CENTRO DE CIENCIAS BÁSICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

TESIS

APROXIMACIÓN CON POLIEDROS PARA OBJETOS 3D POR MEDIO DE DETECCIÓN DE PUNTOS DOMINANTES

PARA OPTAR POR EL GRADO DE MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

Miguel Vázquez Martín del Campo

TUTOR

Dr. Hermilo Sánchez Cruz

Dr. Mario Alberto Rodríguez Díaz (Co-tutor)

ASESOR

Dr. Rogelio Salinas Gutiérrez

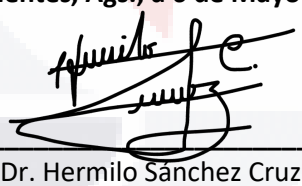
Aguascalientes, Ags., a 28 de Mayo de 2021

M. en C. Jorge Martín Alférez Chávez
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como TUTOR designado del estudiante **MIGUEL VAZQUEZ MARTIN DEL CAMPO** con ID 266176 quien realizó la tesis titulada: **APROXIMACIÓN CON POLIEDROS PARA OBJETOS 3D POR MEDIO DE DETECCIÓN DE PUNTOS DOMINANTES**, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirlo así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 6 de Mayo de 2021.



Dr. Hermilo Sánchez Cruz
Tutor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría Técnica del Programa de Posgrado

M. en C. Jorge Martín Alférez Chávez
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como CO-TUTOR designado del estudiante **MIGUEL VAZQUEZ MARTIN DEL CAMPO** con ID 266176 quien realizó la tesis titulada: **APROXIMACIÓN CON POLIEDROS PARA OBJETOS 3D POR MEDIO DE DETECCIÓN DE PUNTOS DOMINANTES**, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirlo así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 4 de Mayo de 2021

mard81

Dr. Mario Alberto Rodríguez Díaz
Co-tutor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría Técnica del Programa de Posgrado

M. en C. Jorge Martín Alférez Chávez
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como ASESOR designado del estudiante **MIGUEL VAZQUEZ MARTIN DEL CAMPO** con ID 266176 quien realizó la tesis titulada: **APROXIMACIÓN CON POLIEDROS PARA OBJETOS 3D POR MEDIO DE DETECCIÓN DE PUNTOS DOMINANTES**, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirlo así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 4 de Mayo de 2021.

_____
Dr. Rogelio Salinas Gutiérrez
Asesor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría Técnica del Programa de Posgrado

DICTAMEN DE LIBERACIÓN ACADÉMICA PARA INICIAR LOS TRÁMITES DEL EXAMEN DE GRADO

Fecha de dictaminación dd/mm/aaaa: 24/05/2021

NOMBRE: MIGUEL VAZQUEZ MARTIN DEL CAMPO **ID** 266176

PROGRAMA: MAESTRÍA EN CIENCIAS CON OPCIÓN A LAS COMPUTACIÓN, MATEMÁTICAS APLICADAS **LGAC (del posgrado):** Inteligencia Artificial

TIPO DE TRABAJO: (X) Tesis ( ) Trabajo Práctico

TITULO: APROXIMACIÓN CON POLIEDROS PARA OBJETOS 3D POR MEDIO DE DETECCIÓN DE PUNTOS DOMINANTES

IMPACTO SOCIAL (señalar el impacto logrado): Implementar un método computacional para representar objetos 3D mediante poliedros y proponer criterios de error que demuestren su eficacia.

INDICAR SI NO N.A. (NO APLICA) SEGÚN CORRESPONDA:	
Elementos para la revisión académica del trabajo de tesis o trabajo práctico:	
Si	El trabajo es congruente con las LGAC del programa de posgrado
No	La problemática fue abordada desde un enfoque multidisciplinario
Si	Existe coherencia, continuidad y orden lógico del tema central con cada apartado
Si	Los resultados del trabajo dan respuesta a las preguntas de investigación o a la problemática que aborda
Si	Los resultados presentados en el trabajo son de gran relevancia científica, tecnologíca o profesional según el área
Si	El trabajo demuestra más de una aportación original al conocimiento de su área
Si	Las aportaciones responden a los problemas prioritarios del país
N.A.	Generó transferecia del conocimiento o tecnológica
Si	Cumple con la ética para la investigación (reporte de la herramienta antiplagio)
El egresado cumple con lo siguiente:	
Si	Cumple con lo señalado por el Reglamento General de Docencia
Si	Cumple con los requisitos señalados en el plan de estudios (créditos curriculares, optativos, actividades complementarias,estancia, predoctoral, etc)
Si	Cuenta con los votos aprobatorios del comité tutoral, en caso de los posgrados profesionales si tiene solo tutorpodrá liberar solo el tutor
N.A.	Cuenta con la carta de satisfacción del Usuario
Si	Coincide con el título y objetivo registrado
Si	Tiene congruencia con cuerpos académicos
Si	Tiene el CVU del Conacyt actualizado
N.A.	Tiene el artículo aceptado o publicado y cumple con los requisitos institucionales (en caso que proceda)
En caso de Tesis por artículos científicos publicados	
N.A.	Aceptación o Publicación de los articulos según el nivel del programa
N.A.	El estudiante es el primer autor
N.A.	El autor de correspondencia es el Tutor del Núcleo Académico Básico
N.A.	En los artículos se ven reflejados los objetivos de la tesis, ya que son producto de este trabajo de investigación.
N.A.	Los artículos integran los capítulos de la tesis y se presentan en el idioma en que fueron publicados
N.A.	La aceptación o publicación de los artículos en revistas indexadas de alto impacto

Con base a estos criterios, se autoriza se continúen con los trámites de titulación y programación del examen de grado:

Sí X

No

FIRMAS

Elaboró:

* NOMBRE Y FIRMA DEL CONSEJERO SEGÚN LA LGAC DE ADSCRIPCION: Dr. Hermilo Sánchez Cruz

NOMBRE Y FIRMA DEL SECRETARIO TÉCNICO: Dr. Hermilo Sánchez Cruz

*** En caso de conflicto de intereses, firmará un revisor miembro del NAB de la LGAC correspondiente distinto al tutor o miembro del comité tutoral, asignado por el Decano**

Revisó:

NOMBRE Y FIRMA DEL SECRETARIO DE INVESTIGACIÓN Y POSGRADO: Dra. Haydee Martínez Rubalcaba

Autorizó:

NOMBRE Y FIRMA DEL DECANO: M. en C. Jorge Martín Alférez Chávez

Nota: procede el trámite para el Depto. de Apoyo al Posgrado

En cumplimiento con el Art. 105C del Reglamento General de Docencia que a la letra señala entre las funciones del Consejo Académico: .... Cuidar la eficiencia terminal del programa de posgrado y el Art. 105F las funciones del Secretario Técnico, llevar el seguimiento de los alumnos.

Contenido

Lista de Figuras

Lista de Tablas

Resumen

Esta investigación presenta un nuevo método para la representación de objetos binarios de tres dimensiones por medio de una aproximación poliedral. El método se compone de varios pasos, y comienza con un objeto 3D que es discretizado para poder manejarlo como un conjunto de capas de voxeles, en donde algunas de ellas son seleccionadas y tratadas como objetos bidimensionales, de las capas seleccionadas se extrae el contorno de la figura y su código de cadena. Cada código de cadena se evalúa con un método para encontrar los puntos dominantes por medio de una gramática libre de contexto, y así encontrar una nube de puntos en tres dimensiones de todas las capas seleccionadas. Los puntos de esta nube son unidos estratégicamente para dar forma al poliedro aproximado que representa al objeto 3D inicial. Finalmente, se adaptan unos criterios de error que permiten evaluar el poliedro aproximado obtenido respecto al objeto original para conocer, qué tan parecidos son entre ellos y cuanta información se redujo.

El método se aplicó a un conjunto de objetos, en donde se extrajeron algunas características del poliedro aproximado y algunas otras del objeto discretizado, y con base en la combinación de las mismas se agruparon los objetos empleando algunas métricas y técnicas de aprendizaje no supervisado.

Abstract

This research presents a new method for the representation of three-dimensional binary objects by polyhedral approximation. The method consists of several steps, it begins with a 3D object that is discretized to be able to handle it as a set of voxel slices, where some of them are selected and treated as two-dimensional objects. From the selected slices, the proposed method gets the chain code from the contour, and uses the existing context-free grammar method to find the dominant points from each slice, obtaining a point cloud from all the selected slices. The points of this cloud are strategically joined to create the approximate polyhedron that represents the initial 3D object. Finally, some error criteria are adapted to evaluate the approximate polyhedron against the original object, in order to know how similar they are to each other and how much information was reduced.

The method was applied to a set of objects, we got some characteristics from the approximate polyhedron and some others from the discretized object, and based on their combination, the objects were clustered using some metrics and unsupervised learning techniques.

1. Introducción

La vista es quizá el sentido para importante para la mayoría de los seres vivos del reino animal, debido a que este es el canal por el que se percibe la mayor cantidad de información del mundo exterior.

El proceso de la vista comienza cuando un rayo de luz llega a un cuerpo del mundo exterior, este absorbe parte de ese rayo de luz y otra parte la refleja en modo de onda, estas ondas son las que percibe el ojo humano, estas ondas rebotadas son las que permiten distinguir colores, textura y formas, es decir, que si el objeto absorbe por completo el rayo de luz, este sería invisible para el ojo humano. Cuando las ondas llegan al ojo humano, este las convierte en señales que se envían al cerebro.

La visión, que erróneamente se suele confundir con la vista, va más allá del proceso ya descrito, ya que aunque hace uso de la vista, es un proceso más dinámico en el que se incluyen procesos de pensamiento, comprensión de lo que se ve y como se reacciona ante esto.

La inteligencia artificial, a grandes rasgos, es poder imitar en un objeto no vivo la inteligencia humana, pero que sería de esta inteligencia sin el proceso que contiene el canal por el que el humano recibe la mayor cantidad de información del exterior, es ahí donde nace la visión artificial, que no es más que poder captar objetos del mundo real y procesarlos con el fin de poder comprender y reproducir información.

El incremento en la demanda en aplicaciones que utilizan el tratamiento de objetos en el dominio de 2 y 3 dimensiones, ha generado un reto para la visión por computadora, y se logra al optimizar el procesamiento de estos objetos, para eso se hace uso de los descriptores de las imágenes, los cuales representan todo el objeto con la menor información redundante posible.

Los descriptores de las imágenes pueden variar, van desde el interior de las figuras, *e.g* el esqueleto [1][2], o la superficie o contorno [3][4]. Para ambos casos se puede recuperar parcial o totalmente, según corresponda, el objeto original. La representación por el contorno o superficie basada en los códigos de cadena [5][6][7][8], describen un objeto por una secuencia de símbolos que significan un segmento de recta de tamaño único con una orientación. Existen diversos códigos de cadena para objeto de 2 y 3 dimensiones, relativos y absolutos a la orientación del objeto.

La aproximación poligonal es otra representación de los objetos de 2 dimensiones [9][10][11][12] y es poliedral para objetos de 3D, en el cual, a partir de un conjunto puntos del contorno, llamados puntos dominantes que son seleccionados estratégicamente, se

unen por líneas rectas formando así un polígono, que visualmente representa el objeto original. Debido a que el contorno está formado únicamente por líneas rectas, incluso en donde el contorno presentaba más detalles, este método es tolerable a un cierto error, ya que el polígono formado y el objeto original, aunque visualmente son muy parecidos, no son estrictamente idénticos.

A lo largo del contorno, se pueden observar esquinas, rectas y curvas, pero son las curvas y esquinas quienes proporcionan mayor información del objeto original, es por eso, que en esas zonas se concentran la mayor cantidad de puntos. El desarrollo de un método que permita encontrar ese conjunto de puntos dominantes no es una tarea fácil, es por eso que anteriormente se han presentado varios trabajos sobre eso. Attneave realizó una de las primeras propuestas [13], la cual estaba basada en la teoría de que cualquier esquina sería un punto dominante. Estos algoritmos para encontrar los puntos dominantes pueden ser clasificados por su enfoque, los cuales tienen 3 grandes grupos, secuencial, dividir y combinar, y heurístico.

En el enfoque secuencial, Massod [14] propone un conjunto de puntos de ruptura iniciales, en que el existe un cambio de dirección en el contorno, para después eliminar en cada iteración un punto y evaluar el error, así hasta que se alcanza el error deseado. Ray y Ray [15], determinan los segmentos de línea más largos posibles con el error mínimo posible. Kurozumi y Davis [16], propusieron un método que deriva los segmentos aproximados minimizando la distancia máxima entre un conjunto dado de puntos y el segmento correspondiente. Teh y Chin [17], determinaron la región de soporte para cada punto en función de sus propiedades locales y calcularon su importancia relativa (curvatura) y finalmente, detectaron puntos dominantes mediante un proceso de supresión no máxima.

En el enfoque de dividir y combinar se encuentra, Ramer [18], quien presentó un método recursivo que comienza con una segmentación de límites iniciales, y en cada iteración el segmento se divide en el punto que tiene la distancia más lejana del segmento correspondiente a menos que el error de aproximación exceda el error tolerable especificado previamenete. Held et al. [19], propusieron una técnica de división y combinación en la que se utilizó la diferencia de pendiente para dividir segmentos y estos se combinaron según los criterios de significación perceptiva.

Para el enfoque de búsqueda heurística, se han empleado algoritmos dinámicos [20][21], algoritmos genéticos [22][23], búsqueda tabú y colonia de hormigas [24]; y de los más recientes Fernández García [25], propone un nuevo método automático y no paramétrico para aproximaciones poligonales que se basa en una nueva versión simétrica del conocido método Ramer y aplica un nuevo método con umbral adaptativo para obtener los puntos dominantes.

A pesar que existen métodos de representación poliedral para objetos 3D [26], no existe algún método que genere el poliedro a partir de una nube de puntos dominantes hasta el momento de realizar esta investigación.

El procesamiento de objetos 3D es una tarea más complicada que en el caso de objetos 2D, sin embargo, tomando la ventaja de que un objeto 3D puede representarse como un conjunto de capas de un grosor mínimo, se propone el uso de un método con un gramática libre de contexto [27], para obtener una nube de puntos, los cuales son unidos para crear un poliedro.

1.1. Estructura de la tesis

El presente documento está formado por los siguientes capítulos:

- Capítulo 2. Formulación de la investigación. En este apartado se determina el objetivo de la investigación a partir de las hipótesis y preguntas de investigación planteadas.
- Capítulo 3. Preliminares. En este capítulo se explican los temas fundamentales del marco teórico utilizados a lo largo de la investigación.
- Capítulo 4. Método de Aproximación Poliedral a partir de una Nube de Puntos. En este capítulo se presenta un método para la creación de un poliedro a partir de una nube de puntos obtenida por capas, así como las adaptaciones y nuevas propuestas de criterios de error para evaluar el método.
- Capítulo 5. Resultados Experimentales. En este apartado se aplica el método propuesto a un conjunto de objetos y se presentan sus resultados.
- Capítulo 6. Agrupamiento y Reconocimiento de Objetos 3D a partir de sus Características y Poliedro Aproximado. En este capítulo se extraen diversas características de un conjunto de objetos y se aplican diversas métricas y técnicas de aprendizaje para determinar el parecido entre objetos a partir de un poliedro obtenido.
- Capítulo 7. Conclusiones. En esta sección se explican los resultados que se obtuvieron de la investigación y se plantea el trabajo para futuras investigaciones.

2. Formulación de la Investigación

2.1. Objetivo

Desarrollar un método para obtener un poliedro aproximado que represente debidamente a un objeto binario y proponer criterios de error que demuestren la eficacia del método.

2.2. Objetivos Específicos

- Obtener un conjunto puntos dominantes calculados por capas para objetos 3D.
- Reconstruir un poliedro aproximado que represente debidamente al objeto.
- Adaptar los criterios de error para evaluar el método propuesto.
- Establecer una técnica que permitan distinguir objetos a partir de una nube puntos.

2.3. Hipótesis

- Obteniendo puntos dominantes calculados por capas, se puede reconstruir un poliedro aproximado que represente debidamente al objeto.
- Es posible extraer características de un poliedro que permita distinguir un objeto de otro.
- Se puede obtener un poliedro con el cual se puede realizar un reconocimiento eficaz.
- Es posible evaluar el poliedro respecto al objeto original.

2.4. Preguntas de investigación

- ¿Es posible reconstruir un objeto a partir de una nube de puntos por capas?
- ¿Existe un método para seleccionar las capas de un objeto de 3D que contienen más información?
- ¿Se puede hacer un proceso de reconocimiento a partir de un poliedro?
- ¿Qué características se pueden extraer de un poliedro que permitan distinguirlo de los demás objetos?
- ¿Existen valores de error entre el objeto original y el poliedro formado?

3. Preliminares

En el presente capítulo se describen los principales conceptos básicos utilizados de la visión artificial, principalmente para objetos 3D; y algunas de las definiciones de aprendizaje automático.

3.1. Objeto 3D

Un objeto 3D, es un objeto situado en un espacio tri-dimensional, es decir, tres dimensiones, que son ancho, altura y profundidad. Existe una diversidad de formatos para el manejo de objetos 3D en la computación, estos son algunos de los más comunes.

- PLY. El formato PLY fue creado en los años 90 en la universidad de Stanford y lo llamaron *polygon file format*, es por eso que es conocido como el formato del triángulo de Stanford, ya que contiene un listado de polígonos e información sobre estos, tal como el color, transparencia y textura. Es utilizado para almacenar la información recopilada de escáneres 3D [28].
- OBJ. Es de los formatos más comunes para objetos 3D y puede ser compatible con la mayoría de software para objetos 3D, fue desarrollado por Wavefront Technologies, contiene las coordenadas de los vértices, mapas de textura y caras poligonales [29].
- STP. Es un formato utilizado para la transferencia de datos 3D entre programas CAD y CAM con el estándar para el intercambio de datos de producto (STEP, por sus siglas en inglés) [30].
- WRL. El formato lenguaje de modelado de realidad virtual (WRL, por sus siglas en inglés), está basado en texto para la descripción de objetos 3D, es el formato estándar para el manejo de gráficos interactivos en 3D en la web [31].

3.2. Objeto Voxelizado

Un objeto voxelizado es un objeto 3D que se encuentra en un mallado 3D, con coordenadas cortesianas $c(x,y,z)$, creando un conjunto de voxeles que forman el objeto, por lo que un voxel (volumen pixel element), es la unidad cúbica mínima, de este tipo de objetos, un voxel puede tomar dos intensidades, 0 y 1, por lo que se dice que es 1-voxel si la intensidad es uno, y 0-voxel si es cero [32].

Un objeto voxelidado se obtiene a partir de un proceso de voxelización, el cual consiste en convertir un objeto 3D de los formatos descritos en la sección anterior, lo cuales tienen formas continuas, en un objeto formado por voxeles, existen algunos softwares que permiten hacer este proceso, en el que se admiten los formatos 3D de la sección anterior y generar un archivo con el objeto ya voxelizado, *e.g*. Binvox.

3.3. Distancia Euclidiana

La distancia Euclidiana entre dos punto $p, q \in \mathbb{R}^N$ con coordenadas $p = (p_1, p_2, \dots, p_N)$ y $q = (q_1, q_2, \dots, q_N)$, está dada por [33]:

$$d(p,q) = \sqrt{\sum_{n=1}^{N} (p_n - q_n)^2}$$

3.4. Distancia de Hausdorff

La distancia de Hausdorff H entre dos conjuntos A y B, es la máxima distancia dirigida de Hausdorff $\check{H}$, dada por:

$$H(A,B) = \max\{\check{H}(A,B), \check{H}(B,A)\},$$

la distancia dirigida de Hausdorff $\check{H}$ es la distancia máxima entra cada punto $p \in A$ y su vecino más cercano de $q \in B$, esto es:

$$\check{H}(A,B) = max_{p \in A}\left\{min_{q \in B}\{\|p,q\|\}\right\},$$

donde $\|p,q\|$ es la distancia Euclidiana entre los puntos p y q. Cabe mencionar que $\check{H}(A,B) \neq \check{H}(B,A)$, por lo tanto las distancia dirigida de Hausdorff no es una métrica pero la distancia de Hausdorff sí lo es [34].

3.5. Vecindades

Las vecindades se pueden encontrar tanto en 2D y 3D, es decir, entre pixeles y voxeles, las opciones que tenemos para 2D son:

- Vecindad 4. Dos pixeles pueden estar en vecindad 4 cuando ambos pixeles comparten una arista, es decir, únicamente puede cambiar en una unidad solo una coordenada.
- Vecindad 8. Dos pixeles pueden estar en vecindad 8 cuando ambos pixeles comparten un artista o vértice, para esto, puede cambiar una o las dos coordenadas en una unidad.

En los tipos de vecindades en 3D que existen, son:

- Vecindad 6. Dos voxeles pueden estar en vecindad 6 cuando ambos voxeles comparten una cara de voxel, para esto, únicamente puede cambiar una coordenada en una unidad entre ambos voxeles.
- Vecindad 18. Dos voxeles pueden estar en vecindad 18 cuando ambos voxeles comparten una cara o arista de voxel, para esto, únicamente puede cambiar una o dos coordenada en una unidad entre ambos voxeles.
- Vecindad 26. Dos voxeles pueden estar en vecindad 18 cuando ambos voxeles comparten una cara, arista o vertice del voxel, para esto, puede cambiar una, dos o tres coordenada en una unidad entre ambos voxeles.

