



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IDENTIFICACIÓN AUTOMÁTICA DE ESTRUCTURAS REGULARES FORMADAS
POR PATRONES GEOMÉTRICOS REPETITIVOS EN SUPERFICIES 3D

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

FELIPE ALEJANDRO VARGAS GARCIA

PROFESOR GUÍA:
IVÁN SIPIRÁN MENDOZA

MIEMBROS DE LA COMISIÓN:
BENJAMÍN BUSTOS CÁRDENAS
JOSÉ SAAVEDRA RONDO

SANTIAGO DE CHILE
2023

Resumen

En el campo de la arqueología, el estudio de reliquias y alfarería indígena es una tarea recurrente que involucra un gran volumen de muestras. La clasificación de estas muestras bajo criterios estructurados potencia el posterior análisis, sin embargo, gran parte de esta clasificación se realiza a mano, lo que reduce la eficiencia del proceso.

En este trabajo se estudian diversos objetos arqueológicos, dando principal énfasis a los detalles grabados o tallados en su superficie. El objetivo es clasificar las muestras arqueológicas con base en las macroestructuras formadas a partir de estos detalles y desarrollar una forma de reconocimiento y clasificación asistida por software que potencie el proceso de estudio de objetos arqueológicos.

Mediante el análisis de puntos de interés, obtenidos de investigaciones anteriores y el procesamiento de los datos recopilados, se busca construir descriptores que permitan relacionar estos objetos para poder buscar similitud entre ellos e identificar posibles estructuras regulares formadas por los patrones presentes en la superficie de los objetos y desarrollar un programa que pueda automatizar la obtención de esa información.

Los resultados obtenidos a partir del proceso de análisis construido muestran que es factible un proceso automático de clasificación de macroestructuras. Dentro de los resultados presentados, se puede ver como las muestras efectivamente presentan diversos grados de similitud donde se verifica su relación.

Tabla de Contenido

1. Introducción	1
2. Descripción del Problema	2
3. Objetivos	3
3.1. Objetivo General	3
3.2. Objetivos Específicos	3
4. Estado del Arte	4
4.1. Caracterización de patrones geométricos	4
4.2. Grafos en visión computacional	6
4.3. Análisis de grafos	7
5. Descripción de la Solución	8
5.1. Contexto	8
5.2. Conjunto de datos	8
5.3. Anotaciones	9
5.4. Modelado	10
5.5. Texturizado	12
5.6. Visualización de anotaciones	13
5.7. Cálculo de centroides	13
5.8. Proyección de centroides	15
5.9. Relación entre centroides	17

5.10. Heatmap	17
5.11. Vector de características	18
5.12. Pipeline	20
6. Experimentos	21
6.1. Clasificación	21
6.2. Comparación	23
7. Resultados	24
7.1. Clasificación	24
7.2. Comparación	26
8. Discusión	47
8.1. Clasificación	47
8.2. Comparación	52
8.3. Trabajo futuro	54
9. Conclusión	55
Bibliografía	56

Capítulo 1

Introducción

La arqueología es la ciencia que se dedica a estudiar las sociedades antiguas describiendo su cultura, actividades y como se relacionaban con su entorno, principalmente mediante el análisis de restos materiales que por razones multifactoriales se han conservado a través del tiempo. Vasijas, vajilla, cacharrería y otros productos derivados de la actividad alfarera constituyen parte del pilar fundamental de esta área de estudio, dando resultado a grandes volúmenes de información sobre la cual es deseable realizar diferentes tipos de análisis. El desarrollo de actividades intelectuales sobre variadas fuentes de información hace que sea útil recurrir a recursos que hagan el proceso más expedito, por lo que apoyarse en herramientas tecnológicas, específicamente software, es un hecho que presenta valor para esta área.

Dentro de las características de los restos arqueológicos a estudiar, uno de los factores más interesantes son los detalles grabados y/o dibujados en la superficie de los mismos, los patrones o motivos presentes en los objetos pueden permitir relacionar diferentes objetos en una misma categoría para posterior análisis, proceso que realizado de forma manual puede significar un gran esfuerzo por parte de los expertos. Sin embargo, una herramienta de identificación y clasificación de patrones presenta una dificultad importante que se centra en el origen de los elementos que componen los patrones, las figuras que se ven representadas en los objetos no son idénticas unas con otras, ya que, con un alto grado de certeza, fueron confeccionadas a mano, lo que significa que símbolos que la intuición indica que deben ser iguales presentan claras diferencias en color, orientación y/o tamaño.

Para tratar información proveniente de fuentes como las enunciadas (estatuillas, loza, jarras, etc.) es imperativo realizar un proceso de discretización de la información de la que se dispone para dar posibilidad al uso de software. Por lo que se propone proyectar la superficie a estudiar en un plano, lo que permite estudiar los detalles presentes en la superficie que no necesariamente se reflejan en la estructura del objeto, como esbozos, patrones o dibujos.

Desarrollar heurísticas que permitan la generación de nueva información y su fácil interpretación, apoyando estos procedimientos con asistencia de software es de alto interés para el área de la arqueología, ya que su efectiva elaboración, permitiría un manejo más ágil de la información arqueológica que se recolecta y ayudaría a tener resultados objetivos basados en modelos matemáticos, impulsando así la eficiencia y la convencionalidad de la investigación en el campo.

Capítulo 2

Descripción del Problema

Dentro del campo de la arqueología, el estudio de remanentes de la alfarería y otras actividades humanas es una actividad común que necesita realizar análisis sobre grandes volúmenes de información, actualmente, este análisis se realiza de forma completamente manual, sacrificando tiempo importante. Adicionalmente, todas las conjeturas que se realizan al respecto son completamente subjetivas, dado que no se cuenta con fórmulas matemáticas o recetas para extraer información de esta clase de fuentes y, por tanto, cada especialista puede tener una postura diferente frente a un resultado.

Es por esto que el principal problema a resolver en esta experiencia es la falta de métodos y/o herramientas de clasificación sobre esta clase de fuentes de información, buscando plantear una posible solución para probar la factibilidad de diseñar un algoritmo que sea capaz de clasificar esta clase de información mediante un análisis objetivo basado en la información numérica que sea posible obtener.

Hablando específicamente de este trabajo, es importante mencionar entonces que la problemática que busca abordar esta investigación cubre un punto interesante al tener en cuenta la forma de analizar la información de la que se dispone, la identificación de macroestructuras sobre los patrones geométricos identificados en cada uno de los objetos que componen el conjunto de datos no solo depende de la presencia de patrones geométricos en los objetos, también es necesario considerar la distribución de los mismos en la superficie de los objetos en cuestión. Con base en esto, es correcto mencionar que es necesario desarrollar una nueva estrategia para analizar los datos, realizando una inspección relacional y estructural de los patrones geométricos existentes, de forma que el resultado obtenido sea útil.

Capítulo 3

Objetivos

3.1. Objetivo General

Desarrollar un algoritmo que a partir de la información de los patrones geométricos presentes en la superficie de un objeto representado mediante una malla de superficie en tres dimensiones, permita identificar de forma automática la presencia de macroestructuras regulares formadas por la geometría repetitiva presente en la malla y dar la posibilidad de poder visualizar estas nuevas estructuras, para ser aplicado principalmente al área de la arqueología.

3.2. Objetivos Específicos

- Diseñar un modelo de discretización de la información relacionada a la posición de patrones geométricos en la superficie de los objetos del conjunto de datos.
- Relacionar los patrones geométricos ubicados en la superficie de los objetos del conjunto de datos utilizando la información discretizada obtenida mediante el modelo diseñado.
- Modelar un vector de características con el objetivo de representar las relaciones entre patrones geométricos de cada objeto.
- Definir un criterio de comparación mediante el cual poder determinar similitud entre dos vectores de características.
- Definir una clasificación general para categorizar los objetos en clases segregadas a partir del estudio de sus vectores característicos.
- Implementar un algoritmo capaz de generar vectores característicos de un objeto dentro del conjunto de datos y clasificarlo en su correspondiente categoría de acuerdo al criterio de clasificación empleado.

Capítulo 4

Estado del Arte

4.1. Caracterización de patrones geométricos

Pattern Symetry and Colored Repetition in Cultural Contexts. [10]

En este artículo se discute acerca de cómo en la manufactura de las culturas antiguas se encontraba presente la distribución simétrica de elementos, mayormente se trata de la presencia del color y las formas de las confecciones textiles y objetos que pueden construirse de esta base. Se presenta el uso del color como elemento utilizado para romper la simetría estructural original de los objetos dando paso a nuevos grados de simetría representado por patrones repetitivos de color o simplemente buscando el quiebre de la simetría dependiendo del simbolismo que buscaba representar una pieza en particular.

Context-based Surface Pattern Completion of Ancient Pottery. [5]

En este trabajo se busca completar de forma digital objetos de alfarería que muestran daño por diversas con el objetivo de generar una nueva textura en la que se repare el daño presente generando un objeto reconstruido, facilitando su posterior estudio. El proceso de reparación comienza con un objeto texturizado, el cual presenta daño o desgaste, mostrando brechas en su textura. En esta textura se estima la presencia de secuencias de patrones geométricos que deberían estar presentes a partir de las formas geométricas visibles y sus propiedades, como el tamaño y distribución en la superficie del objeto, una vez identificado el espacio donde debería existir un patrón, se completa usando los patrones de la textura real.

The Geometry of African Art II. A Catalog of Benin Patterns. [1]

En este artículo se estudian y clasifican los patrones geométricos encontrados en objetos arqueológicos africanos. A partir de los objetos encontrados, se construyen diferentes clases arquetípicas de patrones geométricos y se clasifican los objetos originales, los resultados de la experiencia logran establecer una relación entre los patrones repetitivos presentes en los objetos estudiados y la cultura específica a la que dicho objeto corresponde.

The Mosaic Patterns of H. J. Woods. [2]

En este artículo se presentan patrones geométricos dicromáticos, los cuales forman macroestructuras denominadas mosaicos. Estos mosaicos corresponden con una distribución repetitiva y uniforme de figuras geométricas dispuestas de forma cohesiva, dentro de la publicación estos mosaicos se categorizan utilizando notaciones matemáticas que representan una relación proporcional de las formas geométricas que las componen, utilizando como principal criterio la distribución de color.

Groups and Geometry in the Ceramic Art of San Ildelfonso. [3]

Como en la publicación anterior, en este artículo se muestra y clasifican patrones geométricos dicromáticos formados por la distribución uniforme de figuras geométricas en cerámica histórica. Se utilizan las notaciones matemáticas estipuladas en dicha publicación y se categorizan los patrones geométricos encontrados utilizando las clases ya establecidas. Adicionalmente, se estudian utilizando nuevos criterios de forma y tamaño en patrones monocromáticos, se logran categorizar en las clasificaciones originales, reafirmando la clasificación planteada relacionada con los mosaicos arquetípicos.

4.2. Grafos en visión computacional

Semantic Object Parsing with Graph LSTM. [7]

En este trabajo se utilizan redes neuronales para generar grafos relacionales a partir de imágenes con el objetivo de poder identificar los elementos presentes en la imagen.

Se utiliza un algoritmo de aprendizaje LSTM, una extensión específica de las redes neuronales que permite mantener un estado de guardado de un estímulo o un resultado durante un largo periodo de tiempo relativo a la duración del experimento al que la red se somete, haciendo uso de esto, a partir de una imagen se realiza una extracción de características de donde resulta un mapa de píxeles que se relacionan por proximidad, luego de pasar por redes convolucionales, se abusa de las propiedades de LSTM para obtener propiedades globales con las que se construyen los experimentos para entrenar redes neuronales capaces de identificar objetos dentro de la imagen original.

Spatial Temporal Graph Convolutional Networks for Skeleton-based Action Recognition. [12]

El objetivo de esta investigación corresponde a la identificación de acciones dentro de un video. Para lograr esto realizan un estudio sobre grafos utilizando redes convolucionales, estimaciones y modelaciones de cinemática aplicada a estructuras esqueléticas.

Para lograr esto, a partir del video se obtienen características importantes de la estructura de un sujeto usando estrategias de reconocimiento de esqueleto, a partir de estos esqueletos se generan grafos relacionales entre los puntos importantes de la estructura, realizado a todos los cuadros del video se tiene una descripción del movimiento a través del tiempo, sobre esta descripción se realizan estimaciones para determinar la acción en el video original.

Videos as Space-time Region Maps. [9]

Como en la publicación anterior, en este trabajo el objetivo principal es identificar acciones dentro de un video, utilizando también redes convolucionales, a diferencia de la investigación anterior se utiliza relaciones de tiempo y espacio para ver cómo interactúan los elementos en el video y de aquí estimar que ocurre.

En esta investigación se utilizan redes convolucionales 3D para obtener información importante considerando profundidad en el video, creando nodos con los cuales se forman grafos que representan una secuencia de cambios en los estados relacionales de los nodos, sobre estos grafos se realiza el estudio con redes neuronales para estimar la acción con base en el cambio que se evidencia en los grafos estudiados.

4.3. Análisis de grafos

HyperFoods: Machine Intelligent Mapping of Cancer-beating Molecules in Food. [8]

En esta investigación se utiliza el estudio de grafos para buscar coincidencias moleculares entre alimentos y fármacos con base en las relaciones genéticas que los componen, con el objetivo de identificar alimentos que puedan ser beneficiosos para combatir el cáncer y así reforzar las investigaciones que estipulan que el cáncer puede ser prevenido con dietas especializadas.

La información con la que se cuenta corresponden a los perfiles moleculares de diversos alimentos y drogas que fueron representados como grafos describiendo sus relaciones intermoleculares. Para tratar los datos, se diseña un algoritmo de aprendizaje no supervisado que trabaja sobre los grafos construidos con los que se genera un nuevo conjunto de datos que representa las posibles interacciones entre las moléculas estudiadas y el genoma humano. Finalmente, se diseña un algoritmo de aprendizaje supervisado que permite clasificar la nueva información para determinar si las moléculas en cuestión presentan propiedades anticancerígenas.

Neurogenetics Profiles Delineate Large-scale Connectivity Dynamics of the Human Brain. [4]

En este artículo se comenta como mediante el uso de grafos, se conforman diferentes experiencias para comprender mejor el funcionamiento del cerebro humano, específicamente la actividad neuronal a través de redes localizadas y redes distribuidas en diversas zonas del cerebro. En estas experiencias se analizan grafos construidos de tal forma que representen las conexiones entre neuronas en un cerebro humano, pudiendo evidenciar cómo se comportan los impulsos eléctricos por las diversas zonas del cerebro una vez que se aplican diversos estímulos sensoriales utilizando análisis de grafos especializado en la caracterización de los patrones de conectividad funcional entre las regiones sensoriales del cerebro y el resto del órgano.

Graph Dynamical Networks for Unsupervised Learning of Atomic Scale Dynamics in Materials. [11]

Este trabajo tiene como principal objetivo utilizar algoritmos no supervisados de aprendizaje para predecir la dinámica a escala atómica de diferentes tipos de materiales. En las experiencias descritas en el artículo se utilizan grafos ponderados donde las relaciones entre los nodos representan las relaciones entre átomos y las aristas contienen información respecto a las propiedades del enlace atómico formado. Mediante modelos matemáticos que dictaminan el comportamiento esperado de los átomos y la teoría de grafos, se logra diseñar un algoritmo de aprendizaje no supervisado que logra predecir el movimiento de los átomos y ruptura y formación de enlaces atómicos al interior de diferentes materiales.

