (i) el objetivo de este método es detectar simetrías de objetos cuya

representación son nubes de puntos; (ii) además, el método trabaja con información faltante del objeto; (iii) primero, se calcula la nube de puntos del objeto; (iv) luego, el método toma una muestra de puntos del esqueleto hasta obtener una cobertura de este, siendo estos puntos usados para el análisis de simetría; (v) después, se seleccionan los pares de puntos son agrupados, estos puntos son llamados lectores de simetría; (vi) una vez obtenidos los puntos del paso anterior, se realizan cuatro filtros; (vii) el primero de los filtros es el de similitud, donde un par de nodos del objeto pasan por el filtro, para verificar si la similitud de las funciones del diámetro de una figura (*Shape Diameter Function*) es menor a un umbral determinado; (viii) el segundo de los filtros es el de ramificación, el cual verifica que dos puntos provengan de la misma ramificación del esqueleto; (ix) el tercer filtro es el de perfil de distancia, en donde se calcula un histograma con los descriptores de distancias entre los puntos analizados y los puntos extremos al resto de puntos terminales del esqueleto; (x) luego, se filtra los puntos que posean una mayor distancia euclidiana; (xi) el último filtro consiste en el soporte de simetría, el cual verifica y analiza si estos electores pueden pertenecer a la misma transformación; (xii) el conjunto de puntos obtenidos al final es una aproximación de las correspondencias simétricas entre nodos del esqueleto; y (xiii) finalmente, se calcula una matriz de correspondencias simétricas, dentro de esta matriz se acumula evidencia de votos para la transformación de simetrías y su descomposición espectral ayuda la búsqueda de simetrías. El resultado de este método se puede observar en la Figura 2.6.

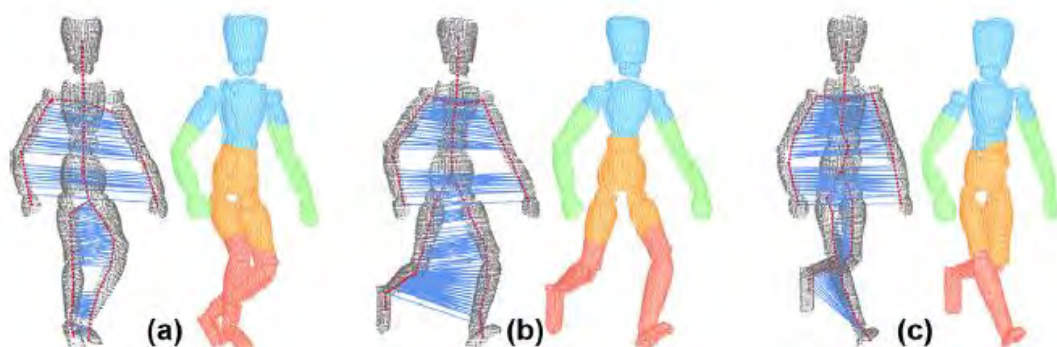


Figura 2.6: Detección de la simetría de las articulaciones de una muñeca de madera. Extraído de (*Jiang, Xu, Cheng, & Zhang, 2013*)

2.3.3. Optimización K-sparse (*K-sparse Optimization*)

Según lo propuesto por Mavridis, Sipiran, Andreadis y Papaioannou (Mavridis, Sipiran, Andreadis, & Papaioannou, 2015): (i) este método tiene como objetivo la reconstrucción de objetos arqueológicos con partes faltantes; (ii) se realiza una detección de simetrías del objeto y copia la geometría que falta basando en simetrías detectadas del objeto, luego se intenta completar al objeto de manera iterativa hasta completar las partes faltantes del objeto. En la Figura 2.7 se observa el proceso de reconstrucción del objeto por cada iteración

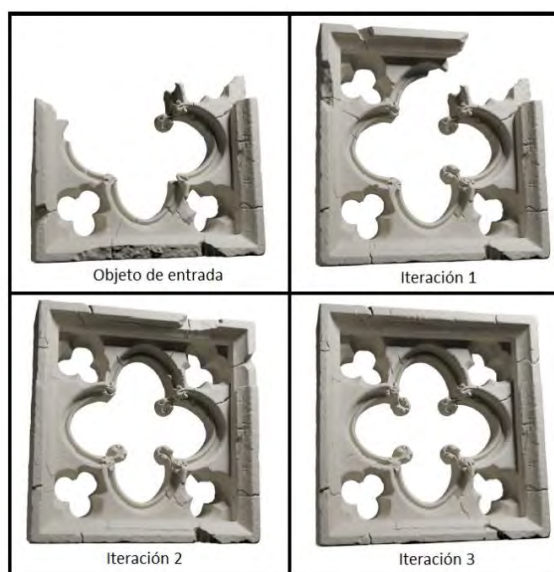


Figura 2.7: Reconstrucción de un objeto mediante el uso de simetrías. Extraído de (Mavridis, Sipiran, Andreadis, & Papaioannou, 2015)

2.3.4. La simetría aproximada de mallas 3D (*Approximate Symmetry in Partial 3D Meshes*)

Según lo propuesto por Sipiran, Gregor y Schreck (Sipiran, Gregor, & Schreck, 2014): (i) este método hace énfasis en la teoría de difusión del calor sobre superficies de objetos para el análisis de la existencia de correspondencias simétricas; (ii) el método consta de dos pasos muy bien definidos; (iii) en primer lugar, una función realiza el cálculo sobre la malla del objeto, de tal manera que los máximos locales de esta función son posibles correspondencias simétricas; (iv) en segundo lugar, se genera un conjunto de planos candidatos y mediante un sistema de votación se determinan los planos con mayor probabilidad de representación de simetrías. En la Figura

2.8 se observa el proceso de reconstrucción del objeto.

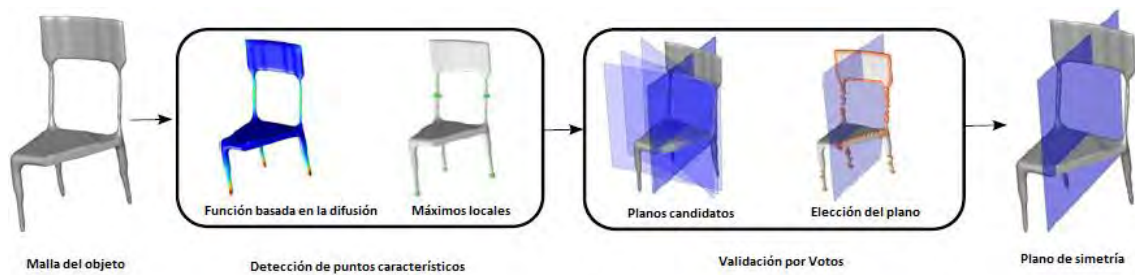


Figura 2.8: Proceso que sigue el método para la detección del plano de simetría en una malla que representa al objeto. Extraído de (*Sipiran, Gregor, & Schreck, 2014*)

2.4. Conclusiones del estado del arte

Actualmente, existen muchos métodos para la detección y reconstrucción de un objeto arqueológico mediante el uso de la simetría. Sin embargo, muchos de estos necesitan ciertas características para poder trabajar en óptimas condiciones, como ya se ha mencionado en la problemática. Estos métodos presentan varios problemas, debido a que algunos necesitan que el objeto esté completo, demandan muchos recursos computacionales o pueden demorar mucho tiempo para obtener un resultado que no necesariamente puede ser el correcto. Además, existen proyectos de investigación como Cutlab3D (Cutlab3D, s.f.) y Presious (Presious, s.f.) que se dedican a la reconstrucción de objetos arqueológicos; sin embargo, no plantean la interacción con el usuario, debido a que son problemas de investigaciones recientes. Añadido a esto, según (Gregor, Bauer, Perakis, & Schreck, 2015) siempre se ha tratado de observar hasta qué punto se puede llegar con la reconstrucción de objetos arqueológicos usando los métodos automáticos actualmente presentes.

Como se mencionó anteriormente, la interacción del usuario beneficiaría de manera productiva la velocidad de procesamiento de los métodos de difusión de calor y simetrías rotacionales. De esta manera, lo que se busca es que los métodos reduzcan su campo de análisis con ayuda del usuario.

3. Marco conceptual

En este apartado se presentarán las principales definiciones que serán útiles para los lectores de este documento comprendan la terminología utilizada. Se cubren conceptos de métodos y conceptos básicos de simetría.

3.1. Simetrías

Simetría es un concepto universal en la naturaleza, ciencia y arte. En el mundo físico, las simetrías geométricas y regularidad estructural se presentan en distintas escalas.

Una simetría es una transformación no trivial T que aplicada sobre un objeto O 3D lo deja igual al original. Formalmente, T es una simetría si $T(O) = O$. Según la clasificación propuesta por Mitra, Pauly, Wand y Ceylan (Mitra, Pauly, Wand, & Ceylan, 2013) la simetría se pueden clasificar en:

- **Simetría intrínseca**

Una transformación T es una simetría intrínseca si la distancia entre dos puntos de un objeto O en 3D es preservada. Es decir, dados dos puntos p_i y p_j en O , se tiene que $d_g(p_1, p_2) = d_g(T(p_1), T(p_2))$. Siendo d_g la distancia geodésica.

- **Simetría extrínseca**

Una transformación extrínseca es aquella que preserva la métrica Euclidiana en el espacio de la forma 3D. En este caso, dado dos puntos p_i y p_j de un objeto 3D, una transformación T es extrínseca si se tiene $\|p_i - p_j\| = \|T(p_i) - T(p_j)\|$. La transformación T preserva las distancias Euclidianas entre pares de puntos del objeto 3D.

- **Simetría global**

Sea O un objeto 3D. Una transformación T es una simetría global se tiene que $T(O) = O$.

- **Simetría parcial**

Sea O el objeto 3D. Una transformación T es una simetría parcial si existen

dos regiones $O_i \subseteq O$ y $O_j \subseteq O$, tal que $T(O_i) = O_j$.

Como cualquier sub-dominio del objeto puede ser considerado como un objeto 3D, la simetría se puede dar dentro de estos sub-dominios. Además, la definición de simetría parcial es más general que la definición de simetría global mencionada previamente, ya que si $O_i = O_j = O$ resulta la definición de simetría global.

3.2. Mapas Mezclados (Blended Maps)

Los mapas mezclados combinan múltiples mapas intrínsecos mezclados con baja dimensionalidad que permite establecer una transformación entre dos superficies con baja distorsión (Kim, Lipman, & Funkhouser, 2011). En la Figura 3.1 se observa las distorsiones de pesos.

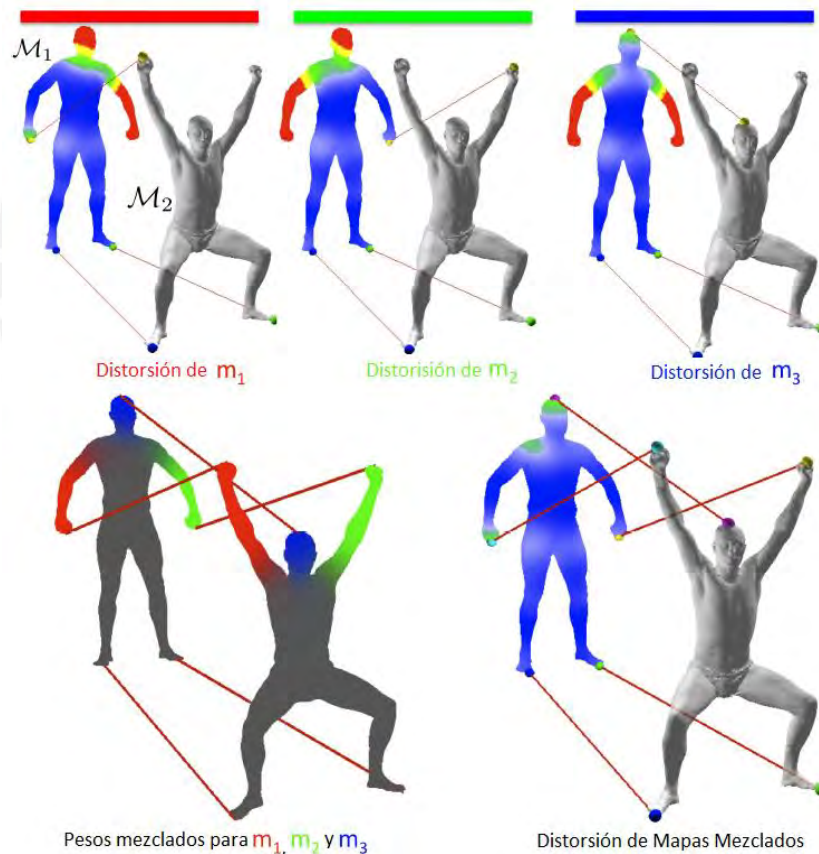


Figura 3.1: Se muestran las distorsiones de los pesos, la mezcla de ambos y resultado de aplicar Mapas Mezclados. Extraído de (Kim, Lipman, & Funkhouser, 2011)

3.3. Métodos de puntos característicos y simetrías rotacionales

Este método consiste de 4 pasos y empieza pre-procesando el objeto para

reducir su tamaño y suavizar la superficie. Luego, el operador de Harris 3D se utiliza para encontrar los puntos característicos de la malla. A continuación, para detectar el eje de rotación se utilizará el método RANSAC (Fischler & Bolles, 1981). El último paso consiste en reconstruir el objeto mediante la rotación de este sobre un eje.

- **Paso 1: Pre-procesamiento**

Debido a que un objeto tiene demasiados vértices que lo componen, se tiene como primer paso la reducción de la malla para evitar una sobrecarga computacional más adelante. Para lograr esto se utiliza el algoritmo propuesto por Garland (Fischler & Bolles, 1981). Después, el objeto que era representado con vértices y triángulos cambia a una representación con grillas y conjunto de nivel. Una vez estos dos últimos datos son calibrados se logrará eliminar las fisuras diminutas en el objeto. Como resultado de este paso se obtiene una malla sin defectos.

- **Paso 2: Detección de puntos característicos**

Una vez que el objeto ha sido simplificado y la gran mayoría de fisuras han sido removidas, como paso siguiente se debe detectar los puntos característicos. Estos puntos son los puntos más representativos de un objeto y sirven para poder simplificar el procesamiento de la malla. Para encontrar estos puntos se utilizó el operador de Harris3D (Sipiran & B., 2011). Este operador mide el cambio de una superficie ante pequeños cambios en varias direcciones alrededor de un punto. Por lo tanto, un vértice del objeto que sea muy distinto a vértices cercanos a este poseerá cambios abruptos al aplicar el operador de Harris 3D y este se escogería como punto característico.

Para nuestro propósito, se necesita analizar el borde circular de un objeto con simetría rotacional; sin embargo, el objeto no está completo y esto representa un problema porque el borde puede estar roto o incompleto. Por este motivo, el operador de Harris3D brinda apoyo a este tipo de problemas porque analiza sobre vecindarios locales.

Cabe recalcar, que el método de Harris 3D utiliza parámetros los cuales son:

- Un parámetro k con valor de 0.04
- Para el análisis de puntos vecinos se utiliza un radio de 1% de la longitud de objeto.
 - El parámetro de selección de puntos característicos es de 5% el número de vértices.
- **Paso 3: Detección de la estructura circular**

Con los puntos detectados en el paso anterior, se tiene como objetivo detectar el círculo parcial oculto en este conjunto de puntos característicos. Como es posible la presencia de puntos aislados, se decide utilizar el método de RANSAC (Fischler & Bolles, 1981). Este método es aplicado para tratar de encontrar el mejor modelo que cubra la mayor cantidad de puntos de interés y de esta manera formar un círculo con estos puntos. Se realizan iteraciones para encontrar el mejor círculo utilizando tres puntos aleatorios. Finalmente, con el método de RANSAC se busca encontrar el círculo que posea la mayor cantidad de puntos característicos, el eje de simetría es calculado ya que pasa por el centro del círculo y su orientación es representada por la normal del plano que contiene este círculo.

- **Paso 4: Reconstrucción**

Usando el eje encontrado en el paso número tres, se puede reconstruir el objeto por rotación alrededor de este eje. Sin embargo, la rotación no siempre se alinearán de manera correcta con el objeto original. Para esto se sigue el método de Pauly (Pauly, Mitra, Giesen, Gross, & Guibas, 2005), que propone que la transformación del objeto debería ser tal que el modelo rotado deformado debe poseer una alineación lo más cercana al objeto original. Esta transformación consiste en que para cada vértice de la malla rotada, existe un vector que almacena el desplazamiento con respecto al objeto original. Finalmente, se posee al objeto reconstruido utilizando el eje de simetría.

3.4. Método de difusión de calor y simetrías rotacionales

La teoría de difusión del calor estudia la manera en como el calor se propaga sobre una superficie en función del tiempo. Este análisis del calor está relacionado con la estructura del objeto. Para lo cual se propone una función

de difusión del calor definida sobre los puntos de cada superficie:

$$H(x, t) = \sum_{i=0}^{\infty} \left(\frac{1 - e^{-\lambda_i t}}{\lambda_i} \right) \bar{v}_i(x)^2$$

Dónde:

- λ_i Es el i-ésimo valor propio del operador Laplaciano del objeto.
- $\bar{v}_i(x)^2$ Es el i-ésimo vector propio del operador Laplaciano del objeto.
- t Es la variable de tiempo usada para la definición del grado de localidad del proceso de difusión.

Para lo cual se desea hallar el eje de simetría del objeto. Y se utiliza la función H definida sobre una superficie del objeto que es invariante a la transformación de simetría R . Es decir, si se poseen dos puntos $x \in \mathcal{O}$ e $y \in \mathcal{O}$ y se cumple que $y = R_{\varepsilon, \theta}(x)$, para algún θ , entonces $H(x) = H(y)$.

- Paso 1: Análisis del objeto

Este método consiste en definir funciones sobre la superficie del objeto 3D que son preservadas mediante transformación isométrica. Esta se basa en el uso de funciones del operador Laplaciano de la superficie del objeto 3D. Estas funciones con valor propio asociado de magnitud baja representan frecuencias bajas y respectivamente es igual en el caso de temperaturas altas. Asimismo, las frecuencias bajas tienen relación a comportamientos globales de la superficie del objeto y las frecuencias altas, con detalles del objeto. Sin embargo, con la problemática planteada del proyecto de fin de carrera donde los objetos analizados no están completos, trae como consecuencia que las características globales puedan verse afectadas.

- Paso 2: Detección de simetría

La detección empieza con el cálculo de la función H extraída de (Sipiran, Gregor, & Schreck, 2014). Para el análisis de la existencia del eje de simetría, se toma en cuenta un conjunto posible de puntos. Estos puntos fueron seleccionados mediante un método de muestreo uniforme. Para cada conjunto de puntos, se buscó un conjunto de puntos que posean valor similar

al ser calculadas en la función H (Función de difusión de calor). Estos puntos seleccionados deben formar una circunferencia y deben ser varios.

Luego, se utilizó el método RANSAC para poder determinar cuál era el mejor círculo formado con los puntos anteriores. Se encuentran muchos círculos donde no todos comparten el mismo eje rotacional. Por lo tanto, se realiza un proceso de agrupamiento de normales (Sipiran & Bustos, 2013), donde la agrupación se da por la orientación que posee la normal. Aquí se analiza que tan cerca está el centro de un círculo al eje de simetría aproximado encontrado por todos los círculos. De esta forma, se obtiene el mejor eje de simetría.

- Paso 3: Resultado

En el siguiente paso se observa el resultado de aplicar el método. Previamente se encontró los ejes de rotación para lograr la reconstrucción. En la Figura 3.2 se muestra el resultado de aplicar el método.