3.6. Camino de Voxeles

Dados dos voxeles A y B, ambos del mismo objeto 3D, se dice que existe un camino entre ambos puntos si es posible llegar del punto A al punto B por medio de alguna secuencia de voxeles de las vecindades 3D que se describen en el punto anterior. Cabe mencionar que no importa cantidad de saltos que se den entre voxeles, lo único que importa es llegar al punto B desde A.

3.7. Morfología Matemática

La morfología matemática es una técnica con base en la teoría de conjuntos utilizada en el tratamiento figuras digitales, permite la modificación de una figura por medio de distintos operadores. Estos operadores requieren de un elemento estructurante, el cual es una vecindad de valor binario multidimensional, que se compone de un punto de origen, que puede ser un voxel o pixel, dependiendo de la dimensión de la figura, y va

a identificar el punto del objeto que se está procesando, y los demás puntos del elemento estructurante permiten la evaluación del punto de origen.

Los dos principales operadores son:

- *Erosión*, una imagen A y un elemento estructurante B, ambos binarios, la erosión de A por B consiste en el conjunto de todos los elementos x para los cuales B trasladado por x está contenido en A:

$$A \ominus B = \{x | B_x \subseteq A\}$$

- *Dilatación,* una imagen A y un elemento estructurante B, ambos binarios, la dilatación de A por B se define como:

$$A \oplus B = \{x | (B^c)_x \cap A \neq \emptyset\}$$

3.8. Aprendizaje Automático

El aprendizaje automático o mejor conocido como *machine learning,* es un área de la inteligencia artificial en que la que se tiene el objetivo de permitir que las máquinas aprendan por si mismas para fomentar su autonomía en la realización de una cierta tarea, proporcionando una cierta capacidad cognitiva a la maquina por medio de un entrenamiento y adaptación de ciertos algoritmos respecto a una entrada de datos a el sistema.

El machine learning consiste en la capacidad del sistema para identificar una serie de patrones determinados en volúmenes de datos en una gran cantidad de parámetros, es por eso que se dice que el machine learning es un maestro del reconocimiento de patrones.

El aprendizaje que logran las máquinas va a depender de los datos que se ingresen al sistema, es ahí donde comienza el aprendizaje, con un algoritmo en su programación capaz de adaptarse a los datos de entrada, y así, poder predecir un posible escenario y generar una respuesta. Se dice que las maquinas aprenden por sí misma, pero esto no significa que las máquinas hacen todo el trabajo, sino que aprenden a resolver una tarea sin ser explícitamente programas para ese tarea, es decir, fueron aprendiendo como resolverlo por medio de los ejemplos o datos que se le proporcionaron, aprenden de la experiencia con base en análisis de datos [35].

Existen diferentes algoritmos que permiten este tipo de aprendizaje y estos son agrupados, en función de cómo operan y la salida de los mismos, en dos grandes grupos:

- Aprendizaje Supervisado. En este tipo de algoritmos, los datos de entrada a la máquina para el entrenamiento se encuentras etiquetados, es por eso que se le llama supervisado, ya que ese etiquetado inicial corresponde, en la mayoría de los casos, a una persona. El algoritmo que utiliza la máquina tiene la capacidad de asignar alguna de esas etiquetas que se ingresaron en los primeros datos a otro conjunto de datos. Los algoritmos más comunes de este tipo de aprendizaje son:

    - redes neuronales artificiales,
    - máquinas de soporte vectorial, y
    - árboles de decisión.

- Aprendizaje No Supervisado. En los algoritmos de este tipo, a diferencia del aprendizaje supervisado, no cuenta con la información del etiquetado sobre las categorías, por lo que solo adaptan el modelo predictivo tomando en cuenta los datos de entrada, buscando similitudes entre ellos. Estos algoritmos no tienen como objetivo detectar un tipo de dato en específico, sino que a partir de los datos de entrada, tratan de buscar elementos que se parezcan y así poder agrupar los que tiene mayor similitud. Los algoritmos más comunes de este tipo son:

    - k-means, y
    - clustering jerárquico.

4. Método de Aproximación Poliedral a partir de una Nube de Puntos

El método propuesto consta de varios pasos, comenzando con la selección de corte de un objeto voxelizado, continuando con el proceso de cada corte para encontrar su conjunto de puntos dominantes para obtener una nube de puntos de todo el objeto. Finalmente, unimos los puntos para crear el poliedro y calcular el error cometido.

4.1. Selección de Capas y Componentes Conectados

El método propuesto comienza con un objeto 3D, puede ser en diversos formatos, tales como: .obj o .ply, en la Fig 4.1. se puede observar el objeto Brain en formato .ply.

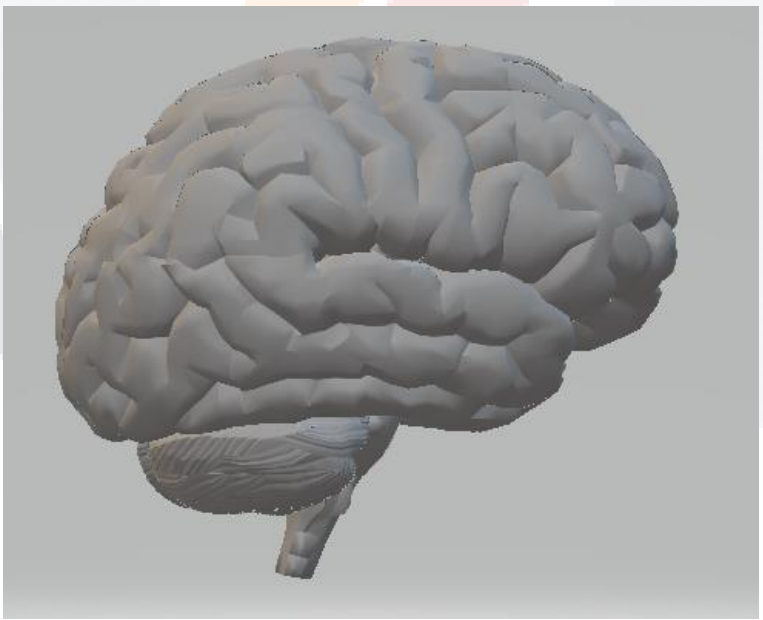


Fig. 4.1 Objeto 3D

El objeto 3D es discretizado con la ayuda del software Binvox, en el cual se le configura el tamaño del objeto de salida, dando como resultado un objeto voxelizado. En

la Fig. 4.2 se puede ver el Brain voxelizado a $128 \times 128 \times 128$, cabe mencionar que el objeto puede ser voxelizado a cualquier tamaño, pero se utilizó este tamaño porque es lo suficientemente grande para mantener los detalles del objeto original y lo suficientemente pequeño para que el procesamiento sea rápido. Para la voxelización, el objeto debe tener una orientación en donde lo más largo del cuerpo del objeto sea paralelo al eje Z, así como sus extremidades y ramas, principalmente aquellas que tienen un grosor muy pequeño, en medida de lo posible. Para ello se emplea la función de rotación de Binvox.

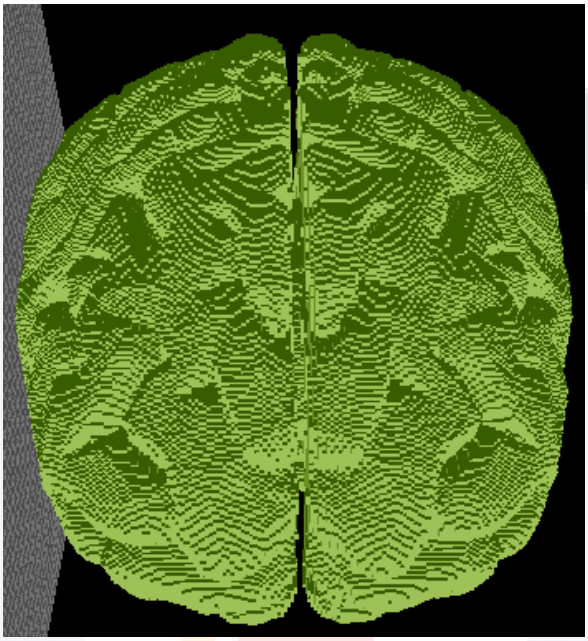


Fig. 4.2 Brain voxelizado

El objeto voxelizado es tratado como una caja cerrada compuesta por 1-voxeles (parte de la figura) y 0-voxeles (fondo), organizada por capas en el eje Z, en donde es recorrido desde el fondo hacia arriba en búsqueda del primer 1-voxel. Cuando ese 1-voxel es localizado, todos los voxeles en esa capa, es decir, voxeles con ese mismo valor en Z, forman parte de la primera capa. Las capas siguientes con seleccionadas recorriendo hacia arriba el objeto en intervalos de N, obteniendo una capa cada N capas recorridas. En el valor de N se recomienda que sea el número entero más cercano a la raíz cuadrada de la cantidad de voxeles por el eje recorrido, es decir, el eje Z, y que cumpla con el residuo de (1) sea igual a cero, esto para que la distancia entre capas sea igual entre ellas,

$$\frac{(Cantidad\ de\ voxeles\ en\ Z)}{N}. \qquad (1)$$

En este caso, la cantidad de capas en el eje Z es 128, así que $\sqrt{128}$ = 11.31, los valores enteros más cercanos son 11, 12, 10, 13, 9, 14, 8, 15, 7, y 16, en ese orden, estos posibles valores son evaluados en (1) como valor de N, pero el primer valor que cumple que el residuo de la razón (1) sea igual a cero es 8, y el segundo es 16, así que $N = 8$ para esta explicación. En Fig. 4.3, la tercera capa del Brain puede observarse.

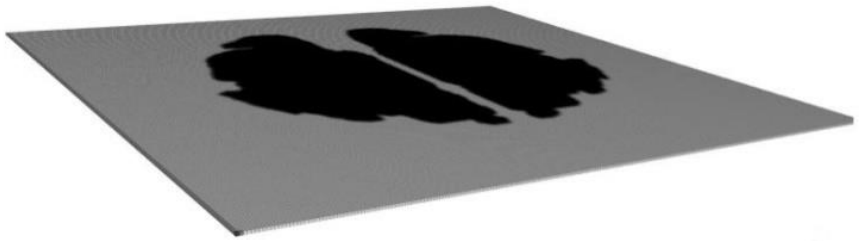


Fig. 4.3 La tercera capa del Brain voxelizado

Una vez que se ha obtenido el conjunto de capas seleccionadas, cada capa es convertida en un objeto de dos dimensiones debido a que tienen un grosor de 1, para esto el objeto es visto en un mallado compuesto de *r* filas y *c* columnas, donde cada celda, ahora llamada pixel, puede tomar dos valores en su intensidad, 0 y 1, 0 si el pixel es parte del fondo o 1 si es parte de la figura. En la Fig. 4.4 se puede observar la tercera capa del Brain.

Fig. 4.4 Tercera capa del Brain en 2D

Ahora que se tienen todas las capas en 2D, es necesario un procesamiento a cada una para identificar los componentes conectados en la imagen, para esto toda la imagen es recorrida en búsqueda de un 1-pixel, cuando es localizado se etiqueta ese pixel como parte del componente conectado y se envía a un proceso recursivo llamado Inundación, en donde se evalúan todos los pixeles vecinos en vecindad 8, si alguno de ellos tiene un 1 en su intensidad, se etiqueta como parte del mismo componente y se envía al mismo proceso para que evalúen sus vecinos, la recursividad termina cuando todos los vecinos son evaluados y todos ellos tiene un 0 en su intensidad. Para finalizar, el resto de la imagen es recorrida en búsqueda de otro 1-pixel sin evaluar, si es localizado, este va a formar parte de otro componente conectado. En Fig. 4.5 se puede observar en rojo al pixel P con valores (x, y), para identificar sus vecinos en vecindad 8, basta con cambiar el valor de sus coordenadas en una unidad por lo menos una de ellas, en color gris se observa los 8 vecinos del pixel P.

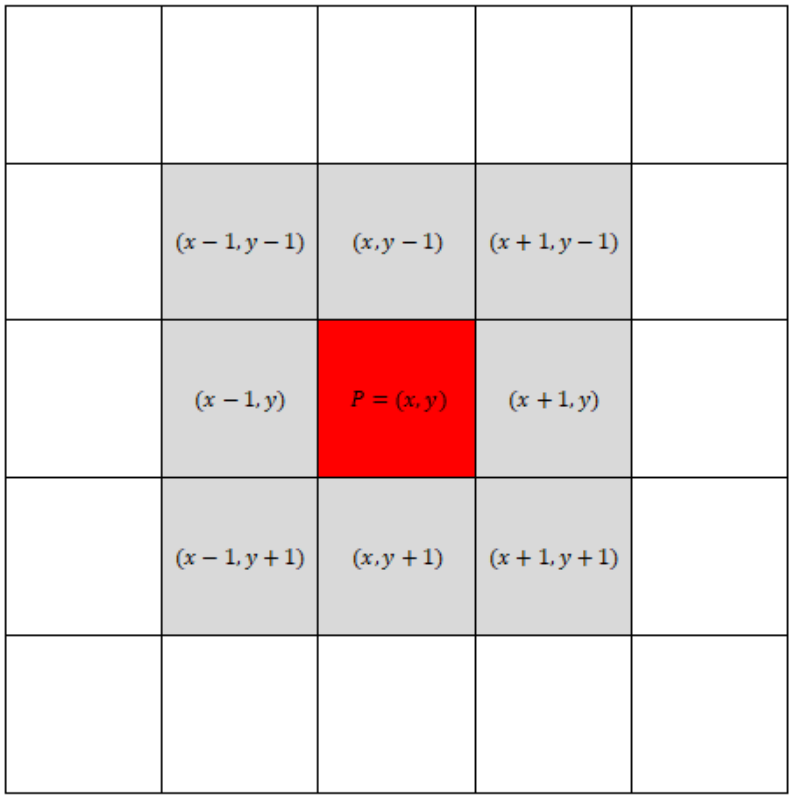


Fig. 4.5 Vecindad 8 de un pixel

A continuación se presenta el algoritmo del método recursivo inundación. Y en Fig. 4.6 se puede observar como la tercera capa contiene dos componentes conectados.

Algoritmo 4.1 Método recursivo para la inundación

```
1.  Inundación (Pixel P(x,y), Componente C)
2.      Etiquetar P como parte del componente C
3.      Si P(x-1,y-1)=1 y P(x-1,y-1) no está etiquetado entonces
4.            Inundación (P(x-1,y-1), C)
5.      Si P(x-1,y)=1 y P(x-1,y) no está etiquetado entonces
6.            Inundación (P(x-1,y), C)
7.      Si P(x-1,y+1)=1 y P(x-1,y+1) no está etiquetado entonces
8.            Inundación (P(x-1,y+1), C)
9.      Si P(x,y-1)=1 y P(x,y-1) no está etiquetado entonces
10.           Inundación (P(x,y-1), C)
11.     Si P(x,y+1)=1 y P(x,y+1) no está etiquetado entonces
12.          Inundación (P(x,y+1), C)
13.     Si P(x+1,y-1)=1 y P(x+1,y-1) no está etiquetado entonces
14.          Inundación (P(x+1,y-1), C)
15.     Si P(x+1,y)=1 y P(x+1,y) no está etiquetado entonces
16.          Inundación (P(x+1,y), C)
17.     Si P(x+1,y+1)=1 y P(x+1,y+1) no está etiquetado entonces
18.          Inundación (P(x+1,y+1), C)
```

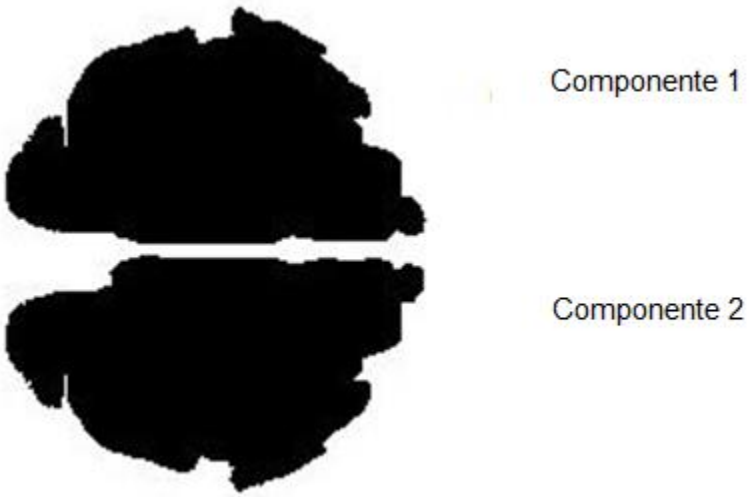


Fig. 4.6 Dos componentes de la tercera capa del Brain

Ya que se tienen identificados los componentes conectados, se van a separar, con el objetivo de tratar cada uno de los componentes conectados de manera independiente en los siguientes pasos.

4.2. Selección del Conjunto de Puntos Dominantes

Ya que se tienen todos los componentes conectados identificados, es necesario obtener el contorno de cada uno de ellos, para esto se hace uso de la morfología matemática con el elemento estructurante de Fig. 4.7 en la resta de conjuntos definida por:

$$[Contorno\ de\ la\ figura] = [Figura\ origial] - [Figura\ erosionada\ con\ Fig.1]. \quad (2)$$

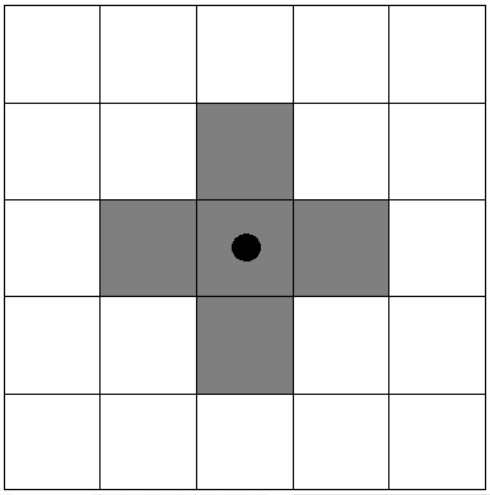


Fig. 4.7 Elemento estructurante para la erosión

Para la explicación de esta parte del método, se utilizó Fig. 4.8 por su simpleza. En Fig. 4.9 (a) se puede observar cual fue el resultado de erosionar Fig.4.8 con el elemento estructurante de Fig. 4.7 y en Fig. 4.9 (b) se observa el contorno obtenido después de aplicar (2).