Capítulo 5

Descripción de la Solución

5.1. Contexto

La Universidad Tecnológica de Graz, Instituto de Computación Gráfica y Visualización del Conocimiento, Austria, de la mano con el Departamento de Ciencias de la Computación, Universidad de Chile, desarrollan la Investigación “A Benchmark Dataset for Repetitive Pattern Recognition on Textured 3D Surfaces” [6]. Este trabajo se orienta al desarrollo de un sistema de etiquetado de patrones asistido por software, aplicación de diferentes estrategias para el reconocimiento de patrones y la evaluación de los mismos para comprobar su eficacia. Es de esta investigación de donde se obtiene la información utilizada en la presente experiencia, proponiendo como punto de partida los objetos usados y sus etiquetas obtenidas del sistema de etiquetado desarrollado.

5.2. Conjunto de datos

En esta investigación se trabaja con un subconjunto de un total de 963 objetos en el museo Josefina Ramos de Cox en Lima, Perú. Ochenta y dos mallas triangulares texturizadas en tres dimensiones se obtuvieron mediante escaneo con un escáner de luz estructurada [6]. La colección comprende varias clases de formas, como jarras, cántaros, cuencos, ollas y jarrones. Estos objetos se atribuyen a varias culturas precolombinas latinoamericanas, de las cuales se puede caracterizar con estilos artísticos únicos, en formas y pinturas.

Estas mallas fueron normalizadas para estar contenidas en un cubo unitario y alineadas con el eje Y+. Se estima que se presentan 130.000 triángulos por modelo. Cabe destacar de forma adicional que estos objetos con el paso del tiempo se han deteriorado o roto, detalles que se han reflejado en las mallas resultantes del proceso de escaneo, lo que hace que esto sea otro desafío para su trato y procesamiento.

5.3. Anotaciones

Cada objeto perteneciente al conjunto de datos trabajado presenta patrones geométricos distribuidos por su superficie, cada una de estas ocurrencias es denominada entidad de patrón. Un grupo de entidades similares forma un arquetipo de patrón, pero a pesar de pertenecer al mismo arquetipo, muestran variaciones de tamaño, forma, color y/o orientación. La información referente a estos registros se guarda en anotaciones, una anotación se refiere a un elemento en un diccionario que almacena la información de entidades y arquetipos.

Para estructurar la información de una entidad, se almacena la información obtenida de comparar dicha entidad y una entidad de control del arquetipo con el cual corresponde.

```
Pattern_Entity: {  
    reflection : bool  
    orientation : float  
    scale      : float  
    face_ids   : [int]  
}
```

Los atributos para una entidad de patrón significan:

Atributo	Significado	Valor
reflection	Indica si la entidad está en su estado de reflexión con respecto a la entidad de control.	0 1
orientation	Almacena la orientación de la entidad con respecto al modelo de su arquetipo en radianes.	$r \in [0, 2\pi)$
scale	Almacena el tamaño de la entidad con relación al patrón de control.	$s \in (0, +\infty)$
face_ids	Lista que almacena los índices de las caras que conforman la entidad dentro de la malla de superficie.	$[f_1, \dots, f_n] f_i \in \mathbb{N}_0$

La información referente a un arquetipo está contenida en un formato similar, identificando las propiedades más importantes de un arquetipo.

```
Pattern_Archetyp: {  
    object_id      : int  
    fold_symmetry : int  
    sample_entity : Image  
    entities       : [Patter_Entity]  
}
```

Los atributos para un arquetipo de patrón son:

Los datos construidos con la estructura indicada anteriormente se almacenan en formato JSON, debido a ser un formato altamente escalable y con un amplio soporte en términos de parseo.

Atributo	Significado	Valor
object_id	Identificador del arquetipo.	$x \in \mathbb{N}_0$
fold_symmetry	Indica el grado de simetrías que la entidad de control presenta.	$x \in \mathbb{N}_0$
sample_entity	Es la imagen de la entidad de control del arquetipo	Image
entities	Lista que almacena una colección de <i>Pattern_Entity</i> que corresponden al arquetipo.	[<i>Patern_entity</i>]

5.4. Modelado

Los archivos que almacenan las mallas triangulares sobre las cuales se trabaja en esta investigación tienen el formato *PLY*, formato comúnmente utilizado en el manejo de información de modelado 3D.

Para poder acceder a la información se hace uso de la librería *plyfile*, la que permite parsear de forma directa la información de un archivo de extensión *.ply* y se construye una clase capaz de rescatar toda la información y la metadata de los archivos.

Cada archivo *PLY* almacena la información de uno de los objetos del conjunto de datos y se compone de la siguiente información:

```
comment TextureFile: str
element vertex: int {
    property x : float
    property y : float
    property z : float
    property nx : float
    property ny : float
    property nz : float
}
element face: int {
    property vertex_indices : list <int>
    property texcoord       : list <float>
}
```

Los atributos almacenados en el archivo *PLY* indican:

Atributo	Propiedad	Significado	Valor
TextureFile		Nombre de la imagen que almacena el texturizado.	String
vertex		Identificador del vértice	$id \in \mathbb{N}_0$
	x	Posición del vértice en el eje x	$x \in \mathbb{R}_0$
	y	Posición del vértice en el eje y	$y \in \mathbb{R}_0$
	z	Posición del vértice en el eje z	$z \in \mathbb{R}_0$
	nx	Componente del vector normal del vértice en el eje x	$nx \in \mathbb{R}_0$
	ny	Componente del vector normal el vértice en el eje y	$ny \in \mathbb{R}_0$
	nz	Componente del vector normal del vértice en el eje z	$nz \in \mathbb{R}_0$
face		Identificador de la cara	$x \in \mathbb{N}_0$
	vertex_indices	Lista de identificadores de vértices que componen la cara	$[v_1, \dots, v_3] v_i \in \mathbb{N}_0$
	textcoord	Lista de puntos bidimensionales que indican el sector de la textura que corresponde con la cara	$[t_1, \dots, t_3] t_i \in \mathbb{R}^2$

Habiendo comprendido y relacionado la información contenida en uno de los archivos, es posible generar la visualización de la malla triangular del objeto contenido en dicho archivo, haciendo uso de los métodos para mallas de superficie de *polyscope*, resulta una visualización como la siguiente:

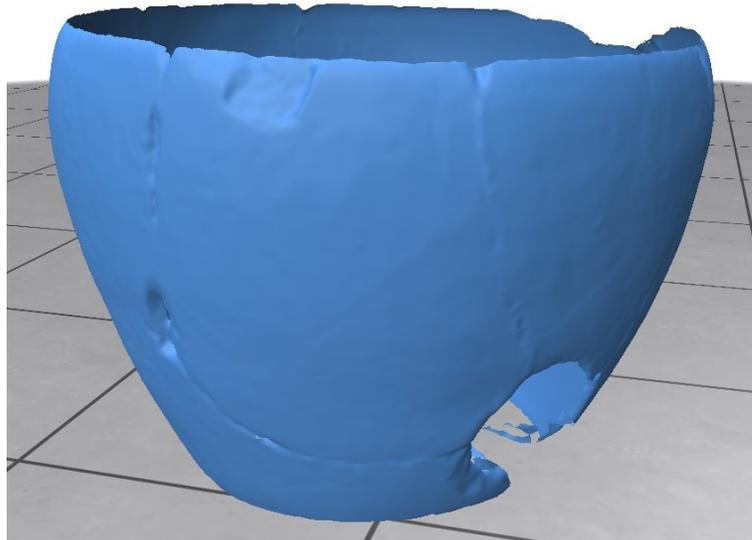


Figura 5.1: Visualización de una malla a partir de la información en su archivo .ply

5.5. Texturizado

A partir de la información anteriormente mencionada, también se da cuenta de la presencia de una imagen que contiene la textura del objeto, sin embargo, *polyscope* no cuenta con soporte para objetos texturizados, por lo que para poder texturizar el objeto se hace uso de otra de las propiedades de *polyscope* sobre las mallas de superficie, *color quantity* es la propiedad que asigna color a la malla de superficie y existen dos formas de colorear una superficie, asignar color a las caras o asignar color a los vértices, al asignar color a las caras, cada una de las caras se representa en su totalidad con un color uniforme lo que resulta en un objeto escamoso, por otro lado, el asignar colores a los puntos, produce que el color de los vértices se difumine en las caras hasta llegar al centro de ella, lo que produce una textura mucho más suave por lo que se decide trabajar con este proceso de textura.

Para lograr este texturizado se usa la información contenida en la metadata del objeto correspondiente a la imagen de la textura, las coordenadas que se especifican para cada vértice de acuerdo a la cara de la que forma parte y los vectores normales a la superficie en cada punto, resultando en la visualización de un objeto texturizado.



Figura 5.2: Visualización de una malla texturizada

5.6. Visualización de anotaciones

Dentro de los archivos JSON de cada uno de los objetos se tienen las instancias para cada uno de los patrones identificados y registrados en determinado objeto y para cada instancia se conocen las caras que forman parte de ella, de la misma manera que se logró entonces construir la visualización del objeto, pueden construirse parches de la superficie que corresponden a cada uno de los patrones presentes en el objeto, permitiendo entonces la representación de los objetos y los patrones previamente registrados para cada uno.



Figura 5.3: Visualización de una malla texturizada con sus anotaciones

5.7. Cálculo de centroides

Para comenzar el estudio estructural de los patrones definidos en cada uno de los objetos, la principal característica a tener en consideración es su distribución a lo largo de la superficie de la malla, por lo que lo más importante es determinar la posición de un patrón en el objeto. Debido a que la malla representa una superficie de densidad uniforme, para obtener la posición de un patrón basta calcular su centroide, esto se consigue con el promedio aritmético de la posición de los vértices que forman parte de una cara en particular.

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \frac{\sum_{i=0}^n \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}}{n} \quad (5.1)$$

Donde x_c, y_c, z_c son las coordenadas del centroide de una cara en los ejes x, y, z respectivamente, mientras que x_i, y_i, z_i son las coordenadas de un vértice que forma parte de la cara en cuestión, n es el número de vértices presentes en la cara y entonces considerados para el cálculo del centroide.

Sin embargo, la superficie sobre la que se trabaja tiene 3 dimensiones y la superficie de los objetos es mayormente cóncava. Calcular el centroide de los patrones resulta en un punto al interior del objeto, mientras más grande el patrón, más alejado está el centroide de la superficie del objeto.

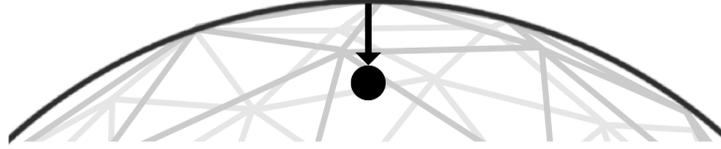


Figura 5.4: Ejemplo de cálculo de centroide en 2D

Para conseguir que el centroide calculado corresponda con la superficie del objeto, se realiza una proyección del centroide para llevarlo a la superficie, para determinar el vector necesario para desplazar el centroide se realiza un análisis de componentes principales.

Para realizar este análisis se comienza con las coordenadas calculadas para el centroide y las coordenadas de los puntos que fueron usados para calcularlas, a partir de esta información se forma la matriz de covarianza respecto al centroide.

$$Cov(c) = \sum_{i=0}^n \begin{pmatrix} |x_c - x_i| & |y_c - x_i| & |z_c - x_i| \\ |x_c - y_i| & |y_c - y_i| & |z_c - y_i| \\ |x_c - z_i| & |y_c - z_i| & |z_c - z_i| \end{pmatrix} \quad (5.2)$$

Se calculan los vectores y valores propios de esta matriz de covarianza con la ayuda de la librería *numpy*, una vez obtenidos los valores y vectores propios se puede identificar el vector que representa la transformación necesaria para llevar el centroide a la superficie del objeto, este vector corresponde al vector propio que se relaciona con el menor valor propio obtenido, entonces sea u el menor valor propio y \vec{v} el vector propio correspondiente a u , se tiene que el vector de translación tal que el centroide está en la superficie es:

$$Tr(c) = u\vec{v} = u \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \quad (5.3)$$

Donde se obtiene un vector que entrega la magnitud de la transformación en tres dimensiones. Dado que el análisis de componentes principales describe el conjunto de puntos en nuevas componentes que se pueden relacionar con las coordenadas cartesianas en las que originalmente estaba representado.

Así, se puede lograr, junto con la representación de los centroides calculados como una nube de puntos sobre el objeto texturizado y etiquetado, la visualización de la posición de los patrones identificados en el objeto.



Figura 5.5: Objeto texturado con sus centroides calculados y proyectados a su superficie

5.8. Proyección de centroides

Para continuar con el análisis estructural de los patrones, se plantea reducir el problema a un problema en dos dimensiones, proyectando el objeto en un plano, de forma similar a la proyección cilíndrica de Lambert para el globo terráqueo, usando un cilindro centrado en eje y para proyectar el objeto y posteriormente aplanar, esto abusando del supuesto que los objetos son mayormente redondos y, por tanto, siempre contenidos en el cilindro propuesto.

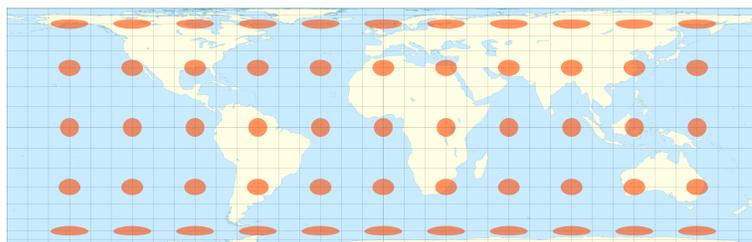


Figura 5.6: Proyección cilíndrica de Lambert

Si bien esta proyección produce una deformación sobre el objeto, se determina que dicha deformación no producirá un error significativo dado que los objetos, a diferencia de globo terráqueo, no son esferas, lo que hace que la deformación de la proyección en las partes superior e inferior de ellos no sea excesiva.

Para proyectar los puntos al cilindro se tienen las siguientes relaciones:

$$x_c = \frac{x_o}{d} \quad y_c = y_o \quad z_c = \frac{z_o}{d} \quad d = \sqrt{x_o^2 + z_o^2} \quad (5.4)$$

Donde x_c, y_c, z_c son las coordenadas de un punto en el cilindro y x_o, y_o, z_o las coordenadas de un punto del objeto, la altura de un punto se mantiene y las coordenadas en las dimensiones x, z se normalizan a una circunferencia de radio 1.

Luego, para pasar las coordenadas cilíndricas a un plano, se plantean las siguientes relaciones:

$$x_p = \begin{cases} \arccos(x_c) & \text{si } z_c \geq 0 \\ 2\pi - \arccos(x_c) & \text{si } z_c < 0 \end{cases} \quad y_p = y_c \quad (5.5)$$

Finalmente, a partir de puntos presentes contenidos en un cubo de lado 2 se llega a un plano de lados 2 y 2π definido por las siguientes relaciones:

$$x_p = \begin{cases} \arccos(\frac{x_o}{d}) & \text{si } z_c \geq 0 \\ 2\pi - \arccos(\frac{x_o}{d}) & \text{si } z_c < 0 \end{cases} \quad y_p = y_o \quad d = \sqrt{x_o^2 + z_o^2} \quad (5.6)$$

La proyección de los centroides al cilindro se ve representado de forma gráfica como sigue:

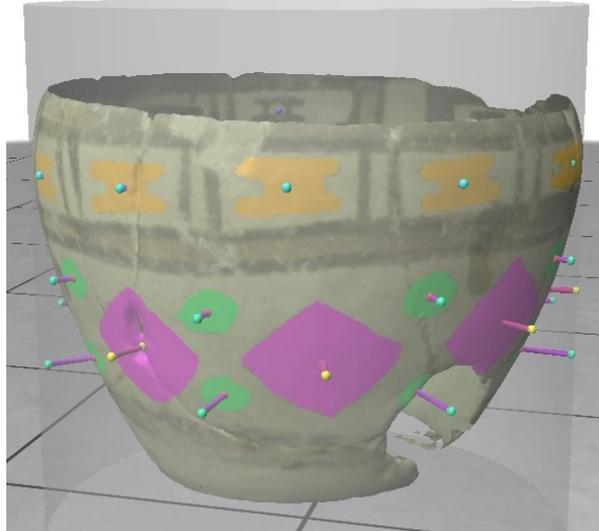


Figura 5.7: Proyección de los centroides a un cilindro de radio 1

5.9. Relación entre centroides

Para estudiar las estructuras regulares que se forman a partir de los puntos proyectados, se relacionan usando grafos $k - nearestneighbor$, los k vecinos más cercanos a cada punto, de esta forma se obtienen matrices de adyacencia que se utilizarán para posteriormente identificar macroestructuras.