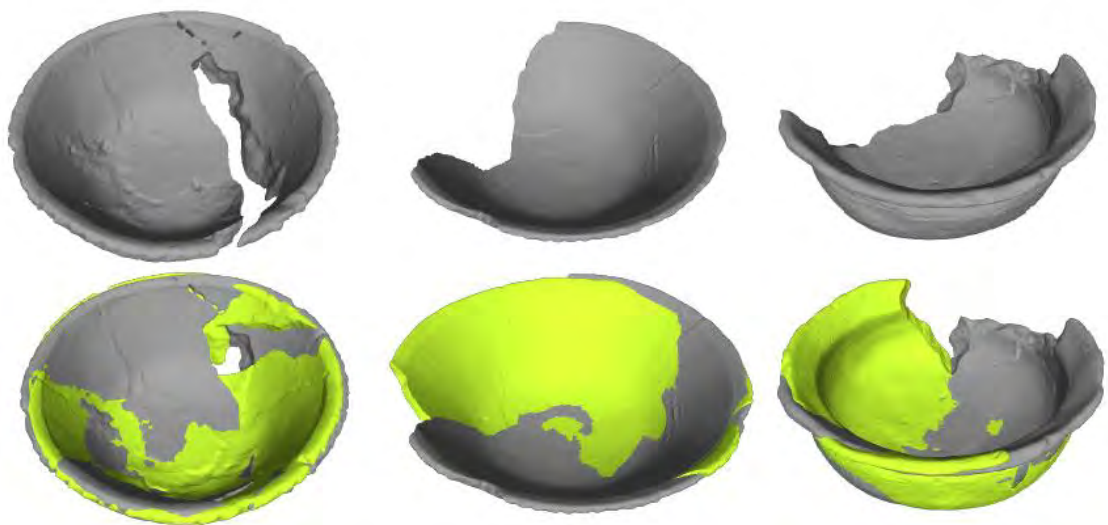


Figura 3.2: Resultado del método de difusión de calor utilizando el eje de rotación encontrado.

3.5. Método de características locales en imágenes para detección de simetrías

Según lo propuesto por (Beltrán Castañón & Sipirán Mendoza) el objetivo de este método es trabajar con objetos a los que les faltan partes, por lo cual el análisis 2D es el encargado de tratar con este problema de reconstrucción;

En el primer paso, se genera un conjunto de imágenes de profundidad desde distintos puntos de visión y en diferentes orientaciones. Se crea un mapa de profundidad en el cual cada pixel representa el valor de la profundidad, debido a que la profundidad representa algo de la geometría 3D del objeto analizado. Sin embargo, el cálculo de mapas de profundidad es un proceso costoso computacionalmente.

En el segundo paso, se procede a detectar la simetría en las imágenes. En la cual se trabajan con descriptores que son creados por cada pixel que no sea el fondo de la imagen. Todos estos descriptores sirven para detectar posibles líneas de simetría en el mapa de profundidad. Se invierte la plantilla para formar una plantilla nueva que es verticalmente simétrica a la plantilla original. Dentro de esta nueva plantilla se crea un conjunto de descriptores por cada pixel que no sea el fondo. Se comparan estos descriptores, si la distancia Hamming es pequeña entre los descriptores, significará que los pixeles son altamente simétricos. El resultado de las líneas de simetría se observan en la Figura 3.3.

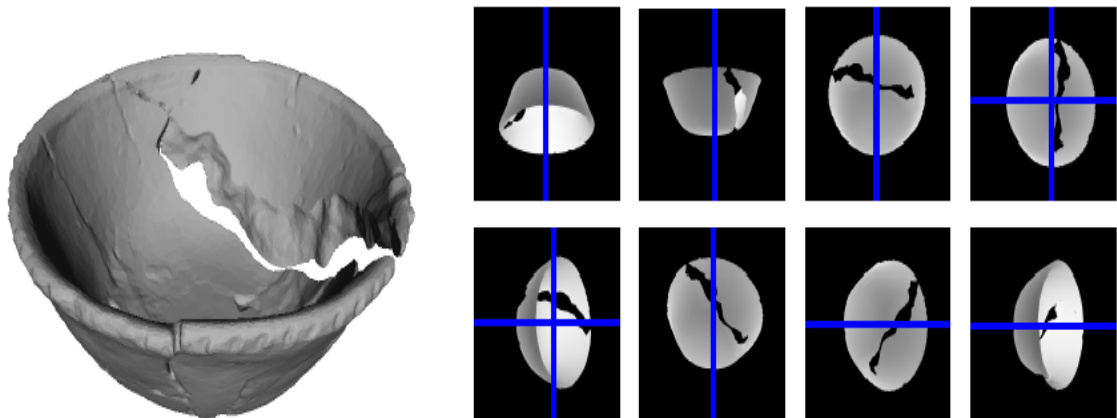


Figura 3.3: Líneas de simetría en los mapas de profundidad.

3.6. Rotación alrededor de un eje arbitrario

Es la rotación de un conjunto de puntos alrededor de un eje en el plano cartesiano. Para obtener la ecuación de (Kovács, 2012) (i) Asume que un eje arbitrario en un espacio cartesiano pasará por los puntos $P_0(x_0, y_0, z_0)$ y $P_1(x_1, y_1, z_1)$ como se aprecia en la Figura 3.4.

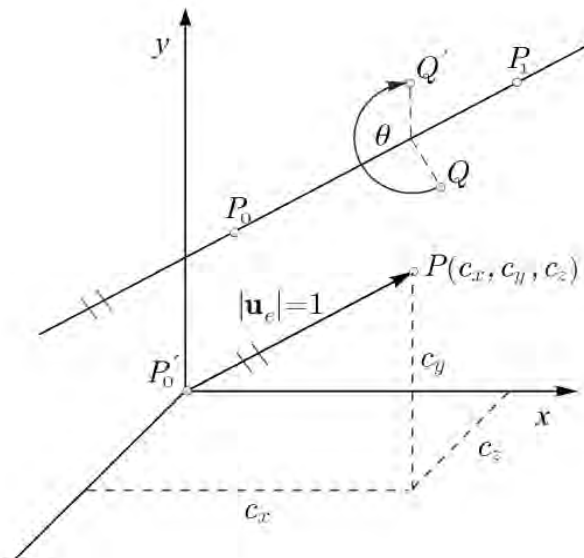


Figura 3.4: Rotación alrededor de un eje arbitrario

(ii) La rotación alrededor de un eje arbitrario con un ángulo θ es lograda siguiendo los siguientes pasos:

- Trasladar el punto $P_0(x_0, y_0, z_0)$ al origen de coordenadas.
- Realizar las rotaciones apropiadas para hacer que el eje de rotación coincida con el eje Z.
- Rotar a través del eje Z utilizando el ángulo θ
- Realizar la inversa de la transformación de rotación.
- Realizar la inversa de la translación

Por simplicidad, se calcula el vector unitario u_e como que se aprecia en la siguiente ecuación:

$$u_e = \frac{P_1 - P_0}{|P_1 - P_0|} = (c_x, c_y, c_z)$$

Luego los puntos del vector u_e se representan utilizando cosenos como se

muestra en la siguiente ecuación:

$$c_x^2 + c_y^2 + c_z^2 = 1$$

$$\cos \phi_x = c_x \quad \cos \phi_y = c_y \quad \cos \phi_z = c_z$$

La representación las ecuaciones presentadas anteriormente se pueden ver reflejadas en la Figura 3.5.

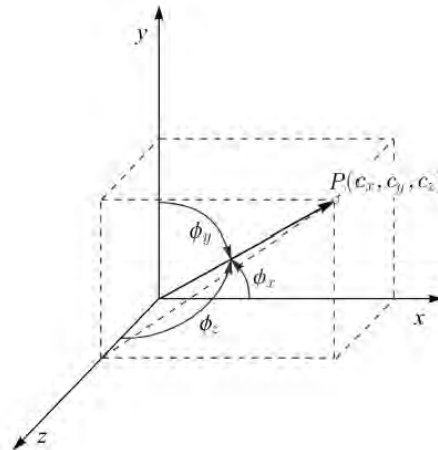


Figura 3.5: Dirección de cosenos

Luego el siguiente paso consiste de dos rotaciones sucesivas alrededor del eje X con el ángulo θ_x y del eje Y con el ángulo θ_y .

La primera rotación con el eje X se muestra en la Figura 3.6.

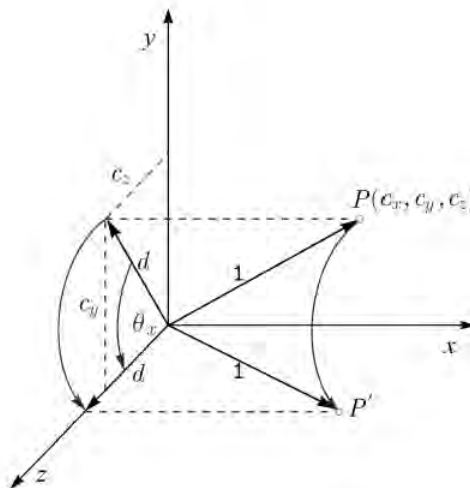


Figura 3.6: Rotación alrededor del eje X

Donde la distancia d puede representada con la siguiente ecuación $d = \sqrt{c_y^2 + c_z^2}$, con este cálculo se pueden calcular los valores de $\sin \theta_x$ y $\cos \theta_x$. Los valores de $\sin \theta_x$ y $\cos \theta_x$ son $\frac{c_y}{d}$ y $\frac{c_z}{d}$ respectivamente. Y la matriz de rotación alrededor del eje X es la siguiente ecuación:

$$\mathbf{R}_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_z/d & -c_y/d & 0 \\ 0 & c_y/d & c_z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Después se realiza la segunda rotación de manera similar alrededor del eje Y con el ángulo θ_y . Representación de la rotación se observa en la Figura 3.7.

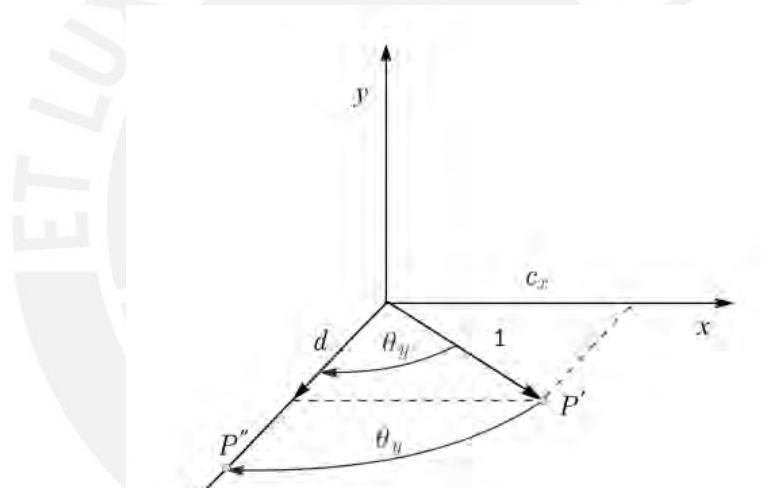


Figura 3.7: Rotación alrededor del eje Y

De esta manera se obtiene los valores de $\sin \theta_y$ y $\cos \theta_y$ será d para ambas variables. Estos valores son obtenidos del análisis de la Figura 3.7. Y la matriz de rotación alrededor del eje Y es la siguiente ecuación:

$$\mathbf{R}_y(-\theta_y) = \begin{bmatrix} d & 0 & -c_x & 0 \\ 0 & 1 & 0 & 0 \\ c_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(iii) Además, se propone la siguiente matriz de translación:

$$\mathbf{T}(-\mathbf{p}_0) = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(iv) Para completar la transformación con todo el cálculo previo se tiene la siguiente ecuación:

$$M = T_{inv}(-p_0)R_{x\ inv}(\theta_x)R_{y\ inv}(-\theta_y)R_z(\theta)R_y(-\theta_y)R_x(\theta_x)T(-p_0)$$

Sin embargo, el autor del informe propone el uso de la herramienta Maple el cual es una herramienta para la enseñanza de gráficos en computadoras, su utilidad es demostrada en los siguientes informes de (Kovács, Using Maple in teaching of computer graphics, 1995) y (Kovács, Using some mathematical program in computer graphics teaching, 1996). Utilizando la formula se arroja como respuesta el siguiente valor para M :

$$M = \begin{bmatrix} \cos\theta + c_x^2(1 - \cos\theta) & c_x c_y(1 - \cos\theta) - c_z \sin\theta & c_x c_z(1 - \cos\theta) + c_y \sin\theta & 0 \\ c_y c_x(1 - \cos\theta) + c_x \sin\theta & \cos\theta + c_y^2(1 - \cos\theta) & c_y c_z(1 - \cos\theta) - c_x \sin\theta & 0 \\ c_z c_x(1 - \cos\theta) - c_y \sin\theta & c_z c_y(1 - \cos\theta) + c_x \sin\theta & \cos\theta + c_z^2(1 - \cos\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.7. Rotación alrededor de un plano arbitrario

Rotación alrededor de un plano arbitrario, como se menciona en (Kovács, 2012) puede ser deducida de la sección anterior 3.6. (i) Asumiendo que se tienen tres puntos que pertenecen al plano arbitrario los cuales son $P_0(x_0, y_0, z_0)$, $P_1(x_1, y_1, z_1)$ y $P_2(x_2, y_2, z_2)$. Con estos puntos y lo planteado anteriormente se sigue el siguiente procedimiento:

- Paso 1: Trasladar el plano de reflexión al origen del sistema de coordenadas con la ayuda del punto $P_0(x_0, y_0, z_0)$.
- Paso 2: Realizar rotaciones para hacer que el vector normal del plano de reflexión coincida con el eje $+Z$.
- Paso 3: Reflejar el objeto a través del plano coordenado $Z = 0$.

- Paso 4: Realizar la inversa de la transformación combinada de la rotación del paso número 2.

- Paso 5: Realizar la inversa de la translación del paso 1.

(ii) Se calcula la normal de producto cruz de los vectores $a = P_1 - P_0$ y $b = P_2 - P_0$. Luego se calcula su vector unitario. El cálculo se observa en la siguiente ecuación:

$$n_e = \frac{a \times b}{|a \times b|} = (c_x, c_y, c_z)$$

(iii) En el paso 2 la rotación de matrices será igual como en la sección 3.6. Y se posee una matriz completada la transformación que es la siguiente:

$$M = T(p_0)R_x(-\theta_x)R_y(\theta_y)R_{reflect}^{(x,y)}R_y(-\theta_y)R_x(\theta_x)T(-p_0)$$

Donde el valor $R_{reflect}$ es el siguiente:

$$R_{reflect} = \begin{bmatrix} 1 - 2c_x^2 & -2c_x c_y & -2c_x c_z & -2c_x d \\ -2c_x c_y & 1 - 2c_y^2 & -2c_y c_z & -2c_y d \\ -2c_x c_z & -2c_y c_z & 1 - 2c_z^2 & -2c_z d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde el valor de d es igual a $d = -c_x x_0 - c_y y_0 - c_z z_0$.

4. Objetivo general

Desarrollar un software para la reparación de objetos arqueológicos en 3D basados en su simetría usando los métodos de puntos característicos y simetrías rotacionales, difusión de calor y características locales.

4.1. Objetivos específicos

Los objetivos específicos para lograr el objetivo general propuesto son:

- Objetivo 1:

Implementar los métodos de visualización de un objeto 3D como ayuda de la visualización de la simetría de un objeto.

- Objetivo 2:

Diseñar una interfaz gráfica para la interacción entre el usuario y los métodos de reconstrucción de un objeto arqueológico mediante el uso de simetrías.

- Objetivo 3:

Evaluar la reconstrucción realizada mediante el uso del método extraído de Aproximación de detección de simetrías en Mallas 3D (*Approximate Symmetry Detection in Partial 3D Meshes*) (Sipiran, Gregor, & Schreck, 2014).

4.2. Resultados esperados

- [RE1] Resultado esperado 1 para el objetivo 1: Programa que permita la visualización del objeto arqueológico analizado.
- [RE2] Resultado esperado 2 para el objetivo 1: Visualización de las guías de simetría que engloban al objeto, el paralelepípedo que encierra al objeto.
- [RE3] Resultado esperado 3 para el objetivo 1: Barra de color que indica si un objeto es simétrico en un plano de corte.
- [RE4] Resultado esperado 4 para el objetivo 2: Software integrado con la visualización e interacción con el objeto.
- [RE5] Resultado esperado 5 para el objetivo 2: Visualización de la reconstrucción del objeto e implementación del algoritmo de reconstrucción.
- [RE6] Resultado esperado 6 para el objetivo 3: Medición cuantitativa de

la precisión al reconstruir un objeto 3D mediante la interacción de usuario comparada con los métodos automáticos.

4.3. Herramientas, métodos, metodologías y procedimientos

En esta sección se describen las herramientas y metodologías que se utilizó para cumplir los objetivos específicos que se plantearon en la sección anterior.

En la Tabla 4.1, se muestra un mapeo de las herramientas que se usaron para la obtención de los resultados propuestos en este proyecto de fin de carrera.

Tabla 4.1: Mapeo de herramientas y métodos versus resultados esperados.

Resultado esperado	Herramientas y métodos a usarse
RE1	OpenGL es una herramienta utilizada para desarrollo de aplicaciones graficas interactivas en 2D y 3D. (OpenGL, s.f.)
RE2	CGAL es un proyecto de software que proporciona fácil acceso a los algoritmos geométricos en la biblioteca de C++. (The Computational Geometry Algorithms Library, 2016).
RE3	<p>OpenVDB es una biblioteca de C++ de código abierto que posee un conjunto de herramientas para el almacenamiento y manipulación de datos volumétricos dispersos en mallas tridimensionales. (OpenVDB, s.f.)</p> <p>Qt es un marco de referencia orientado a objetos utilizado para el desarrollo de programas que utilicen una interfaz gráfica de usuario. (Qt, s.f.)</p> <p>Librería boost es un conjunto de bibliotecas preparadas para extender y estandarizar las funcionalidades del lenguaje C++. (Boost C++ Libraries, s.f.)</p> <p>C++ es un lenguaje de programación que permite la manipulación de objetos. Además, es uno de los lenguajes de programación más populares para el manejo de aplicaciones gráficas. (cplusplus, s.f.)</p>

RE4	<p>Qt es un marco de referencia orientado a objetos utilizado para el desarrollo de programas que utilicen una interfaz gráfica de usuario. (Qt, s.f.)</p>
RE5	<p>Método de rayo-superficie (<i>ray casting</i>) es la proyección de un rayo a una superficie. (Roth, 1982)</p> <p>Librería boost es un conjunto de bibliotecas preparadas para extender y estandarizar las funcionalidades del lenguaje C++. (Boost C++ Libraries, s.f.)</p> <p>C++ es un lenguaje de programación que permite la manipulación de objetos. Además, es uno de los lenguajes de programación más populares para el manejo de aplicaciones gráficas. (cplusplus, s.f.)</p>
RE6	<p>Método extraído de (Gregor, y otros, 2014) brinda un número especificando si la reconstrucción buena.</p> <p>C++ es un lenguaje de programación que permite la manipulación de objetos. Además, es uno de los lenguajes de programación más populares para el manejo de aplicaciones gráficas. (cplusplus, s.f.)</p> <p>Librería boost es un conjunto de bibliotecas preparadas para extender y estandarizar las funcionalidades del lenguaje C++. (Boost C++ Libraries, s.f.)</p> <p>TBB ‘Threading Building Blocks’ es una librería desarrollada por Intel que permite escribir programas en paralelos en C++. (Threading Building Blocks (TBB), s.f.)</p>

4.3.1. Herramientas

En esta sección se describen las herramientas utilizadas durante el desarrollo del presente proyecto de fin de carrera.

- **OpenGL:** según la página web de OpenGL (OpenGL, s.f.) (i) es una

herramienta utilizada para desarrollo de aplicaciones gráficas interactivas en 2D y 3D. (ii) Esta interfaz posee muchas funciones para dibujar escenas tridimensionales a partir de geométricas simples. Se utilizará OpenGL con una versión mayor 4.0. La visualización del programa se encuentra en la Figura 4.1.