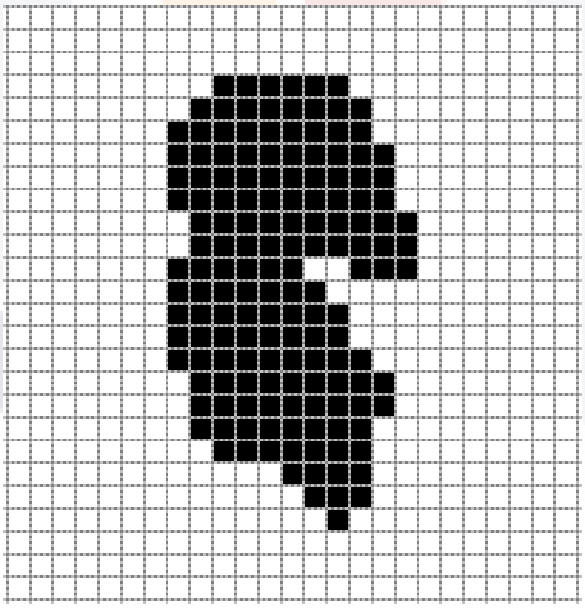


Fig. 4.8 Objeto simple en 2D

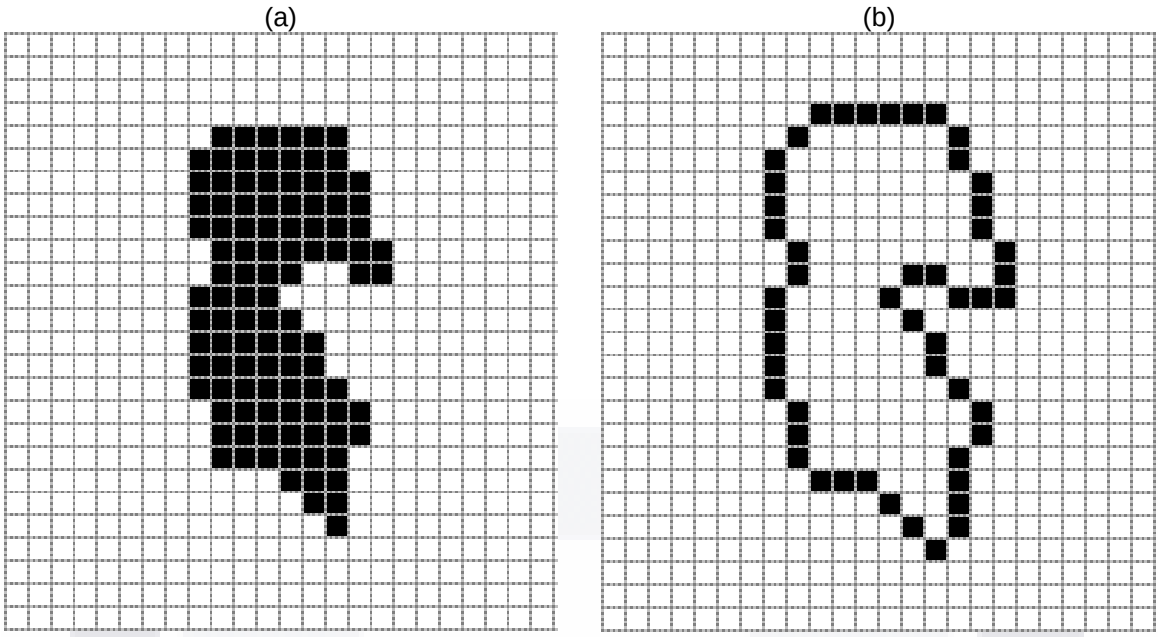


Fig. 4.9 (a) Objeto erosionado; (b) Contorno de la figura

Ya que se obtuvo el contorno de cada figura, ahora se debe de obtener el código de cadena AF8 de todo el contorno de cada uno de los componentes conectado, se utiliza AF8 debido a que es un código de cadena invariante bajo las trasformaciones de rotación y traslación, con base en dos vectores: un vector de referencia y un vector de cambio. AF8 tiene ocho direcciones de cambio, como se puede ver en Fig. 4.10, y cada dirección de cambio está etiquetado con un símbolo del alfabeto:

$$\Sigma_{AF8} = \{a, b, c, d, e, f, g, h\}$$

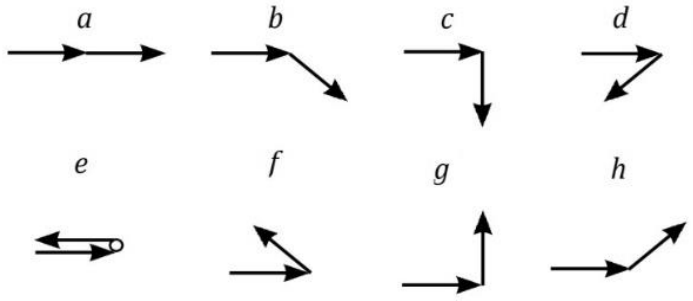


Fig. 4.10 Símbolos del código de cadena AF8

Utilizando el código de Fig. 4.10 en el contorno de Fig. 4.9 (b) se obtiene el siguiente código de cadena AF8:

$$C_{AF8} = aaaabbhbahbacabhhgabhabbhaabcaahabbahbaaabhhbaabab.$$

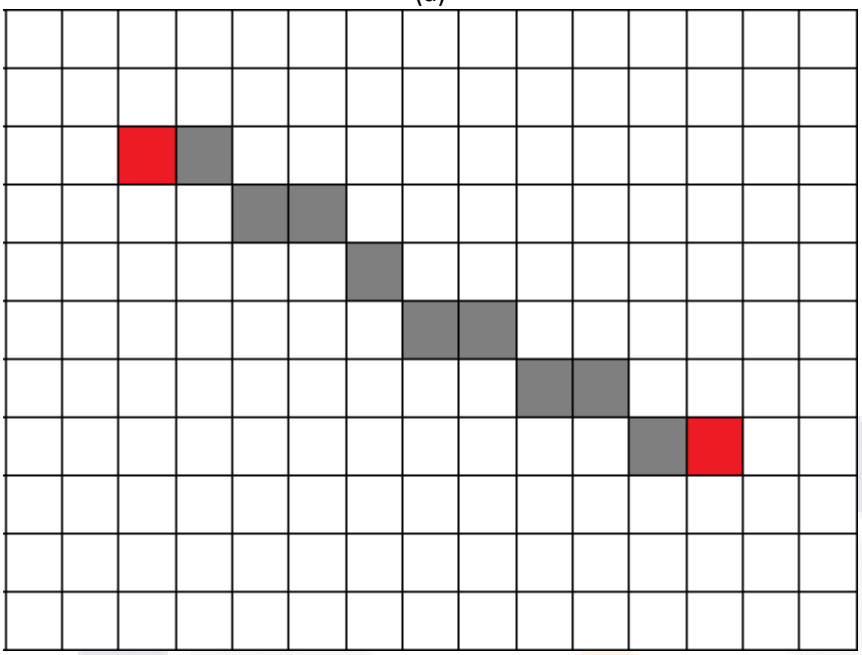
En la Tabla 4.1 se muestra la frecuencia de cada uno de los símbolos de alfabeto Σ_{AF8} anteriormente mencionado. Como se puede observar, los símbolos más comunes son *a, b* y *h*.

Tabla 4.1. Frecuencia de los símbolos de código de cadena AF8

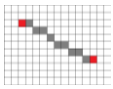
Símbolo	**Frecuencia**
a	21
b	16
c	2
d	0
e	0
f	0
g	1
h	10
Total:	50

El símbolo *a* por sí mismo representa una línea recta, pero una combinación con los otros dos símbolos, *b* y *h*, también se puede crear una recta, visualmente hablando, aunque en el mundo discreto se puede ver claramente que no se trata de una recta, pero si ese conjunto de pixeles se encuentra un poco más alejado del ojo humano o más pequeño, se identificaría a ese segmento como una recta, en Fig. 4.11 se puede observar dos ejemplos de combinación de los tres símbolos, *a, b* y *h,* que pueden formar rectas para el ojo humano, a pesar de que no lo sean, las Fig. 4.11 (a) y (b), son las mismos segmentos, al igual que Fig. 4.11 (c) y (d) pero están a diferente tamaño, por lo que en Fig. 4.11 (b) y (d) el ojo humano las reconoce como rectas.

(a)

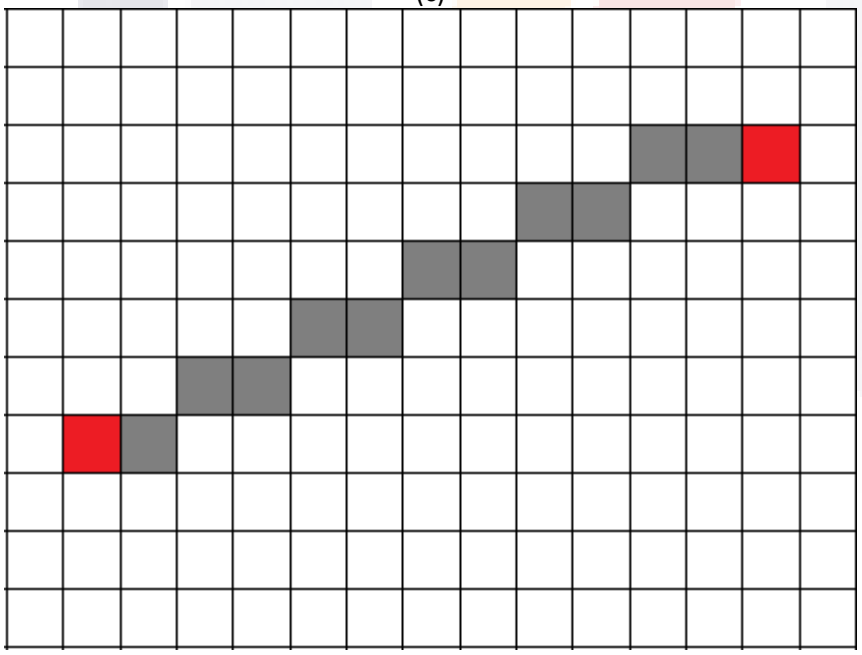


(b)

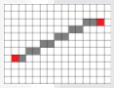


$C_{AF8} = bhbahbhbh$

(c)



(d)



$C_{AF8} = hbhbhbhbhba$

Fig. 4.11 Ejemplos de rectas para el ojo humano

Identificar esos segmentos del contorno que para el ojo humano son rectas, es el objetivo de esta sección del método, encontrar los *segmentos de recta digital* (DSS, por sus siglas en inglés, digital straight segment) [27]. Para esto es necesario evaluar el código de cadena del contorno C_{AF8} de cada figura en el lenguaje L (3):

$$L = \{xa^p(bha^q)^r, xa^p(hba^q)^r | x \in \{a, b, c, d, e, f, g, h\}\}, \quad (3)$$

donde x representa el inicio de un DSS, esto es, un punto dominante, p, q y r representan los valores máximos de apariciones de un símbolo o conjunto de símbolos en paréntesis a los largo del código de cadena del DSS, $a, b, \ldots, h$ son símbolos del alfabeto de AF8 [27].

Para hacer la evaluación del código de cadena en (3), se toma el primer símbolo de la cadena y se evalúa en L, y cada de que cumpla con las reglas de L, se va a ir añadiendo un símbolo más a esta nueva cadena de manera iterativa, hasta que al añadir un símbolo ya no se cumplan las reglas de L, en ese momento se crea el primer DSS, que va desde el primer símbolo añadido, hasta el último símbolo con el que fue aceptado por L, y el siguiente DSS va a comenzar el símbolo que no fue aceptado por el DSS anterior, así sucesivamente hasta que se termine con toda la cadena C_{AF8}.

L es un lenguaje creado a partir de la *gramática libre de contexto* G, dado por la 4-tupla:

$$G = (V, \Sigma_{AF8}, S, P),$$

donde V es el conjunto finito llamado alfabeto de símbolos no terminales, Σ es el conjunto finito llamado alfabeto de símbolos terminales, S es el estado inicial y P es el conjunto de reglas de producción dadas por [27]:

$$S \rightarrow xAB|xAC$$

$$B \rightarrow bhAB| \in$$

$$C \rightarrow hbAC| \in$$

$$A \rightarrow aA| \in,$$

donde $\in$ es la cadena vacía.

L también pueden representarse por medio de un *Autómata Finito Determinístico* (AFD), ya que su evaluación es un sistema determinístico, es decir, por cada estado que se encuentre, solo existe un camino por cada letra del alfabeto Σ, este AFD está formada por una 5-tupla $(Q, \Sigma, q_0, \delta, F)$, donde Q es el conjunto de estados, Σ es el alfabeto, q_0 es el estado incial en donde $q_0 \in Q$, δ son las funciones de transición, y F es el conjunto de estado finales en donde $F \in Q$ [36]. En la Fig.4.12 se puede observar el AFD.

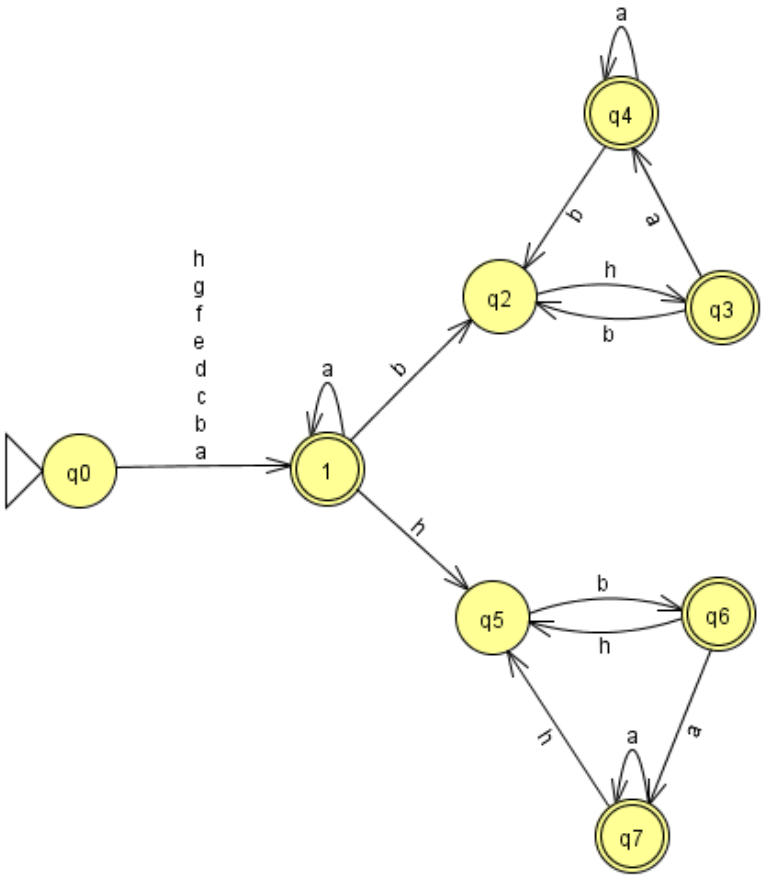


Fig. 4.12 AFD de $\boldsymbol{L}$

Se implementó el AFD propuesto en el software JFLAP para evaluar algunas cadenas si son o no DSS, en la Fig. 4.13 se pueden observar las cadenas con su resultado, en donde si el resultado es *Accept*, la cadena evaluada es un DSS y si es *Reject*, no lo es.

Input	Result
aaaaabhabhabhaaaag	Reject
aaaabhb	Reject
aaaa	Accept
bbh	Accept
bahba	Accept
cabh	Accept
h	Accept
gabha	Accept
bbhaa	Accept
b	Accept
caa	Accept
ha	Accept
b	Accept
bahbaaa	Accept
b	Accept
hhbaa	Accept
ba	Accept
b	Accept
bhbhbhbhbhbhbh	Reject
aaaaaabhbhbhabhabhabhaaaabhabhabha	Accept
ac	Reject

Fig. 4.13 Verificación de cadenas en el AFD

Como ya se mencionó, el código de cadena debe ser evaluado en L, por lo que el código C_{AF8} presentado anteriormente de Fig. 4.9 (b) queda dividido en DSS de la siguiente manera:

$$C_{AF8} = aaaa/bbh/bahba/cabh/h/gabha/bbhaa/b/caaha/b/bahbaaa/b/hhbaa/ba/b,$$

y en términos de x como lo marca L queda de la siguiente manera:

$$C_{AF8} = xaaaxbhxahbaxabhxxabhaxbhaaxxaaxaxxahbaaaxxhbaaxax,$$

donde cada DSS tiene diferente valores de p, q y r, en Tabla 4.2 se puede observar los valores de cada uno así como su equivalencia en L.

Tabla 4.2. DSS y sus valores de $\boldsymbol{p}, \boldsymbol{q}$ y $\boldsymbol{r}$

		Valores por DSS		
DSS	**Equivalencia**	$\boldsymbol{p}$	$\boldsymbol{q}$	$\boldsymbol{r}$
$xaaa$	xa^3	3	0	0
xbh	$xa^0(bha^0)^1$	0	0	1
$xahba$	$xa^1(hba^1)^1$	1	1	1
$xabh$	$xa^1(bha^0)^1$	1	0	1
x	xa^0	0	0	0
$xabha$	$xa^1(bha^1)^1$	1	1	1
$xbhaa$	$xa^0(bha^2)^1$	0	2	1
x	xa^0	0	0	0
xaa	xa^2	2	0	0
xa	xa^1	1	0	0
x	xa^0	0	0	0
$xahbaaa$	$xa^1(hba^3)^1$	1	3	1
x	xa^0	0	0	0
$xhbaa$	$xa^0(hba^2)^1$	0	2	1
xa	xa^1	1	0	0
x	xa^0	0	0	0
	Máximos:	3	3	1

Cabe mencionar que estos valores puede tomar cualquier valor, es decir, no tiene un tope máximo para esta parte de método, por lo que la única restricción sobre los valores es que $p, q, r \geq 0$

El siguiente paso es ubicar los puntos dominantes encontrados en el componente conectado original, en Fig. 4.14 se pueden observar los puntos dominantes encontrados en C_{AF8} de color rojo.

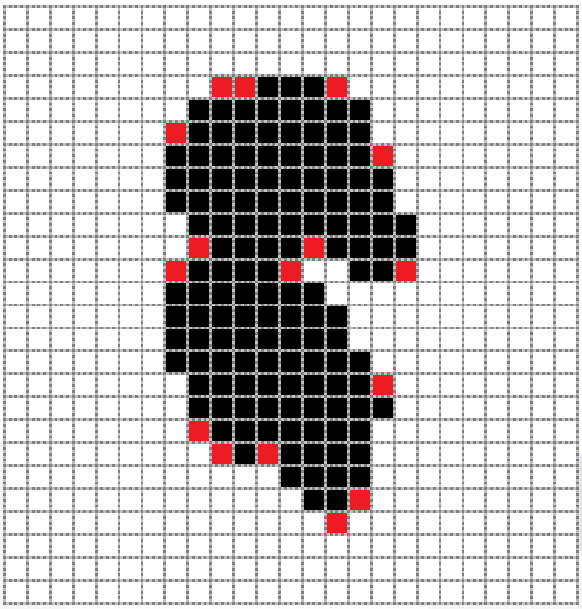


Fig. 4.14 Verificación de cadenas en el AFD

Ahora un caso real por lo que se retoma el componente 2 de la tercera capa del Brain, en Fig. 4.15 se puede observar el contorno del componente.

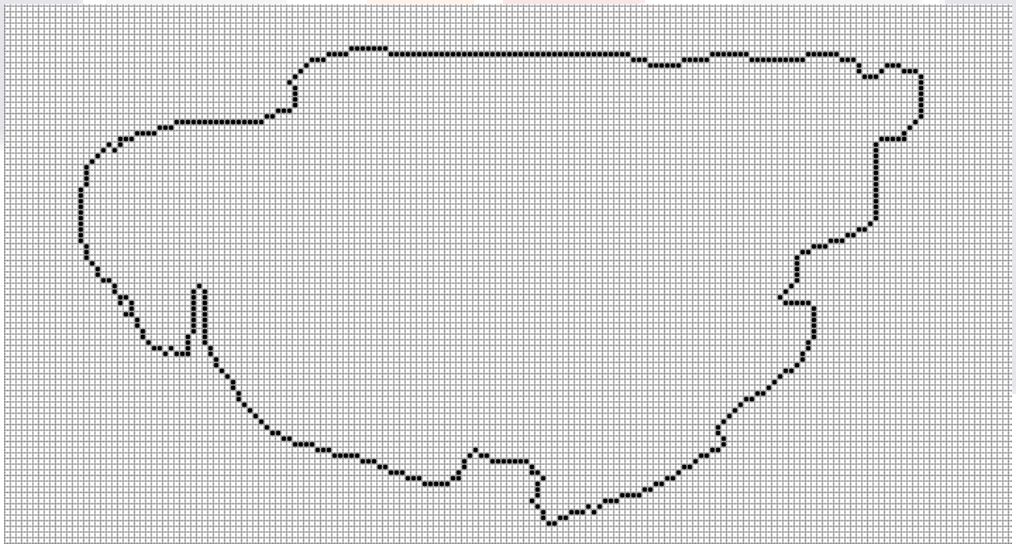


Fig. 4.15 Contorno del componente 2 de la tercera capa del Brain

El código de cadena del contorno de este componente es el siguiente:

$$C_{AF8} = aaaaabhaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa$$
$$aaaaaaaaaaabhabhaaaahbaaahbaaaaabhaaaaaaaah$$
$$baaaabhabbhhahababhabbaaaaaabaahcaaahhaaaaa$$
$$aaaaaaabahbhbhbahbahbhhaaabaaghaaabbaaaabhbh$$
$$babhaaabhbhaaaahhbbbhbhabhabhbhbhbaahbahacg$$
$$bahbhbbbhachabghchhaaaaabhbgahbhchbaaabhabh$$
$$abhbhabhaaabhabhaabhbhbaaaaabhbhaabhbhbaaaa$$
$$aaaahghaaaaaabhaacabgchbabhbhbahgdhbhhbbhab$$
$$ahbaaaaaaaabhaabaaacghcahbaahbahbaaaaaaaaaa$$
$$aaaahbhbahhaahcaaabahbaa.$$

En Fig. 4.16 se pueden observar los puntos dominantes encontrados en el componente 2 de Fig. 4.6, la fecha roja indica el inicio del objeto, es decir, donde comienza el código de cadena.