La relación de los puntos se calcula sin ningún problema en el espacio 3D en el que se encuentran originalmente con la ayuda de la librería *sklearn*, la matriz obtenida permite fácilmente relacionar los puntos en el plano equivalente a la proyección cilíndrica de los centroides.

5.10. Heatmap

Con el fin de permitir la visualización de los puntos y las relaciones entre ellos, se construye un mapa de calor representando en diferentes colores cada uno de arquetipos de patrones existentes en un objeto, para esto se construye una imagen y utilizando una función de distancia se calcula la saturación de cada pixel.

$$S(p) = \left(1 - \left(\frac{d(p,c_i)}{r}\right)^t\right)^h \quad (5.7)$$

La función de saturación de un pixel p entonces está relacionada con la función distancia d entre p y los centroides y líneas que los relacionan (c_i) además de un radio r dentro del cual el pixel es efectivamente saturado y los términos t y h cumplen la función de suavizar la función de saturación.

El resultado del proceso descrito resulta en la siguiente imagen:

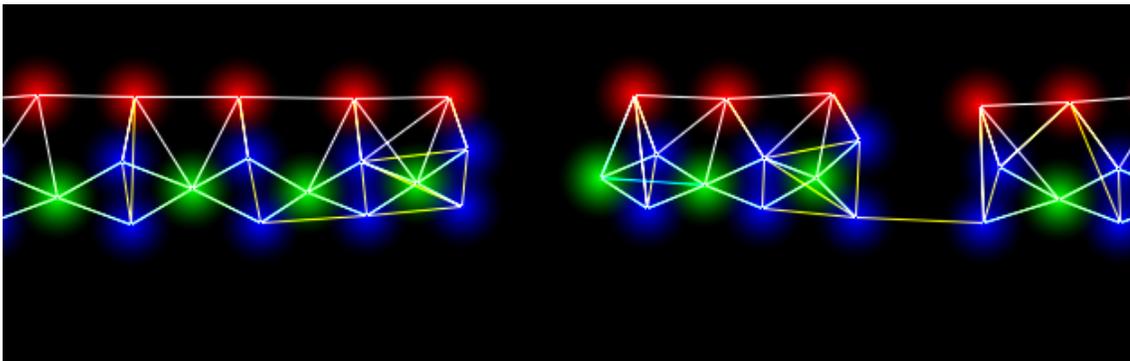


Figura 5.8: Heatmap generado a partir del objeto de muestra

Cabe destacar que esta imagen cumple exclusivamente la función de permitir visualizar el grafo a analizar. En los lineamientos de este propósito, la imagen construida se puede utilizar para texturizar la visualización del cilindro de proyección usando el mismo proceso de texturizado usado sobre el objeto, resultando en la siguiente visualización.

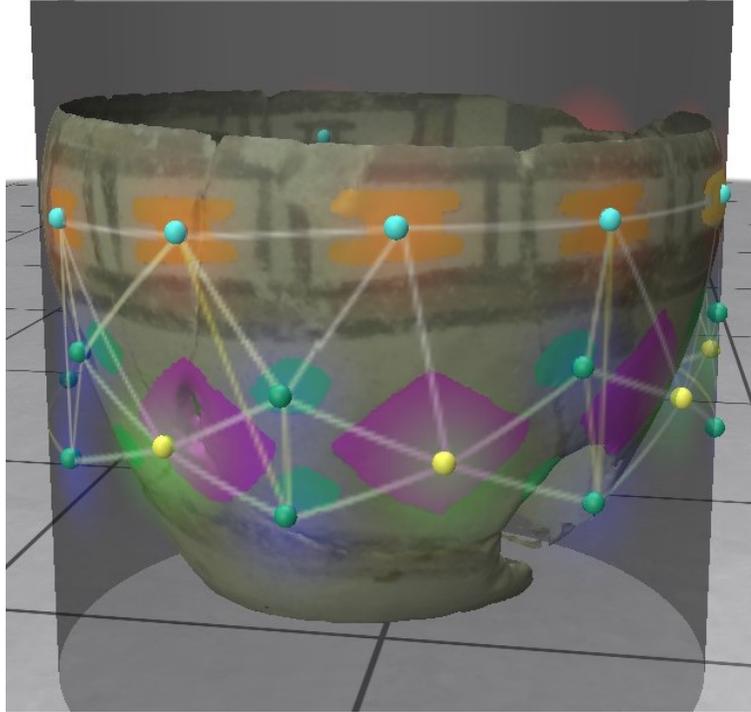


Figura 5.9: Texturizado del heatmap sobre el cilindro de proyección

5.11. Vector de características

Para comparar las estructuras formadas por los centroides y sus vecinos y poder realizar una comparación con los demás objetos que forman parte del conjunto de datos, se construye un vector de características que consiste en un vector de largo 8 donde cada dimensión del vector representa la proporción de ángulos que hace en media un centroide perteneciente a un arquetipo específico de un objeto con sus centroides vecinos, pudiendo estos últimos ser o no de su mismo arquetipo.

El vector construido tiene dimensión 8 pues se opta por agrupar los ángulos en 8 clases de equivalencia, que son: 0 , $\pi/4$, $\pi/2$, $3\pi/4$, π , $5\pi/4$, $3\pi/2$ y $7\pi/4$, entonces un ángulo α formado entre el centroide y su vecino se puede clasificar en una de estas clases de equivalencia X de la siguiente forma:

$$\alpha \in [X] \quad \text{si} \quad \left(X - \frac{\pi}{8}\right) < \alpha \leq \left(X + \frac{\pi}{8}\right) \quad (5.8)$$

Este vector de características, al considerar los ángulos formados para su construcción, ignora la longitud de los vectores que relacionan a los centroides, por lo que se pueden identificar patrones regulares de cualquier tamaño.

En un objeto, se construye un vector de características para cada arquetipo de patrones presentes en él, para esto, se realiza la media aritmética de los vector de características de todos los centroides originados de las ocurrencias del mismo arquetipo, lo que también permite identificar estructuras sin importar la densidad de centroides.

Con lo dicho anteriormente, el vector de características D de un centroide c se define de forma tal que:

$$D(c) = \frac{\begin{pmatrix} X_1 \\ \vdots \\ X_8 \end{pmatrix}}{k} \quad (5.9)$$

Donde X_i corresponde a la cantidad de vecinos cuyos ángulos con c forman un ángulo β donde $\beta = X_i$ y k es la cantidad de vecinos relacionados con c . Representado de forma gráfica, este vector de características puede verse como sigue:

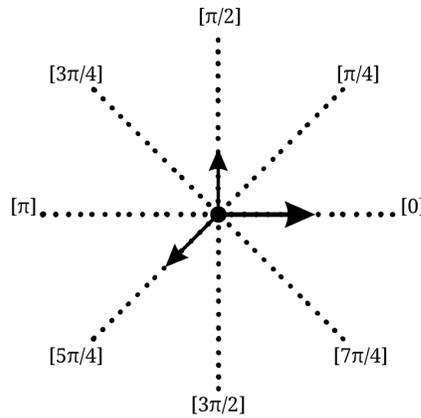


Figura 5.10: Representación gráfica del vector de características construido

El vector de características de un arquetipo en particular de un objeto, se define como:

$$D(ob)[A_i] = \sum_{c_a=0}^n \frac{D(c_a)}{n} \quad (5.10)$$

Donde $D(ob)[A_i]$ es el vector de características del objeto según el arquetipo A_i presente en el objeto y c_a corresponde a cada centroide originado de un patrón perteneciente al arquetipo A_i

Para comparar dos objetos, se calcula la distancia de la siguiente forma:

$$d(D(ob_1)[A_i], D(ob_2)[A_e]) = \sum_{j=1}^8 |D(ob_1)[A_i][X_j] - D(ob_2)[A_e][X_j]| \quad (5.11)$$

Obteniendo una distancia para cada par de arquetipos que se puedan comparar entre los objetos, la menor distancia encontrada indica que tan parecidos son las estructuras que forman los patrones pertenecientes a los arquetipos comparados en el resultado considerado.

5.12. Pipeline

Un objeto del conjunto de datos se encuentra contenido en un conjunto de archivos que describen sus características estructurales, los cuales son: archivo de forma en formato PLY, que caracteriza la superficie del objeto; archivo de texto en formato JSON, que describe los patrones identificados en la superficie del objeto y archivo de imagen en formato JPG que representa la textura del objeto. A partir de estos archivos se construye el modelo texturizado del objeto que tiene como fin validar la correcta interpretación de los archivos. Con esta información se calculan los centroides de cada patrón y se proyectan a la superficie de un cilindro, con esta información se construye un grafo relacional que muestra las relaciones entre patrones más cercanos para finalmente construir el vector de características asociado a los macro patrones del objeto.

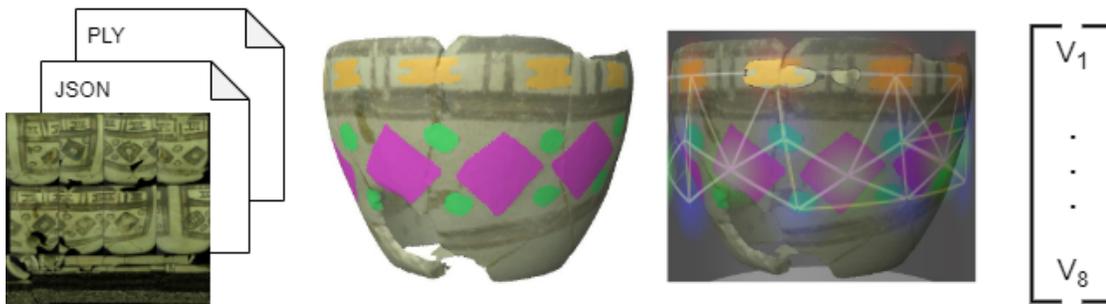


Figura 5.11: Procesamiento de la información del conjunto de datos

Capítulo 6

Experimentos

6.1. Clasificación

Una vez que los objetos tienen un vector de características asociado a cada arquetipo de patrón en ellos, se busca clasificar macro patrones similares en la misma categoría. Debido a las características del conjunto de datos, es imposible saber de antemano cuáles son los diferentes tipos de macro patrones con los que un objeto puede corresponder. Para abordar esta situación se utilizará DBSCAN, un algoritmo de clusterización espacial que es capaz de encontrar el número de grupos que se forman en un conjunto a partir de la distribución de densidad de los elementos de dicho conjunto. Se opta por DBSCAN por sobre otros algoritmos de clusterización principalmente porque resuelve la problemática planteada anteriormente a partir de dos parámetros. Estos parámetros son: el número mínimo de elementos relacionados para formar un grupo y una distancia de tolerancia que dictamina si dos elementos pertenecen al mismo grupo.

Originalmente, el vector de características tiene 8 dimensiones, pero con el objetivo de facilitar la visualización de los resultados y simplificar el criterio de distancia a emplear dentro del algoritmo DBSCAN, los vectores se representarán en un espacio de dos dimensiones. Como primer acercamiento, se buscó utilizar el vector de forma directa para representar el patrón correspondiente a él en el espacio de dos dimensiones utilizando cada dimensión como un vector de posición con una determinada magnitud en los ejes x e y de la siguiente forma.

$$f(D) = (\sum_{k=1}^8 D_k \cos(\frac{k\pi}{4}), \sum_{k=1}^8 D_k \sin(\frac{k\pi}{4})) \quad (6.1)$$

Donde D corresponde a un vector de características.

Esta transformación no clasifica los patrones correctamente, ya que produce que todos los grafos que presentan elementos relacionados con cualquier grado de simetría respecto al elemento considerado tengan la misma imagen en el plano de dos dimensiones, algunos ejemplos simplificados de grafos que tendrían este comportamiento son los siguientes.

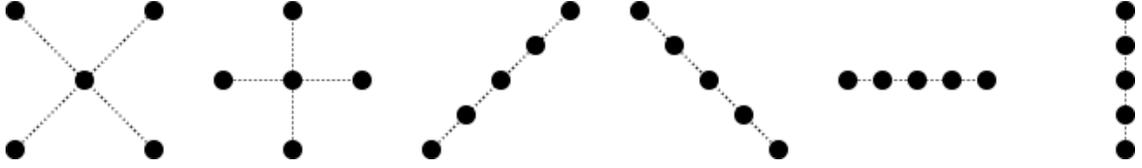


Figura 6.1: Ejemplos de patrones que no debiesen ser parte de la misma clase

Suponiendo que los elementos en la figura [6.1] son la estructura de los macro patrones presentes en los objetos, por mera observación se puede decir que no pertenecen a la misma clase. Para solucionar este problema se considera modificar la función de transformación utilizada para cambiar la orientación de cada una de las dimensiones del vector resultando en un coeficiente de ponderación que produce que el comportamiento antes mencionado no se produzca, esta función se ve de la siguiente manera.

$$f(D) = (\sum_{k=1}^8 D_k \cos(\frac{k\pi}{4} \cdot \frac{1}{2}), \sum_{k=1}^8 D_k \sin(\frac{k\pi}{4} \cdot \frac{1}{2}))$$

$$f(D) = (\sum_{k=1}^8 D_k \cos(\frac{k\pi}{8}), \sum_{k=1}^8 D_k \sin(\frac{k\pi}{8})) \quad (6.2)$$

De esta forma, el algoritmo DBSCAN aplicado sobre el conjunto de datos usará la distancia euclidiana en dos dimensiones como criterio de relación.

Los puntos obtenidos mediante la función definida constituyen el conjunto de datos sobre el cual se ejecutará DBSCAN. Los parámetros necesarios para el funcionamiento del algoritmo se determinarán de forma exploratoria y los resultados obtenidos de la aplicación del algoritmo dependerán de dichos parámetros.

Se espera que luego de aplicar el algoritmo DBSCAN sobre el conjunto de datos definido, se obtenga un conjunto segregado de elementos que dicte los macro patrones presentes en el conjunto de objetos original.

6.2. Comparación

Con el objetivo de corroborar el resultado de la experiencia anterior se utiliza el criterio de comparación definido en [5.11] de donde se espera que los objetos más cercanos a un objeto de control sean objetos que bajo el criterio de clasificación obtenido con DBSCAN pertenezcan a la misma categoría. Se decide utilizar la fórmula de distancia descrita en [5.11] dado que es un método de comparación que se utiliza directamente sobre los vectores de características de cada objeto antes de reducir la dimensión del vector, ya que de forma directa si se utiliza un procedimiento para determinar el vecino más cercano dentro del espacio de dos dimensiones se obtendrá seguramente un objeto perteneciente a la misma categoría que el objeto de control.

Para este conjunto de experimentos se seleccionan algunos de los objetos del conjunto de datos de forma aleatoria para construir un conjunto de objetos de control y se determinará la categoría de los objetos más cercanos a cada uno con base en el criterio de comparación mencionado anteriormente.