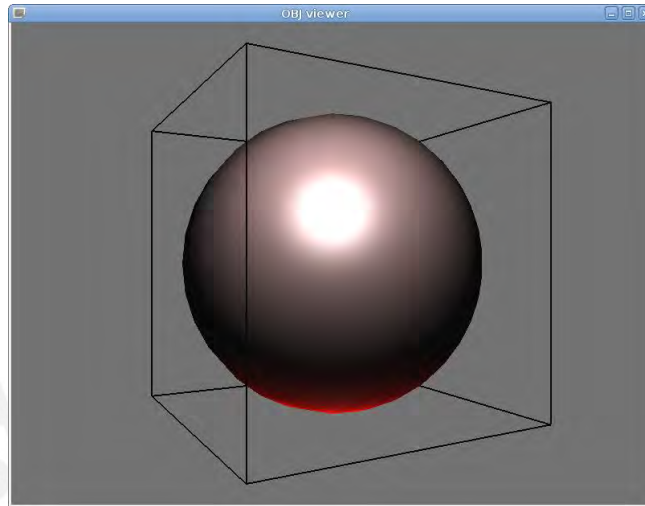


Figura 4.1: Visualización de un objeto 3D. Extraído de (*OpenGL Programming, s.f.*)

- **CGAL:** según la página web de CGAL (The Computational Geometry Algorithms Library, 2016): (i) es un marco de referencia orientado a objetos utilizado para el desarrollo de programas que utilicen una interfaz gráfica de usuario; (ii) es utilizada en varias áreas de necesidad de computación geométrica; y (iii) la librería ofrece muchas estructuras de datos para realizar la visualización de los objetos. La visualización del programa se encuentra en la Figura 4.2.

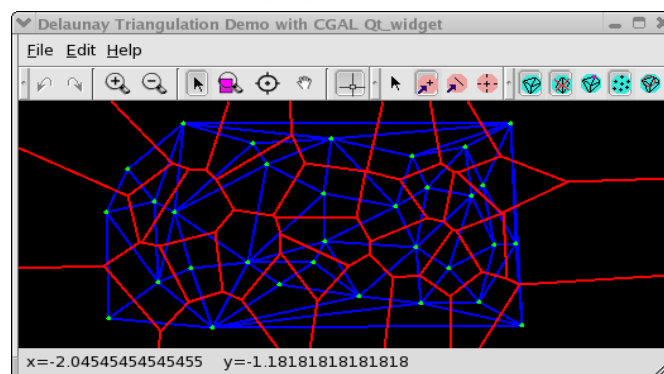


Figura 4.2: Visualización del marco de referencia de CGAL. Extraído de (*CGAL course, s.f.*)

- **Qt:** según la página de Qt (Qt, s.f.): (i) es un marco de referencia orientado a objetos, direccionado para el desarrollo de programas que utilicen una interfaz gráfica de usuario. La visualización del programa se encuentra en la Figura 4.3.

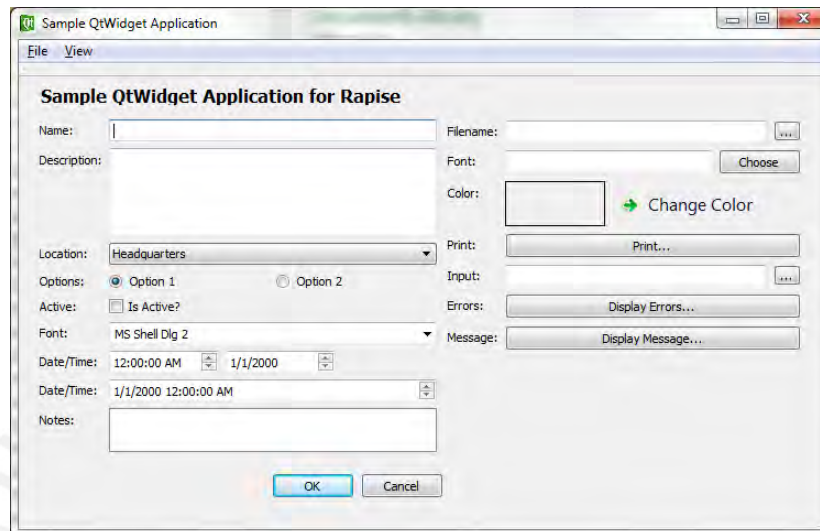


Figura 4.3: Visualización del programa de Qt. Extraído de (*Qt Framework, s.f.*)

- **C++:** según (cplusplus, s.f.): es un lenguaje de programación, que es la evolución del lenguaje de programación C. Además, agrega programación orientada objetos, añadiendo esta característica a su predecesor C. Cabe recalcar que es el lenguaje más popular relacionado a gráficos en computadoras.

Esta herramienta se utilizó para realizar todos los resultados. La interfaz, el algoritmo de reconstrucción y la evaluación estarán realizadas bajo el lenguaje de programación C++.

- **Librería Boost:** según (Boost C++ Libraries, s.f.) es un conjunto de bibliotecas preparadas para extender y estandarizar las funcionalidades del lenguaje.

Esta herramienta se utilizó para realizar todos los resultados y fue de ayuda para el manejo de las estructuras que maneje la interfaz

propuesta en el presente documento.

- **OpenVDB:** según (OpenVDB, s.f.) un código libre del lenguaje de programación C++ que posee un conjunto de herramientas para el almacenamiento y manipulación de datos volumétricos dispersos en mallas tridimensionales.

Esta herramienta se utilizó para poder trabajar los objetos que se desean analizar, por lo que es de mucha ayuda para los resultados esperados número 1, 2 y 3. Cabe recalcar que la terminología que trabaja la librería son mallas tridimensionales para referirse a la estructura de un objeto.

- **TBB ‘Threading Building Blocks’:** según (Threading Building Blocks (TBB), s.f.) es una herramienta desarrollada por Intel que permite aprovechar el paralelismo de los procesadores. Además, permite realizar asignaciones escalables de memoria y programación de tareas.

Esta herramienta permitió ejecutar las funciones y estructuras de manera eficiente del resultado esperado 6.

4.3.2. Métodos

A continuación, se presentan los métodos que se emplearon para la reconstrucción del objeto analizado, la interacción con el usuario y el método de cuantificación del resultado de la reconstrucción del objeto.

- **Método de rayo-superficie** es la proyección de un rayo a la superficie de un objeto y permite resolver gran variedad de problemas relacionados con gráficos en computadoras y geometría computacional (Roth, 1982)
- **Método extraído de** (Gregor, y otros, 2014), el resultado de este método es brindar un número en el cual se indique qué tan buena fue

la reconstrucción del objeto utilizando la interacción del usuario o si fue mejor reconstruirlo utilizando métodos automáticos. El criterio de evaluación es la congruencia entre el objeto reparado y el objeto original completo. En las cuales mediante el método de evaluación se poseerán valores para analizar la congruencia que van desde cero (en caso el objeto original no concuerda en nada con el reparado) a uno (en caso el objeto original calza correctamente con el reparado). Además, se tiene la medida del grado de geometría faltante el cual indica cuanto volumen se ha sido removida al objeto. Por lo tanto, mientras más bajo sea este valor, más difícil será su restauración.



5. Alcance y limitaciones

En esta sección, se explica el alcance que tendrá el proyecto de tesis con respecto a la problemática planteada. Además, se explica las limitaciones que influyeron en la realización y finalización del presente proyecto de tesis.

5.1. Alcance

Este proyecto de fin de carrera presenta una alternativa de solución al problema de reconstrucción de piezas arqueológicas. Es decir, ayuda a la reconstrucción de piezas arqueológicas tomando como apoyo la información que el usuario pueda brindar. Además, facilita la decisión si la información brindada por el usuario es óptima para la reconstrucción del objeto. Para lograr este objetivo, se tomaron en cuenta datos de entrada que serán importantes para la adecuada reconstrucción. Estos datos de entrada son la malla que representa al objeto arqueológico y los datos que pueda ingresar el usuario en la interacción con esta malla.

A continuación, se procedió con la implementación de los métodos de visualización de los objetos. Luego, se diseñó la interfaz gráfica que permite la interacción con el usuario. En este paso es donde se procedió a utilizar los métodos de reconstrucción de objetos mencionados previamente. Y finalmente se realiza un análisis para observar si la reconstrucción utilizando los datos ingresados fue del todo correcta.

Se tiene como información de entrada para las pruebas de la interfaz los datos de objetos arqueológicos provenientes del Museo Larco (Museo Larco, s.f.). Además, cada objeto posee más de 500 mil vértices con aproximadamente más de 1 millón de aristas. Vale la pena mencionar que los objetos pertenecen a la cultura Inca, quienes no conocían sobre el uso de la técnica de la rueda para la fabricación de cerámicos.

Los objetos que se analizaron poseen un posible plano de reconstrucción paralelo a algún plano cartesiano.

5.2. Limitaciones

La siguiente lista muestra las limitaciones de este proyecto de fin de carrera:

- El tiempo de ejecución de los métodos de reconstrucción puede variar dependiendo de la malla del objeto a analizar y de las características del hardware de la computadora. Esto se debe a la complejidad computacional de los métodos que se utilizan para la reconstrucción.
- El tiempo de reconstrucción del objeto puede verse afectado por los datos de entrada por parte del usuario, ya que estos datos puede que no ayuden para la detección de simetrías y por lo tanto no brinda ayuda para la reconstrucción del objeto.
- La visualización de los objetos y su interacción con estos dependió de las características que poseía el computador.
- De los tres métodos que se utilizan para la reconstrucción de piezas arqueológicas no se tiene una manera de poder clasificar qué método es mejor de aplicar para ciertos datos de entrada.



6. Justificación y viabilidad

En esta sección, se detalla la justificación y viabilidad del proyecto de tesis. Con la justificación presentada se da sustento del proyecto de tesis.

6.1. Justificación del proyecto de fin de carrera

El presente proyecto de fin de carrera, surge por la dificultad de reconstruir piezas arqueológicas que no se encuentren completas en su totalidad. Existen un conjunto de métodos que tienen como objetivo principal detectar si el objeto es simétrico y la mayoría de métodos no brinda una solución óptima en la mayoría de los casos. Al no poseer el objeto completo y que muchos de los métodos no brinden solución, se afecta negativamente el proceso de reconstrucción de objetos. Por lo tanto, el principal objetivo del proyecto de fin de carrera es brindar una alternativa de solución mediante la implementación de un software para la reparación de objetos arqueológicos en 3D basados en su simetría, permitiendo la interacción del usuario en el proceso de reconstrucción de piezas arqueológicas y evaluar la reconstrucción del objeto con la interacción por parte del usuario.

Finalmente, el proyecto contribuye en la reconstrucción de las piezas arqueológicas ayudando a los arqueólogos a no realizar maquetados manuales de la reconstrucción de la pieza arqueológica, proceso que podría tomar más de un día o no tener una reconstrucción de manera efectiva.

6.2. Análisis de viabilidad del proyecto de fin de carrera

En el siguiente apartado se presentan factores que influyen en la viabilidad del proyecto de fin de carrera.

6.2.1. Viabilidad técnica

Se cuenta con los conocimientos necesarios para alcanzar los objetivos del presente proyecto de fin de carrera. Se han llevado cursos a lo largo de carrera de Ingeniería informática en los que se han empleado el lenguaje C++ y manejo de visualización de objetos en 3D.

Además, se cuenta con el apoyo por parte del asesor que domina los temas, métodos y técnicas que se utilizaron para este proyecto de fin de carrera.

6.2.2. Viabilidad temporal

Con respecto a la viabilidad temporal, el presente proyecto debe de cumplir el tiempo de un año. Durante el periodo de 1 año se debe de asignar tareas y seguir el cronograma establecido por los cursos de Proyecto de tesis 1 y Proyecto de tesis 2.

6.2.3. Viabilidad económica

El presente proyecto de fin de carrera no presentará costos debido a que las herramientas necesarias para la ejecución del proyecto.

Las herramientas mencionadas anteriormente como OpenGL, CGAL, OpenVDB, Qt, Librería boost y TBB 'Threading Building Blocks' herramientas de código abierto por lo cual no añade un costo al proyecto de tesis. Además, los objetos escaneados fueron brindados por el Museo Larco.

6.3. Conclusión

En conclusión, se cuenta con los conocimientos necesarios para iniciar el proyecto de fin de carrera el cual puede ser completado con la documentación disponible. Las herramientas usadas se encuentran disponibles. Adicionalmente, se cuenta con el apoyo del asesor, el cual conoce muy bien las herramientas y metodologías. En otras palabras, se pudo afirmar que el proyecto de fin de carrera si era viable.

7. Implementación de la interfaz gráfica y visualización del objeto

En este capítulo se presenta los resultados obtenidos y las decisiones tomadas durante el desarrollo del proyecto de fin de carrera. Es decir, se describe el proceso realizado para la entrega del producto final.

7.1. Programa que permita la visualización del objeto arqueológico analizado

Este resultado consiste en el desarrollo de una interfaz gráfica que reciba como entrada un archivo (.OFF) con los puntos característicos de un objeto y permita al usuario visualizarlo en 3D. Esto se logró mediante la implementación de la visualización del objeto considerando la luminosidad para brindar una percepción más realista. Con ello, esta interfaz tiene como finalidad permitir el desarrollo de los objetivos que se presentan en las siguientes secciones de este documento. Conviene subrayar, que uno de estos objetivos fue que la interacción del usuario con la interfaz posea una curva de aprendizaje baja; por consiguiente, facilitará la utilización del software de reconstrucciones basado en el algoritmo.

En esta sección, se explica el proceso de desarrollo que se llevó a cabo y los resultados obtenidos para alcanzar el resultado esperado número 1.

7.1.1. Desarrollo

Para la obtención del resultado esperado 1, se siguieron los siguientes pasos en secuencia:

- Realizar la interfaz gráfica mediante la herramienta Qt.
- Implementar las clases principales como la que representa el objeto, el contenedor de estos objetos, entre otras.
- Implementar los métodos de lectura del objeto.
- Implementar el método que permita la visualización del objeto.
- Implementar la interacción entre el ratón y el objeto, permitiendo la rotación de este último.

7.1.2. Resultado obtenido

Luego de realizar cada uno de los puntos anteriormente mencionados, se

obtuvo la interfaz gráfica con la visualización del objeto correspondiente. Asimismo, se detallan cada una de las vistas correspondientes a la interfaz.

- **Visualización de la interfaz**

A continuación, se muestra la Figura 7.1. La cual es el resultado final de la interfaz que se utiliza, mostrando las diferentes funcionalidades que posee. Entre ellas, el módulo de lectura del objeto 3D.

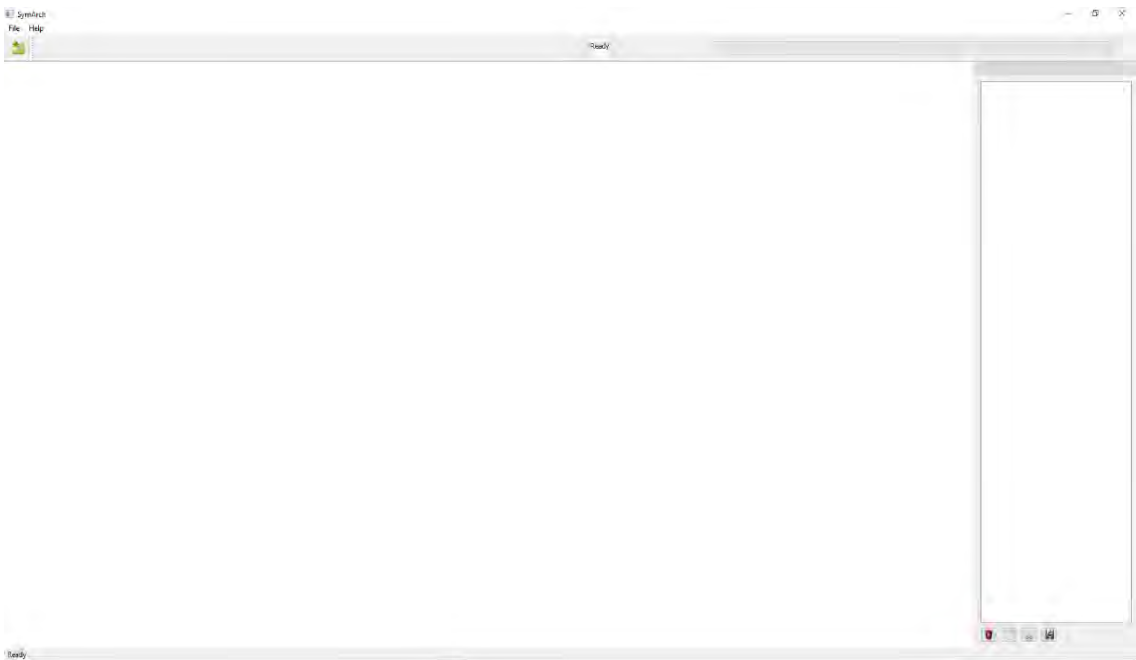


Figura 7.1: Interfaz inicial

En la Figura 7.2, se observa la funcionalidad de lectura. Primero se selecciona la opción “**File**” ubicada en el menú de tareas. Luego, se selecciona el archivo con formato “**.off**” que uno desee. Cabe recalcar que el archivo debe poseer el formato “**.off**”, documento que posee las aristas y vértices del objeto que se analizará.

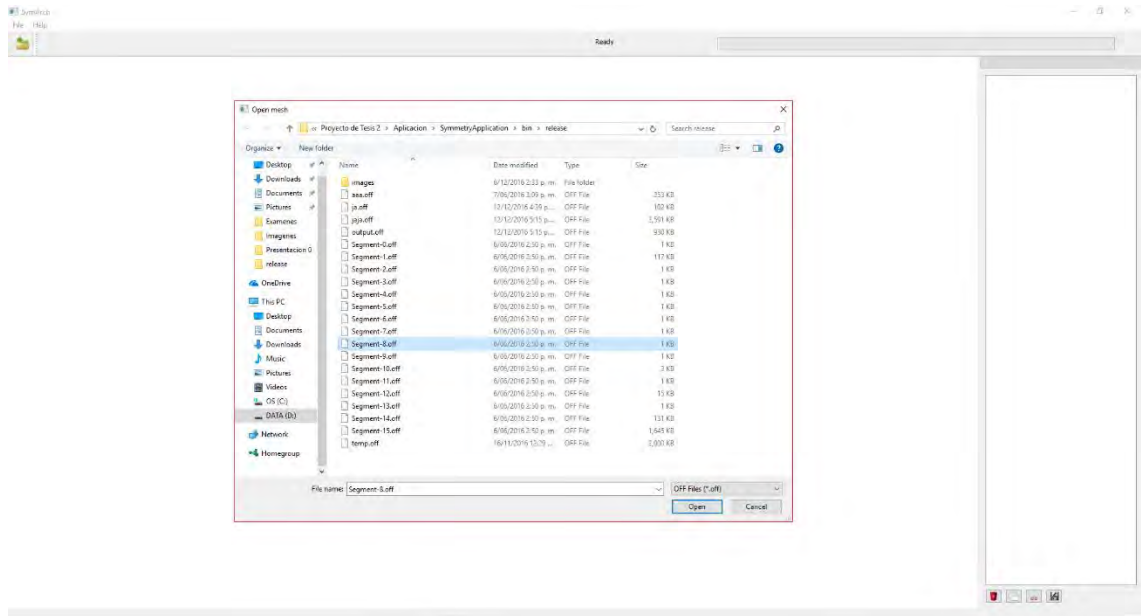


Figura 7.2: Módulo de lectura que ofrece la interfaz.

Después de haber seleccionado el archivo, en la Figura 7.3 se observa la barra de color verde. Esta nos indica el porcentaje que ha llevado la lectura de este archivo y cuanto falta para que se pueda visualizar el objeto.