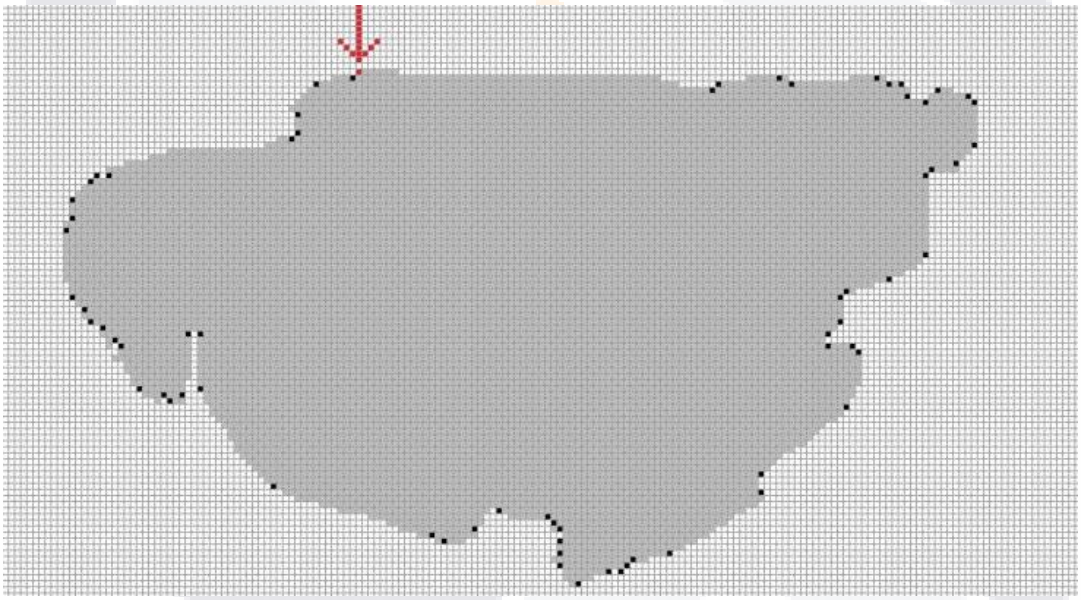


Fig. 4.16 Puntos dominantes del componente 2 de la tercera capa del Brain

Se deben encontrar todos los puntos dominantes de todos los componentes conectados de todas las capas, dando como resultado de esta parte del método, un conjunto de puntos dominantes a organizados a diferente nivel.

4.3. Creación del Poliedro

Una vez que se ha obtenido el conjunto de puntos dominantes, es decir, una nube de puntos, el siguiente paso es la creación del poliedro por medio de la unión de estos puntos dominantes. Para hacer la unión de los mismos, se realiza con dos diferentes maneras que trabajan en conjunto: la primera manera de hacerlo es dentro de cada una de las capas, en donde, todos los puntos dominantes de un mismo componente son unidos entre sí para formar un polígono por cada componente. En la Fig. 4.17 se puede observar el polígono creado a partir de componente 2 de la tercera capa.

Fig. 4.17 Polígono del componente 2 de la tercera capa

Se van a crear tantos polígonos como componentes existan en todas las capas de los pasos anteriores, hasta obtener un conjunto de polígonos a diferentes niveles como se puede ver en la Fig. 4.18.

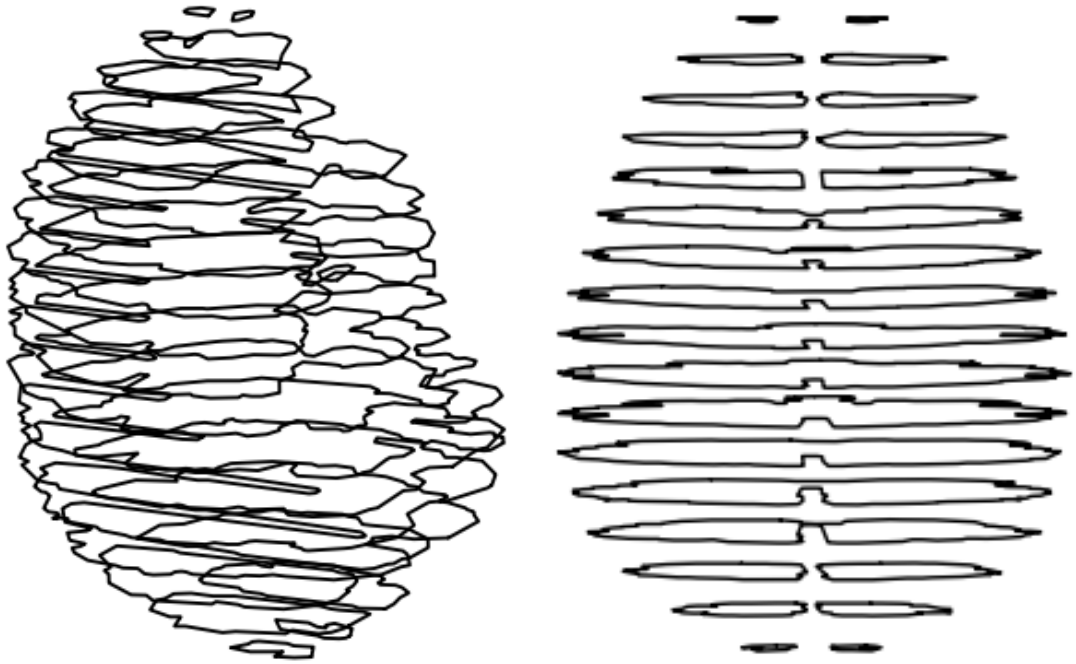


Fig. 4.18 Conjunto de polígonos del Brain desde diferentes perspectivas

Ya que se tiene el conjunto de polígonos, se procede a la segunda manera de unir los puntos dominantes, que es entre capas, uniendo todos los polígonos, para esto, se realizan dos procesos. El primero proceso, llamado hacia arriba, aplica desde la primera capa hasta la penúltima, en la que cada polígono es recorrido por completo, tomando dos puntos dominantes contiguos $p_i = (x_i, y_i, z_i)$ y $p_j = (x_j, y_j, z_j)$en la capa z, de los cuales es necesario calcular su punto medio p_m, a través de la siguiente formula:

$$p_m = \left(\frac{x_i + x_j}{2}, \frac{y_i + y_j}{2}, \frac{z_i + z_j}{2}\right), \tag{4}$$

este punto es utilizando para medir la distancia con los vértices de los polígonos de la capa superior, es decir, se mide la distancia del punto p_m con todos los puntos dominantes de la capa $z + 1$, y se extrae el punto p_k, que es el punto que tiene la menor distancia a p_m. El siguiente paso es unir lo puntos p_i y p_j con el punto p_k, formando así un plano triangular con los tres puntos. En la fig. 4.19 se puede observar como dos puntos contiguos de la capa z se unen un punto de la capa $z + 1$.

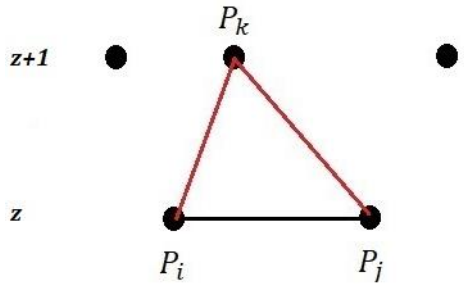


Fig. 4.19 Unión de capas

Ya que se ha terminado el proceso hacia arriba de la unión de puntos dominantes entre capas, el resultado sería algo parecido a la fig. 4.20. Toda la unión de las capas hacia arriba se renderizó para que fuera más fácil la visualización de los planos, como se puede ver en la fig. 4.21.

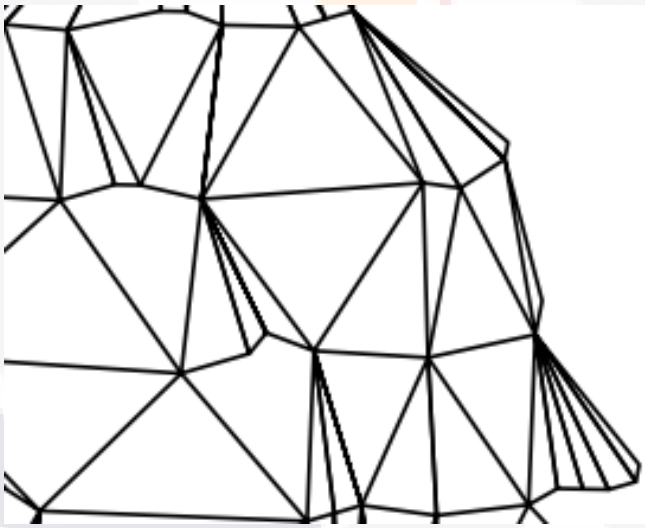


Fig. 4.20 Ejemplo de unión de capas hacia arriba

Fig. 4.21 Unión de capas hacia arriba renderizado

Como se puede observar en la Fig. 4.21 ahora es necesario cubrir esos huecos que se forman para completar el poliedro, para lo que se realiza el segundo proceso de unión de capas, llamado hacia abajo, el cual va aplicar desde la última capa hasta la segunda capa. Este proceso consiste, al igual que el proceso hacia arriba, en recorrer todos los polígonos, tomando dos puntos contiguos p_i y p_j en una capa z , pero a diferencia del primer proceso, no se van a unir con el punto dominante de la capa inferior que tenga la menor distancia a su punto medio p_m, debido a que si se hace de esa manera, podrían empalmarse algunos planos, dejando huecos al no cerrarse el poliedro.

Para evitar la anterior, se van a tener 3 escenarios posibles al unir los dos puntos contiguos p_i y p_j de la capa z, con un punto de la capa $z-1$; en el primero se va a recorrer la capa inferior, $z-1$, en búsqueda de un punto dominante en el que ya exista un arista con el punto p_i y otra arista con el punto p_j, si es posible encontrar ese punto, este es el punto con el que se hace el plano triangular, p_k, debido a que no es posible

encontrar más de un punto en esa situación. En Fig. 4.22 se puede observar como p_k tiene arista tanto con p_i como con p_j.

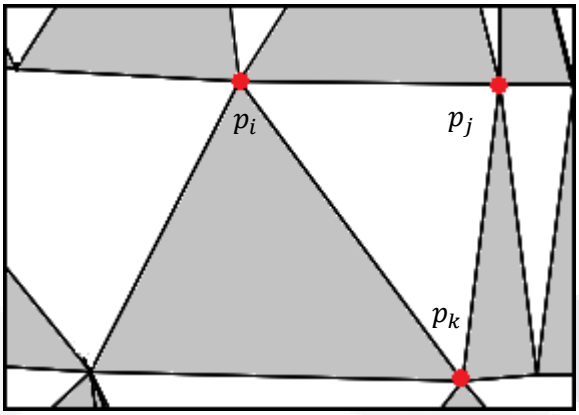


Fig. 4.22 Unión con $\boldsymbol{p_k}$ cuando ya se tienen dos aristas

En caso de que no exista ese punto, y solo existan puntos con una arista, ya sea con el punto p_i o con el punto p_j, ese conjunto de puntos serán evaluados en búsqueda del punto con la menor distancia al punto medio p_m de los puntos p_i y p_j. En Fig. 4.23 y 4.24 se puede observar que solo tiene unión con un punto de la capa de abajo por medio de una arista, ya sea de p_i o p_j.

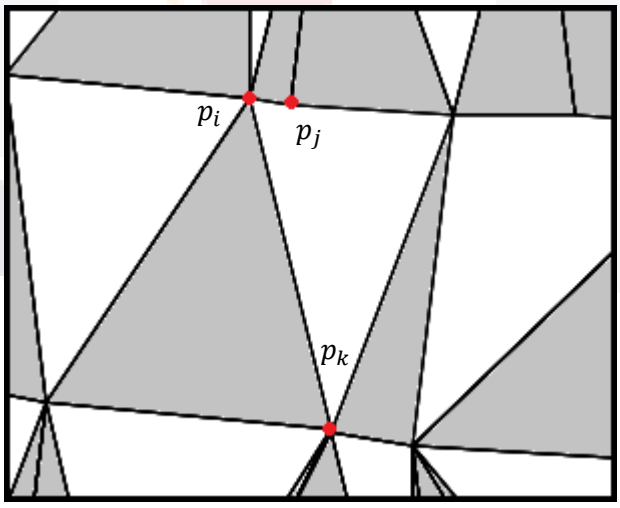


Fig. 4.23 Unión con $\boldsymbol{p_k}$ cuando solo $\boldsymbol{p_i}$ tiene aristas

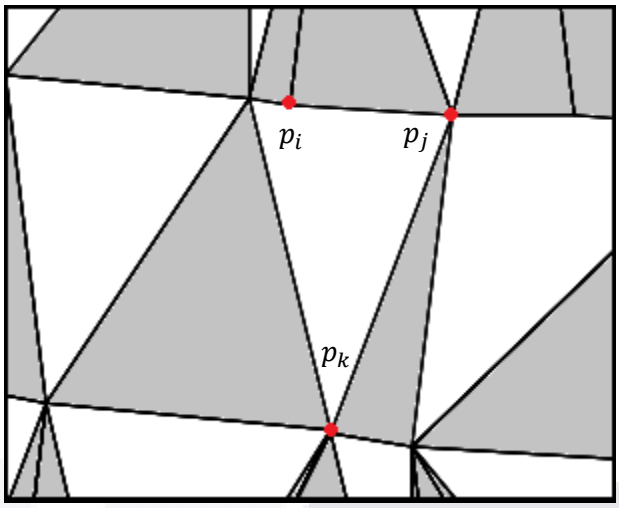


Fig. 4.24 Unión con $\boldsymbol{p_k}$ cuando solo $\boldsymbol{p_j}$ tiene aristas

Si no existe ningún punto que contenga al menos una arista a los puntos p_i y p_j, entonces ahora sí, se realiza una evaluación como en el proceso hacia arriba, el punto que tenga la menor distancia al punto medio p_m de los puntos p_i y p_j. En Fig. 4.25 se puede observar como ninguno de los puntos p_i y p_j tiene un arista a un punto de la capa inferior.

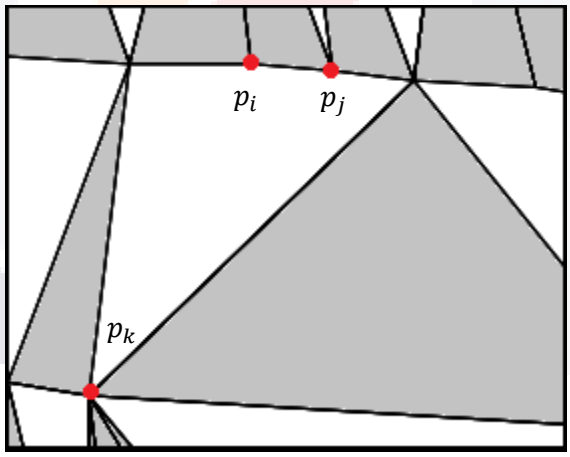


Fig. 4.25 Unión con $\boldsymbol{p_k}$ cuando no se tienen aristas con la capa inferior

Para ambos recorridos se debe de considerar que para unir dos puntos de dos componentes diferentes, debe existir un camino de voxeles en vecindad 26 en el objeto voxelizado entre ellos sin pasar por otros componentes, esto es debido a que el objeto 3D puede presentar algunas extremidades, ramas o algo parecido a eso, y los componentes de esa parte no deben ser unidos a otras capas, aun cuando existan componentes en

niveles superiores o inferiores, esto no significa que estos componentes se encuentran aislados, al contrario, solo marca el final de la extremidad en un sentido, pero sí está unido en el otro sentido del recorrido.

En Fig. 4.26 se puede observar un simple ejemplo de polígonos creados en diversas capas, en donde el objeto tiene una extremidad; los polígonos de color negro, forman la parte principal del objeto, los polígonos de color azul y rojo, forman la extremidad, y el polígono gris es en donde se une la extremidad al cuerpo principal, ahí se puede observar como la extremidad va a terminar en el polígono rojo, es decir, situados en el polígono rojo, solo se va a unir con el polígono azul superior y no con un polígono negro de la capa inferior, debido a que en el objeto principal no hay un camino de 1-voxeles en entre un polígono negro y el polígono rojo sin que pase por el polígono gris.

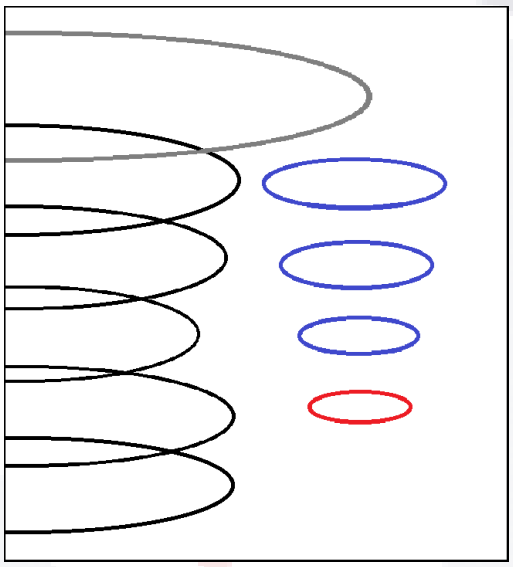


Fig. 4.26 Unión de capas en extremidades

Ya que se han hecho todas las triangulaciones de puntos dominantes y todos los puntos han sido unidos, el resultado es el poliedro final. En Fig. 4.27 se puede observar el poliedro del Brain obtenido desde diferentes perspectivas y en Fig. 4.28 se renderizó el poliedro para la mejor visualización de los planos.

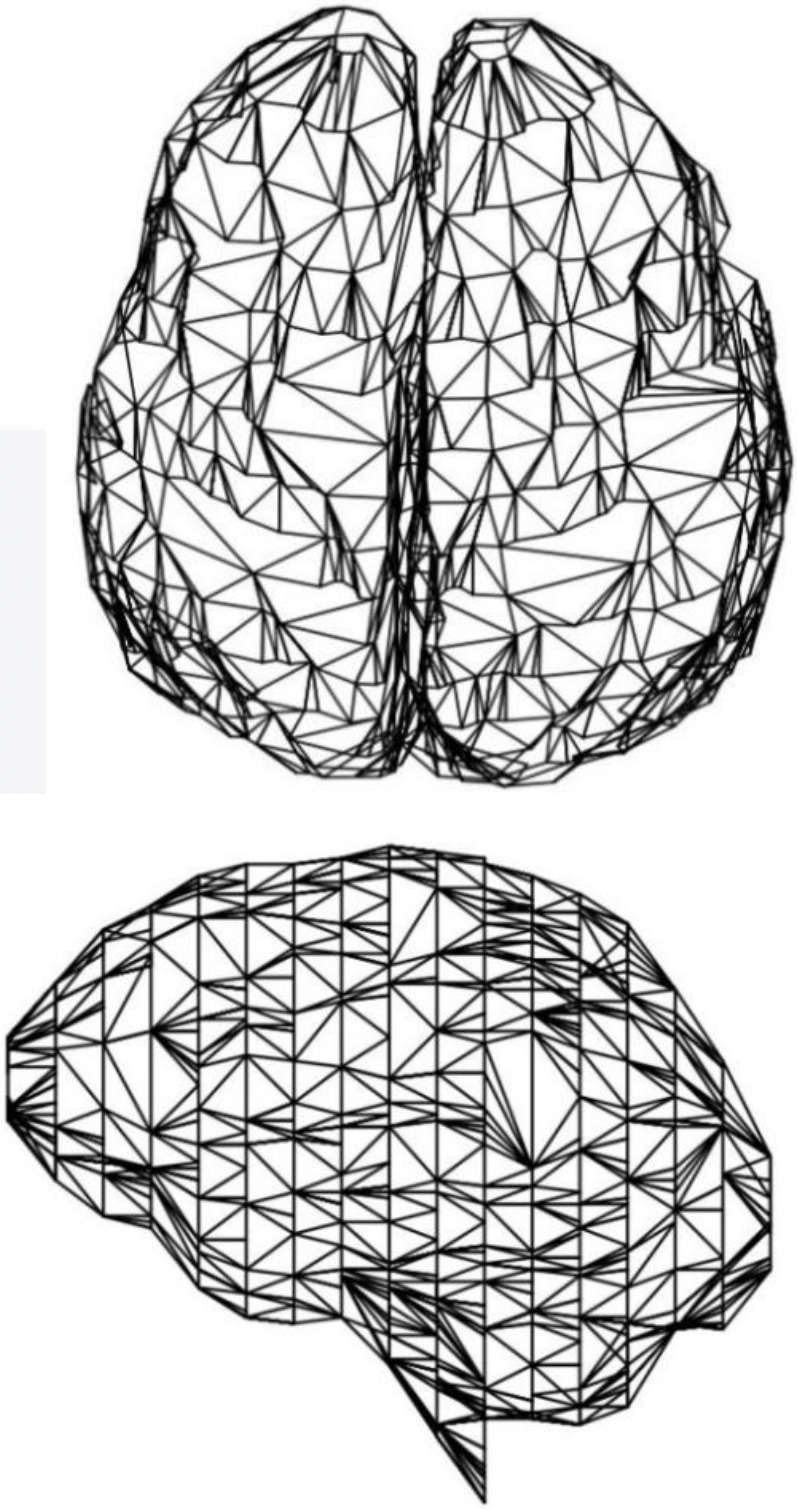


Fig. 4.27 Poliedro del Brain

Fig. 4.28 Poliedro del Brain renderizado

4.4. Cálculo del error

Es necesario evaluar el poliedro obtenido para conocer qué tan bueno es el método propuesto, por ejemplo qué tan diferente es el poliedro obtenido del objeto original, cuanta información se redujo del objeto original al poliedro, entre otros. La evaluación más rápida que se puede hacer es comparar el poliedro obtenido con el objeto original visualmente, pero esta evaluación es muy subjetiva por lo que se implementaron algunos criterios cuantitativos para la evaluación y se muestran a continuación.