Capítulo 7

Resultados

7.1. Clasificación

Una vez los objetos del conjunto de datos han sido procesados, se tienen los vectores de características de cada objeto y mediante la función definida en [6.2] los vectores se distribuyen en el espacio de dos dimensiones. La representación visual del nuevo espacio obtenido se ve como sigue.

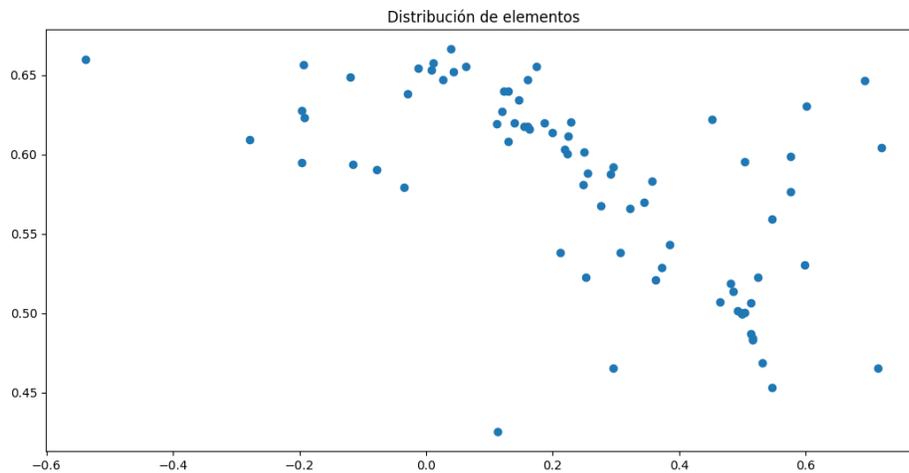


Figura 7.1: Conjunto de elementos luego de la transformación

Sobre este nuevo conjunto se aplica el algoritmo DBSCAN, para poder hacer uso de él es necesario determinar los parámetros sobre los cuales el algoritmo trabaja. Para la primera prueba, se consideró utilizar el número de elementos mínimo para un cluster como el 5% del total del conjunto, dado que en la totalidad de los objetos del conjunto de datos se consideran 113 arquetipos en total, se utiliza el valor 5; para el parámetro de tolerancia, se utiliza un 1% de la distancia máxima entre dos objetos dada la definición del vector de características, por esto se utiliza el valor 0.02. Con estas condiciones se obtiene el siguiente resultado.

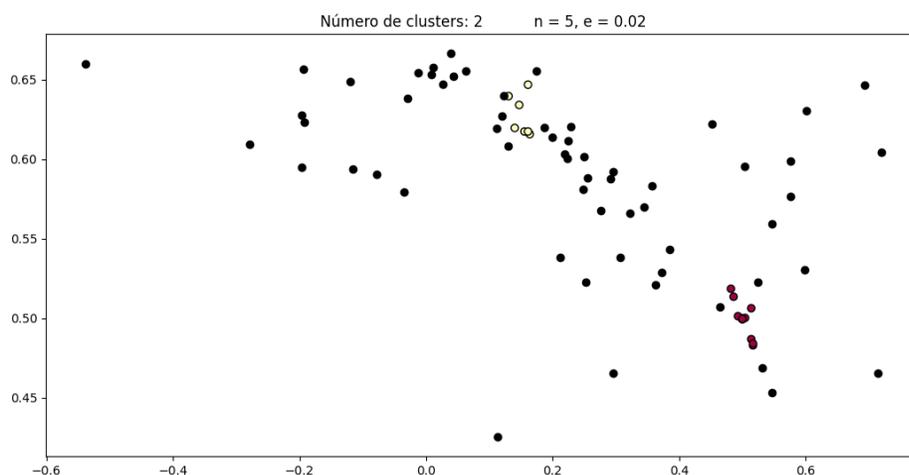


Figura 7.2: Clusterización con DBSCAN con parámetros $n = 5$ y $e = 0.02$

Teniendo en consideración el resultado anterior se generan nuevas condiciones aumentando la distancia de tolerancia, ya que en el resultado anterior se aprecia una gran cantidad de puntos considerados ruido. Se aplica entonces el algoritmo usando una distancia de tolerancia 0.04 y manteniendo el número de elementos mínimo para formar un cluster con un valor de 5, obteniendo el siguiente resultado.

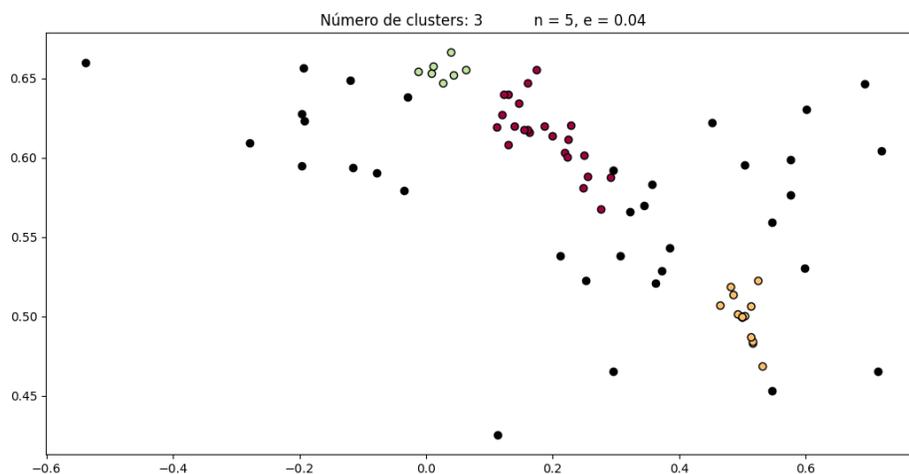


Figura 7.3: Clusterización con DBSCAN con parámetros $n = 5$ y $e = 0.04$

Aun considerando 34 puntos, es decir, el 30% de los elementos como ruido, se decide disminuir el número mínimo de elementos para formar un cluster a 3, aproximadamente un 2.5% del total de la muestra y manteniendo el parámetro de tolerancia en 0.04, se llega al siguiente resultado.

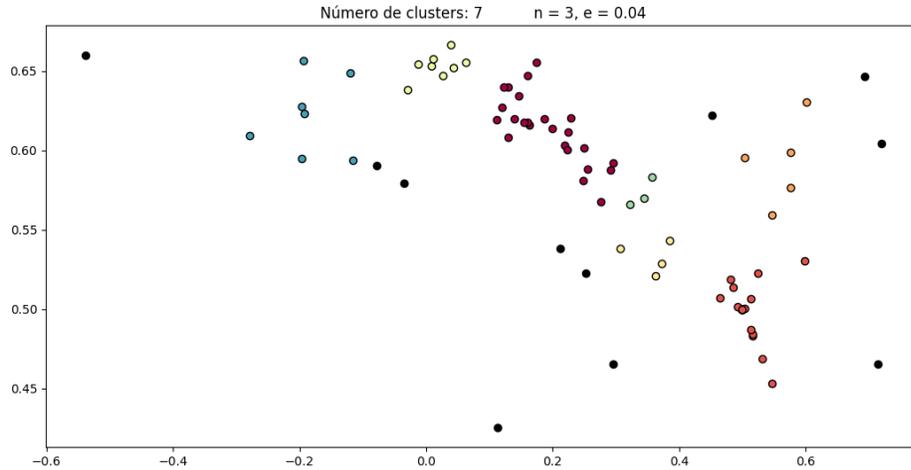


Figura 7.4: Clusterización con DBSCAN con parámetros $n = 3$ y $e = 0.04$

En este último resultado, hay 7 clusters y solo 11 puntos son considerados ruido, que corresponde a menos del 10% del total de la muestra.

7.2. Comparación

Para identificar los diferentes tipos de estructuras y patrones regulares formados por las diversas etiquetas presentes en los objetos, se calcula para cada objeto y cada uno de sus arquetipos, el descriptor construido y se comparan usando la distancia definida, se opta por corroborar sobre el vector descrito originalmente, ya que buscar los objetos más cercanos luego de la transformación al plano de dos dimensiones entrega la misma información que el algoritmo de clasificación, un objeto similar se encontrará en la vecindad del elemento.

A continuación se presentan cinco objetos seleccionados al azar del total de objetos considerados. Para cada objeto de control se presentan los macro patrones presentes en él y su respectivo vector de características y transformación al espacio de dos dimensiones. En la sección de cada uno de los objetos de control se presentan cinco objetos adicionales, los cuales representan los objetos más cercanos al objeto de control según la distancia entre los vectores de características de sus macro patrones, estos objetos se destacan también en el espacio de dos dimensiones construido para posteriormente determinar si pertenece a la misma categoría del objeto de control.

Objeto 1

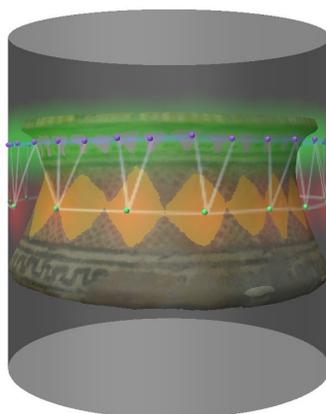


Figura 7.5: Objeto 1 con patrones resaltados y mapa de calor sobrepuesto.

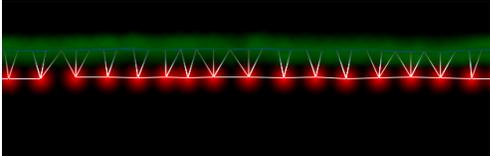
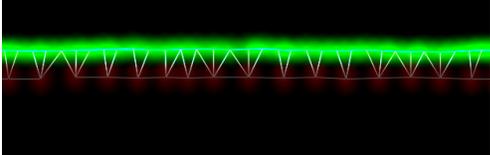
Macro patrón	Vector	Transformación
	$[0.1875$ 0.0937 0.3906 0.1328 0.1953 0.0000 0.0000 $0.0000]$	$[0.6011$ $0.6301]$
	$[0.4053$ 0.0000 0.0000 0.0000 0.3901 0.0000 0.1666 $0.0378]$	$[0.2524$ $0.5224]$

Tabla 7.1: Macro patrones del objeto 1, vectores de características y transformaciones.

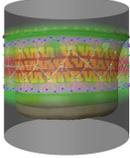
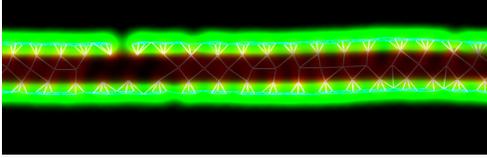
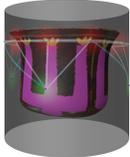
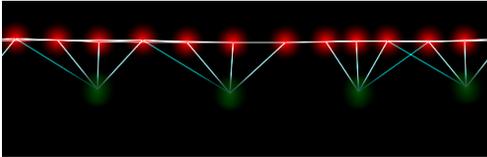
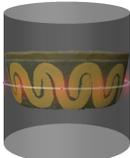
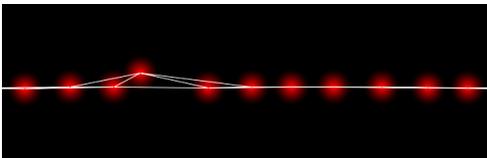
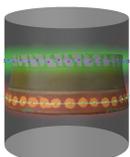
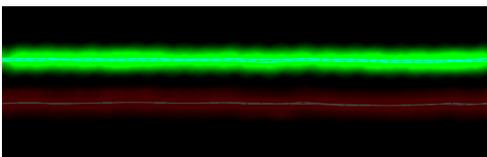
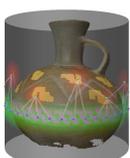
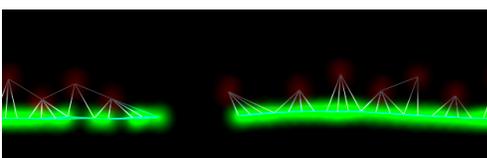
Objeto	Macro patrón	Vector	Distancia	Transformación
		[0.3825 0.0331 0.0388 0.0549 0.3797 0.0293 0.0445 0.0369]	0.3124	[0.3848 0.5433]
		[0.4583 0.0000 0.0000 0.0000 0.2916 0.0937 0.0833 0.0729]	0.3636	[0.2961 0.4651]
		[0.4772 0.0000 0.0000 0.0227 0.4772 0.0000 0.0000 0.0227]	0.3636	[0.4649 0.5069]
		[0.4970 0.0000 0.0000 0.0000 0.4970 0.0000 0.0059 0.0000]	0.3971	[0.4928 0.5012]
		[0.4314 0.0443 0.0524 0.0403 0.4314 0.0000 0.0000 0.0000]	0.4090	[0.5249 0.5227]

Tabla 7.2: Macro patrones más cercanos a aquellos en el objeto 1.

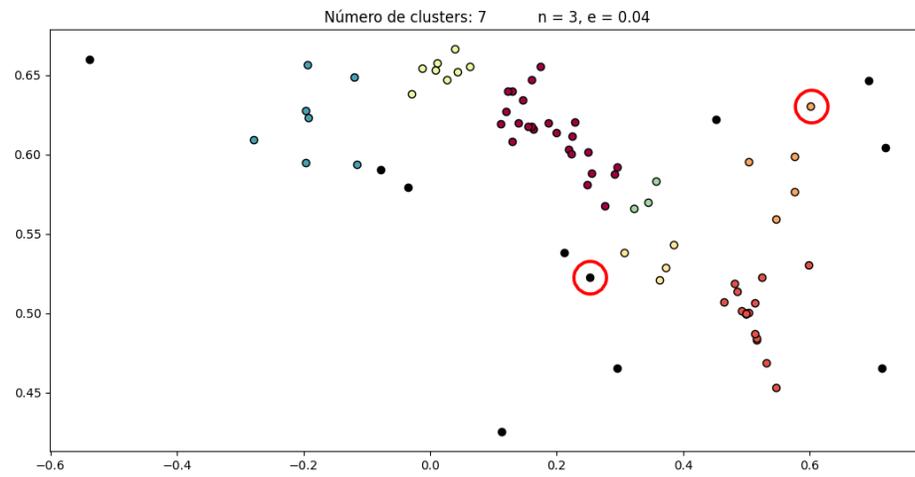


Figura 7.6: Distribución de los macro patrones del objeto 1 en el espacio bidimensional.

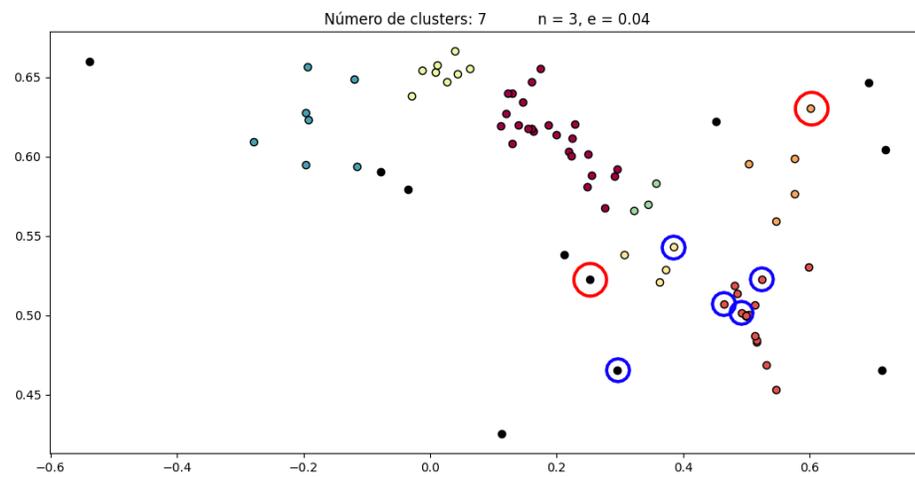


Figura 7.7: Distribución de los macro patrones cercanos al objeto 1 en el espacio.