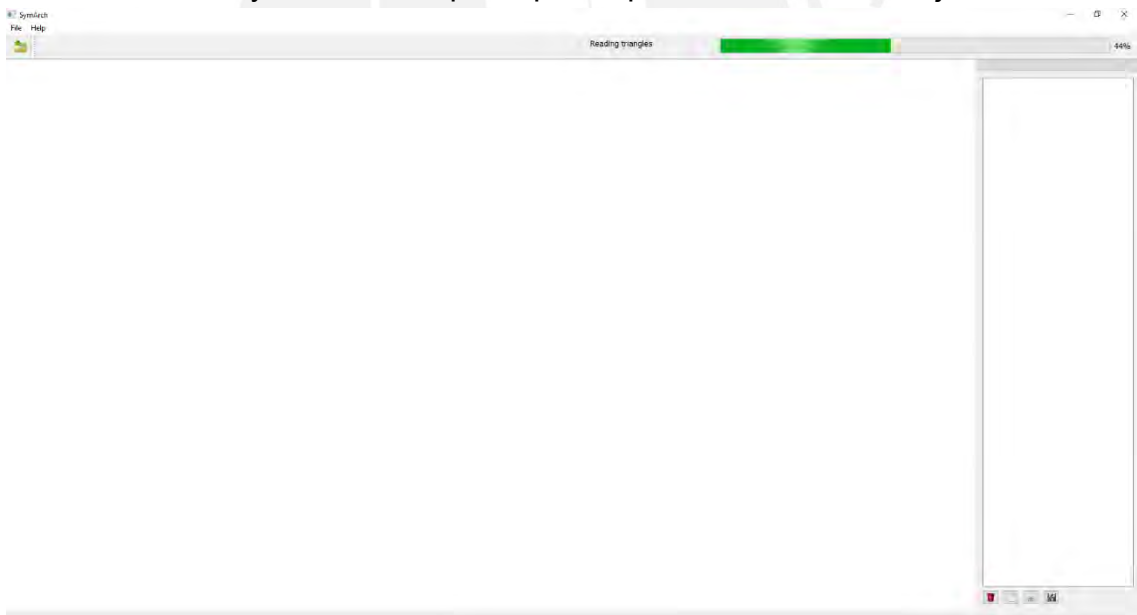


Figura 7.3: Lectura del objeto 3D.

- Visualización del objeto

En las siguientes imágenes, se muestran el resultado obtenido por parte de la implementación de la visualización.

En la Figura 7.4, se observa el resultado de la lectura del archivo (.OFF)

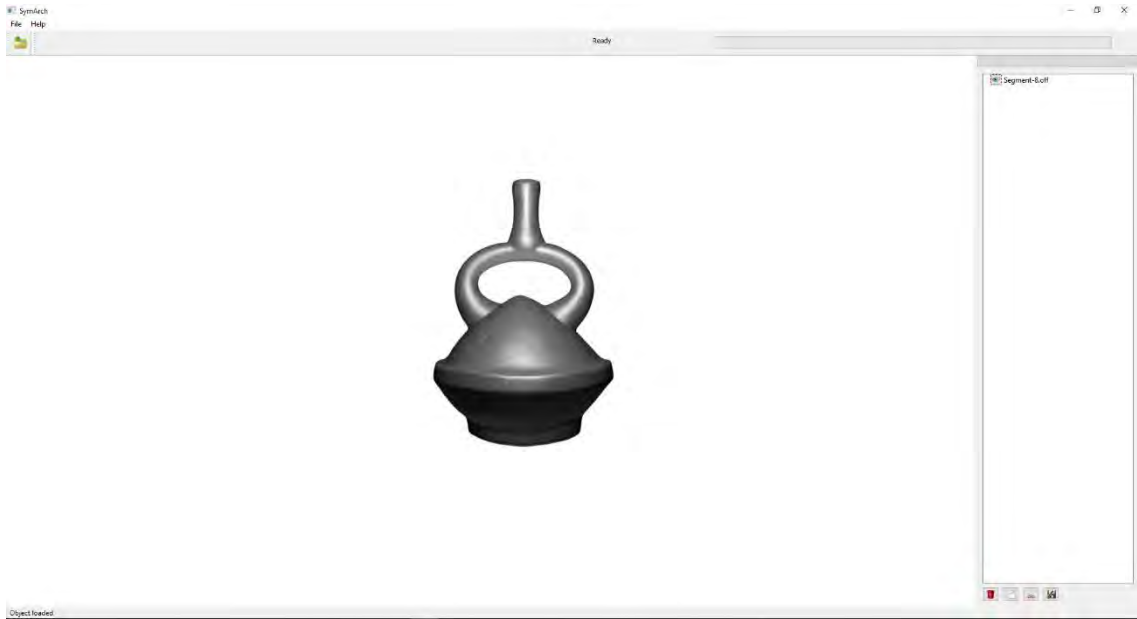


Figura 7.4: Visualización del objeto 3D.

Además, se puede observar la textura, brillo y luminosidad del objeto, que brinda mayor realismo a su representación.

Finalmente, la animación 3D del objeto permite que este pueda ser rotado en todos los ángulos posibles para poder observar al objeto desde distintas vistas. Esto se puede observar en la Figura 7.5.

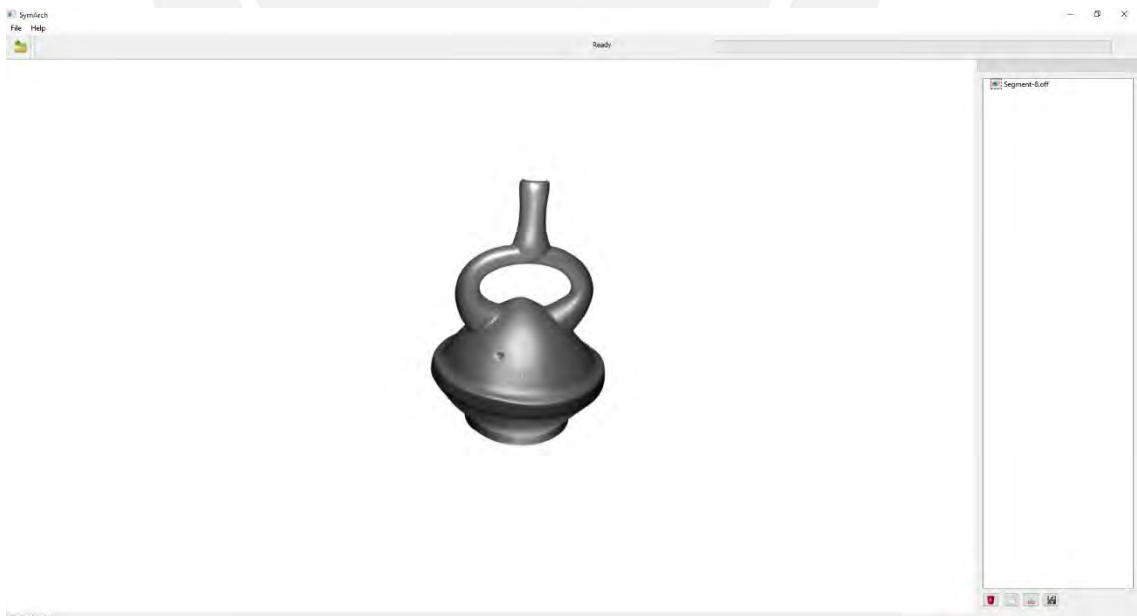


Figura 7.5: rotación del objeto 3D

7.2. Visualización de las guías de simetría que engloban al objeto, el paralelepípedo que encierra al objeto.

Este resultado consiste en el desarrollo de guías de simetrías correspondientes al objeto. Estas tendrán la forma de un paralelepípedo que encerrará al objeto en su totalidad. En otras palabras, este paralelepípedo servirá como referencia para que el usuario pueda trazar posibles planos de simetría en el objeto.

7.2.1. Desarrollo

Para la obtención del resultado esperado 2, se siguieron los siguientes pasos en secuencia:

- Calcular la longitud máxima del objeto en sus tres dimensiones, permitiendo definir las dimensiones del paralelepípedo.
- Implementar guías de simetría en forma de paralelepípedo.
- Implementar la actualización de las guías de simetría cuando se rote el objeto.

7.2.2. Resultado

- **Visualización de guías de simetría**

En primer lugar, se realizó un paralelepípedo que contenga al objeto; sin embargo, este tuvo problemas en relación con el movimiento del objeto. Por ejemplo, cuando se realizaba algún movimiento con el ratón sobre el objeto, este rotaba pero el paralelepípedo no mantenía sincronía con el objeto. Es decir, no se lograba apreciar una sincronía entre objeto y guía de simetría.



Figura 7.6: Visualización de las guías de simetría que encierran al objeto.

Finalmente, se realizó un cambio en el código de visualización de las guías de simetría. Este cambio consistió en que no solo se consideraría el tamaño del objeto para realizar las guías de simetría, sino que se utilizará la posición de los puntos del objeto cada vez que eran mostrados de forma 2D. Por lo tanto, si tomamos un punto en el centro de la base del objeto, este punto cambiará sus coordenadas cada vez que se arrastre el ratón, pero se moverá con el objeto haciendo que este no pierda su forma original. Se siguió la misma lógica para las guías de simetría y se eliminó ese efecto de desincronización. Podemos apreciar en la Figura 7.6 las guías de simetría y en la Figura 7.7 como estas guías de simetría se mueven en relación al objeto.



Figura 7.7: Objeto 3D rotado y asociado con la guías de simetría

7.3. Barras de color que indica si un objeto es simétrico en un plano de corte

Este resultado consiste en el desarrollo de barras de color que tendrán una forma cilíndrica y su funcionalidad es representar qué regiones del objeto son simétricas si se les traza un plano de corte. Este comportamiento es representado en la Figura 7.8.

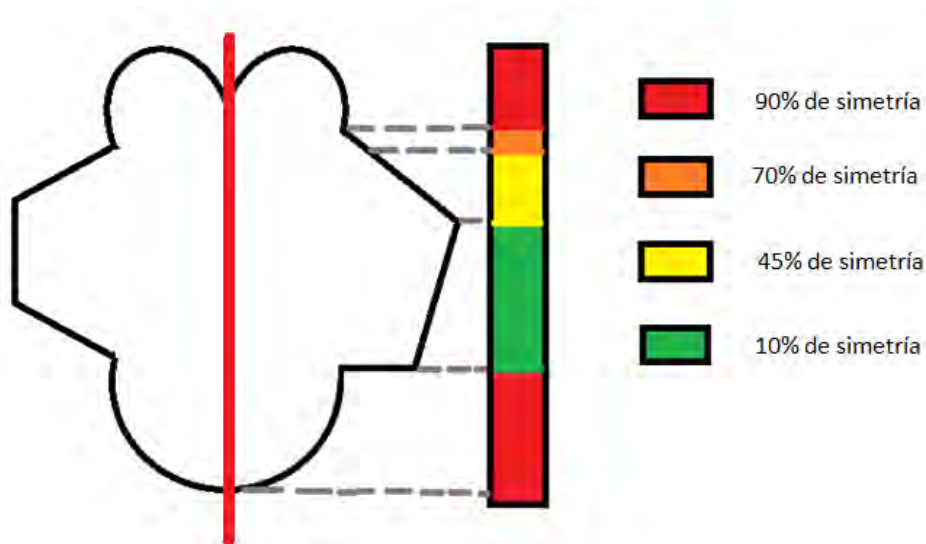


Figura 7.8: Representación de la barra de color.

En la Figura 7.8, se representa el uso de la barra color con respecto a un plano definido con su respectiva leyenda. Con esto, se busca brindar al usuario una ayuda sobre las posibles simetrías que puede tener el objeto analizado.

7.3.1. Desarrollo

Para la obtención del resultado esperado 3, se siguieron los siguientes pasos en secuencia:

- Medir los límites del objeto para cada uno de sus ejes, permitiendo calcular la longitud las barras de color.
- Alinear de forma paralela cada barra de color con los ejes cartesianos, para visualizar la simetría en su ancho, largo y profundidad.
- Dibujar la barra de color con las características mencionadas anteriormente.
- Asignar colores a la barra de color, asignando colores basándose en la Figura 7.8.

7.3.2. Resultado

- **Visualización de barras de simetría**

Se realizó un conjunto de barras, estas poseían las siguientes características:

- Las barras son paralelas a los vértices del cubo que encierra al objeto.
- La longitud de las barras de simetría poseen una altura similar la diferencia entre los puntos más alejados respecto a cada eje cartesiano.
- El diámetro de las barras de simetría es 10% la diferencia entre los puntos más alejados en cualquiera de los ejes de simetría.

Además, la representación de las barras de simetría se encuentra en la Figura 7.9, donde se representa un cilindro con pocos puntos para poder explicar la estructura usada.

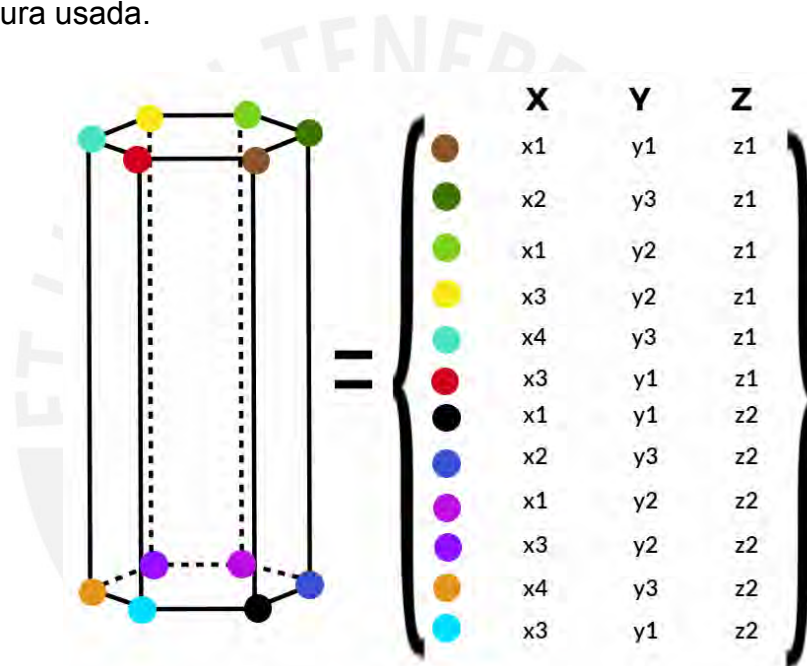


Figura 7.9: Representación del cilindro en un espacio de tres dimensiones

Cabe recalcar que estos puntos siguen un orden de colocación para que más adelante puedan pintarse las caras laterales del cilindro de manera efectiva. Para el almacenamiento de los puntos se utilizó un arreglo, en donde se almacenó los puntos según el orden que muestra en la Figura 7.9 donde los puntos de la parte superior son agregados primero con un z_1 y luego se agregan los de la parte inferior z_2 . Además, se agregó dos nuevos vértices donde se ubica el centro de las tapas superior e inferior del cilindro como se puede observar en la Figura 7.10.

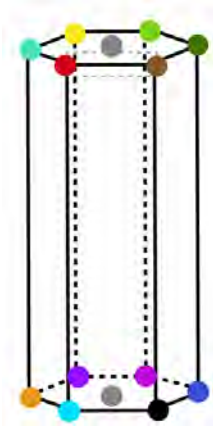


Figura 7.10: Representación de los centros en las tapa superior e inferior.

En la Figura 7.10, estos puntos son colocados para poder pintar de manera más rápida las tapas del cilindro con el método que se explicará más adelante en el presente documento.

Luego con los puntos obtenidos, se realizó la construcción de los triángulos utilizando estos puntos. Estos triángulos son lo que se pintan para poder darle una textura a los cilindros. La creación de los triángulos realizó mediante el método descrito en la Figura 7.11, donde cada punto posee un sentido. Este sentido es importante ya que al calcular la normal del triángulo, este me indicará que cara del triángulo será pintada.

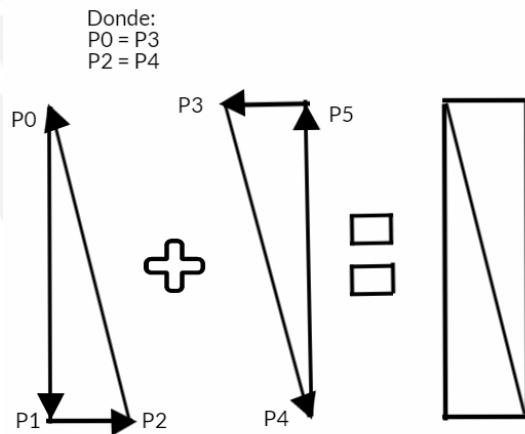


Figura 7.11: Representación de los triángulos que conforman la estructura del cilindro

La representación de la normal de triángulo indica porque lado será pintada esta figura, esta representación se observa en la Figura 7.12, donde el sentido de los puntos que representan al triángulo es muy importante. Estos puntos son almacenados en una estructura que contiene los triángulos relacionados a cada vértice del cilindro.

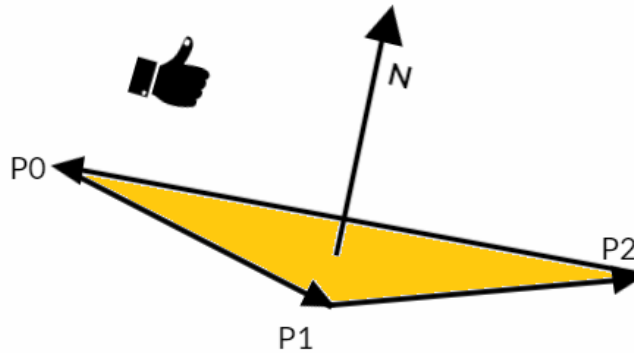


Figura 7.12: La normal del plano que representa al triángulo indica que cara de esta figura será pintada. Siguiendo la regla de la orientación de la mano derecha se indica la dirección en qué orden deben ser colocado los puntos que representan el triángulo.

En la Figura 7.13, se observa un cilindro que representará una de las barras de color. Esas barras se encuentran representadas en cada uno de los ejes cartesianos. Además, se observa que la longitud de las barras de simetría poseen la longitud del objeto 3D. Cabe recalcar que las barras de colores aún no han sido pintadas y se explicará con mayor detalle cómo se realizó el pintado en el siguiente acápite.

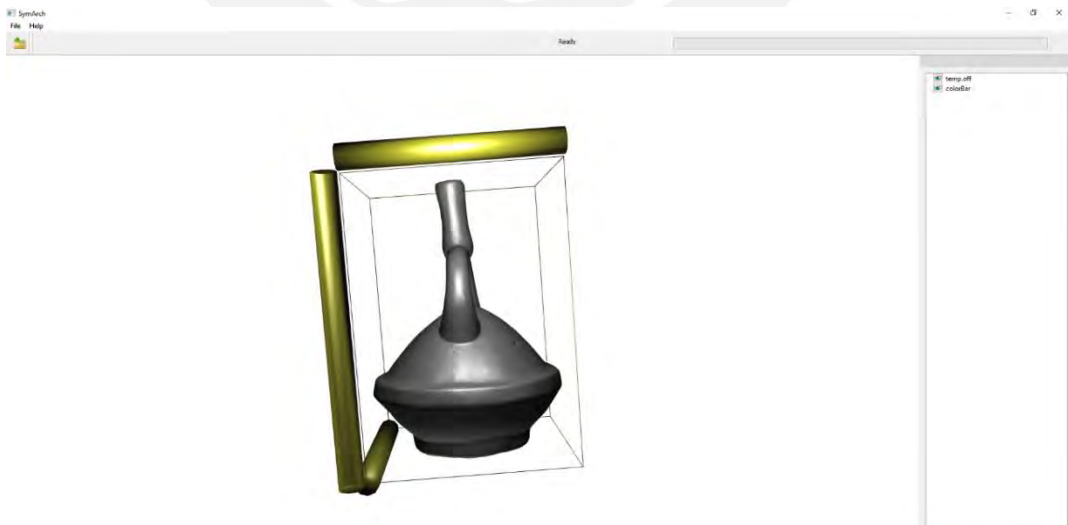


Figura 7.13: Representación de las tres barras de color. Las 3 barras están alineadas a ejes cartesianos.