4.5. ISE en 3D

Normalmente, el *error cuadrático integral* (ISE, por sus siglas en inglés, Integral Square Error) es utilizado para la evaluación de un polígono al objeto original, es decir, en objetos de 2D, con la suma de todas las distancias perpendiculares del contorno al polígono, pero en esta ocasión se adaptó para un escenario en 3D que permite conocer la distorsión total ocasionada por la aproximación poliedral, por lo tanto, el valor debe ser tan pequeño como un error tolerable lo permita, para un buen poliedro. Por lo tanto la nueva adaptación del ISE está dado por:

$$\mathrm{ISE}_{3D} = \sum_{i=1}^{n\mathrm{VOX}} d_i \qquad (5)$$

donde $n\mathrm{VOX}$ es la cantidad de voxeles en la superficie del objeto original voxelizado, d_i es la distancia perpendicular de un voxel de la superficie del objeto X a una de las cara del poliedro obtenido, es decir, a un plano π dado por 3 puntos P, Q y R. En Fig. 4.29 se puede ver como es la distancia de un voxel X a un plano π, cabe mencionar que esta distancia debe ser calculada para todos los voxeles que se encuentren en la superficie del objeto original.

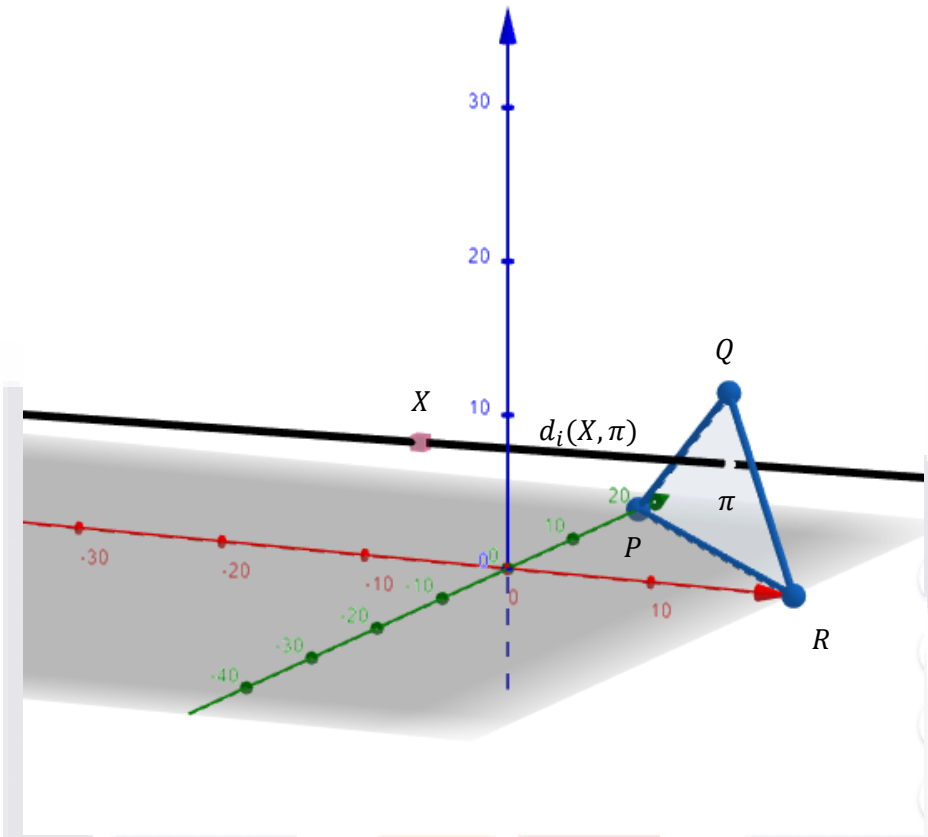


Fig. 4.29 Distancia de un punto a un plano

La distancia d_i esta dada por:

$$d_i(X,\pi) = \frac{|Ax_x + By_x + Cz_x + D|}{\sqrt{A^2 + B^2 + C^2}}, \tag{6}$$

donde $X = (x_x, y_x, z_x)$, es un punto de la superficie del objeto original A, B, C y D son componentes de la ecuación general del plano π y son calculados por las ecuaciones (7) – (10):

$$A = \begin{vmatrix} y_r - y_p & z_r - z_p \\ y_q - y_p & z_q - z_p \end{vmatrix}, \tag{7}$$

$$B = -\begin{vmatrix} x_r - x_p & z_r - z_p \\ x_q - x_p & z_q - z_p \end{vmatrix}, \tag{8}$$

$$C = \begin{vmatrix} x_r - x_p & y_r - y_p \\ x_q - x_p & y_q - y_p \end{vmatrix}, \tag{9}$$

$$D = -x_p \begin{vmatrix} y_r - y_p & z_r - z_p \\ y_q - y_p & z_q - z_p \end{vmatrix} - y_p \begin{vmatrix} x_r - x_p & z_r - z_p \\ x_q - x_p & z_q - z_p \end{vmatrix} - z_p \begin{vmatrix} x_r - x_p & y_r - y_p \\ x_q - x_p & y_q - y_p \end{vmatrix}, \tag{10}$$

donde $P = (x_p, y_p, z_p)$, $Q = (x_q, y_q, z_q)$ y $R = (x_r, y_r, z_r)$ son los puntos que forman el plano π.

4.6. Razón de Compresión

La *razón de compresión* (CR, por sus siglas en inglés, compression ratio) es otro criterio de error, pero este criterio va a permitir conocer cuanta información se redujo del objeto voxelizado a el poliedro obtenido, esto es, que entre menos puntos dominantes, mayor y mejor es la razón de compresión. Este criterio está dado por la ecuación:

$$\mathrm{CR} = \frac{n\mathrm{VOX}}{n\mathrm{DP}}, \tag{11}$$

donde nVOX es la cantidad de voxeles en la superficie del objeto original voxelizado y nDP es el número total de puntos dominantes en el poliedro obtenido.

4.7. FOM

El ISE y CR son criterios que se utilizaron para evaluar la aproximación poliedral, pero cada criterio mide características muy diferentes, por lo que utilizar solo uno de ellos o utilizarlos de manera separada no es suficiente para reconocer un buen método de aproximación poliedral, debido a que puede existir un método que genere un poliedro con

una alta compresión, pero con alta distorsión, o al contrario, un poliedro con muy poca distorsión, pero muy baja compresión.

Para evaluar un método de aproximación es necesario que exista una negociación entre el ISE y CR para encontrar un punto de equilibrio dependiendo las necesidades en la que se vaya aplicar el método, es por eso que en 1993 Sarkar [37] propuso un criterio de error que relaciona los dos ya mencionados criterios, llamado *figura de mérito* (FOM, por sus siglas en inglés, figure of merit), y está dado por:

$$\mathrm{FOM}=\frac{\mathrm{CR}}{\mathrm{ISE}}. \tag{12}$$

4.8. Número de Caras

Los criterios de error ya mencionados (ISE, CR y FOM) han sido ampliamente utilizados en la literatura para la evaluación de métodos de aproximación poligonal, sin embargo, para la evaluación de aproximación poliedral no ha sido publicado un trabajo que permita obtener un criterio de apropiadamente hasta el momento de la escritura de este trabajo. Es por eso que se introduce una nueva definición de criterio de error, llamado *número de caras* $n\mathrm{F}$, el cual va a permitir evaluar únicamente la sección de la creación del poliedro, es decir, la unión de puntos dominantes. En los experimentos realizados con el método propuesto, el número de caras fue aproximadamente al doble del total de puntos dominantes.

5. Resultados Experimentales

El método propuesto fue aplicado a un conjunto de muestra de objetos 3D, de los formatos que se utilizaron fueron archivos con extensión .obj y .ply, en Fig. 5.1 se muestran los 8 objetos utilizados para los experimentos incluido el ya mencionado brain.

Fig. 5.1 Conjunto de muestra de objetos 3D

Los objetos de Fig. 5.1 se voxelizaron con un tamaño de $128 \times 128 \times 128$ voxeles, así que la distancia entre capas fue $N = 8$, se utilizó está medida al voxelizar para que todos los objetos estuvieran a la misma escala, pero se pueden escoger cualquier tamaño al voxelizar, limitado por el software usado para voxelizar y la capacidad de procesamieto, si se escoge otro tamaño, al estar a otra escala, los resultados van a cambiar un poco al tratarse de otro objeto. En Fig. 5.2 se muestran los objetos voxelizados.

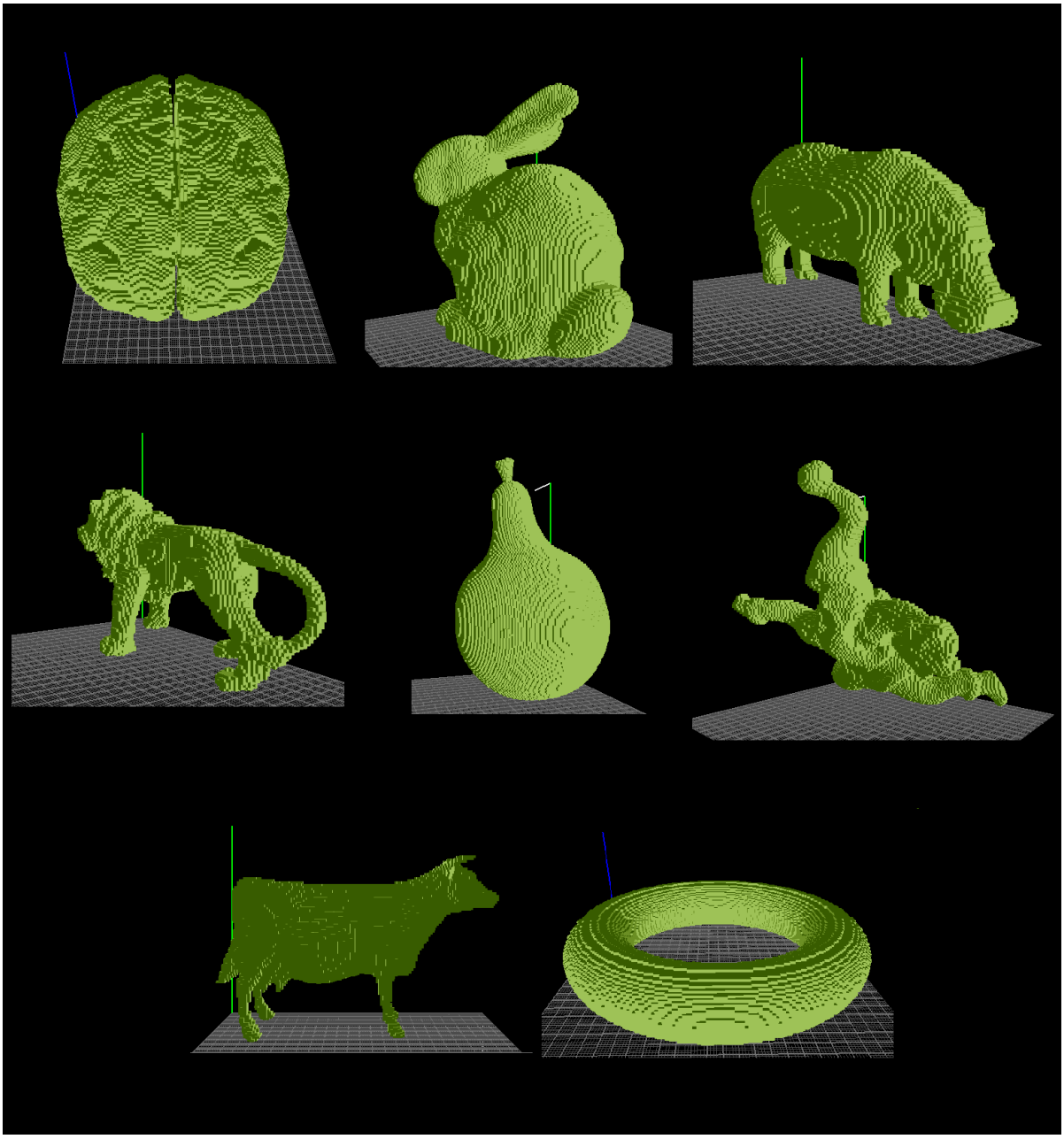


Fig. 5.2 Conjunto de objetos voxelizados

Aplicando el método a estos objetos voxelizados, se obtienen los poliedros de Fig. 5.3, y los criterios de error son mostrados en Tabla 5.1.

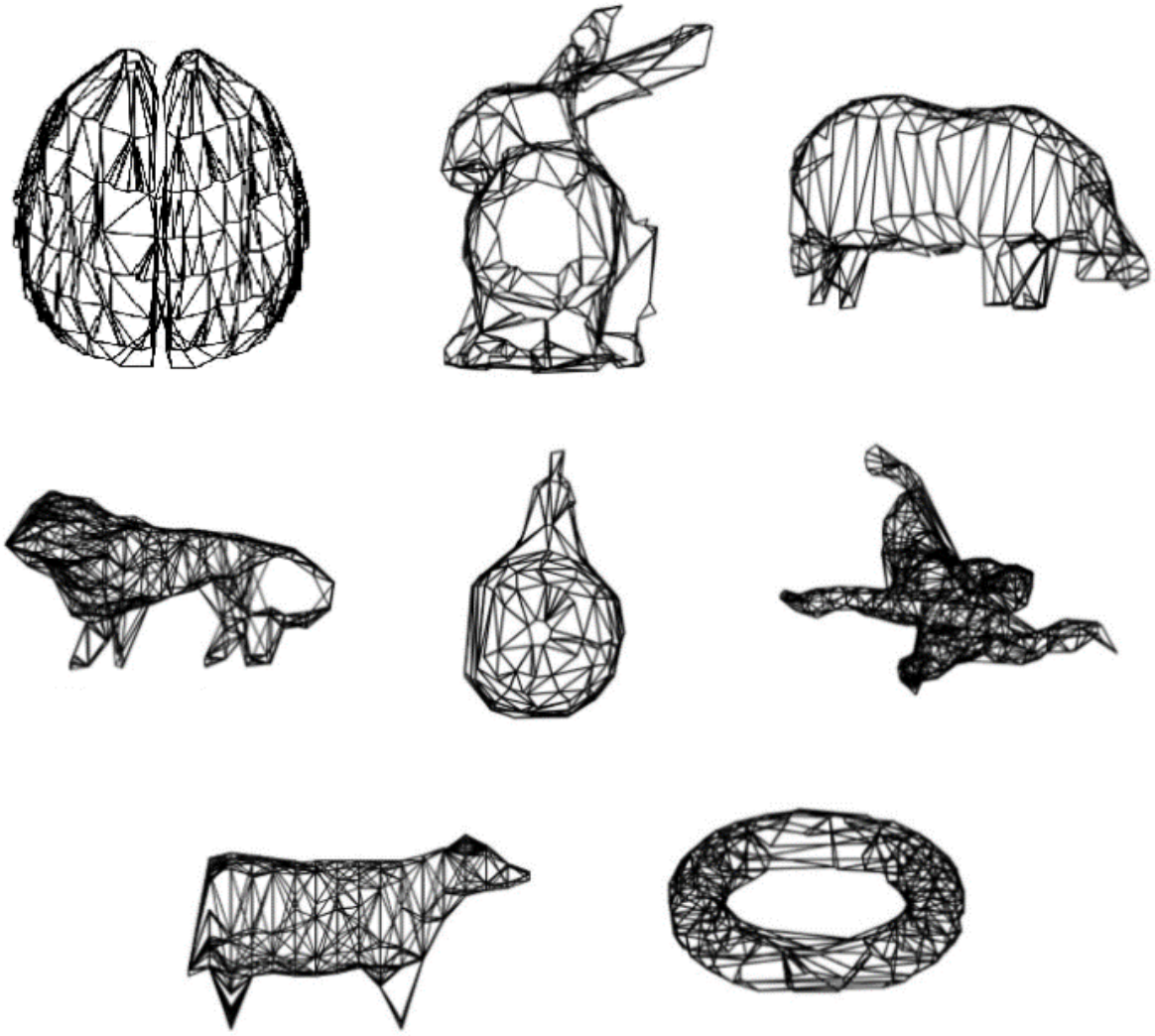


Fig. 5.3 Conjunto de poliedros obtenidos

Tabla 5.1. Criterios de error para los objetos voxelizados a $N = 8$

Objeto	ISE	nDP	CR	FOM	nF
Brain	35919	1380	36.06	0.0010	2706
Bunny	31839	536	61.77	0.0019	1045
Hippo	13631	289	54.70	0.0040	563
Lion	10632	350	35.40	0.0033	672
Pear	17479	240	87.45	0.0050	461
Santa	12307	384	41.67	0.0039	736
Cow	14640	262	50.17	0.0034	512
Torus	23936	328	65.71	0.0027	628

Recordando la sección de selección de capas, especialmente el tamaño entre capas N, se tiene que el valor utilizado fue $N = 8$, pero el siguiente valor posible a utilizar era 16, por lo que se hicieron experimentos a ese tamaño de N, los criterios de error para estos poliedros son mostrados en Tabla 5.2.

Tabla 5.2. Criterios de error para los objetos voxelizados a $N = 16$

Objeto	ISE	nDP	CR	FOM	nF
Brain	66342	710	70.09	0.0010	1368
Bunny	65883	274	120.83	0.0018	536
Hippo	20771	146	108.29	0.0052	274
Lion	18060	170	72.882	0.0040	330
Pear	27209	125	167.92	0.0062	232
Santa	44250	184	86.97	0.0020	356
Cow	28044	117	112.36	0.0040	220
Torus	41624	179	120.42	0.0029	330

Como se puede observar, $n\mathrm{DP}$ cambia inversamente proporcional a N, pero ISE, CR, y $n\mathrm{F}$ cambia directamente proporcional a N, FOM se mantiene relativamente igual debido a que ISE se duplicó y $n\mathrm{DP}$ se redujo por mitad aproximadamente. Como se menciona en la sección 4.1. Selección de Capas y Componentes Conectados, se recomienda utilizar en la distancia entre capas N un entero cercano a la raíz, para tener una buena relación entre el ISE modificado y $n\mathrm{F}$, pero todo va a depender si se requiere mucho o poca definición en el poliedro, esto se podría ajustar modificando N, es decir, un N muy pequeño, va a permitir un poliedro más parecido al objeto original pero con mayor cantidad de cara, y un N más grande, va a generar pocas caras en el poliedro pero este va a tener mayor distorsión.

En los objetos utilizados, se presentó un problema con dos de los objetos, Lion y Santa, esto debido a las extremidades que presentan, ya que no se pudieron alinear, al momento de voxelizar, con el eje Z, y por lo tanto no fue posible cerrar el poliedro por completo en esas zonas. En Fig. 5.4 se puede observar el problema cuando las extremidades tienen un grosor muy pequeño y no están alineadas con el eje Z, se complica la triangulación para unir las capas, debido a que los puntos dominantes más cercanos de la capa superior no son los más factibles para unirlos.

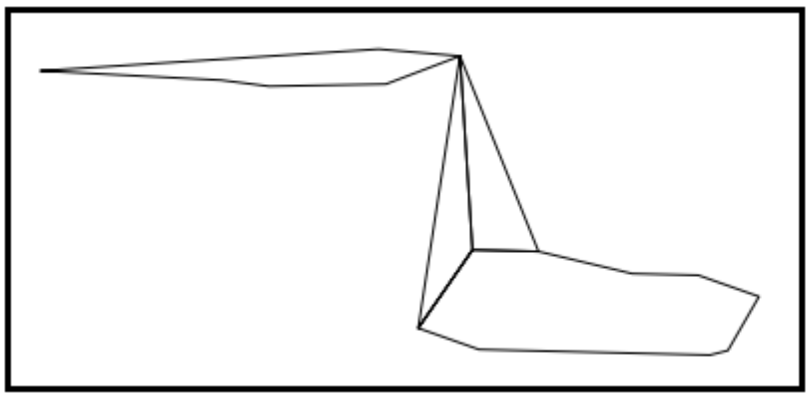


Fig. 5.4 Problema con las capas de extremidades sin estar alineadas

6. Agrupamiento y Reconocimiento de Objetos 3D a partir de sus Características y Poliedro Aproximado

A partir de un objeto voxelizado, se pueden extrar ciertas características y descriptores que permiten identificar el objeto con el que se está tratando, aunado a esto, el poliedro obtenido en las secciones pasadas va a permitir hacer este agrupamiento en conjunto a las características. En este capítulo se presentan algunas características, descriptores y métricas utilizadas para hacer el agrupamiento.

Para el uso de las técnicas de agrupamiento se amplió la muestra que se mostró, utilizando los objetos de Fig. 6.1 en donde todos los objetos ya fueron voxelizados al mismo tamaño $128 \times 128 \times 128$ voxeles.