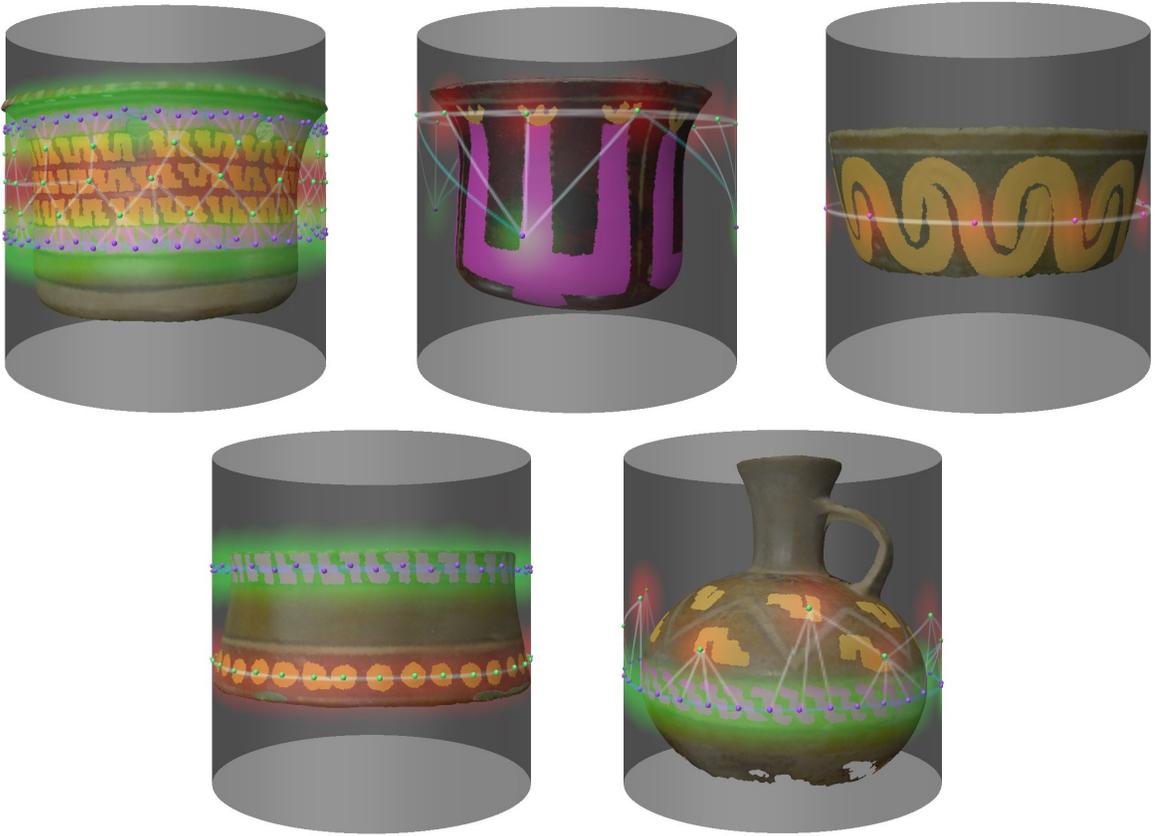


Figura 7.8: Objetos más parecidos al objeto 1 con base en el criterio utilizado.

Objeto 2

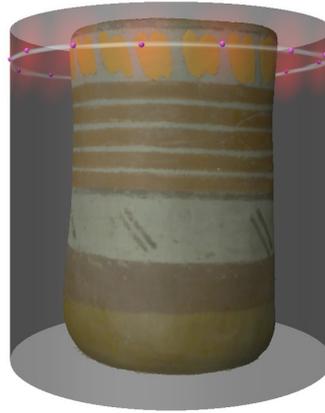


Figura 7.9: Objeto 2 con patrones resaltados y mapa de calor sobrepuesto.

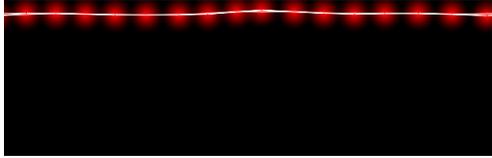
Macro patrón	Vector	Transformación
	$[0.5000$ 0.0000 0.0000 0.0000 0.5000 0.0000 0.0000 $0.0000]$	$[0.5000$ $0.5000]$

Tabla 7.3: Macro patrones del objeto 2, vectores de características y transformaciones.

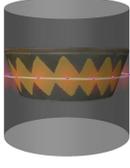
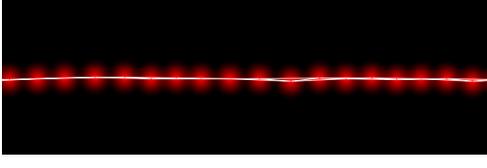
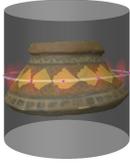
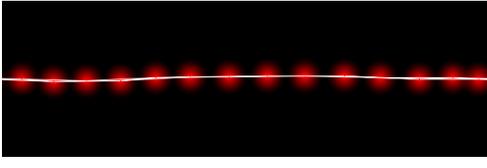
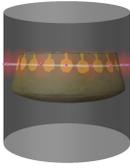
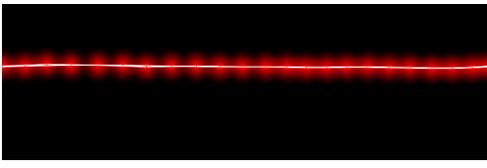
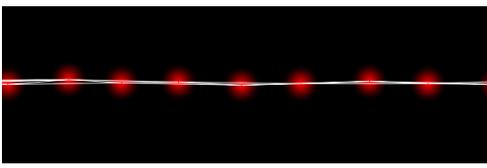
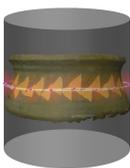
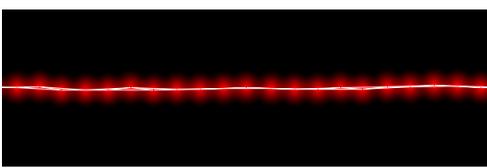
Objeto	Macro patrón	Vector	Distancia	Transformación
		[0.5000 0.0000 0.0000 0.0000 0.5000 0.0000 0.0000 0.0000]	0.0000	[0.5000 0.5000]
		[0.5000 0.0000 0.0000 0.0000 0.5000 0.0000 0.0000 0.0000]	0.0000	[0.5000 0.5000]
		[0.5000 0.0000 0.0000 0.0000 0.5000 0.0000 0.0000 0.0000]	0.0000	[0.5000 0.5000]
		[0.5000 0.0000 0.0000 0.0000 0.5000 0.0000 0.0000 0.0000]	0.0000	[0.5000 0.5000]
		[0.5000 0.0000 0.0000 0.0000 0.5000 0.0000 0.0000 0.0000]	0.0000	[0.5000 0.5000]

Tabla 7.4: Macro patrones más cercanos a aquellos en el objeto 2.

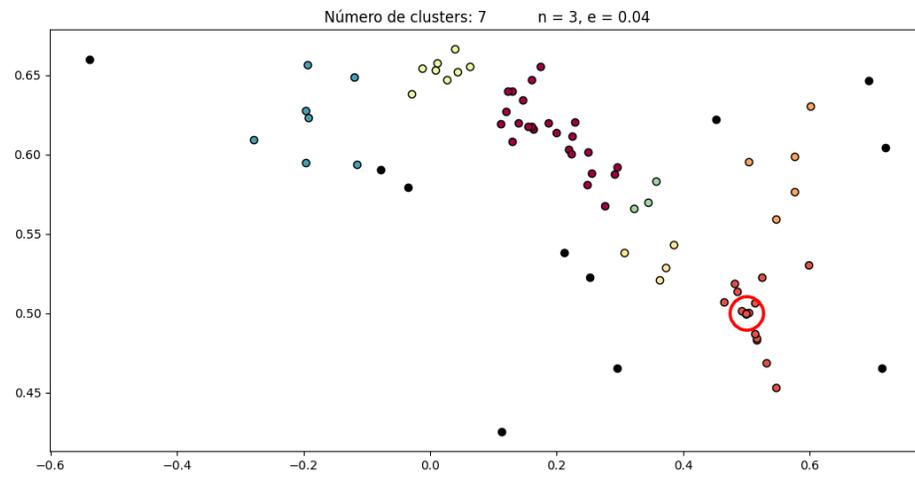


Figura 7.10: Distribución de los macro patrones del objeto 2 en el espacio bidimensional.

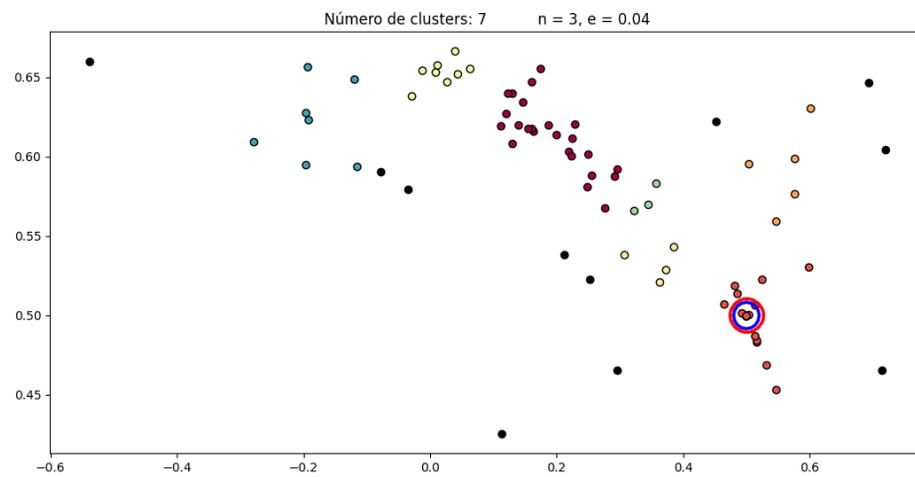


Figura 7.11: Distribución de los macro patrones cercanos al objeto 2 en el espacio.

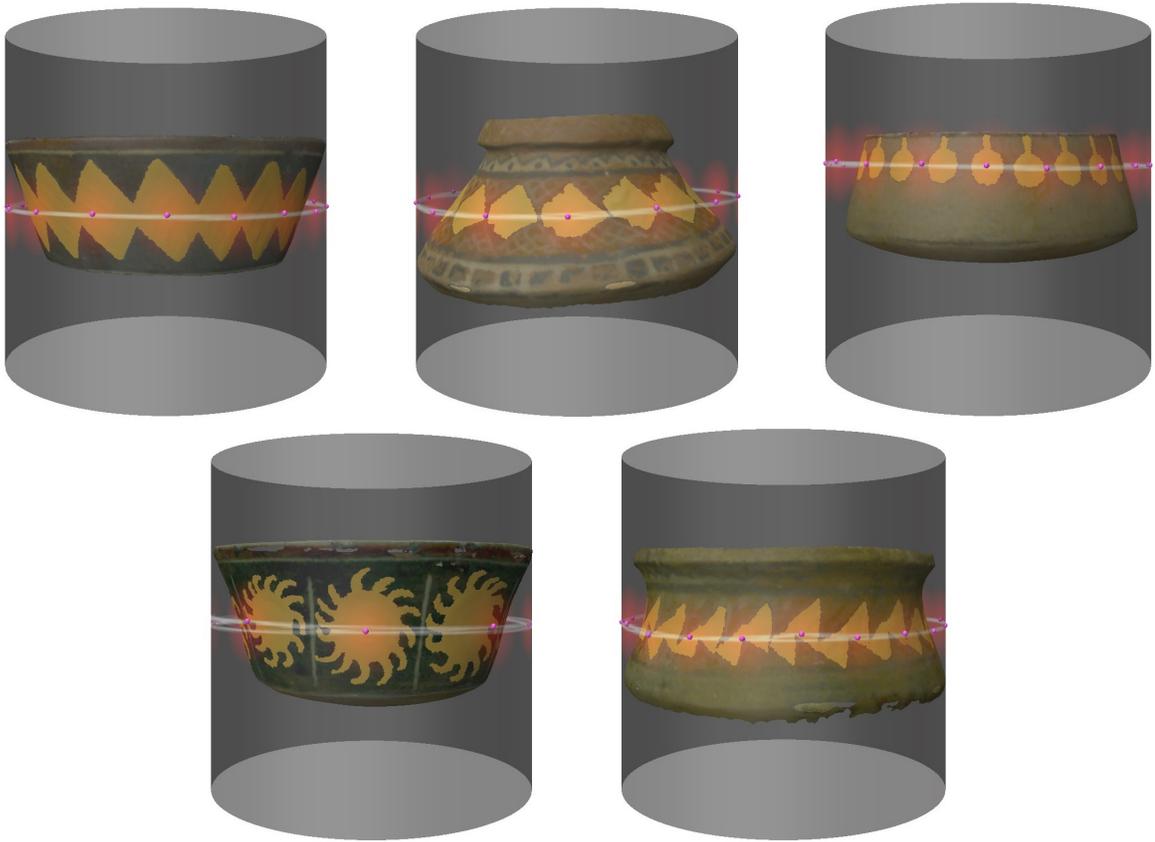


Figura 7.12: Objetos más parecidos al objeto 2 con base en el criterio utilizado.

Objeto 3

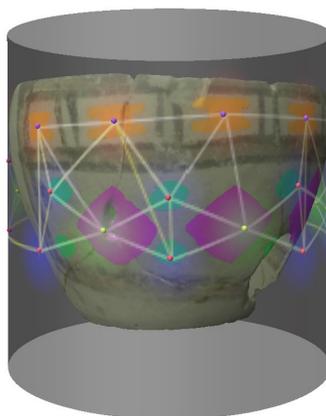


Figura 7.13: Objeto 3 con patrones resaltados y mapa de calor sobrepuesto.

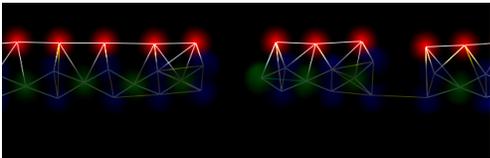
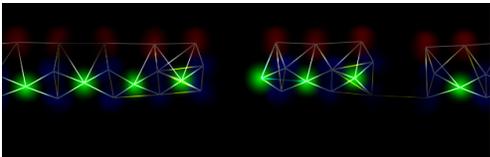
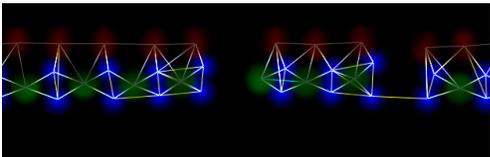
Macro patrón	Vector	Transformación
	[0.1750 0.0000 0.0000 0.0000 0.1625 0.1625 0.3250 0.1750]	[-0.2786 0.6094]
	[0.1875 0.1093 0.0312 0.2500 0.1562 0.1250 0.0000 0.1406]	[0.2285 0.6204]
	[0.1250 0.1388 0.2500 0.0763 0.1250 0.0486 0.1250 0.1111]	[0.2496 0.6013]

Tabla 7.5: Macro patrones del objeto 3, vectores de características y transformaciones.