- Color de las barras de simetría

En primer lugar, se implementó un método el cual tenía como objetivo detectar la simetría que poseía un objeto respecto a cada eje de simetría. Este método analiza el objeto, haciendo que uno de los valores del eje sea constante y los demás varía. Por ejemplo, si el valor de Z era igual a 20, el método analizará todos los valores que estén dentro del rango de $20 \leq Z < 21$. Para estos valores, se analizan los valores de X e Y, verificándose la simetría en ese plano y colocando los colores respectivos a la Figura 7.13. Estos cortes con un plano son representados y explicados en la Figura 7.14 donde se observa los planos de corte realizados a dos cerámicos completamente distintos.

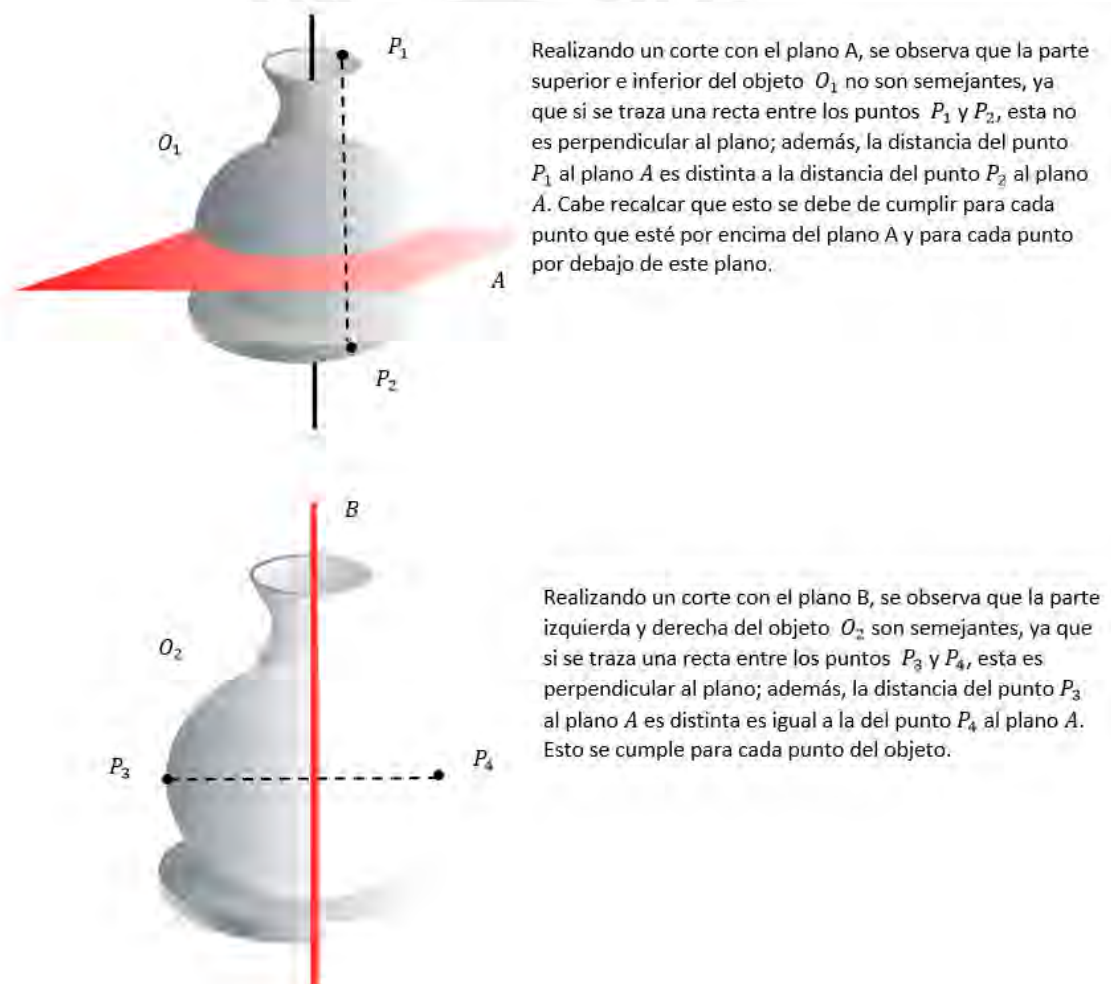


Figura 7.14: Corte del objeto utilizando planos, explicando cuando el plano es óptimo.

El método realiza un recorrido por cada vértice, dentro de este recorrido primero se toman los puntos que posean cierto grado de curvatura superior a 0 y menores a 0.6. Luego se traza una recta entre el punto analizado y todos los puntos que se encuentren al otro lado del plano. Una vez que se realiza este procedimiento se analiza si las distancia entre los dos puntos al plano son iguales. Por lo que si son iguales posee un gran valor de simetría y de esta manera se analiza para cada punto del objeto

El resultado de este método se puede observar en la Figura 7.15, donde se observa la variación entre dos planos Z y se debe al agujero que presenta el objeto visualizado en la Figura 7.16.

```
Application Output
SymArch x
Plano Z = 40 -> La simetría tuvo una aproximación de 85%
Plano Z = 80 -> La simetría tuvo una aproximación de 15%
```

Figura 7.15: Resultado luego de aplicar el método de cálculo de la simetría.

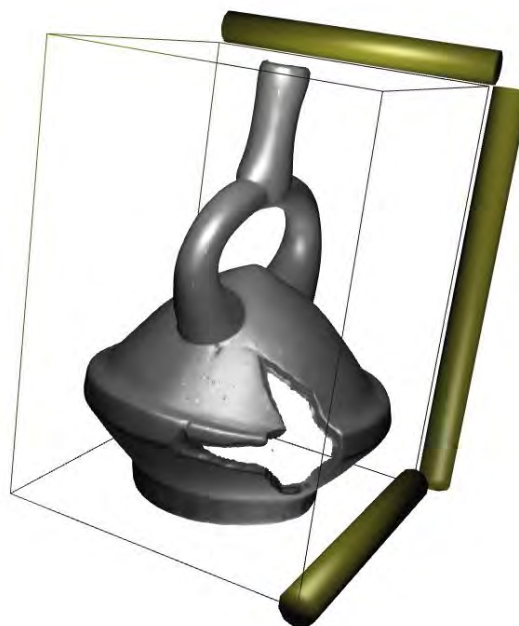


Figura 7.16: Ruptura del objeto analizado

- **Colorear barras de simetría**

Con los vértices que representan a estos cilindros, se realiza el pintado de las barras de simetría, donde se transforma los porcentajes de

aproximación calculados en el punto anterior en colores. Es decir, existe una mayor simetría si la barra muestra color rojo, caso contrario el color será verde.

El método de coloreado de las barras de color se realizó de la siguiente manera:

- Cada barra presenta la estructura presentada en la Figura 7.11, se colocó esa estructura en un arreglo que contenía todos los vértices del objeto.
- Luego se procedió con la colocación del conjunto de puntos en el orden descrito en la Figura 7.11 utilizando la normal como ayuda.
- Finalmente se utilizó un segmento de longitud uno donde se colocó una paleta de colores que esté relacionada con la simetría del objeto.

Finalmente se obtuvo los siguientes resultados que se muestran en la Figura 7.17. Se observa como los colores en las barras de color corresponden a la simetría del objeto. Para este caso las partes de la barra de color verde indican que en el objeto no existe mucha simetría mientras que en las partes rojas existe una mayor simetría.

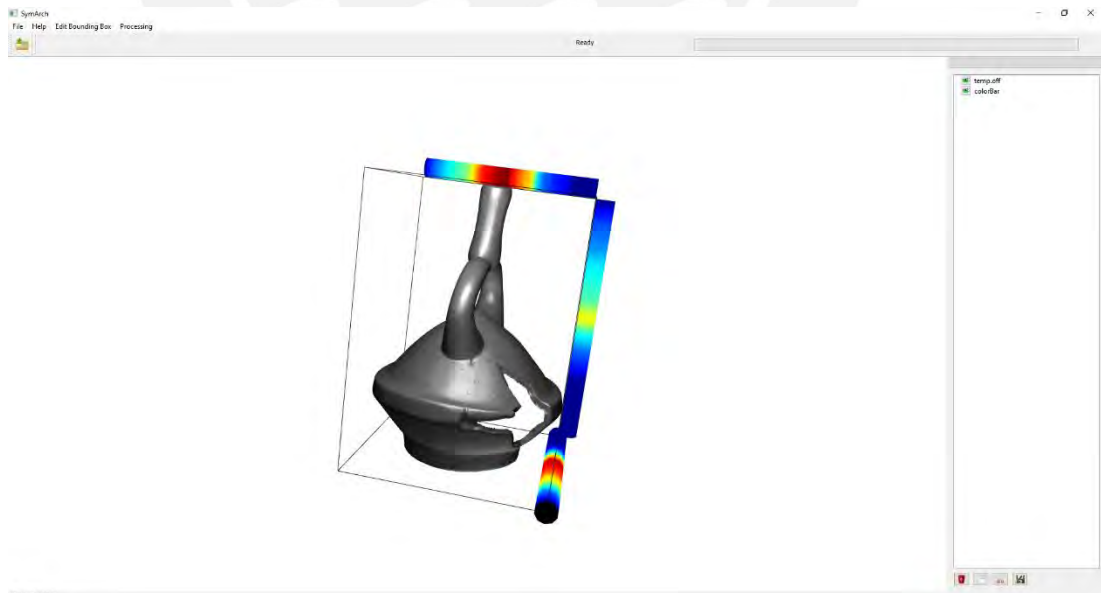


Figura 7.17: Barras de colores donde se muestra la simetría del objeto. Los colores en las barras de simetría indican la simetría del objeto desde cada plano cartesiano.

8. Interacción de entre el objeto analizado y usuario

En este capítulo se presenta la interacción entre el objeto analizado y el usuario. Además, dentro de esta sección se observa la ventaja que existe al utilizar la interacción del usuario. El usuario tendrá como funcionalidad poder interactuar con el paralelepípedo que encierra el objeto y poder escoger los planos más óptimos que en donde el corte de un plano represente la simetría como se explicó en la Figura 7.8. En primer lugar se creó un software integrado con la visualización e interacción con el objeto, en esta interacción se encuentra la funcionalidad de reducción del paralelepípedo realizada por el usuario y la selección de planos que se utilizará para la reconstrucción del objeto.

8.1. Software integrado con la visualización e interacción con el objeto.

Este resultado consiste en la interacción que existirá entre el usuario y la interfaz. El usuario podrá escoger los planos que él considere para la reconstrucción y el volumen del cubo a considerar para la reconstrucción del objeto.

8.1.1. Desarrollo

Para la obtención de este resultado, se siguieron los siguientes pasos en secuencia:

- Reducir el paralelepípedo que contiene al objeto analizado con las medidas que ingrese el usuario.
- Seleccionar el plano para cada eje X , Y y Z por parte del usuario.

8.1.2. Resultado

- **Reducción del paralelepípedo realizada por el usuario**

Con el desarrollo de las guías de simetría descritas en la sección 7.2.2 del presente documento, se realizó la reducción de estas guías mediante la interacción del usuario. Se empezó con la creación del botón que permita esta reconstrucción y solo se encuentra activada si existe un objeto ya previamente visualizado. La opción de reducción las guías de simetría se encuentra

deshabilitado cuando no se posee un objeto en el área de trabajo. Esto se observa en la Figura 8.1.

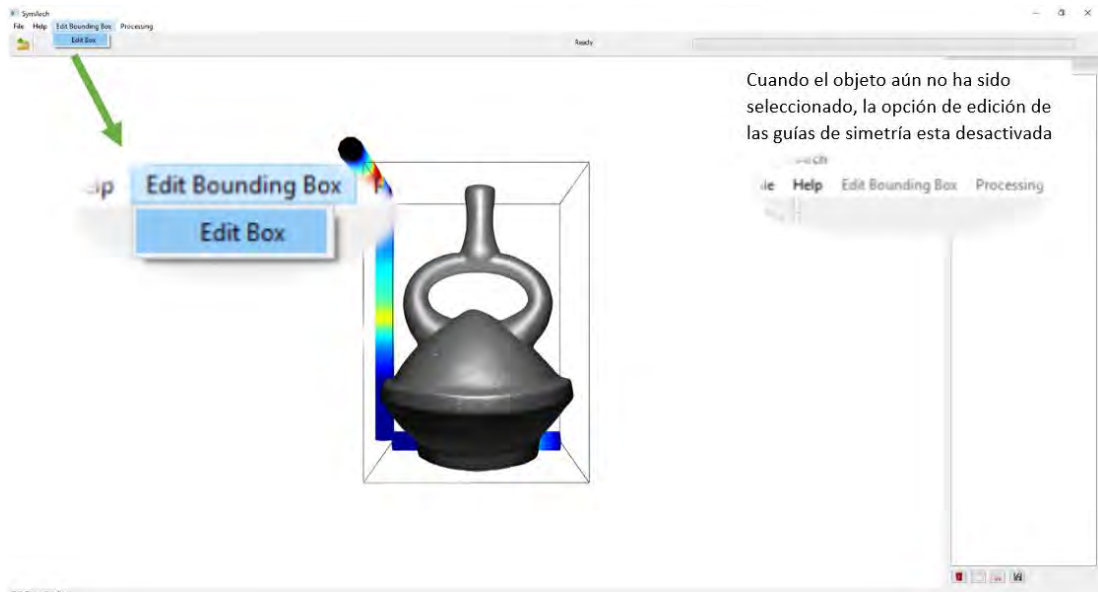


Figura 8.1: Opción de edición de guías de simetría

Para tener este comportamiento se realizó el uso de una variable que era falsa cuando el objeto aún no había sido leído, y verdadera indicando que el objeto ya había sido leído y visualizado en la interfaz.

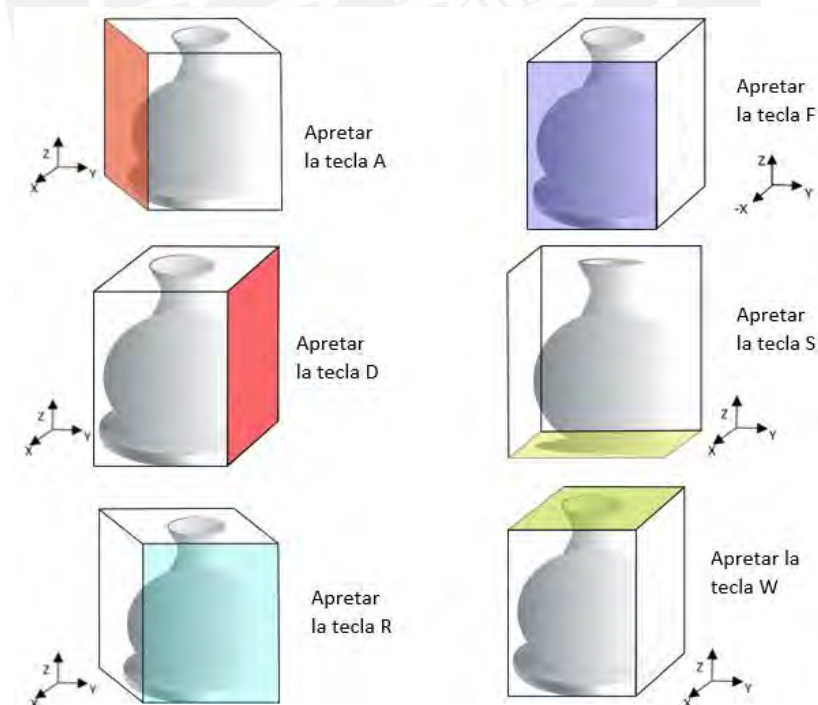


Figura 8.2: Opciones para seleccionar la cara del paralelepípedo que se desea modificar

Una vez el usuario presione la opción de “**Edit Box**” como se observa en la Figura 8.1, la variable mencionada anteriormente cambia de valor a verdadera si y solo se acepta la opción de edición. Una vez que se acepta la opción de editado, se muestra un mensaje indicando las instrucciones que se deben de seguir para realizar la edición de las guías de simetría, las cuales se observan en la Figura 8.2.

Cada sección del paralelepípedo se almacena dentro de una variable, esta variable está asociado a un conjunto de teclas mencionada en la Figura 8.2, brindando al usuario la opción de escoger el plano que desee modificar. El usuario podrá aumentar o disminuir la distancia entre el lado seleccionado y el objeto. Esta funcionalidad se visualiza en la Figura 8.3, donde el usuario disminuyó el lado izquierdo del paralelepípedo. Cabe recalcar que esta funcionalidad está presente para las seis caras del paralelepípedo.

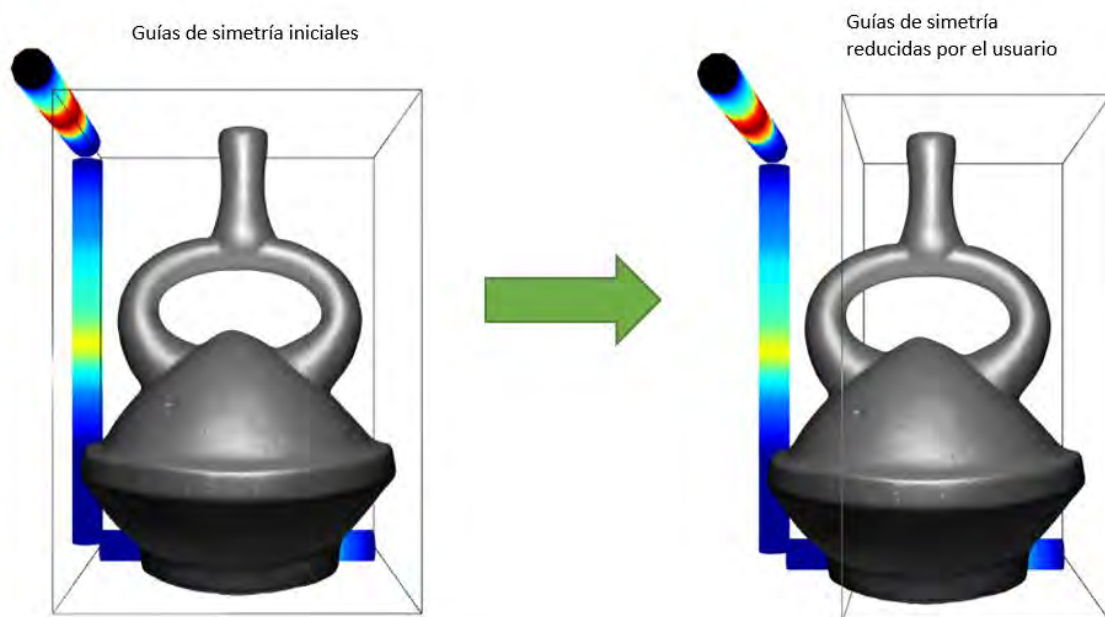


Figura 8.3: Guías de simetría cambiadas por el usuario. Se observa que el usuario disminuyó la cara izquierda del paralelepípedo

- **Selección de planos para la reconstrucción del objeto**

Con las barras de simetría y su respectivo coloreo, se desarrolló la funcionalidad de darle la libertad al usuario de poder escoger los posibles

planos para la reconstrucción del objeto. Primero se creó un botón que permita la selección de los planos de simetría. Esta opción se observa en la Figura 8.3, donde el usuario puede seleccionar la opción de **“Select Planes”** en el menú **“Processing”**. Además, esta opción solo está disponible si el usuario realizó la lectura de un objeto como en el caso de la funcionalidad de

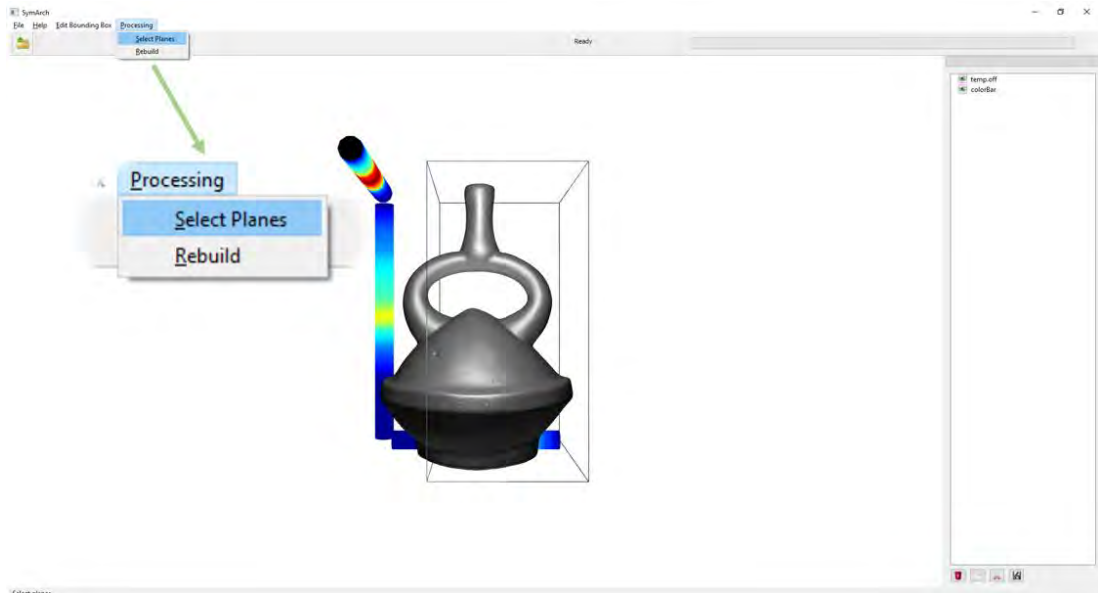


Figura 8.4: Opción que permite la selección de los planos de simetría.