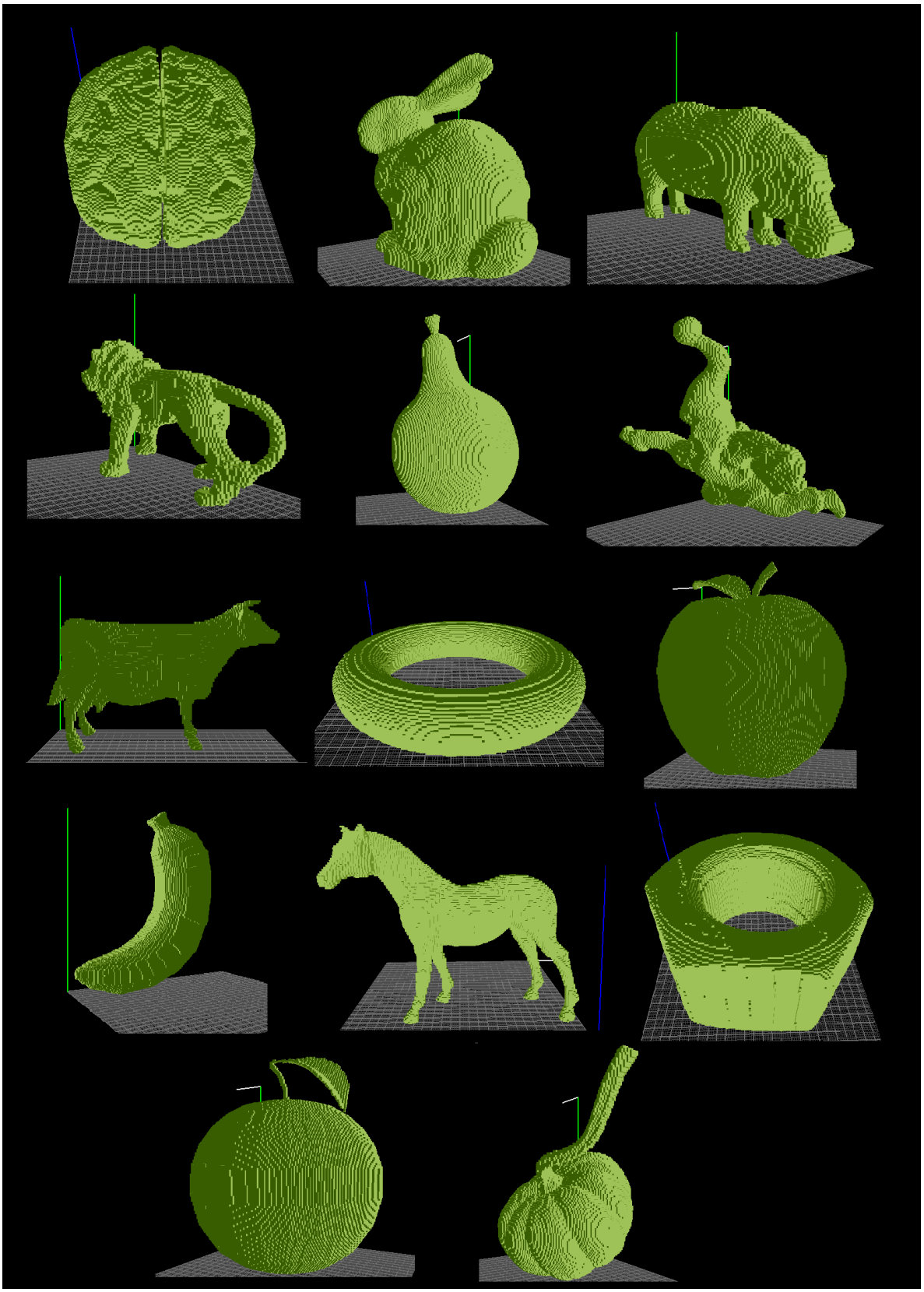


Fig. 6.1 Muestra utilizada para el agrupamiento

6.1. Compacidad Discreta

La compacidad (C_D) es una propiedad intrínseca importante de los objetos, en los objetos 3D esta propiedad relaciona el área de superficie envolvente (A) y el área de la superficie de contacto (A_C), está dada por la fórmula [30]:

$$C_D = \frac{A_C - A_{C_{min}}}{A_{C_{max}} - A_{C_{min}}}, \tag{13}$$

donde A_C es el área de la superficie de contacto, que es la suma de las áreas de las superficies que son comunes a dos voxels, dado por [38]:

$$A_C = \frac{6anVOX - A}{2}, \tag{14}$$

donde a es el área de una cara de un voxel, es decir, 1, nVOX es el número total del voxeles en el objeto, A es el área envolvente del objeto, y es la suma de las áreas de los polígonos del plano externo de los voxeles que forman las caras visibles del objeto 3D, por lo que este valor debe ser calculado al recorrer el objeto y contando las cara que tienen contacto con el exterior, *e.g*. un solo voxel, su área envolvente sería 6, porque son 6 caras que están en contacto con el exterior, es decir, son visibles. $A_{C_{max}}$ y $A_{C_{min}}$ son los máximos y mínimos valores del área de la superficie de contacto, y está dado por [38]:

$$A_{C_{min}} = a(nVOX - 1) \tag{15}$$

Y

$$A_{C_{max}} = 3\left(nVOX - \left(\sqrt[3]{nVOX}\right)^2\right). \tag{16}$$

Estas características fueron calculadas en los objetos de prueba de la Fig. 6.1 y estos resultados pueden ser observados en la tabla 6.1.

Tabla 6.1. Valores de los objetos de prueba

Objeto	**Área Envolvente** A	**Área de contacto** A_C	**No Voxeles en Superficie** $n\mathbf{Vox}$	**No Voxeles Toteles** n_T	**Área de contacto mínima** $A_{C_{min}}$	**Área de contacto máxima** $A_{C_{max}}$	**Compacidad Discreta** C_D
Brain	84620	1701443	49764	581251	581250	1722858.576	0.981240876
Bunny	57270	1230570	33109	419735	419734	1242387.115	0.985635361
Hippo	25668	400554	15811	137796	137795	405384.5176	0.981948031
Lion	22290	176565	12390	62570	62569	182981.7689	0.946710229
Pear	38510	1018226	20990	345827	345826	1022700.339	0.993389705
Santa	30262	254683	16004	89938	89937	263791.8708	0.947606468
Cow	22524	284403	13147	98555	98554	289264.11	0.974510476
Torus	36784	469252	21556	162548	162547	478708.7027	0.970089032
Apple	61830	2124381	34583	718432	718431	2131231.416	0.99515118
Banana	25150	445579	15761	152718	152717	449582.6728	0.98651352
Horse	25038	256140	14120	89553	89552	262654.0691	0.962368624
Nut	60544	1223902	41327	418058	418057	1237400.941	0.983524696
Orange	61060	2091955	33450	707495	707494	2098665.27	0.995176532
Pumpkin	37964	829874	20495	282952	282951	835926.0623	0.989055452

Sobre los mismos objetos de la Fig. 6.1 se todo el método de aproximación poliedral del capítulo anterior para encontrar los puntos dominantes y DSS, con la única restricción sobre los valores fue $p, q, r \geq 0$, los resultados son presentados en la Tabla 6.2, en donde se puede observar que la cantidad de puntos dominantes y DSS son iguales. Adicionalmente se añade la columna de número de planos que conforman el poliedro $n\mathrm{F}$.

Tabla 6.2. Valores obtenidos de la aproximación poliedral sobre la muestra

Objeto	**No de DSS**	$n\mathbf{DP}$	$n\mathbf{F}$
Brain	1380	1380	2706
Bunny	536	536	1045
Hippo	289	289	563
Lion	350	350	672
Pear	240	240	461
Santa	384	384	736
Cow	262	262	512
Torus	328	328	628
Apple	354	354	694
Banana	145	145	279
Horse	213	213	339
Nut	376	376	712
Orange	343	343	669
Pumpkin	382	382	734

Ahora se presenta un análisis entre la relación de los valores presentados que permitan agrupar a los objetos presentados. Uno de ellos es la relación entre área envolvente y número de planos:

$$A/_{n\mathrm{F}}, \tag{17}$$

en el cual los resultados son presentados en Fig. 6.2, en donde se puede observar que existe una cercanía en la relación, debido a que son más parecidos, entre Santa, Cow y Hippo, ya que los tres poseen varias extremidades, el Lion se acerca un poco a este grupo debido a las extremidades, pero se encuentra muy cercano al Brain, ya que la melena del Lion y los detalles en el Brain son muy parecidos. El objeto Pear es el objeto más lejano de los demás, es decir, no es muy semejante a los demás pero es bastante cercano a las otras frutas como Apple, Banana y Orange.

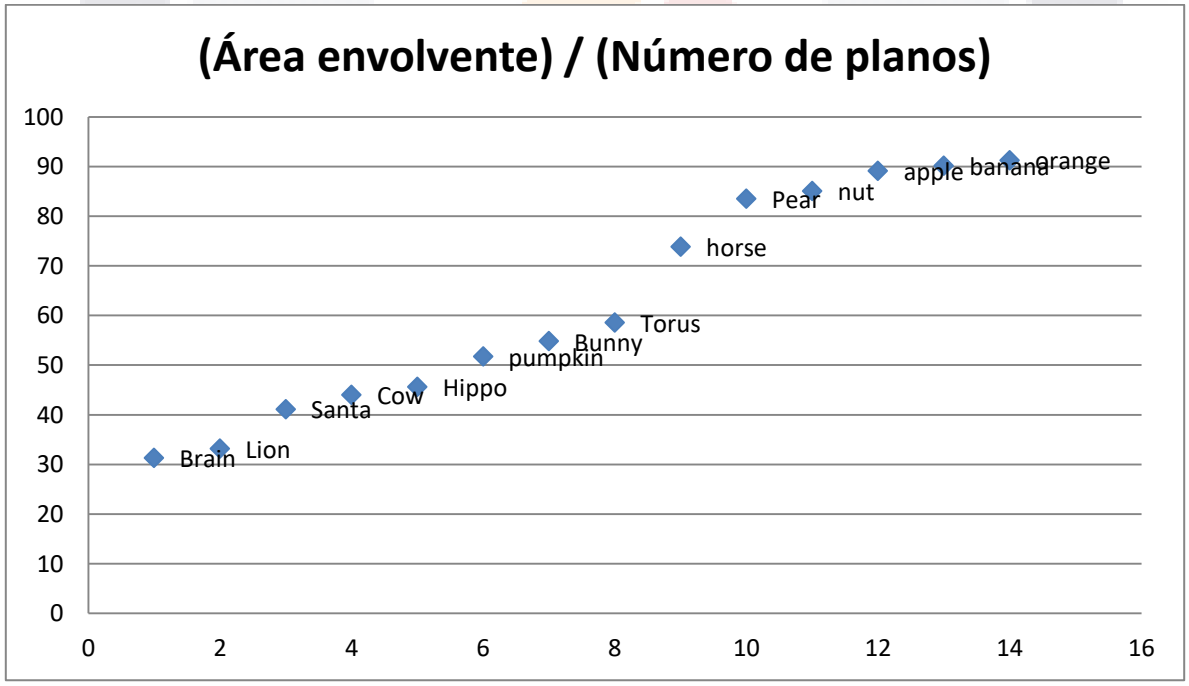


Fig. 6.2 (Área de contacto) / (Número de planos)

Ahora se presenta una relación que incluye área de contacto, número de voxeles en superficie y total voxeles, de la siguiente manera:

$$\frac{n\text{VOX} * n_T}{A_C}, \tag{18}$$

los resultados son presentados en la Fig. 6.3, en donde se ve claramente el grupo de los objetos Lion, Cow, Horse, Hippo y Santa, por lo ya mencionado anteriormente, además de otro grupo de Pumpkin, Pear y Torus, y el último grupo de Bunny, Orange, y Apple, pero ahora el objeto que más se aleja de los demás en el Brain.

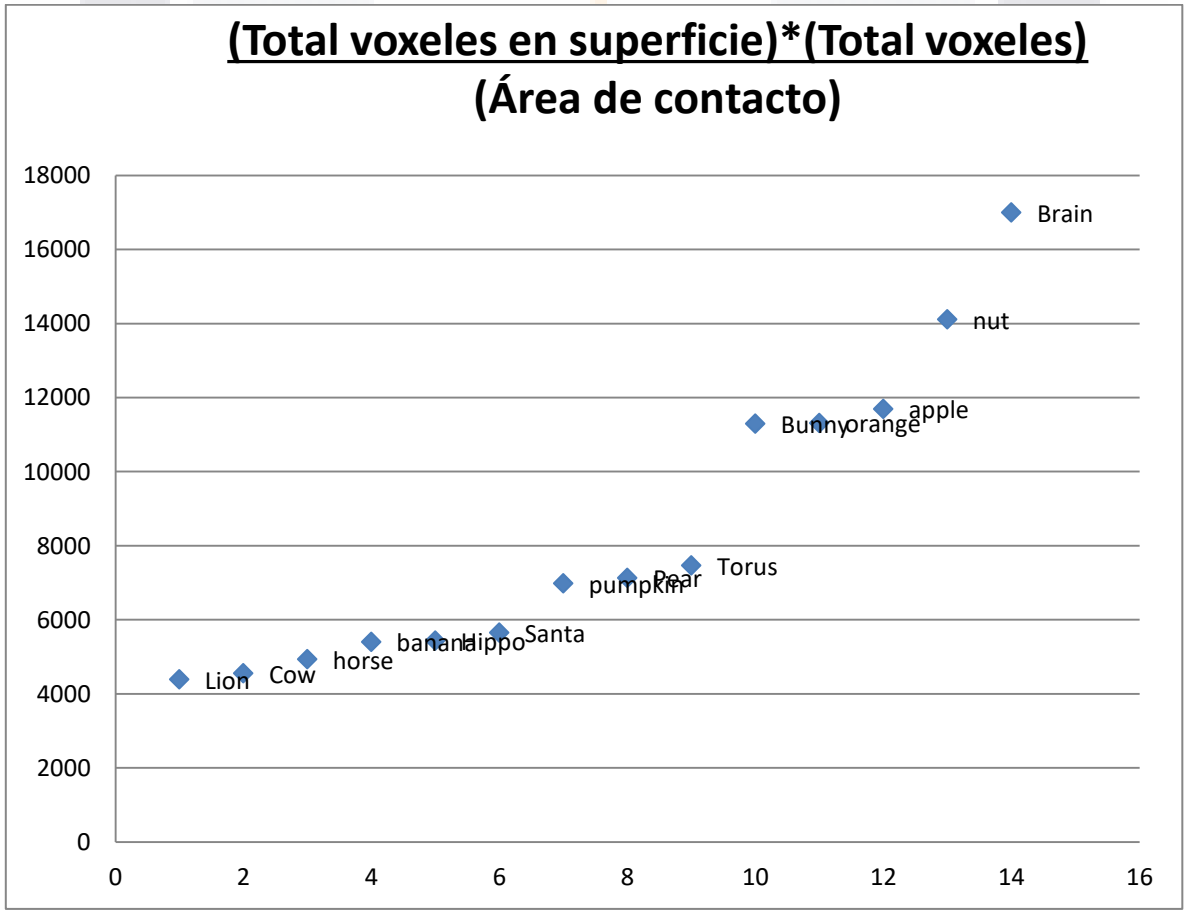


Fig. 6.3 (Total voxeles en superficie)*(total voxeles) / (área de contacto)

Por último, se presenta una relación entre la Compacidad máxima y la Compacidad mínima, ver Fig. 6.4, en donde se rescata el grupo entre Brain, Orange y Apple, ya que los tres presentan una forma redonda, pero se dispersa el grupo con las extremidades, como lo es Horse, Santa y Cow, la relación está dado de la forma:

$$A_{C_{min}} / A_{C_{max}}. \tag{19}$$

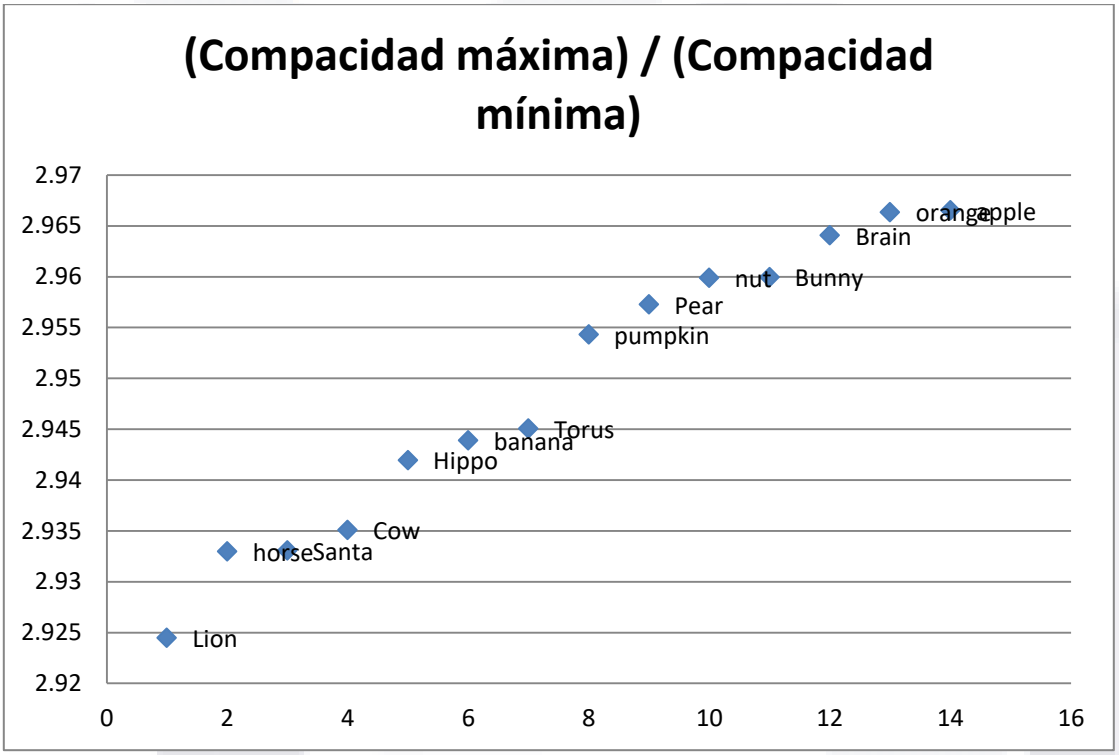


Fig. 6.4 (Compacidad máxima) / (Compacidad mínima)

6.2. K-means

El k-means es una técnica de machine learning que pertenece al aprendizaje no supervisado, es decir, no existe una etiqueta o grupo de salida, en el que se agrupan los objetos en k grupos con base a sus características, el agrupamiento se hace minimizando la suma de distancias entre cada objeto y el centroide de su grupo.

Los datos que se ingresaron al k-means son los de Tabla 6.1, añadiendo una columna más que corresponde al número de caras de Tabla 6.2.

El número del grupos k de salida se envía como parámetro al algoritmo, por lo que en Fig. 6.5 se utilizaron 2 grupos, y se poder ver como se agrupan los objetos Brain, Bunny, Nut, Apple, y Orange; y los objeto en otro grupo son Pear, Pumpkin, Torus, Santa, Lion, Horse, Cow, Hippo, y Banana. Adicional al grupo se contabilizan los objetos que visualmente se considera que no deben pertenecer a ese grupo.

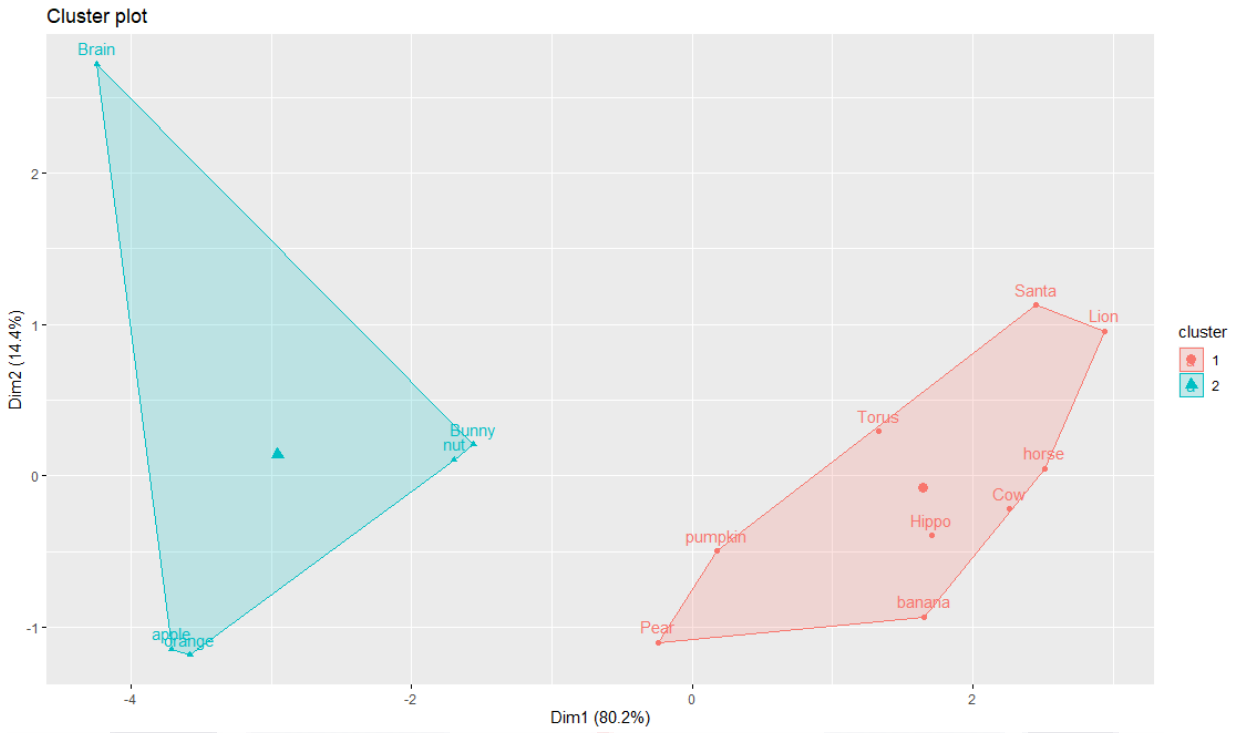


Fig. 6.5 K-means con 2 grupos

Tabla 6.3. Total de errores de K- means con 2 grupos

	Total	Porcentaje
Aciertos	10	71.43
Error	4	28.57

Como ya se mencionó, se puede manipular la cantidad de grupos, y como se puede observar en Fig. 6.5, aún existe demasiada dispersión en los grupo, por lo que en Fig. 6.6 y 6.7, se incrementó la cantidad de clusters, logrado mejores resultados con 4

grupos, de la siguiente forma: grupo 1 (Brain), grupo 2 (Apple, Orange), grupo 3 (Bunny, Nut, Pumpkin, Pear), grupo 4 (Santa, Lion, Horse, Cow, Banana, Hippo, Torus).