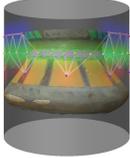
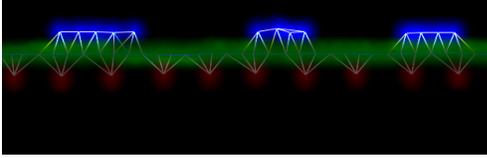
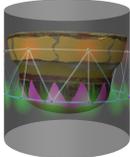
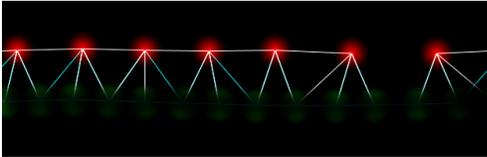
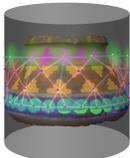
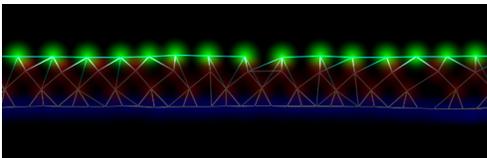
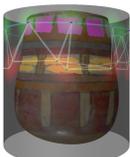
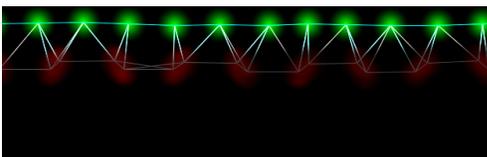
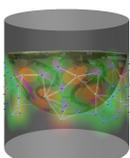
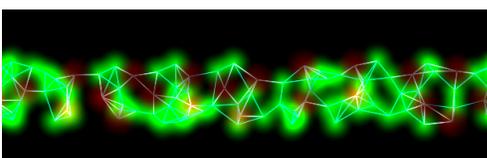
Objeto	Macro patrón	Vector	Distancia	Transformación
		[0.2115 0.0000 0.0000 0.0000 0.2019 0.1153 0.3269 0.1442]	0.1557	[-0.1970 0.5948]
		[0.1964 0.0000 0.0000 0.0000 0.1964 0.1964 0.2857 0.1250]	0.1785	[-0.1962 0.6277]
		[0.1916 0.0000 0.0000 0.0000 0.2250 0.2083 0.1916 0.1833]	0.2666	[-0.1929 0.6231]
		[0.2386 0.0000 0.0000 0.0000 0.2386 0.1363 0.2500 0.1363]	0.2795	[-0.1163 0.5935]
		[0.1322 0.1286 0.1213 0.1358 0.1177 0.1068 0.1358 0.1213]	0.2922	[0.1397 0.6196]

Tabla 7.6: Macro patrones más cercanos a aquellos en el objeto 3.

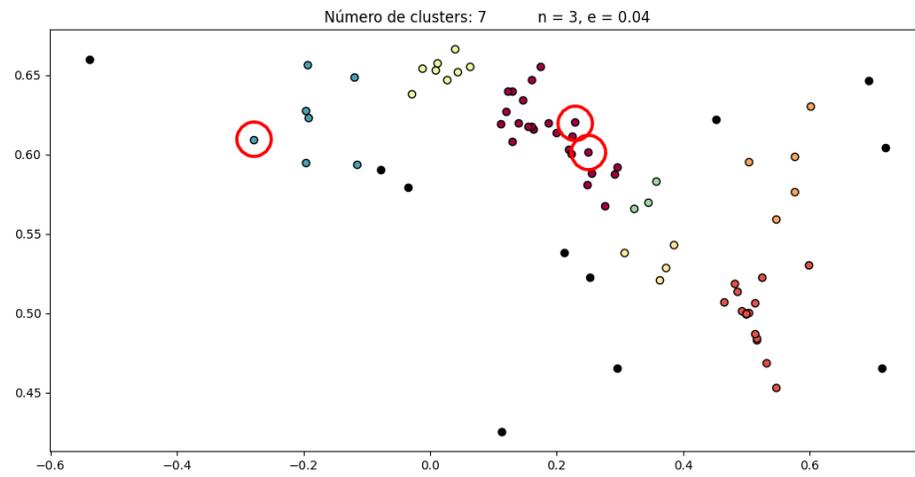


Figura 7.14: Distribución de los macro patrones del objeto 3 en el espacio bidimensional.

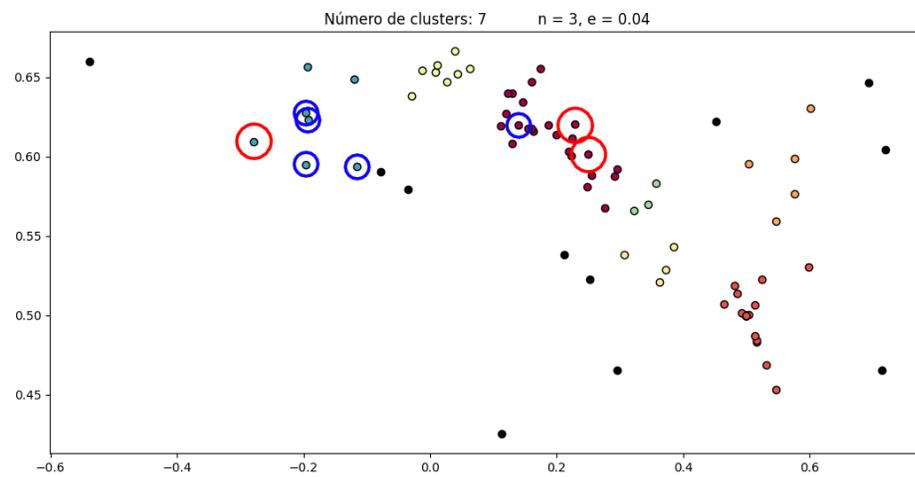


Figura 7.15: Distribución de los macro patrones cercanos al objeto 3 en el espacio.

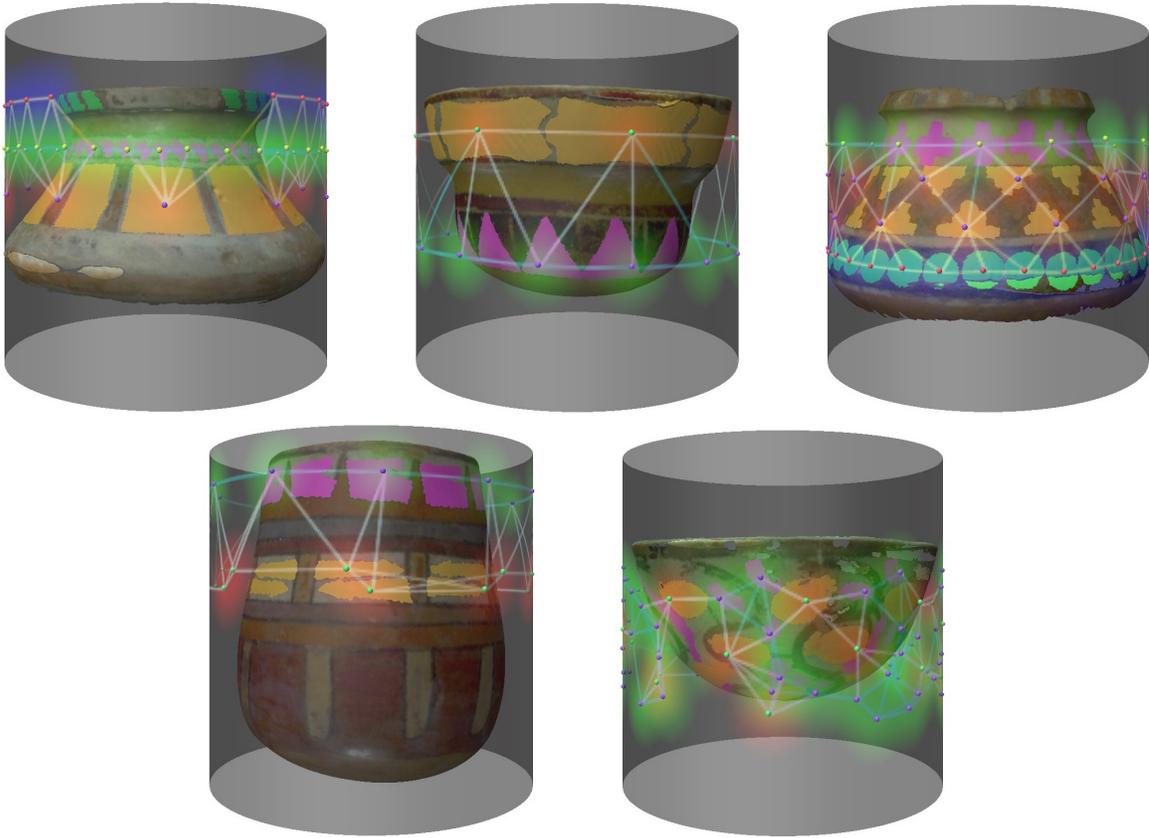


Figura 7.16: Objetos más parecidos al objeto 3 con base en el criterio utilizado.

Objeto 4

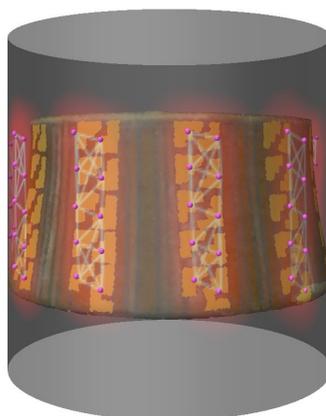


Figura 7.17: Objeto 4 con patrones resaltados y mapa de calor sobrepuesto.

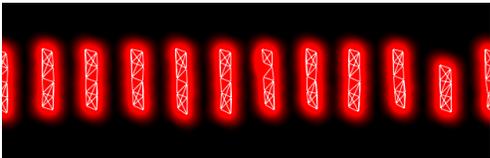
Macro patrón	Vector	Transformación
	[0.0794 0.0725 0.2407 0.1094 0.0794 0.0725 0.2384 0.1071]	[0.0632 0.6553]

Tabla 7.7: Macro patrones del objeto 4, vectores de características y transformaciones.

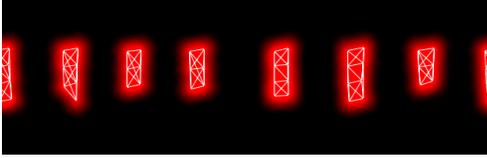
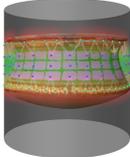
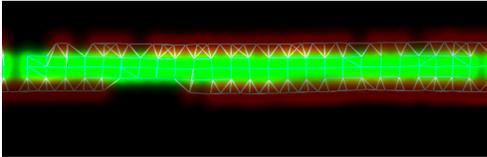
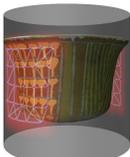
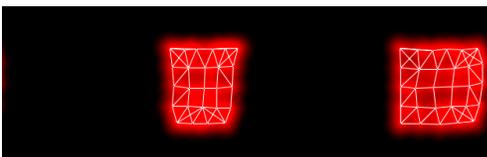
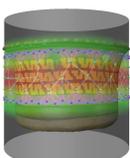
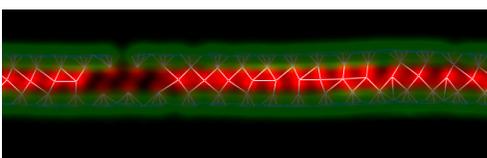
Objeto	Macro patrón	Vector	Distancia	Transformación
		[0.1224 0.0943 0.2474 0.0357 0.1224 0.1147 0.2270 0.0357]	0.3131	[0.1608 0.6468]
		[0.1522 0.0560 0.2339 0.0512 0.1570 0.0448 0.2612 0.0432]	0.3460	[0.1472 0.6340]
		[0.1718 0.0820 0.1289 0.1054 0.1640 0.0625 0.1757 0.1093]	0.3772	[0.1299 0.6079]
		[0.2083 0.0532 0.1851 0.0694 0.2129 0.0347 0.1875 0.0486]	0.5246	[0.2242 0.6117]
		[0.0231 0.1527 0.1574 0.1689 0.0185 0.1504 0.1481 0.1805]	0.5820	[0.0111 0.6572]

Tabla 7.8: Macro patrones más cercanos a aquellos en el objeto 4.

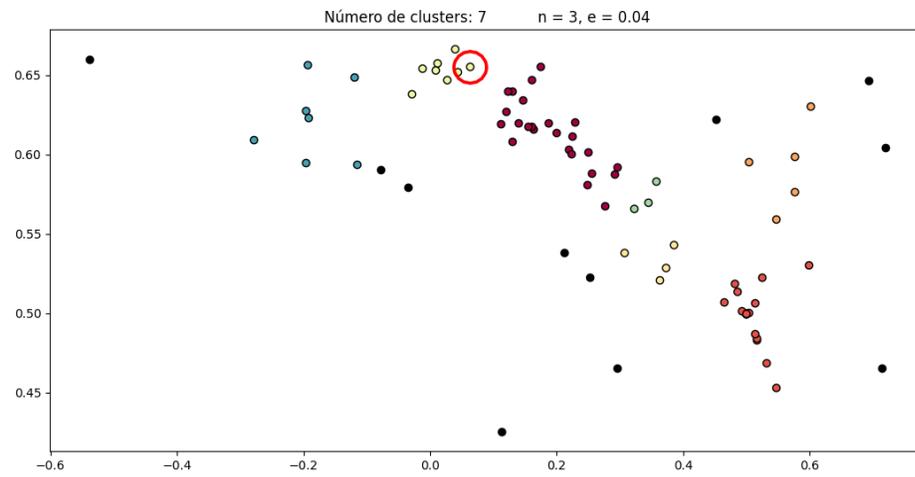


Figura 7.18: Distribución de los macro patrones del objeto 4 en el espacio bidimensional.

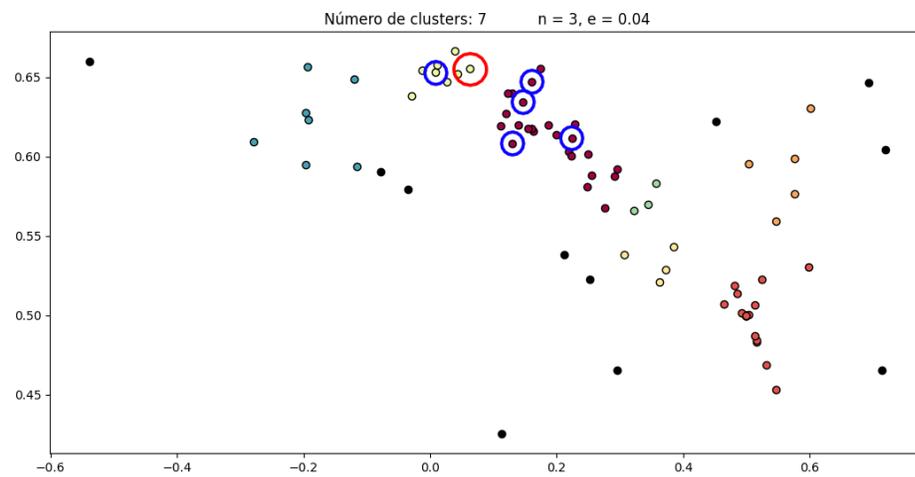


Figura 7.19: Distribución de los macro patrones cercanos al objeto 4 en el espacio.