“Reducción del paralelepípedo realizada por el usuario”.

Al realizar esta opción se abre la siguiente venta de mensaje, por lo que el usuario podrá seleccionar un plano que tiene como objetivo la reducción del volumen a analizar. Este mensaje se observa en la Figura 8.5.

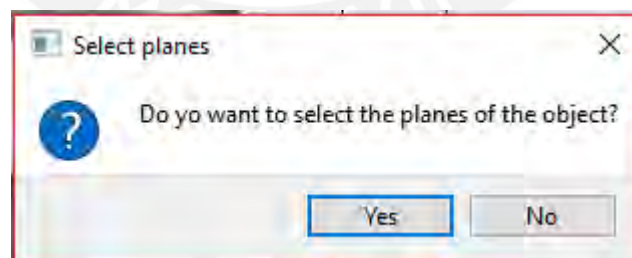


Figura 8.5: Mensaje donde se indica si el usuario desea realizar la selección de los planos.

En caso el usuario seleccione la opción **“No”**, solo realizará la visualización del objeto. Sin embargo, si el usuario escoge la opción **“Yes”** de la Figura 8.5 se procederá a indicar al usuario cuales son los mejores planos con respecto

a cada eje de simetría. En un inicio se mostrará al objeto con sus respectivas barras de simetría y en las barras de simetría se pintará un anillo de color negro, el cual indica el plano de selección que se tiene hasta el momento. Un ejemplo de esta descripción se observa en la Figura 8.6, donde los planos P_1 , P_2 y P_3 son los planos que tiene el usuario hasta el momento. Cuando se realiza la primera llamada a la función “**Select Planes**”, se obtiene lo planos mencionados anteriormente, los mejores planos para escoger simetría respecto a cada eje.

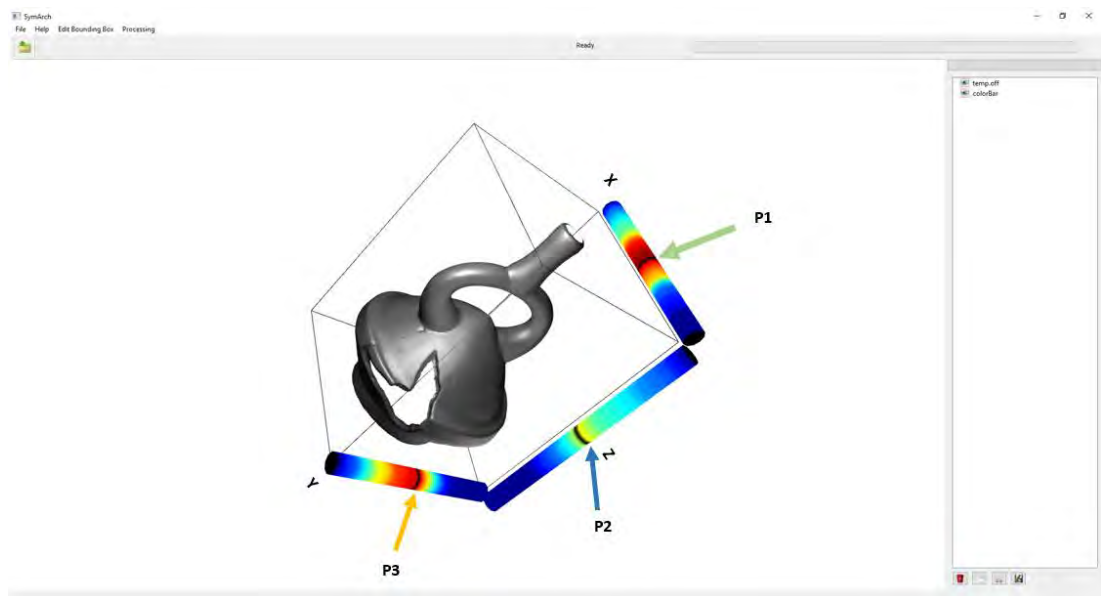


Figura 8.6: Visualización los planos seleccionados hasta el momento. Por defecto estos planos en un inicio son los mejores planos que pueden representar la simetría del objeto en los distintos ejes de simetría.

Luego el usuario podrá seleccionar en que barra desea seleccionar el plano. Para realizar esta funcionalidad el usuario puede seleccionar las teclas X , Y y Z para modificar los anillos de color negro, planos para la simetría, que se encuentran en cada barra. Donde las teclas X , Y y Z modificarán los planos X , Y y Z respectivamente.

Por ejemplo, el usuario desea modificar la barra paralela al eje Z . Para realizar la modificación de este plano apretó la tecla Z y para subir y bajar el anillo que es el plano P_2 como se observa en la Figura 8.6, se selecciona los botones “+” y “-” para poder realizar estas acciones. El resultado de la selección del plano Z se observa en la Figura 8.7, donde se realiza el cambio del plano P_2

a la posición donde apunta la flecha mP_2 , la cual es la posición final del plano seleccionada por el usuario.

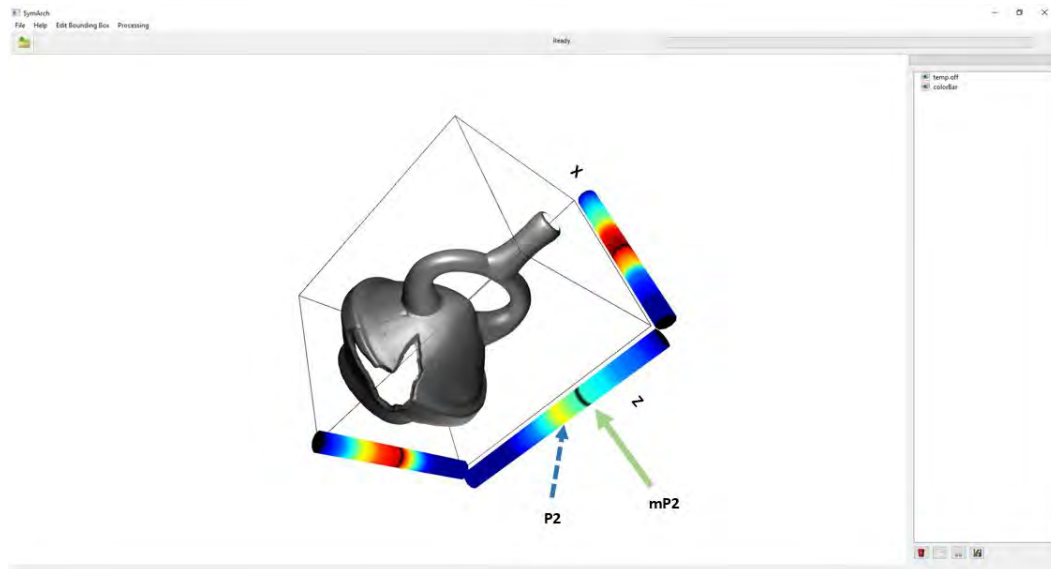


Figura 8.7: Cambio del plano Z, donde P2 era su posición inicial y con la interacción del usuario este plano cambio de posición.



9. Implementación del algoritmo de reconstrucción y análisis del resultado de la reconstrucción del objeto

En este capítulo se presentan los algoritmos que fueron implementados para la reconstrucción del objeto. Se explica su funcionamiento y los pasos en que consiste cada uno de ellos.

9.1. Implementación del algoritmo de reconstrucción

En esta sección se explica la implementación del algoritmo de reconstrucción utilizando la fórmula de rotación de Rodrigues que se encuentra en (Kovács, 2012). El algoritmo de reconstrucción de un objeto basado en un plano arbitrario es el siguiente:

```
REPLICAR_GEOMETRIA_SIMETRICA (plano, eje, coord, v, t)
    punto_referencia <- CREAM_PUNTO_REFERENCIA (eje)
    vertices_hom <- HOMOGENIZAR_VERTICES (v)
    T <- CREAM_MATRIZ_TRANSLACION (punto_referencia)
    Tinv <- CREAM_MATRIZ_TRANSLACION_INV (punto_referencia)
    M <- ASIGNAR_COEFICIENTES_PLANO (plano)
    N <- (0, 0, 1)
    ejeV, c <- OBTENER_VALORES_PREVIOS (M, N)
    x, y, z <- OBTENER_COORDENADAS (ejeV)
    s <- OBTENER_RAIZ_CUADRADA (c)
    tt <- 1 - c
    Q <- MATRIZ_MAPLE (tt, s, x, y, z)
    Qinv <- MATRIZ_MAPLE (tt, -s, x, y, z)
    Ref <- MATRIZ_REFLEXION ()
    TransfBase <- Tinv * Qinv * Ref * Q * T
    vertices_rotados <- TransfBase * vertices_hom
    vertices_salida <- ASIGNAR_VERTICES (vertices_rotados)
    triangulos_salida <- ASIGNAR_TRIANGULOS (t)
    CALCULAR_NORMALES (vertices_salida, triangulos_salida)
```

Figura 9.1: Seudocódigo del algoritmo de reconstrucción

- **Parámetros de entrada**

En la función de la Figura 9.1 se tiene 5 parámetros de entrada, cada uno representa lo siguiente:

1. **plano:** este es el plano arbitrario que el usuario seleccionó para realizar la reconstrucción del objeto y posee la forma de $Ax + By + Cz + d = 0$.
2. **eje:** el eje al que pertenece el plano (Si es X el eje es 1, si es Y el eje es 2 y si es Z el eje es 3).
3. **coord:** Es un número el cual indica el corte del plano escogido con el eje.
4. **v:** los vértices del objeto original al que se le aplica el algoritmo. Donde **v** posee la siguiente estructura $P_i = (x_i, y_i, z_i)$ para cada vértice i del objeto original.
5. **t:** son triángulos que su conjunto forman al objeto. Son las relaciones de cada vértice para poder conformar el objeto a analizar.

- **Creación de punto de referencia**

Se crea un punto de referencia que ayuda a la creación de matrices en los siguientes puntos. Como parámetro de entrada se tiene la variable **eje**, dependiendo que valor tenga esta variable se tomarán ciertas acciones para la creación del punto de referencia. Si el valor de la variable **eje** es 1 (El eje es X) entonces la función retornará $(1, 0, 0)$, en caso el valor de **eje** sea 2 (El eje es Y) entonces la función retornará $(0, 1, 0)$. Finalmente, si el valor del **eje** es 3 (El eje es Z) entonces la función retornará $(0, 0, 1)$.

- **Homogenización de vértices**

Esta función posee como parámetros de entrada los vértices del objeto al cual se está analizando. En la Figura 9.2 se muestra el pseudocódigo para la

homogenización de los vértices.

```
HOMOGENIZAR_VERTICES (v)

    W <- 1

    hom_vertices: <4, CANTIDAD_VERTICES (v)>

    indice <- 0

    Para cada  $V_i:(x_i, y_i, z_i)$  en v:

        INSERTAR_VERTICES_HOM (hom_vertices, índice,  $V_i$ , W)

        indice <- índice + 1
```

Figura 9.2: Seudocódigo de la función de homogenización de vértices

Se crea una variable W igual a 1, esta será la que se añadirá a cada como cuarto parámetro a la conjunto de vértices. También se posee una matriz $hom_vertices$ con dimensiones $4 \times cantidad_vertices$, dentro de esa matriz se tendrá cada vértice homogenizado.

Se recorre cada vértice del objeto analizado, y dentro de este bucle se inserta el vértice homogenizado en la variable $hom_vertices$.

Para poder insertar un vértice homogenizado, se utiliza el procedimiento **INSERTAR_VERTICES_HOM**, este procedimiento recibe como parámetros la variable $hom_vertices$, la variable W que es 1, el índice proveniente de bucle y un vértice. Entro de la función se crea un insertan en la columna i los valores de $V_i:(x_i, y_i, z_i)$ dentro de cada fila. El último valor de la fila debe ser completado con la variable W . La inserción se muestra en elseudocódigo de la Figura 9.3.

```
INSERTAR_VERTICES_HOM (hom_vertices, índice,  $V_i$ , W)

    hom_vertices[0][índice] =  $x_{índice}$ 

    hom_vertices[1][índice] =  $y_{índice}$ 

    hom_vertices[2][índice] =  $z_{índice}$ 

    hom_vertices[3][índice] =  $W$ 
```

Figura 9.3: Seudocódigo de la función de inserción de vértices homogéneos

- **Creación de matriz de translación y su inversa**

Se crea una matriz de translación que se asignará a la variable T y su respectiva matriz inversa T_{inv} . Ambas matrices poseen la misma dimensión 4×4 y poseen la siguiente estructura que se observa en la **(Figura 9.4)**. Las variables X, Y y Z dentro de las matrices pertenecen a las coordenadas del parámetro de la función el cual es el punto de referencia calculado en pasos anteriores.

$$T = \begin{matrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{matrix} \quad T_{inv} = \begin{matrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{matrix}$$

Figura 9.4: La matriz de la izquierda es la matriz de translación T y la matriz de la derecha es la matriz de translación inversa T_{inv}

- **Asignación de coeficientes del plano seleccionado**

Esta función tiene como objetivo asignar los coeficientes de la ecuación del plano seleccionado, donde la ecuación general de un plano P es $Ax + By + Cz + D = 0$. Siendo el plano P el parámetro de entrada. Asimismo, se tomará en cuenta solo planos que sean perpendiculares a cada eje del plano cartesiano, por lo que la cantidad de valores que pueden tomar los coeficientes del plano seleccionado son reducidos a tres tipos de casos. Los casos son los siguientes:

- Cuando el plano P es paralelo al plano XY los valores de los coeficientes A y B serán iguales a cero. Y el valor de retorno de la función será un punto con la siguiente estructura $(0, 0, 1)$
- Cuando el plano P es paralelo al plano XZ los valores de los coeficientes A y C serán iguales a cero. Y el valor de retorno de la función será un punto con la siguiente estructura $(0, 1, 0)$
- Cuando el plano P es paralelo al plano YZ los valores de los coeficientes B y C serán iguales a cero. Y el valor de retorno de la función será un punto con la siguiente estructura $(1, 0, 0)$

- **Obtener el cálculo de valores previos a la función de maple**

Esta función se encarga de algunos cálculos previos para un mejor manejo dentro de la función de Maple (Kovács, 1995) que se revisa en los siguientes pasos del pseudocódigo de la función **REPLICAR_GEOMETRIA_SIMETRICA** Figura 9.1. Como datos de entrada esta función recibe dos parámetros que son los puntos M y N . Dentro de esta función se posee una serie de cálculos como se muestra en el pseudocódigo de la Figura 9.5.

```
OBTENER_VALORES_PREVIOS (M, N)

  prod_cruz <- M x N
  prod_punto <- M * N
  ejeV <- prod_cruz / NORMALIZAR (prod_cruz)
  c <- prod_punto / ( NORMALIZAR (M) * NORMALIZAR (N) )
```

Figura 9.5: Pseudocódigo de la función para la obtención de valores previos a la función de Maple

En la primera línea de la función de la Figura 9.5 se calcula el producto cruz de vectores de los puntos M y N , obteniéndose como resultado otro vector. En la siguiente línea se calcula el producto punto de vectores y se obtiene como resultado un valor escalar. Todo este cálculo se realizó para que las operaciones de las siguientes líneas sean más inteligibles para el lector.

Con los datos calculados en los pasos previos de la función **OBTENER_VALORES_PREVIOS** se procede a calcular los valores para las variables de salida $ejeV$ y c . Para obtener el valor de la variable $ejeV$ se divide el vector **prod_cruz** entre la normalización del vector **prod_cruz**, obteniéndose como resultado un vector unitario que posee la misma dirección que el vector **prod_cruz**. Cabe recalcar que la función normalización tiene como resultado $\sqrt{x^2 + y^2 + z^2}$ donde (x, y, z) es el parámetro de entrada.

Para obtener el valor de la variable c se divide la variable escalar **prod_punto** entre producto de las normalizaciones de los puntos M y N .

- **Obtención de coordenadas**

Posee como parámetro de entrada la variable **ejeV**, da como resultado las coordenadas del punto **ejeV**.

- **Obtención de la raíz cuadrada**

La función arroja como resultado a diferencia entre 1 menos el parámetro de entrada de la función **OBTENER_RAIZ_CUADRADA** al cuadrado.

- **Creación de la matriz de maple**

Se crea una matriz de 4 x 4, donde cada uno de sus valores sigue la estructura de la función maple explicada en la sección 3 Marco conceptual. Esta función tiene como parámetros de entrada la variable **tt** que es 1 menos el valor **c** calculado anteriormente, la variable **s** que es el resultado de la función **OBTENER_RAIZ_CUADRADA**, y los valores de las coordenadas del punto **ejeV** que se obtuvieron en la función **OBTENER_COORDENADAS**. La estructura de la matriz se observa en la siguiente Figura 9.6.

$$Q = \begin{bmatrix} tt * x^2 + c & tt * x * y - s * z & tt * x * z + s * y & 0 \\ tt * x * y + s * z & tt * y^2 + c & tt * y * z - s * x & 0 \\ tt * x * z - s * y & tt * y * z + s * x & tt * z^2 + c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 9.6: Matriz Q que representa la forma final de la fórmula de Rodrigues aplicando Maple

Para calcular matriz **Qinv** solo se cambia el parámetro s de entrada de la función **MATRIZ_MAPLE** por el valor negativo **-s**, luego el procedimiento es similar al de **Q**.

- **Creación de matriz de reflexión**

Es una matriz de dimensiones 4 x 4 que posee la siguiente estructura como se observa en la Figura 9.7.

$$R_{ef} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 9.7: Matriz de reflexión

- **Creación de matriz de transformación**

Esta matriz es creada por el producto de 5 matrices las cuales son la matriz de translación invertida T_{inv} , el resultado de la función de maple invertido Q_{inv} , la matriz de reflexión R_{ef} , el resultado de la función de maple Q y la matriz de translación T . Esta matriz permitirá llevar a cabo la rotación del objeto original.

- **Rotación de los vértices**

Para la rotación de los vértices del objeto original se multiplica la matriz de vértices homogéneos calculado con la función **HOMOGENIZAR_VERTICES**.