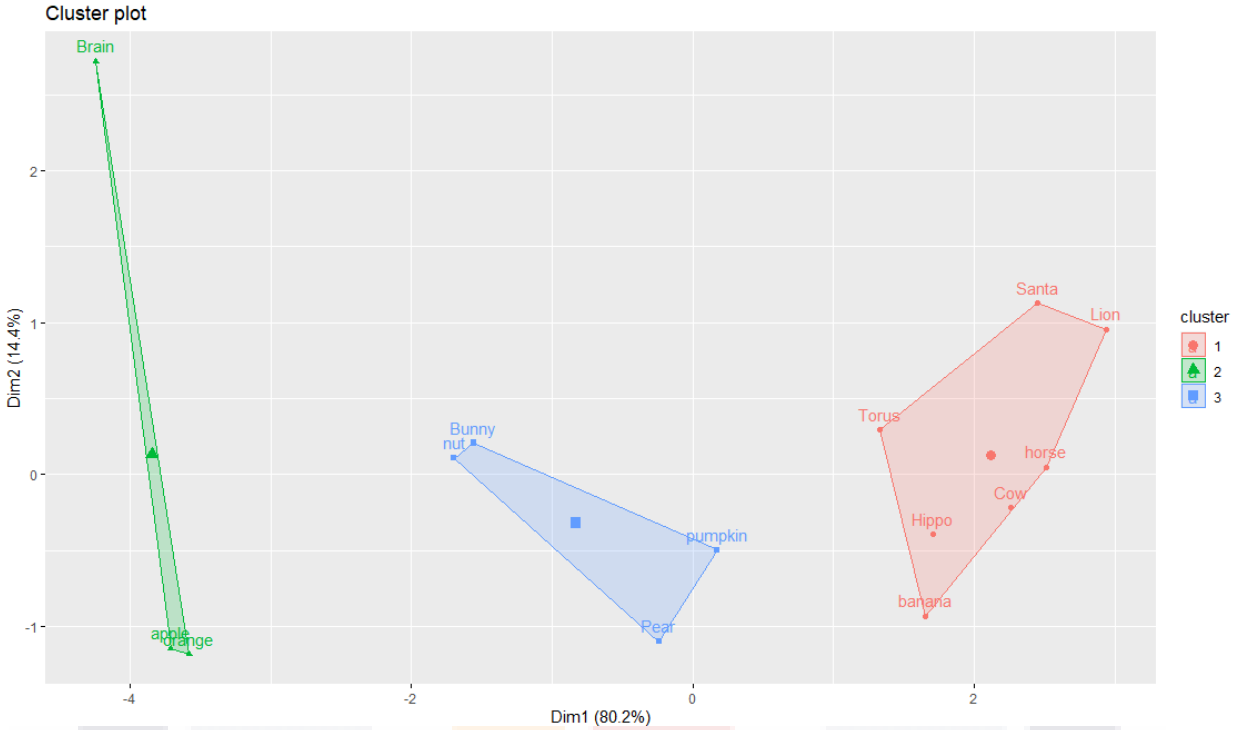


Fig. 6.6 K-means con 3 grupos

Tabla 6.4. Total de errores de K- means con 3 grupos

	Total	Porcentaje
Aciertos	11	78.57
Error	3	21.43

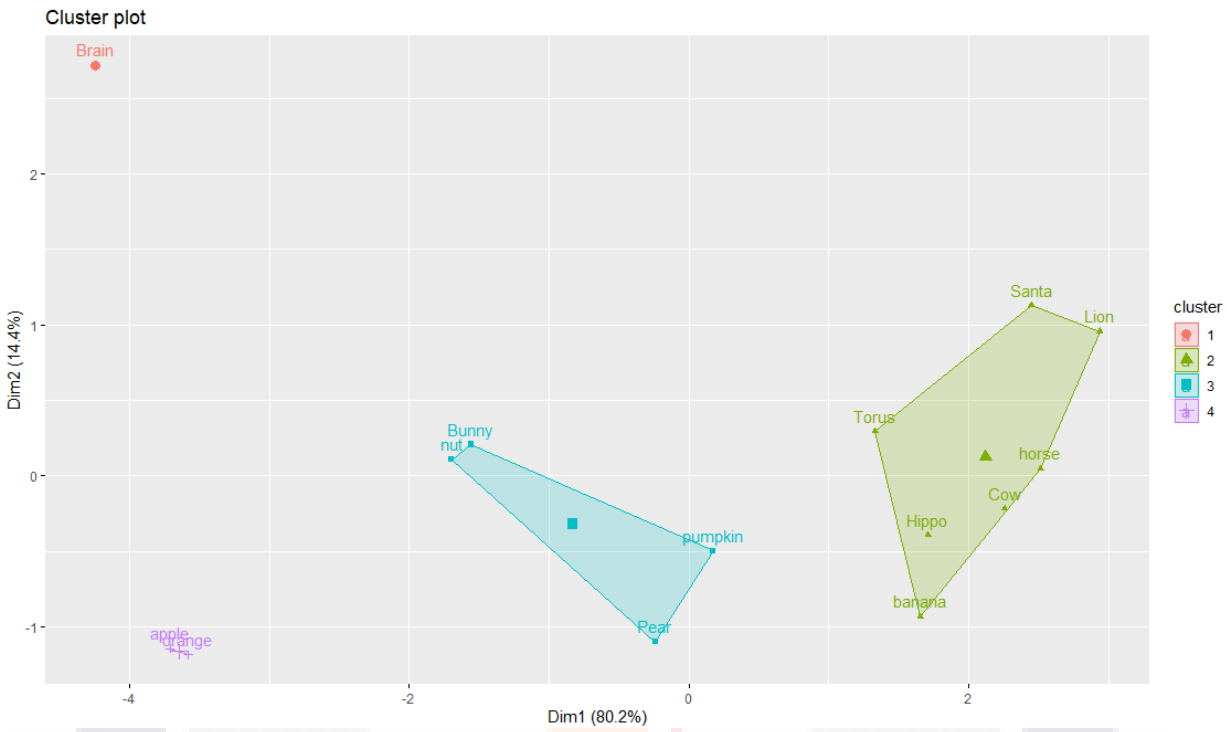


Fig. 6.7 K-means con 4 grupos

Tabla 6.5. Total de errores de K- means con 4 grupos

	Total	Porcentaje
Aciertos	11	78.57
Error	3	21.43

6.3. Clustering Jerárquico

El clustering jerárquico también pertenece al grupo de aprendizaje no supervizado de machine learning, pero a diferencia del anterior no es necesario establecer la cantidad de grupos. La agrupación la hace con base a las distancias entre cada uno de los objetos, buscando que los datos entre los objetos de un clúster sean lo más similares posibles, añadiendo al grupo un objeto que tenga las características más similares, formando así una especie de árbol, cabe mencionar que el valor en el eje Y determina la distancia entre los datos, es decir, entre mayor sea la rama, entre clusters, mayor diferencia tienen los objetos. En Fig. 6.8 se presenta el clúster jerarquico de los objetos de Fig. 1 con los mismos datos que se ingresaron al K-means, en donde se puede observar como se agrupan Apple y Orange para que después se les una Brain, otro clúster muy claro es

Lion, Cow, Santa y Horse, pero los demás objetos que aunque están dentro de un grupo, visualmente no son muy parecidos entre sí.

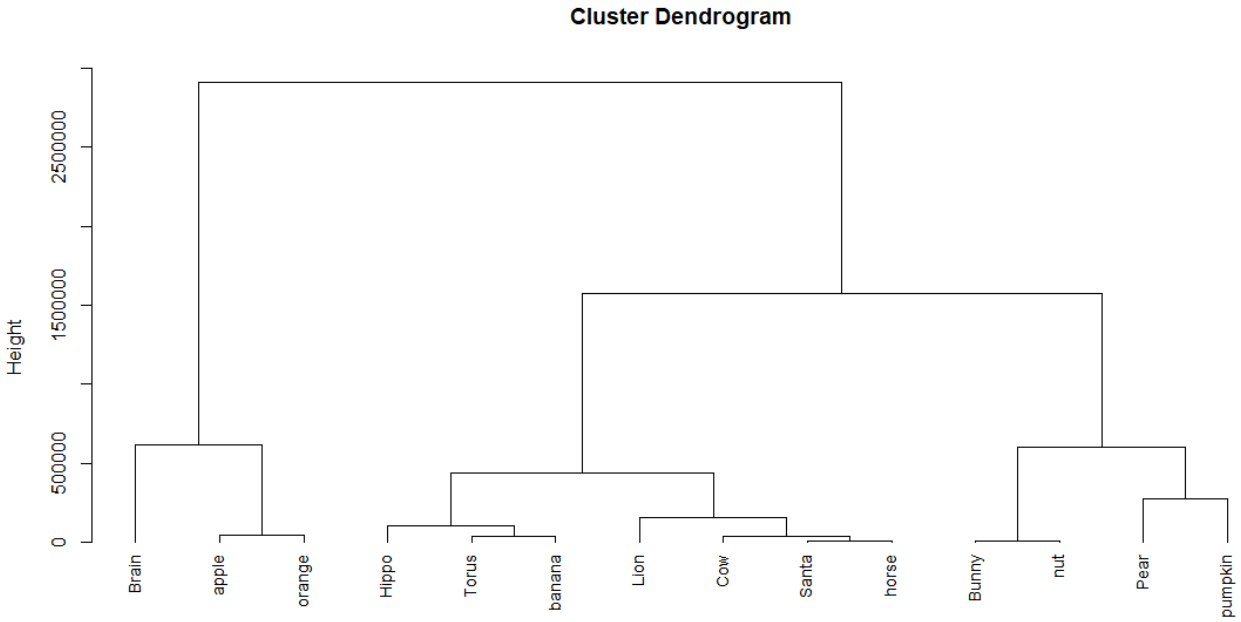


Fig. 6.8 Clustering Jerarquico

Tabla 6.6. Total de errores de clustering Jerarquico

	Total	Porcentaje
Aciertos	8	57.14
Error	6	42.86

6.4. Distancia de Hausdorff Aplicado a los Objetos

La distancia de Hausdorff o métrica de Hausdorff mide, en pocas palabras, que tan lejos está un conjunto de otro conjunto compacto de un espacio métrico. Para este caso, un conjunto está compuesto por su nube de puntos dominantes de un objeto, mientras que el otro conjunto es la nube de puntos dominantes de otro objeto.

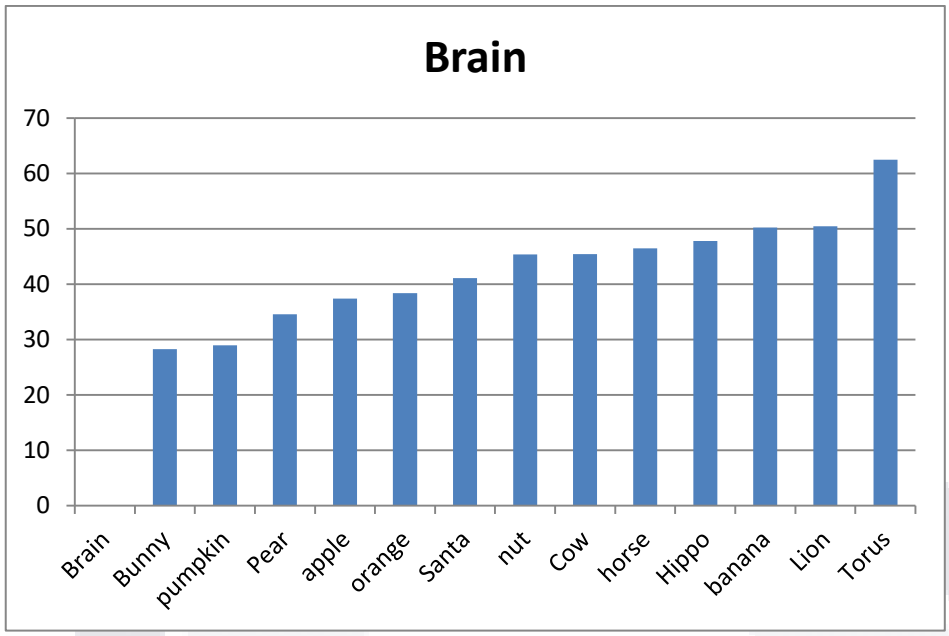
Para hacer la medición de la distancia de Hausdorff entre dos objetos es necesario localizar el centro de masa de los objetos voxelizados, y el objeto con la menor cantidad de puntos dominantes en su nube de puntos debe ser trasladado para que coincidan los

centros de masa, cabe mencionar que antes de voxelizar los objetos, se alinearon todos para tener el mismo sentido, esto gracias a la rotación. Así es que en la tabla 6.7, se muestra un matriz con las distancias entre todos los objetos.

Tabla 6.7. Distancia de Hausdorff entre los objetos

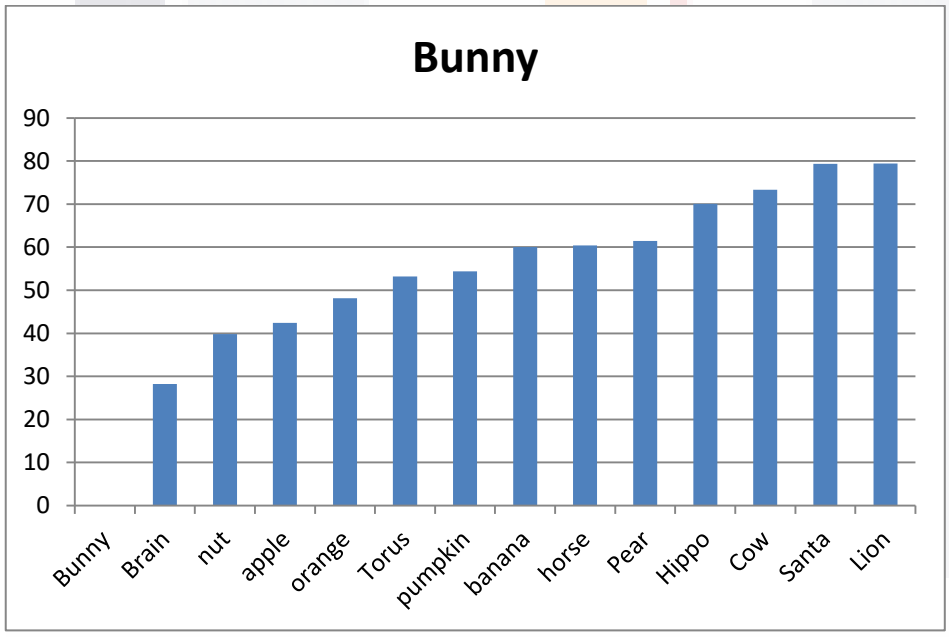
	Brain	Bunny	Hippo	Lion	Pear	Santa	Cow	Torus	orange	banana	apple	pumpkin	nut	horse
Brain	0.00	28.25	47.76	50.44	34.58	41.10	45.41	62.48	38.39	50.22	37.36	28.96	45.36	46.44
Bunny	28.25	0.00	70.03	79.44	61.40	79.34	73.35	53.17	48.13	60.00	42.38	54.40	39.82	60.41
Hippo	47.76	70.03	0.00	39.26	30.28	36.95	37.15	34.29	37.14	49.04	36.52	31.37	27.83	45.18
Lion	50.44	79.44	39.26	0.00	34.80	40.17	29.19	37.65	44.94	39.68	45.16	35.63	31.09	29.44
Pear	34.58	61.40	30.28	34.80	0.00	28.07	63.84	50.55	20.70	36.48	25.29	51.51	32.77	55.11
Santa	41.10	79.34	36.95	40.17	28.07	0.00	60.36	59.61	45.80	55.12	42.73	48.70	43.06	54.15
Cow	45.41	73.35	37.15	29.19	63.84	60.36	0.00	39.55	39.56	43.62	38.57	33.31	30.32	25.13
Torus	62.48	53.17	34.29	37.65	50.55	59.61	39.55	0.00	36.87	48.46	35.34	43.13	26.15	46.11
orange	38.39	48.13	37.14	44.94	20.70	45.80	39.56	36.87	0.00	45.86	22.31	40.56	40.77	55.12
banana	50.22	60.00	49.04	39.68	36.48	55.12	43.62	48.46	45.86	0.00	38.03	36.52	30.89	53.90
apple	37.36	42.38	36.52	45.16	25.29	42.73	38.57	35.34	22.31	38.03	0.00	34.29	45.17	47.78
pumpkin	28.96	54.40	31.37	35.63	51.51	48.70	33.31	43.13	40.56	36.52	34.29	0.00	54.21	52.44
nut	45.36	39.82	27.83	31.09	32.77	43.06	30.32	26.15	40.77	30.89	45.17	54.21	0.00	39.02
horse	46.44	60.41	45.18	29.44	55.11	54.15	25.13	46.11	55.12	53.90	47.78	52.44	39.02	0.00

Esta información permite identificar que tan alejados está una nube de puntos de la otra, es decir, que tan diferentes son ambos objetos, por lo que a continuación se muestran una gráfica por cada objeto en orden de cercanía, permitiendo identificar cual el objeto más cercano y más alejado respecto al objeto que se está evaluando. Se determinó un umbral con valor de 30, para determinar si existe un parecido con el objeto evaluado o no, es decir, si la distancia de Hausdorff entre los dos objetos es menor de 30, se determina que existe un parecido entre ambos objetos y si es mayor del umbral, los objetos no tienen parecido. Adicional a esto, se contabilizaron los aciertos y errores de cada evaluación, para determinar si era un error o acierto se analizan ambos objetos visualmente.