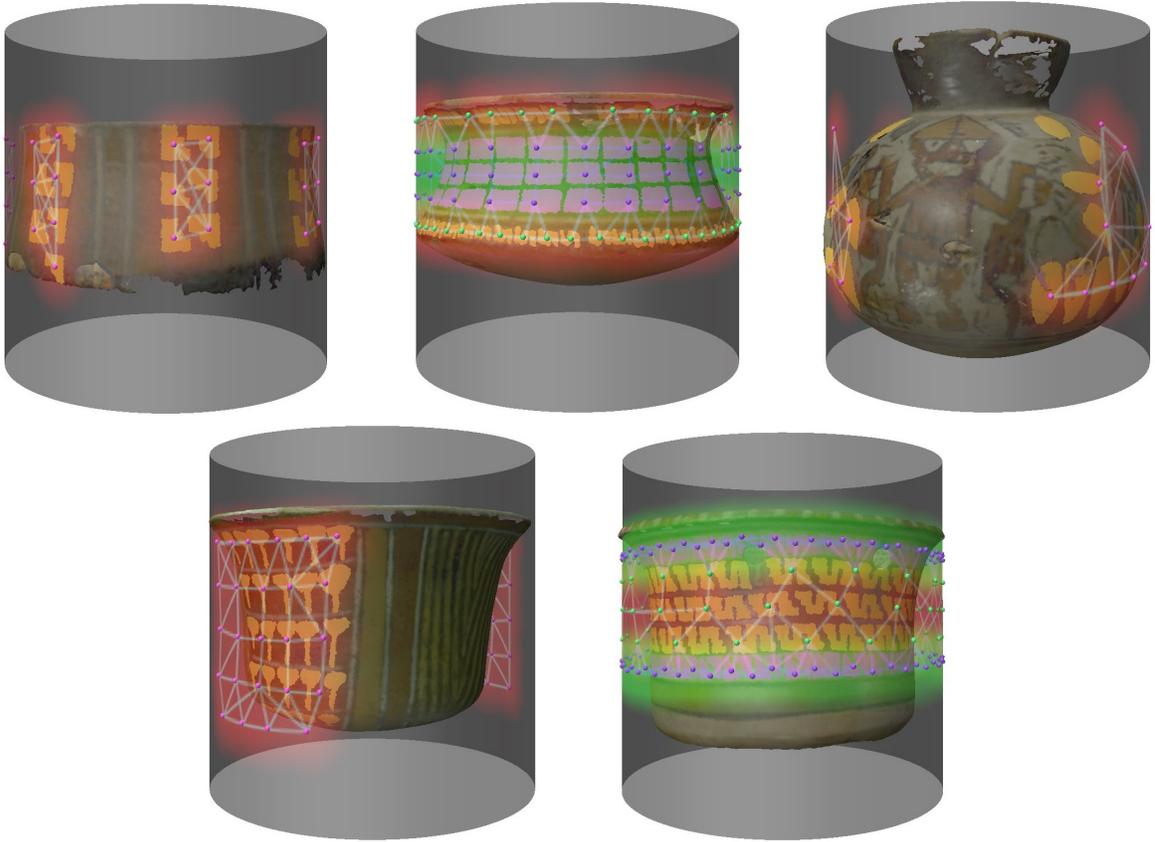


Figura 7.20: Objetos más parecidos al objeto 4 con base en el criterio utilizado.

Objeto 5

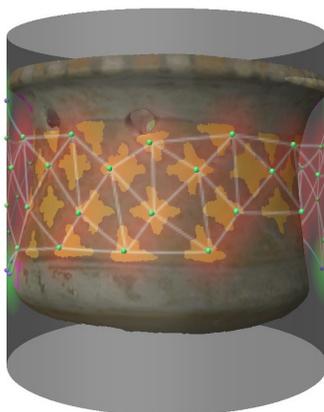


Figura 7.21: Objeto 5 con patrones resaltados y mapa de calor sobrepuesto.

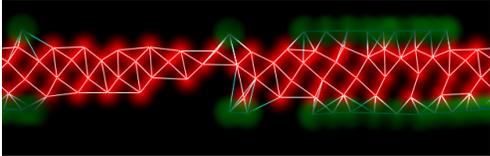
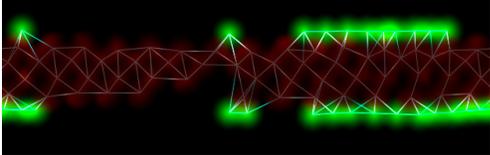
Macro patrón	Vector	Transformación
	$[0.0460$ 0.1754 0.0789 0.1951 0.0350 0.1951 0.0723 $0.2017]$	$[0.0263$ $0.6470]$
	$[0.2307$ 0.0961 0.0913 0.1201 0.2067 0.0576 0.1009 $0.0961]$	$[0.2478$ $0.5806]$

Tabla 7.9: Macro patrones del objeto 5, vectores de características y transformaciones.

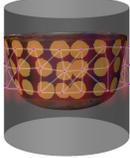
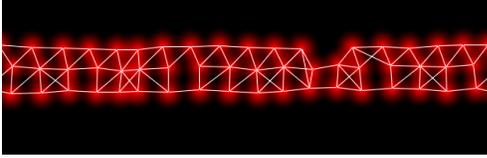
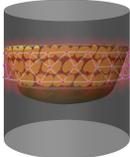
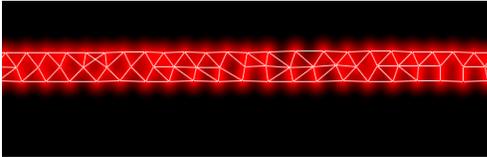
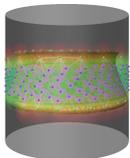
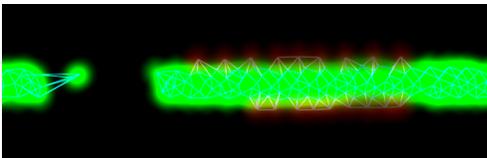
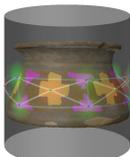
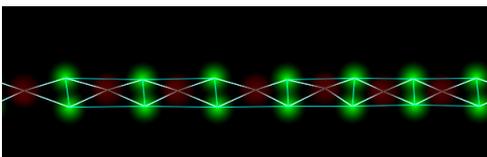
Objeto	Macro patrón	Vector	Distancia	Transformación
		[0.2099 0.0778 0.1273 0.0966 0.2216 0.0518 0.1297 0.0849]	0.1594	[0.2188 0.6030]
		[0.1980 0.1288 0.0711 0.1153 0.2000 0.1134 0.0711 0.1019]	0.1884	[0.2236 0.6003]
		[0.0625 0.1361 0.0914 0.1744 0.0783 0.1679 0.0960 0.1930]	0.1918	[0.0091 0.6532]
		[0.2767 0.0714 0.1250 0.1160 0.2321 0.0357 0.1250 0.0178]	0.2582	[0.3570 0.5833]
		[0.1718 0.0820 0.1289 0.1054 0.1640 0.0625 0.1757 0.1093]	0.2608	[0.1299 0.6079]

Tabla 7.10: Macro patrones más cercanos a aquellos en el objeto 5.

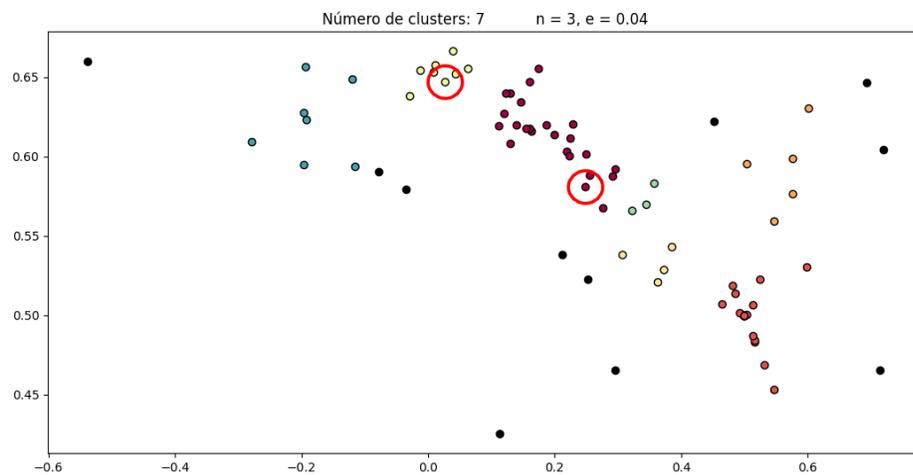


Figura 7.22: Distribución de los macro patrones del objeto 5 en el espacio bidimensional.

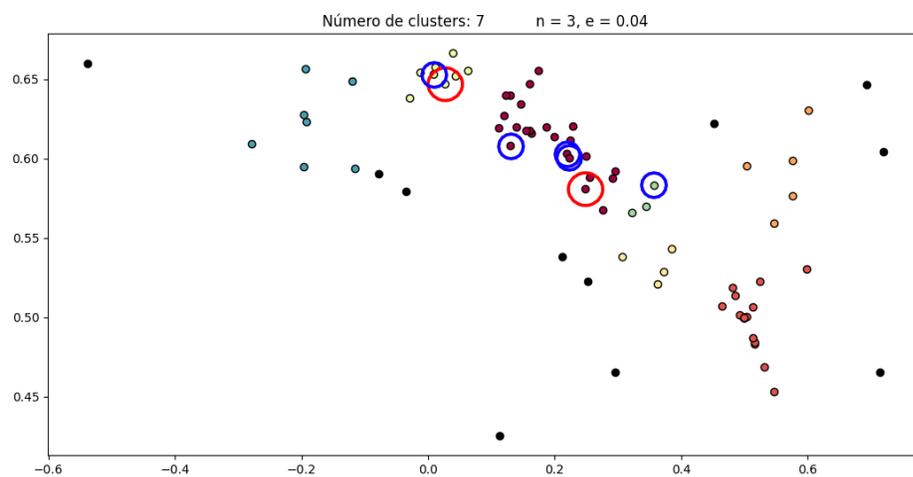


Figura 7.23: Distribución de los macro patrones cercanos al objeto 5 en el espacio.

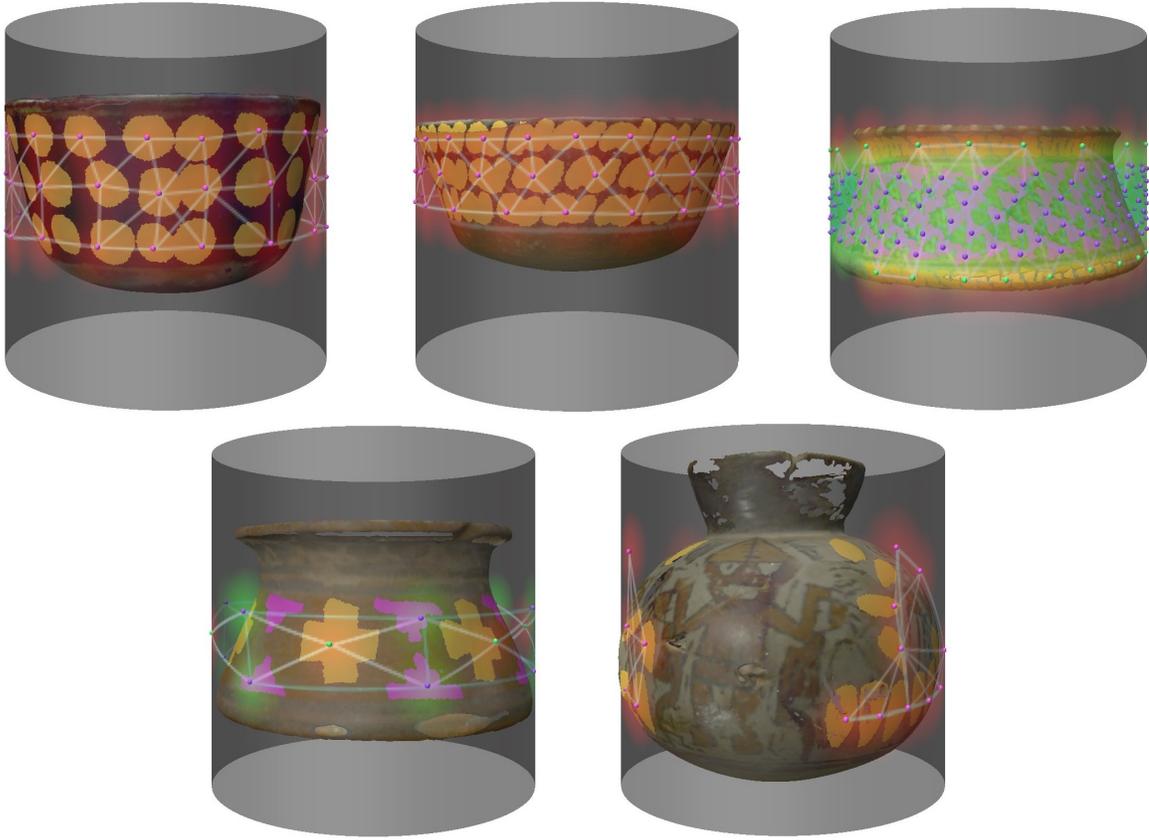


Figura 7.24: Objetos más parecidos al objeto 5 con base en el criterio utilizado.

Capítulo 8

Discusión

8.1. Clasificación

Observando los resultados evidenciados en [7.1] donde se ven 7 clusters, se puede describir cada cluster basándose en los macro patrones que corresponden con dicho cluster, para guiar esta definición, los clusters se enumeran como sigue.

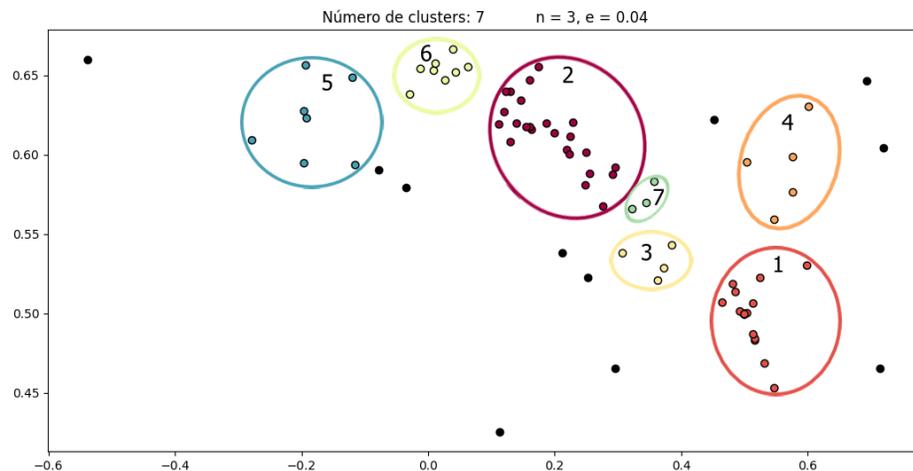


Figura 8.1: Clusters destacados y enumerados

El cluster 1 es el subconjunto más grande, se compone de macro patrones que describen anillos, los patrones se alinean de forma tal que forman circunferencias al rededor del objeto, así este cluster se denomina “anillos”. Algunos ejemplos de objetos pertenecientes a este cluster a continuación.



Figura 8.2: Ejemplos cluster “anillos”

El cluster 2 es el segundo subconjunto más grande, los objetos contenidos en este cluster no muestran una forma completamente definida, pero la mayor parte del tiempo los nodos que se relacionan con otro están uniformemente distribuidos al rededor de este, por esto se les denomina “mallados”. Algunos ejemplos de objetos del cluster son.

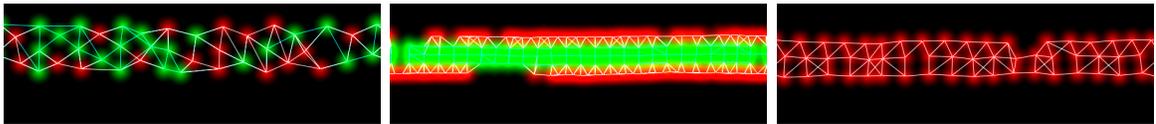


Figura 8.3: Ejemplos cluster “mallados”

El cluster 3 se denomina “zippers”, los objetos dentro de este cluster, presentan patrones que forman anillos que se relacionan con otros patrones o ellos mismos por su proximidad y generalmente se relacionan de forma alternada. Algunos ejemplos de “zippers.” a continuación.