- **Asignación de vértices y triángulos**

En esta sección se asignan los vértices de la parte reconstruida, por lo que el valor de la variable **vertices_rotados** dentro del pseudocódigo (**Figura 9.1**). Una vez que se tienen los vértices rotados estos son almacenados dentro de un objeto el cual posee la información de los vértices rotados. Asimismo, se utiliza la función **ASIGNAR_TRIANGULOS** la cual tiene como parámetro de entrada la variable **t**, este parámetro posee la relación de los vértices para poder conformar al objeto. Con los vértices y conjunto de aristas (los triángulos) de la parte reconstruida se procede a realizar el cálculo de las normales y tenemos la parte reconstruida del objeto analizado.

9.2. Análisis del resultado de la reconstrucción del objeto

En esta sección se analiza la parte el resultado que se obtuvo sin la interacción del usuario. En la Figura 9.8 se observa el resultado de la reconstrucción del objeto utilizando la interfaz.

Cuando el objeto es reconstruido, se genera un archivo llamado text.off donde se posee los puntos de la parte reconstruida. Además, se posee la

funcionalidad de exportación del archivo como se observa en la Figura 9.9, esto genera un archivo **rebuild.off** en el cual es la unión de los puntos reconstruidos con los puntos del objeto original, de esta forma se obtiene una nueva malla con el objeto reconstruido.

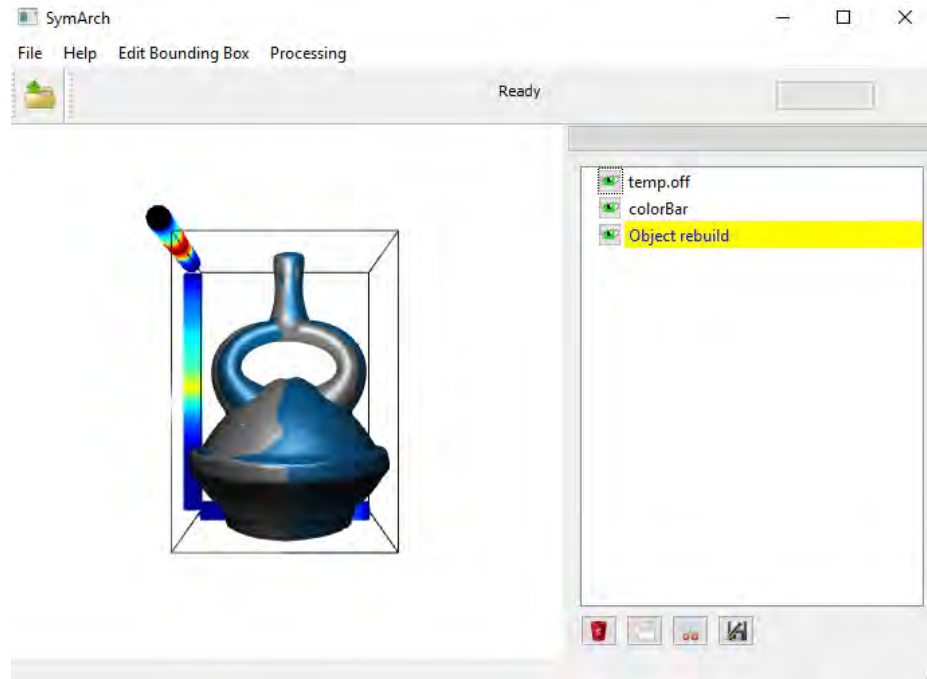


Figura 9.8: Reconstrucción del objeto utilizando interacción del usuario.

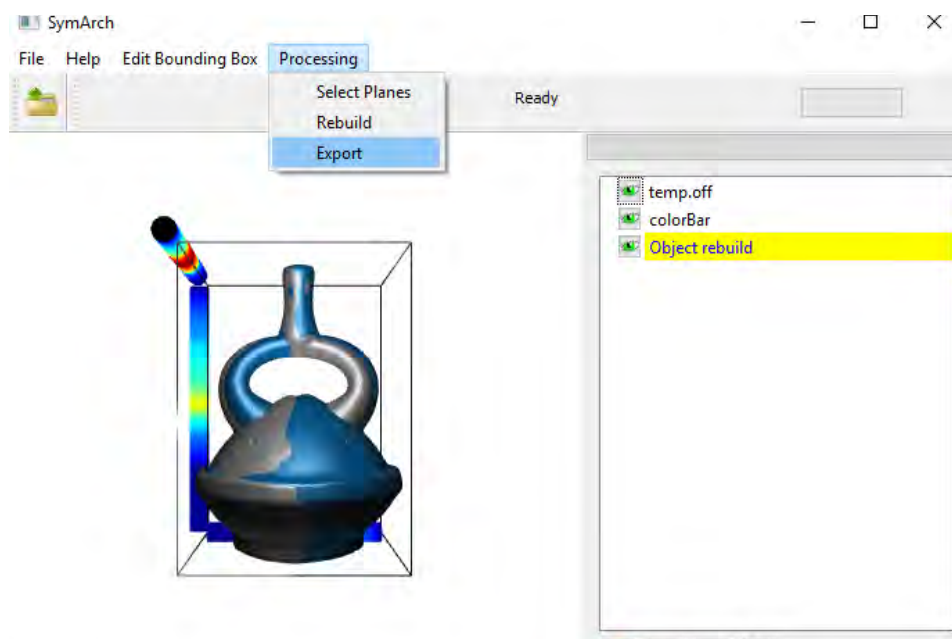


Figura 9.9: Opción de exportar el nuevo objeto reconstruido

9.3. Evaluación de la reconstrucción

En esta sección se analiza la precisión del procedimiento implementado para la reconstrucción de un objeto comparando los resultados con dos algoritmos de reconstrucción encontrados en la literatura, el primero algoritmo es (Mavridis, Sipiran, Andreadis, & Papaioannou, 2015) y el segundo es (Sipiran, Gregor, & Schreck, 2014), ambos explicados en la sección 2 del estado del arte del presente documento. Todos estos algoritmos, incluyendo el algoritmo propuesto dentro de documento serán evaluados con un algoritmo de evaluación desarrollado por Robert Gregor, Danny Bauer, Ivan Sipiran, Panagiotis Peakis y Tobias Schreck en (Gregor, Bauer, Perakis, & Schreck, 2015).

Según lo propuesto realizado en (Gregor, Bauer, Perakis, & Schreck, 2015):

- (i) Se debe lidiar con el problema de que no existe mucha información para poder probar los algoritmos de reconstrucción y que generar la información de objetos dañados puede ser muy costoso ya que depende de distintos factores.
- (ii) Por lo que el informe propone realizar el cálculo de fracturas de cada uno de los objetos.
- (iii) Se realiza el cálculo de un número de objetos que cortaran el objeto a analizar.
- (iv) Luego, se excluye fragmentos que posean un número menor a mil vértices.
- (v) Como resultado del paso anterior, se poseen estructuras con partes faltantes que serán evaluadas y plantean que para la evaluación se tomará como criterio la proporción de los volúmenes que generó cada algoritmo.
- (vi) De manera más detallada en la evaluación plantean lo siguiente: Sea un objeto O_i el cual se desea analizar y $F_i = \{f_i^j\}$ uno de los fragmentos del objeto. Además, los fragmentos descartados f_i . Después de aplicar el algoritmo de reconstrucción se obtiene un objeto completo C_i .
- (vii) Luego se plantea la siguiente fórmula de la Figura 9.10 en la que se evalúa la eficiencia de los algoritmos mediante la división de volúmenes de la intersección del objeto original con el objeto reconstruido entre la unión del objeto original con el objeto reconstruido.

$$E(O_i) = \frac{vol(O_i \cap C_i)}{vol(O_i \cup C_i)}$$

Figura 9.10: Fórmula para la evaluación de la efectividad de los algoritmos de reconstrucción.

Vale la pena resaltar que si el objeto ha sido reconstruido de manera correcta con respecto al original el valor de $E(O_i)$ será igual 1 siendo este el caso ideal. Además, si en caso el objeto original es completamente cubierto por el objeto reconstruido, el valor de E dependerá de que tanto volumen genere la unión de ambos objetos. (viii) Cabe recalcar, que mientras el valor es más cercano a 0, es que la reconstrucción del objeto no es del todo correcta, mientras que si este valor es cercano a 1 la reconstrucción será la ideal.

Este algoritmo de evaluación fue ejecutado utilizando el resultado que genera la interfaz al reconstruir un objeto, el algoritmo propuesto por (Mavridis, Sipiran, Andreadis, & Papaioannou, 2015) y el algoritmo propuesto por (Sipiran, Gregor, & Schreck, 2014), los resultados se muestran en la siguiente Tabla 9.1.

Tabla 9.1: Tabla con los resultados al aplicar ha evaluación a los tres algoritmos de reconstrucción

Objeto analizado	Interfaz propuesta	Mavridis	Sipiran
Objeto 1 (Figura 9.13)	0.671884	0.6494	0.6681
Objeto 2 (Figura 9.12)	0.722566	0.6552	0.5792
Objeto 3 (Figura 9.11)	0.873594	0.7983	0.8712
Objeto 4 (Figura 9.16)	0.863919	0.7686	0.8328
Objeto 5 (Figura 9.15)	0.702871	0.6252	0.6626
Objeto 6 (Figura 9.14)	0.646176	0.5068	0.637
Objeto 7 (Figura 9.18)	0.632744	0.7739	0.724
Objeto 8 (Figura 9.17)	0.602964	0.7712	0.7985
Objeto 9 (Figura 9.19)	0.78054	0.7629	0.7569

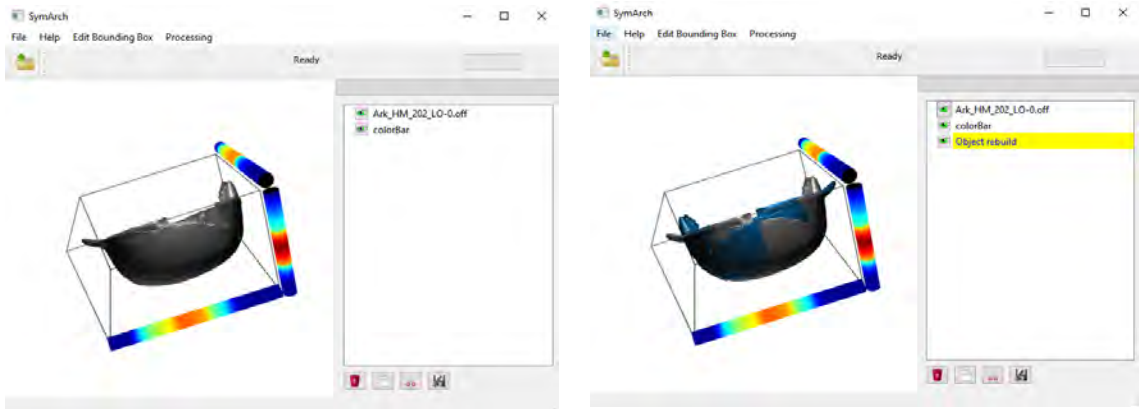


Figura 9.13: Reconstrucción de una vasija, objeto número 1. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 481730 vértices y 963436 aristas

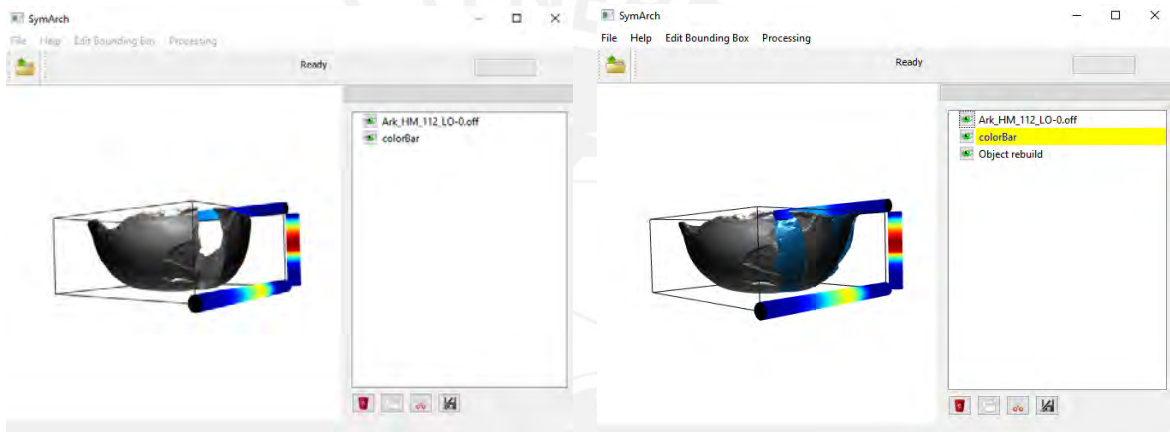


Figura 9.12: Reconstrucción de una vasija, objeto número 2 La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 585878 vértices y 1171744 aristas.

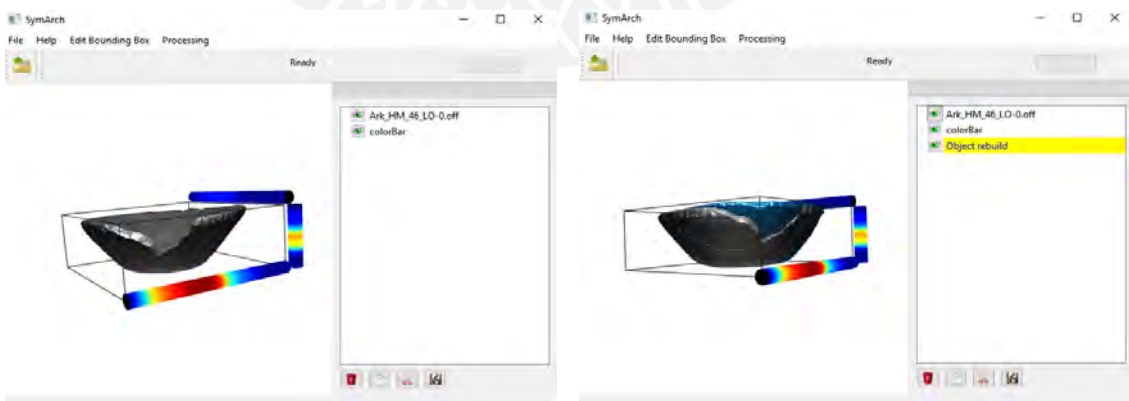


Figura 9.11: Reconstrucción de una vasija, objeto número 3. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 789390 vértices y 1578768 aristas.

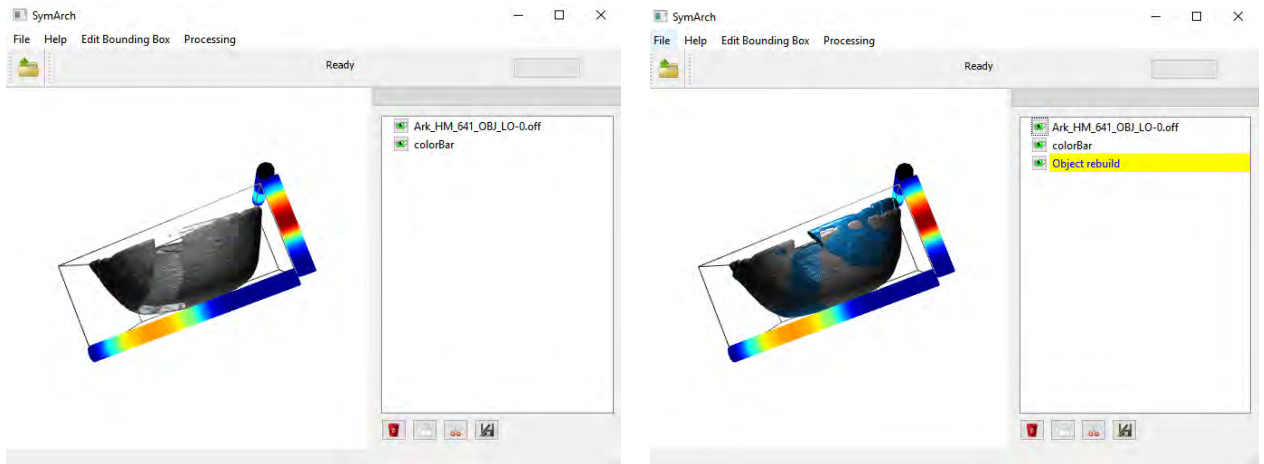


Figura 9.16: Reconstrucción de una vasija, objeto número 4. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 710246 vértices y 1420468 aristas.

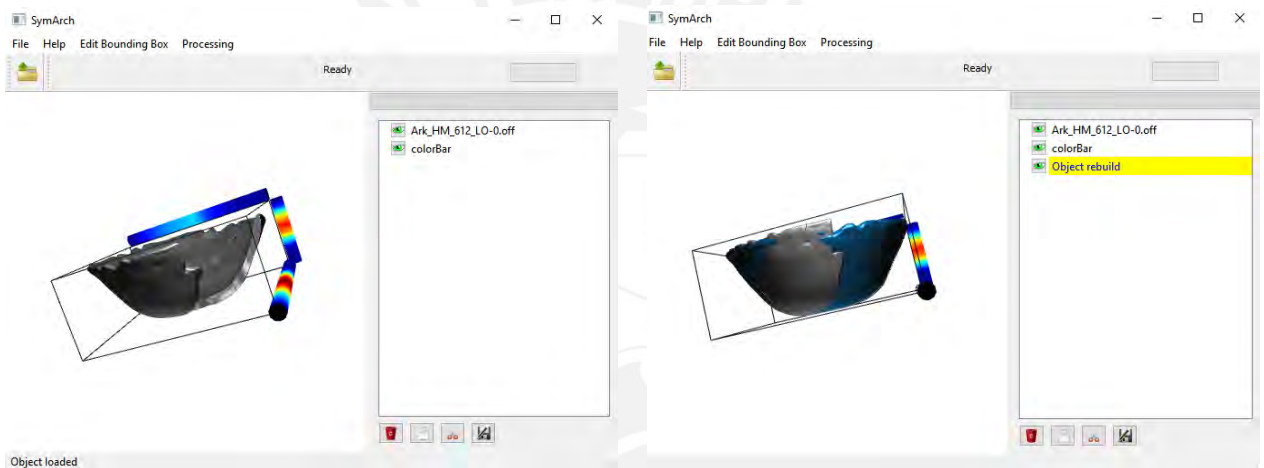


Figura 9.15: Reconstrucción de una vasija, objeto número 5. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 608834 vértices y 1217644 aristas.

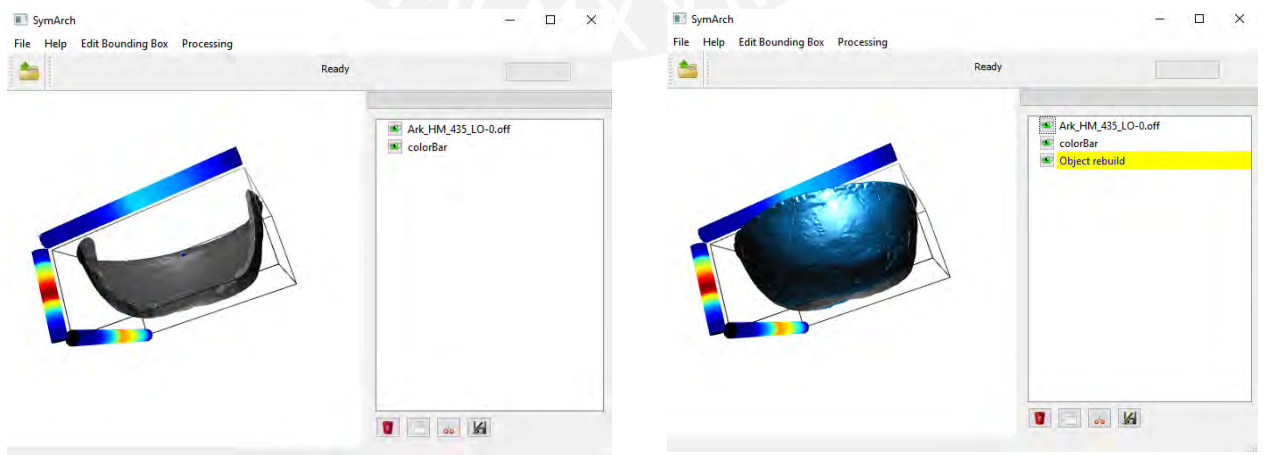


Figura 9.14: Reconstrucción de una vasija, objeto número 6. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 541240 vértices y 1082452 aristas.

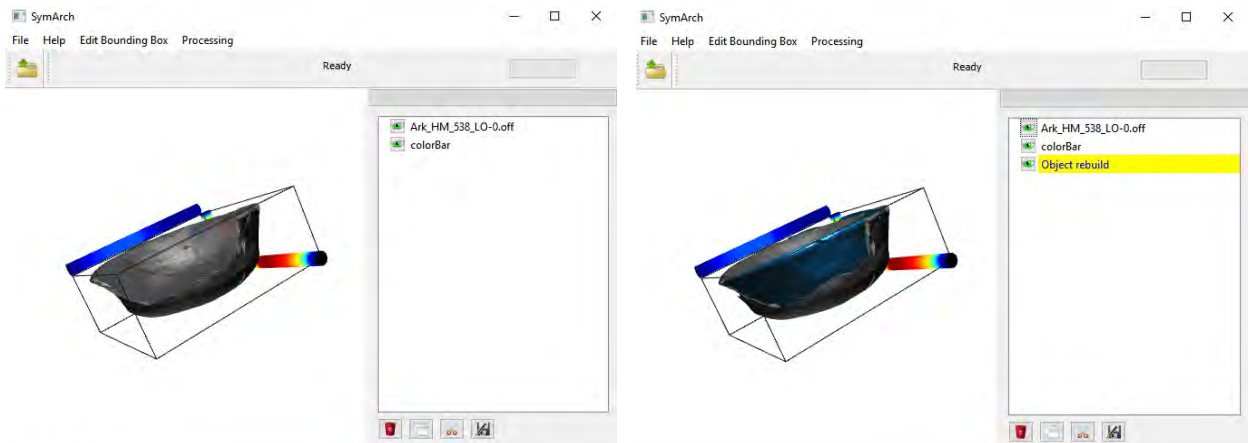


Figura 9.18: Reconstrucción de una vasija, objeto número 7. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 664068 vértices y 1328132 aristas.

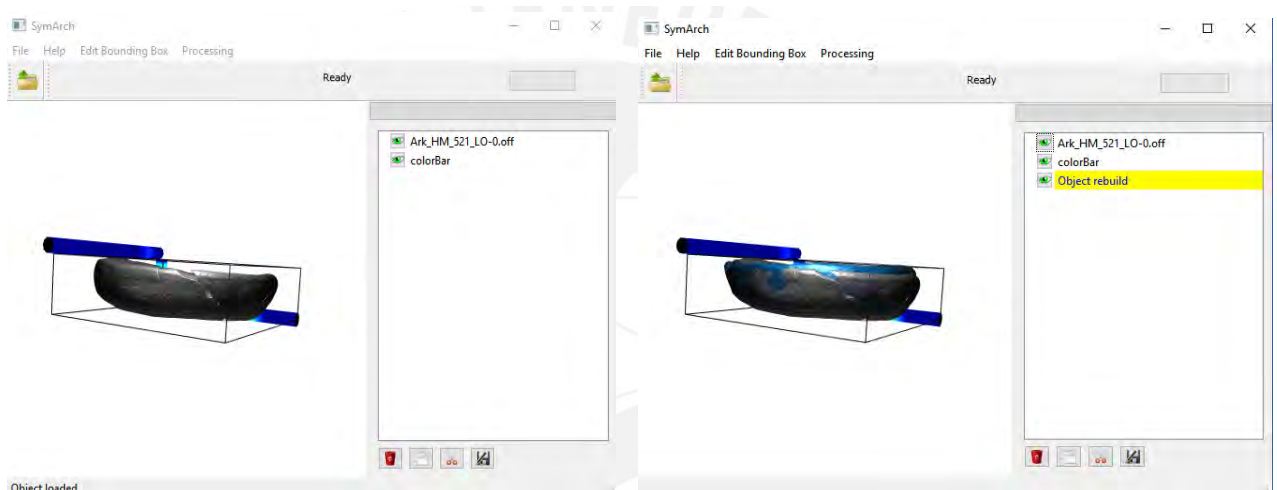


Figura 9.17: Reconstrucción de una vasija, objeto número 8. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 665496 vértices y 1330960 aristas.

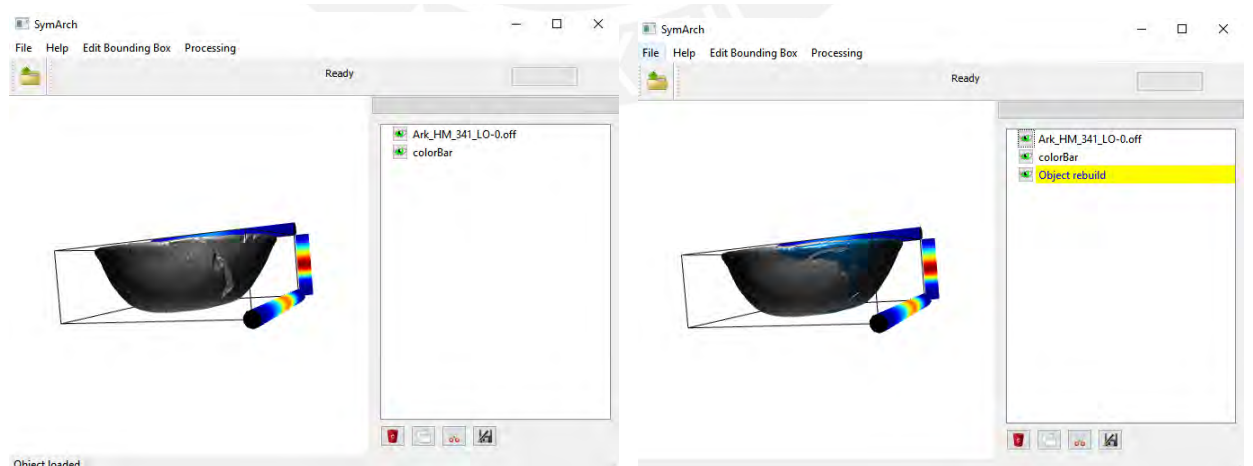


Figura 9.19: Reconstrucción de una vasija, objeto número 9. La imagen de la izquierda se observa el objeto antes de ser reconstruido y la imagen de la derecha muestra al objeto reconstruido. Objeto original con 727042 vértices y 1454060 aristas.

Cabe recalcar que los resultados obtenidos por la interfaz propuesta en este documento fueron óptimos utilizando el método de evaluación postulado en el informe de (Gregor, Bauer, Perakis, & Schreck, 2015) en comparación a los algoritmos de los informes de (Mavridis, Sipiran, Andreadis, & Papaioannou, 2015) y (Sipiran, Gregor, & Schreck, 2014). Además, vale la pena resaltar que la interfaz retornó resultados de la reconstrucción en un tiempo menor al de 4 minutos, mientras que los otros algoritmos pueden tardar un tiempo mayor dependiendo del objeto. Por lo cual se observa la eficiencia y velocidad para realizar la reconstrucción con la ayuda del usuario mediante el uso de la interfaz.



10. Observaciones, conclusiones y trabajos a futuro

En esta sección del documento se presenta las observaciones, conclusiones a las que se ha llegado y trabajos a futuro que se plantean luego del desarrollo de este documento.

- **Observaciones**

Existe un gran interés sobre la reconstrucción de objetos arqueológicos, ya que esto contribuye de manera positiva al estudio de culturas antiguas y su relación con la cerámica.

Existe todo un banco de datos donde se guarda la información de los objetos arqueológicos; sin embargo, la gran mayoría de objetos se encuentran en muy malas condiciones debido a ciertos factores como la erosión de los objetos que estuvieron enterrados, guerras que destruían y dañaban los objetos o la antigüedad de estos objetos.

Existen algoritmos para la reconstrucción de objetos, estos pueden clasificarse de la siguiente manera: métodos basados en correspondencias, métodos basados en transformación, métodos basados en votos y métodos alternativos.

Muchos de los algoritmos de reconstrucción operan bajo casos particulares y en algunos casos requieren de ciertas características como computadoras con un gran poder de procesamiento de información, parámetros difíciles de conseguir, etc.

En la actualidad, la reconstrucción de objetos sigue siendo un área en la cual los investigadores siguen explorando debido a la gran cantidad de información que se encuentra en el día a día y que son objeto de estudio.

- **Conclusiones**

El uso de la simetría para la reconstrucción de un objeto arqueológico es una

característica que puede ser explotada por distintos algoritmos de reconstrucción brindando resultados positivos, puesto que muchos objetos arqueológicos fueron construidos en base a un eje de simetría como se observa en la Figura 9.20.



Figura 10.1: Construcción de cerámico utilizando las manos y técnicas de base giratoria.

Existen una amplia variedad de algoritmos de reconstrucción de objeto, sus resultados pueden ser comparados con la interfaz propuesta en este documento.

La interfaz utilizada para la reconstrucción de objetos arqueológicos brinda resultados positivos en comparación a los algoritmos de (Mavridis, Sipiran, Andreadis, & Papaioannou, 2015) y (Sipiran, Gregor, & Schreck, 2014). Además, el algoritmo propuesto no toma en cuenta un plano de simetría, siendo este un problema en los algoritmos actuales de reconstrucción. De esta forma se mejora el tiempo de respuesta del algoritmo de reconstrucción.

- Trabajos a futuro

Si bien las técnicas usadas para la reconstrucción de un objeto arqueológico mediante el uso de la interfaz solo tomó en cuenta factores físicos como la forma del objeto. Muchos informes sugieren como una muy buena alternativa el análisis de patrones de las pinturas en los cerámicos. Muchos cerámicos antiguos poseen patrones en sus pinturas como en el caso de keros incas como se aprecia en la Figura 9.21. En la figura se puede observar patrones en las líneas, triángulos y cuadrados pueden brindar mayor información para la reconstrucción de un objeto. En caso se tomará los detalles grabados en los cerámicos como un parámetro de entrada para los algoritmos de reconstrucción, brindaría un gran aporte en el tiempo de reconstrucción que tomaría el algoritmo. El algoritmo que usa la interfaz tendría más parámetros a considerar porque puntos es correcto realizar una reconstrucción. Sin embargo, vale la pena recalcar que a medida que se añaden más parámetros al algoritmo, el nivel de complejidad es mayor.



Figura 10.2: Kero con diseños geométricos de líneas horizontales, triángulos y cuadrados concéntricos. Imagen extraída de (Museo Larco, 2010)

Investigaciones que se han propuesto en el informe de (Gregor, Bauer, Perakis, & Schreck, 2015) sugieren que un tratamiento más completo para la reconstrucción de un objeto complejos físicos, procesos mecánicos y químicos. De esta forma, estos factores brindarían un rango más amplio de las condiciones ambientales que se puede esperar con el cambio del tiempo y de tal forma poseer más datos para la reconstrucción. Sin embargo, esto involucra el trabajo de distintas especialidades.

Se recomienda la implementar la funcionalidad de seleccionar planos de simetría no paralelos a los planos cartesianos. De esta manera la interfaz no tendría inconvenientes para la reconstrucción de un objeto que no es paralelo a ningún plano cartesiano.



11. Referencias Bibliográficas

- Beltrán Castañón, C. A., & Sipirán Mendoza, I. A. (s.f.).
- Berner, A., Wand, M., Mitra, N., Mewes, D., & Seidel, H. (2011). *Shape Analysis with Subspace Symmetries*.
- Bokeloh, M., Berner, A., Wand, M., Seidel, H., & Schilling, A. (2009). *Symmetry Detection Using Feature Lines*. Computer Graphics Forum.
- Boost C++ Libraries*. (s.f.). Recuperado el Diciembre de 2016, de <https://www.boost.org/>
- Bray, T. (s.f.). La alfarería imperial inka: una comparación entre la cerámica estatal del área de Cuzco y la cerámica de las provincias. *Revista de antropología Chilena*, 365-374.
- Bustos, B., & Sipiran, I. (2011). *Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes*. Springer-Verlag.
- Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). *Brief: Binary robust independent elementary features*.
- CGAL course*. (s.f.). Obtenido de <https://members.loria.fr/SLazard/CGAL/2004/>
- cplusplus*. (s.f.). Recuperado el 10 de Enero de 2017, de <http://www.cplusplus.com/info/description/>
- Cutlab3D. (s.f.). *Cutlab3D*. Obtenido de <http://www.cutlab3d.de/project.html>
- Fischler, M., & Bolles, R. (1981). *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*.
- Fischler, M., & Bolles, R. (1981). *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*.
- Gregor, R., Bauer, D., Perakis, P., & Schreck, T. (2015). *Automatic 3D Object Fracturing for Evaluation of Partial Retrieval and Object Restoration Tasks - Benchmark and Application to 3D Cultural Heritage Data*.
- Gregor, R., Sipiran, I., Papaioannou, G., Schreck, T., Andreadis, A., & Mavridis, P. (2014). *Towards Automated 3D Reconstruction of Defective Cultural Heritage Objects*. The Eurographics Association.
- Grushko, C., Raviv, D., & Kimmel, R. (2012). *Intrinsic Local Symmetries: A Computational Framework*. n Proceedings of the 5th Eurographics Conference on 3D Object Retrieval.

- Huang, Q., Wand, M., Seidel, H.-P., & Guibas, L. (2014). *Relating Shapes via Geometric Symmetries and Regularities*. ACM Transactions on Graphics.
- Jiang, W., Xu, K., Cheng, Z.-Q., & Zhang, H. (2013). *Skeleton-based intrinsic symmetry detection on point clouds*. Graphical Models.
- Kazhdan, M., Chazelle, B., Dobkin, D., Funkhouser, T., & Rusinkiewicz, S. (2003). *A Reflective Symmetry Descriptor for 3D Models*.
- Kazhdan, M., Funkhouser, T., & Rusinkiewicz, S. (2004). *Symmetry Descriptors and 3D Shape Matching*.
- Kim, V., Lipman, Y., & Funkhouser, T. (2011). *Blended intrinsic maps*.
- Knobloch, P. (2012). Cronología del contacto y de encuentros cercanos de Wari. . *Boletín de Arqueología PUCP*, 69-87.
- Kovács, E. (1995). *Using Maple in teaching of computer graphics*.
- Kovács, E. (1996). *Using some mathematical program in computer graphics teaching*.
- Kovács, E. (2012). *Rotation about an arbitrary axis and reflection through an arbitrary plane*. Annales Mathematicae et Informaticae.
- Lasowski, R., Tevs, A., Seidel, H., & Wand, M. (2009). *A probabilistic framework for partial intrinsic symmetries in geometric data*. In Computer Vision.
- Li, B., Johan, H., Ye, Y., & Lu, Y. (2014). *Efficient View-Based 3D Reflection Symmetry Detection*. New York.
- Lipman, Y., Chen, X., Daubechies, I., & Funkhouser, T. (2010). *Symmetry Factored Embedding And Distance*. ACM Transactions on Graphics.
- Liu, T., G. Kim, V., & Funkhouser, T. (2012). Finding Surface Correspondences. The Eurographics Association and Blackwell Publishing Ltd.
- Liu, T., Kim, V., & Funkhouser, T. (2012). *Finding Surface Correspondences Using Symmetry Axis Curves*.
- Liu, X., Li, S., Liu, R., Wang, J., Wang, H., & Cao, J. (2015). *Properly constrained orthonormal functional maps for intrinsic symmetries*.
- Loop, C. (1987). *Smooth Subdivision Surfaces Based on Triangles*. University of Utah.
- Martinet, A., Soler, C., Holzschuch, N., & François, S. (2006). *Accurate Detection of Symmetries in 3D Shapes*. ACM Transactions on Graphics.
- Mavridis, P., Sipiran, I., Andreadis, A., & Papaioannou, G. (2015). Object Completion using k-Sparse Optimization. *Computer Graphics Forum*. doi:10.1111/cgf.12741

- Mitra, N., Bronstein, A., & Bronstein, M. (2010). *Intrinsic Regularity Detection in 3D Geometry*. Computer Vision.
- Mitra, N., Guibas, L., & Pauly, M. (2007). *Symmetrization*.
- Mitra, N., Pauly, M., Wand, M., & Ceylan, D. (2013). *Symmetry in 3D geometry: Extraction and Applications*. The Eurographics Association and John Wiley & Sons Ltd.
- Museo Larco. (s.f.). Recuperado el Febrero de 2017, de <http://www.museolarco.org/>
- Museo Larco. (2010). Recuperado el 29 de Julio del 2018, de <http://www.museolarco.org/catalogo/ficha.php?id=27994>
- OpenGL. (s.f.). Recuperado el 22 de Octubre de 2016, de https://www.opengl.org/wiki/Getting_Started
- OpenGL. (s.f.). Recuperado el 22 de 10 de 2016, de https://www.opengl.org/wiki/Getting_Started
- OpenGL Programming. (s.f.). Obtenido de https://en.wikibooks.org/wiki/OpenGL_Programming/Bounding_box#/media/File:OpenGL_Tutorial_Bounding_box.png
- OpenVDB. (s.f.). Recuperado el 15 de Enero de 2017, de <http://www.openvdb.org/>
- Pauly, M., Mitra, N., Giesen, J., Gross, M., & Guibas, L. (2005). *Example-based 3d scan completion*.
- Pauly, M., Mitra, N., Giesen, J., Guibas, L., & Gross, M. (2005). *Example-Based 3D Scan Completion*.
- Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., & Funkhouser, T. (2006). *Planar-reflective Symmetry Transform for 3D Shapes*.
- Presious. (s.f.). *Predictive digitization, restoration and degradation assessment of cultural heritage objects*. Obtenido de <http://presious.eu/>
- Qt. (s.f.). Recuperado el 23 de Octubre de 2016, de <https://www.qt.io/>
- Qt Framework. (s.f.). Obtenido de <https://www.inflectra.com/Rapise/Technologies/Qt-Framework.aspx>
- Raviv, D., Bronstein, A., M., B., & Kimmel, R. (2010). *Full and Partial Symmetries of Non-rigid Shapes*.
- Raviv, D., Bronstein, Sapiro, G., Bronstein, A. M., & Kimmel, R. (2010). *Diffusion symmetries of non-rigid shapes*.
- Roth, S. (1982). *Ray casting for modeling solids*.

- (2013). *Shape Matching via Quotient Spaces*.
- Sipiran, I., & B., B. (2011). *Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes*.
- Sipiran, I., & Bustos, B. (2013). *Key-components: Detection of salient regions on 3d meshes*.
- Sipiran, I., Gregor, R., & Schreck, T. (Octubre de 2014). Approximate Symmetry Detection in Partial 3D Meshes. doi:10.1111/cgf.12481
- Tevs, A., Huang, Q., Wand, M., Seidel, H., & Guibas, L. (2014). *Relating Shapes via Geometric Symmetries and Regularities*.
- The Computational Geometry Algorithms Library*. (24 de Octubre de 2016).
Obtenido de <http://www.cgal.org/>
- Threading Building Blocks (TBB)*. (s.f.). Recuperado el 22 de Octubre de 2018, de <https://www.threadingbuildingblocks.org/>
- Thrun, S., & Wegbreit, B. (2005). *Shape from symmetry*.
- Wang, Y., Xu, K., Li, J., Z. H., & Shamir, A. L. (2011). *Symmetry Hierarchy of Man-Made Objects*.
- Xu, K., Zhang, H., Jiang, W., Dyer, R., Cheng, Z., Liu, L., & Chen, B. (2012). *Multi-scale Partial Intrinsic Symmetry Detection*.
- Xu, K., Zhang, H., Tagliasacchi, A., Liu, L., Li, G., Meng, M., & Xiong, Y. (2009). *Partial Intrinsic Reflective Symmetry of 3D Shapes*.
- Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N., Cohen-Or, D., & Chen, B. (2010). *Non-local Scan Consolidation for 3D Urban Scenes*.