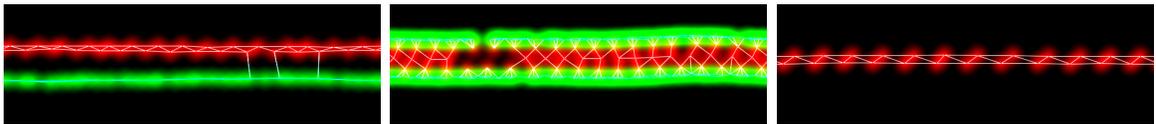


Figura 8.4: Ejemplos cluster “zippers”

El cluster 4 es similar al cluster anterior, denominado “bottom-zipper” dado que los nodos se relacionan de forma parecida al caso anteriormente mencionado, sin embargo, responde a patrones donde los nodos con los que se relacionan están sobre ellos. Ejemplos de este cluster son.

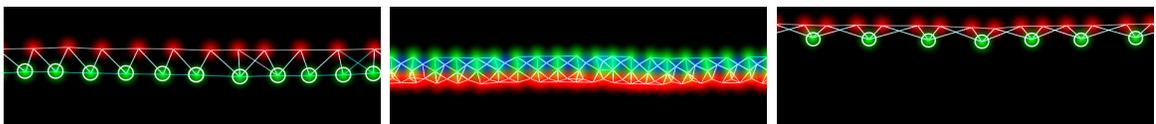


Figura 8.5: Ejemplos cluster “bottom-zippers”

El cluster 5 es el caso análogo del cluster anterior, se denomina “upper-zipper” a los objetos de este cluster, pues se relacionan con patrones que se encuentran bajo ellos. Algunos ejemplos de este cluster a continuación.

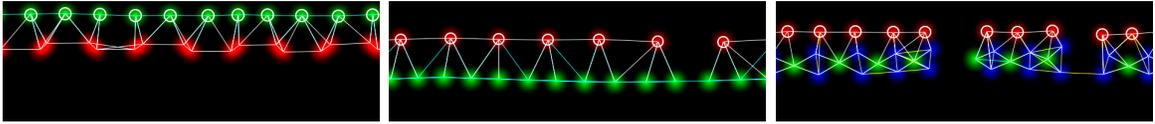


Figura 8.6: Ejemplos cluster “upper-zippers”

El cluster 6 corresponde con objetos que presentan una estructura mayormente ordenada, similar al cluster de “mallados”, sin embargo, estos objetos tienen patrones que se relacionan con sus vecinos formando relaciones en forma de X, por eso se denominan “mallados perfectos”. Algunos ejemplos del cluster a continuación.

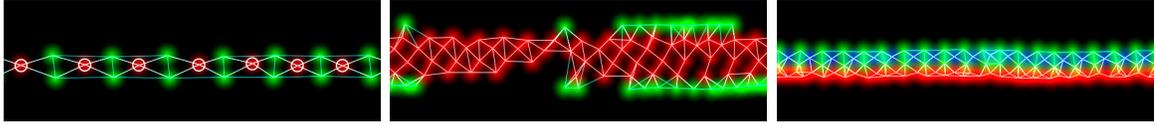


Figura 8.7: Ejemplos cluster “mallados perfectos”

En el cluster 7 y último se encuentran macro patrones que pueden ser considerados ruido, es un cluster muy pequeño que fue construido por la proximidad de sus elementos, pero se pueden ver una particularidad en los elementos que lo componen, este cluster está muy cerca al cluster de “mallados” y los macro patrones que se ven dentro del cluster se pueden considerar uniformes, justificando su posición en el espacio y pudiendo llegar a ser parte de este cluster, también puede darse el caso de que este cluster pueda tener mayores coincidencias en un conjunto de datos más grande, dando paso a otro macro patrón, por ahora este cluster es denominado “ruido mallado”.



Figura 8.8: Ejemplos clusters “ruido mallado”

De las figuras [7.2] y [7.3] es posible destacar que la distribución de elementos en el conjunto de datos estudiado presenta la mayor densidad de elementos en los clusters que representan anillos y mallas, dando a entender que la mayor parte de los macro patrones encontrados corresponden con estas clases.

Dentro de los macro patrones que son considerados como ruido, destacan dos elementos, sus vectores de características tienen la mayor distancia a cualquier otro elemento dentro del conjunto de datos.

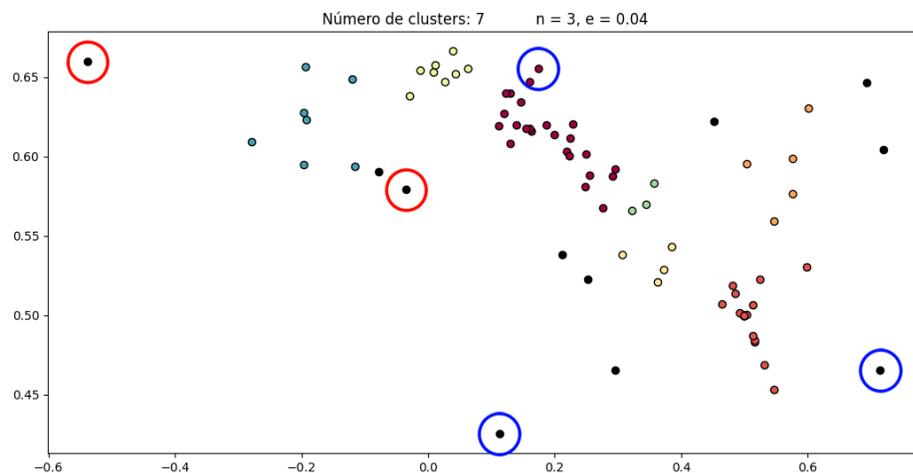


Figura 8.9: Macro patrones con menor coincidencia

Los objetos que corresponden a estos macro patrones con baja coincidencia son los siguientes



Figura 8.10: Objetos con menor coincidencia

Ambos objetos poseen una estructura única dentro del conjunto de datos y así demuestra también su respectivo heatmap.

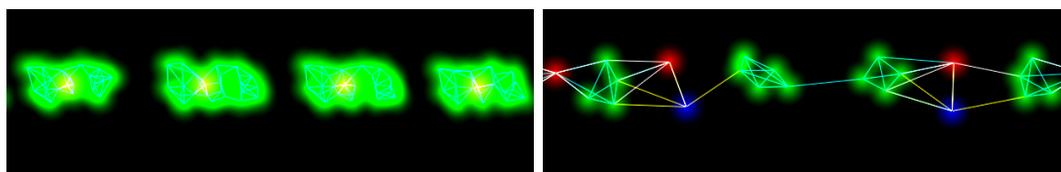


Figura 8.11: Heatmap de objetos con menor coincidencia

8.2. Comparación

De los resultados del experimento de comparación [7.2], se puede ver que buscar similitud usando la distancia euclidiana sobre los vectores de características [6.1] de los macro patrones, entrega patrones que corresponden la mayoría del tiempo con la clase de los macro patrones del objeto en cuestión, de hecho, el resultado que se puede ver en [7.11] es muy estable debido a la concentración de macro patrones pertenecientes a la clase de anillos y a la construcción de sus vectores de características.

A pesar de que estos resultados fueron buenos, si se busca encontrar un objeto que presente macro patrones de la clase correspondiente a los del objeto de prueba, el resultado será mucho mejor si se revisa directamente la vecindad de la distribución en dos dimensiones.

Como experiencia adicional, usando el criterio de comparación utilizado en el experimento de comparación [6.1], se prueba también sobre los objetos que tienen menor coincidencia respecto a la clasificación construida mediante DBSCAN. Los resultados obtenidos están a continuación.

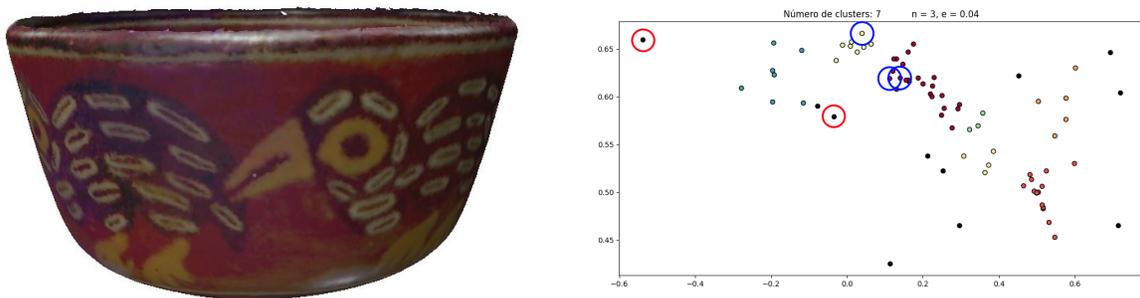


Figura 8.12: Objeto y sus coincidencias destacadas en azul

Las coincidencias obtenidas para el objeto de control [8.12] corresponden con los siguientes objetos:



Figura 8.13: Objetos correspondientes a las coincidencias encontradas

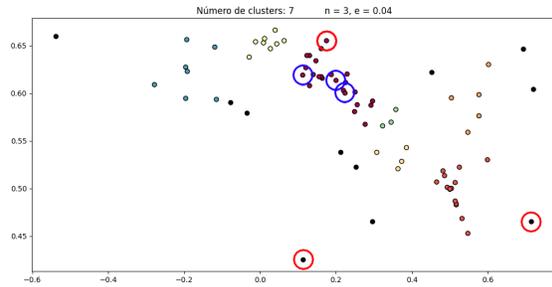


Figura 8.14: Objeto y sus coincidencias destacadas en azul

Las coincidencias obtenidas para el objeto de control [8.14] corresponden con los siguientes objetos.



Figura 8.15: Objetos correspondientes a las coincidencias encontradas

Para ambos objetos, los resultados al comparar con el criterio del vector de características fueron patrones pertenecientes al cluster de mallados, lo que da a pensar que este cluster comprende macro patrones que tienen una distribución no uniforme de ocurrencias, a pesar de que este cluster se identifica por considerar elementos que se distribuyen de forma uniforme. El razonamiento detrás de esta afirmación viene de la forma en el que el vector de características está construido, ya que al calcular un promedio aritmético, se puede tener un vector de características que indique que sus elementos distribuyen de forma uniforme, a pesar de que sus elementos locales no lo hicieran. Otra conclusión válida de este comportamiento es que el cluster de mallados, al presentar una distribución uniforme, los vectores de características que forman parte de él no tienen componentes nulas y, por tanto, siempre serán lo más cercano a un elemento que se considere ruido.

8.3. Trabajo futuro

Dentro de esta investigación se realizaron las experiencias reduciendo la dimensión de los vectores calculados para facilitar la comprensión del comportamiento de los resultados obtenidos, el algoritmo de clusterización utilizado (DBSCAN) puede trabajar en mayores dimensiones, pero es necesario definir una función de distancia significativa para la dimensión sobre la cual trabajar. Esto significa que puede considerarse realizar un análisis similar con un vector de características diferente y en una dimensión que considere las componentes de ese nuevo vector de mejor forma, llegando a nuevas clasificaciones.

También, en este trabajo se analizaron las relaciones de los patrones considerando patrones de diferentes arquetipos, realizar un trabajo similar al presente puede generar resultados interesantes si se consideran o no las relaciones entre patrones que son estrictamente del mismo arquetipo.

Con base en los resultados de esta experiencia, las categorías definidas en esta investigación pueden ser utilizadas como información de entrenamiento para algoritmos de aprendizaje destinados a clasificar elementos de naturaleza similar a los objetos trabajados en ella o a partir de nuevos elementos, generar información que constituya un nuevo sistema de clasificación para destinar de la misma forma.

Mirando el objeto [8.10] se puede ver una característica interesante que sale del alcance de esta investigación, la posibilidad de que macro patrones presentes en objetos estén constituidos de macro patrones que no necesariamente pertenezcan a la misma clase. Dicho esto se puede enfocar un nuevo sistema de clasificación que considere macro patrones como una posible parte de un macro patrón mayor.

Capítulo 9

Conclusión

A partir de lo evidenciado en los experimentos realizados en esta experiencia, se puede concluir que un proceso de clasificación automática de macro patrones en objetos como los del conjunto de datos original es viable, siempre y cuando se cuente con información suficiente para determinar las clases a las que puede pertenecer un macro patrón.

Los resultados de la experiencia se consideran satisfactorios, los objetos del conjunto de datos han podido clasificarse en clases segregadas dependiendo de las estructuras formadas por los patrones geométricos de su superficie, a pesar de que es imposible saber de antemano las clases de macro patrón que pueden existir y se pudo obtener una clara relación entre los macro patrones pertenecientes a una misma clase.

Los objetivos propuestos se consideran como cumplidos, se determina la viabilidad de un algoritmo que pueda clasificar macro patrones de forma automática, además de presentar una posible solución al problema abordado, se refiere como posible solución a la propuesta de este trabajo, pues los resultados tienen margen de mejora, pues esta experiencia aborda un problema nuevo y muchos de los parámetros utilizados han sido establecidos de forma exploratoria, manteniendo aquellas características que demostraban obtener los mejores resultados a lo largo del camino.

El proceso de concepción, desarrollo e implementación llevaron a un algoritmo que cumple su cometido y de forma eficiente, de esta forma, se determina que el proceso tanto teórico como técnico fueron un éxito y un aporte innegable para el resultado obtenido dentro de este trabajo.

Bibliografía

- [1] Donald W. Crowe. The geometry of african art ii, a catalog of benin pattern. *Historia Mathematica*, 2, 1975.
- [2] Donald W. Crowe. The mosaic patterns of h. j. woods. *Computers and Mathematics with Applications*, 12B, 1986.
- [3] Donald W. Crowe and Dorothy K. Washburn. Groups and geometry in the ceramic art of san ildelfonso. *Algebras, Groups and Geometries*, 3, 1985.
- [4] Ibai Diez and Jorge Sepulcre. Neurogenetic profiles delineate large-scale connectivity dynamics of the human brain. *Nature Communications*, 9, 2018.
- [5] Stefan Lengauer, Reinhold Preiner, Ivan Sipiran, Stephan Karl, Elisabeth Tinkl, Benjamin Bustos, and Tobias Schreck. Context-based surface pattern completion of ancient pottery. 2022.
- [6] Stefan Lengauer, Ivan Sipiran, Reinhold Preiner, Tobias Schreck, and Benjamin Bustos. A benchmark dataset for repetitive pattern recognition on textured 3d surfaces. *Eurographics Symposium on Geometry Processing 2021*, 40(5), 2021.
- [7] Xiaodan Liang, Xiaohui Shen, Jiashi Feng, Liang Lin, and Shuicheng Yan. Semantic object parsing with graph lstm. 2016.
- [8] Kirill Veselkov, Guadalupe Gonzalez, Shahad Aljifri, Dieter Galea, Reza Mirnezami, Jozef Youssef, Michael Bronstein, and Ivan Laponogov. Hyperfoods: Machine intelligent mapping of cancer-beating molecules in foods. *Scientific Reports*, 9, 2019.
- [9] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. 2018.
- [10] Dorothy K. Washburn. Pattern symetry and colored repetition in cultural contexts. *Computers and Mathematics with Applications*, 12B, 1986.
- [11] Tian Xie, Arthur France-Lanord, Yanming Wang, Yang Shao-Horn, and Jeffrey C. Grossman. Graph dynamical networks for unsupervised learning of atomic scale dynamics in materials. *Nature Communications*, 10, 2019.
- [12] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. 2018.