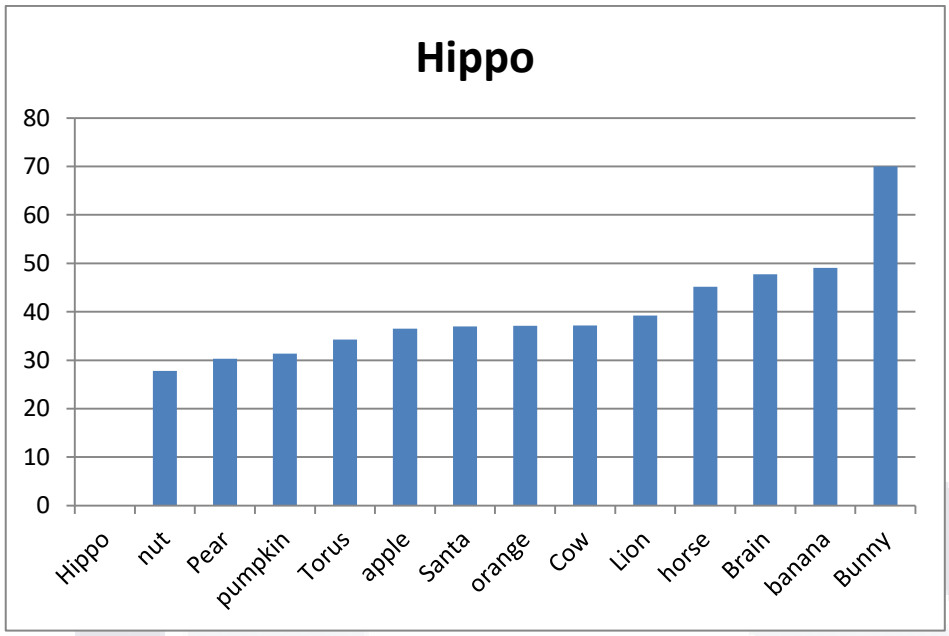
Aciertos: 13
Errores: 0

Fig. 6.9 Comparación de la distancia de Hausdorff de Brain



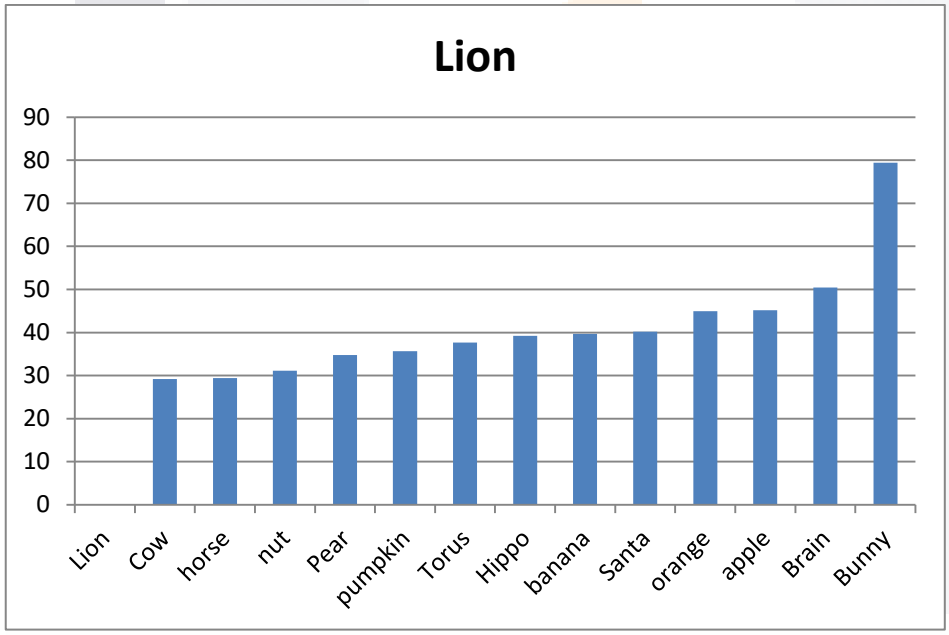
Aciertos: 13
Errores: 0

Fig. 6.10 Comparación de la distancia de Hausdorff de Bunny



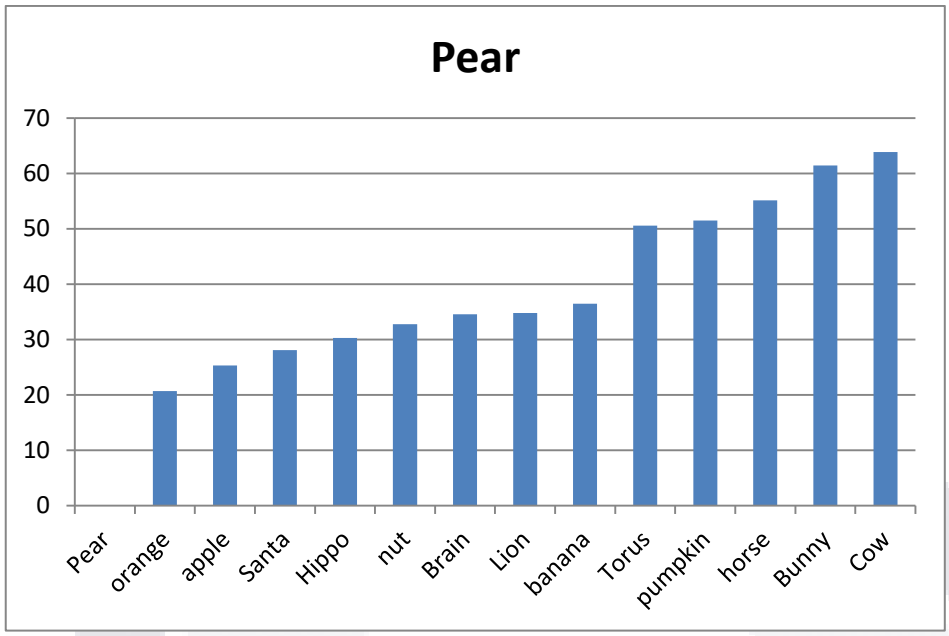
Aciertos: 9
Errores: 4

Fig. 6.11 Comparación de la distancia de Hausdorff de Hippo



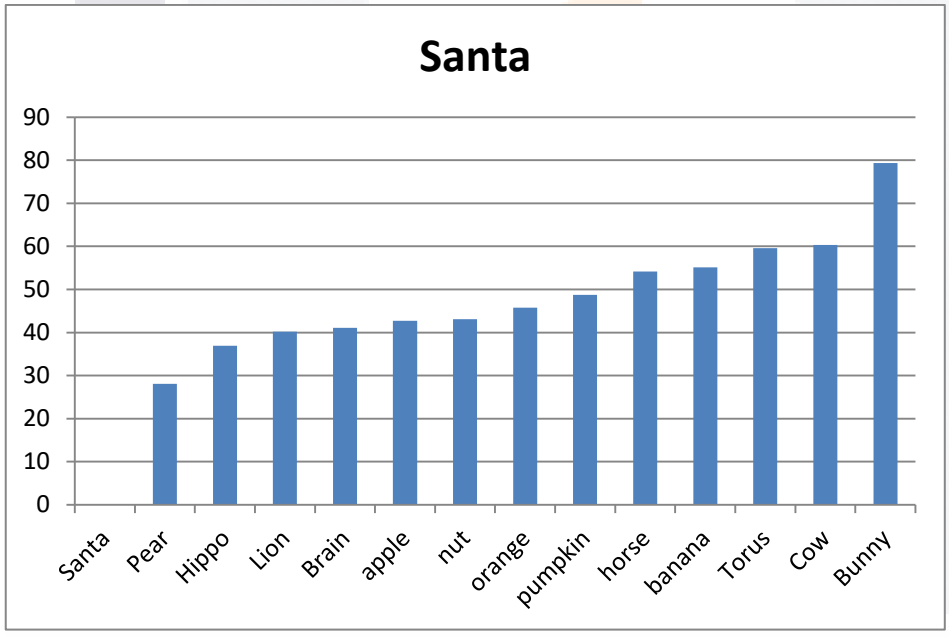
Aciertos:12
Errores:1

Fig. 6.12 Comparación de la distancia de Hausdorff de Lion



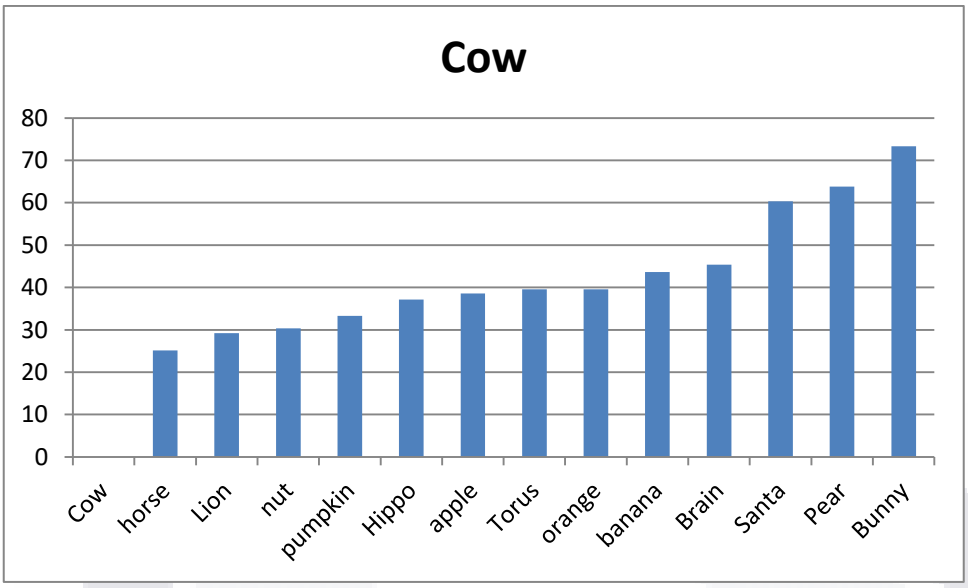
Aciertos:12
Errores:1

Fig. 6.13 Comparación de la distancia de Hausdorff de Pear



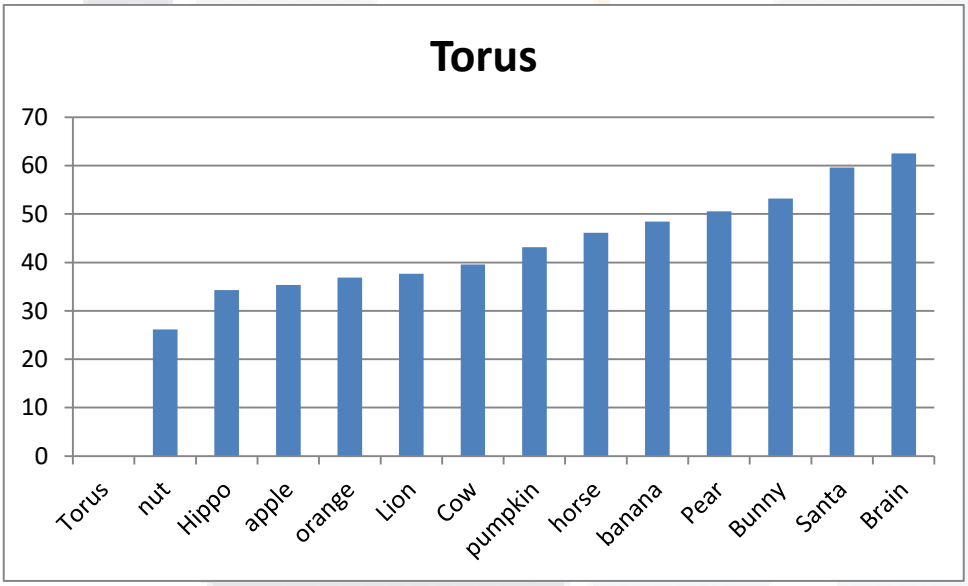
Aciertos:12
Errores:1

Fig. 6.14 Comparación de la distancia de Hausdorff de Santa



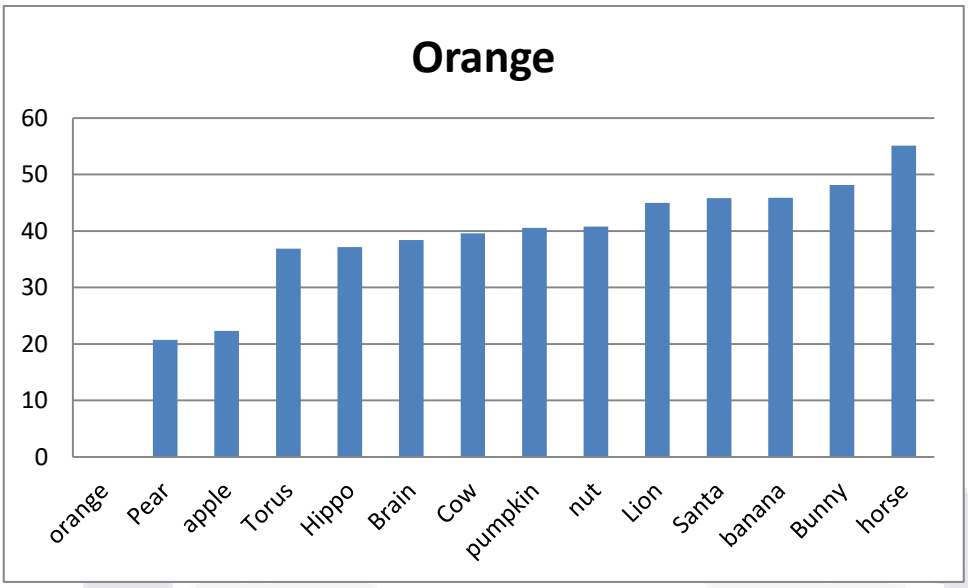
Aciertos:12
Errores:1

Fig. 6.15 Comparación de la distancia de Hausdorff de Cow



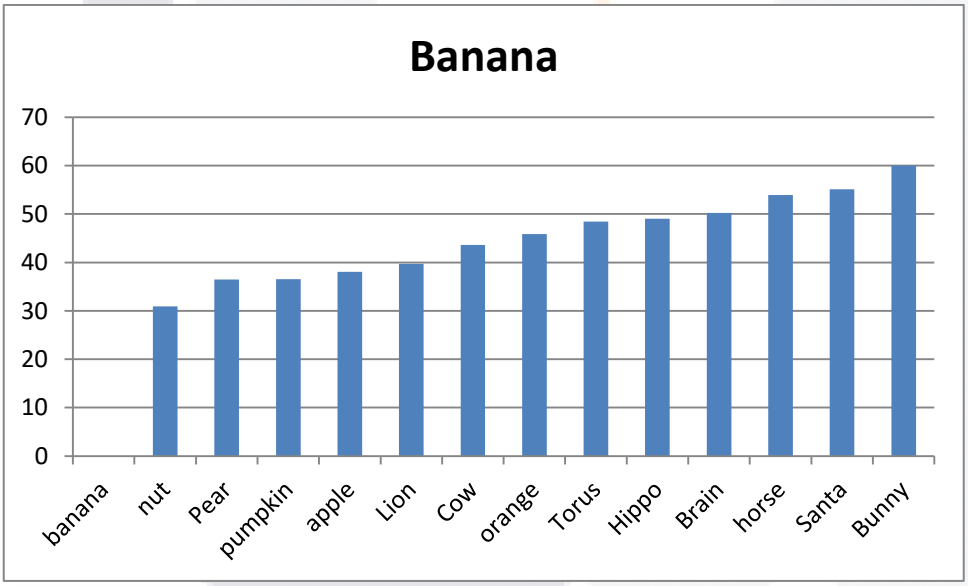
Aciertos:13
Errores:0

Fig. 6.16 Comparación de la distancia de Hausdorff de Torus



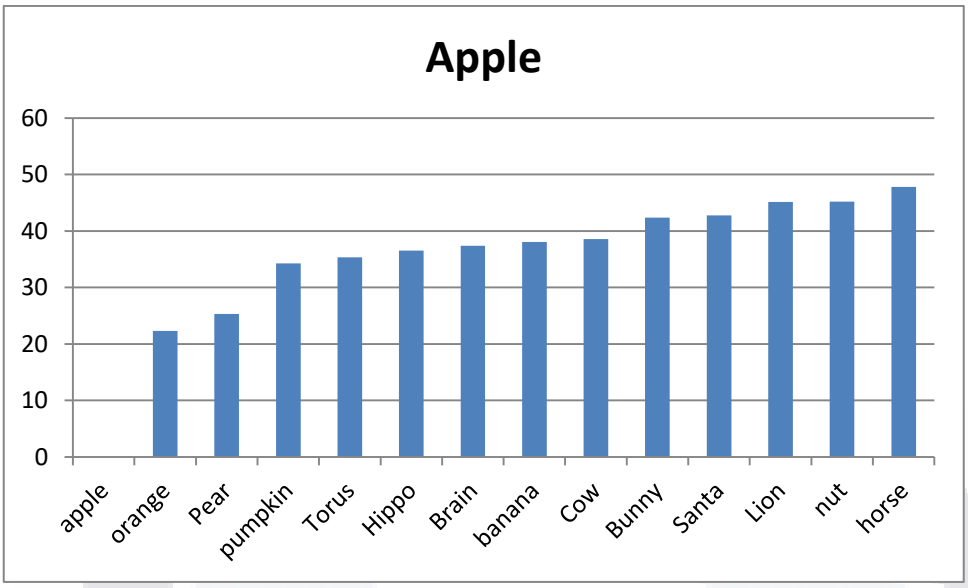
Aciertos:13
Errores:0

Fig. 6.17 Comparación de la distancia de Hausdorff de Orange



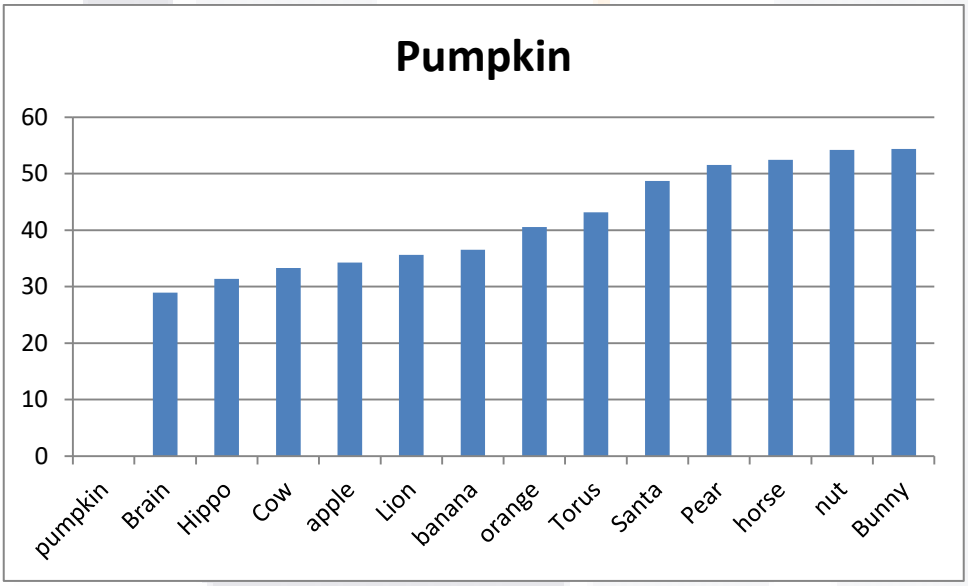
Aciertos:13
Errores:0

Fig. 6.18 Comparación de la distancia de Hausdorff de Banana



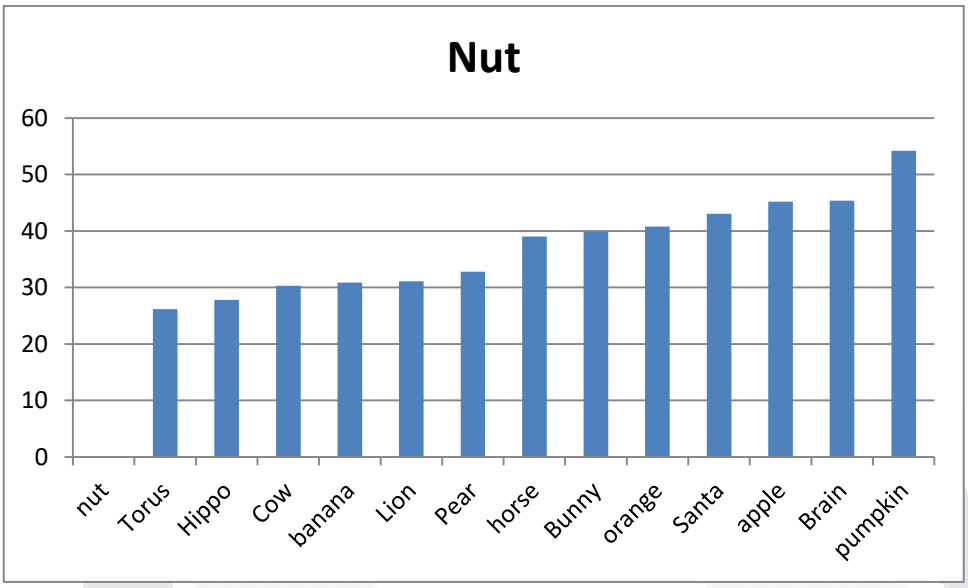
Aciertos:13
Errores:0

Fig. 6.19 Comparación de la distancia de Hausdorff de Apple



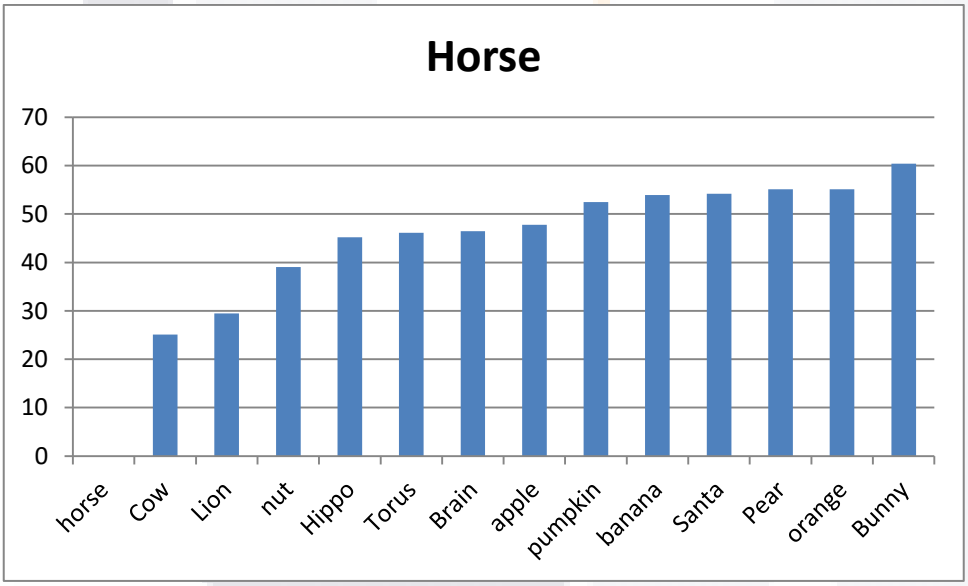
Aciertos:13
Errores:0

Fig. 6.20 Comparación de la distancia de Hausdorff de Pumpkin



Aciertos:12
Errores:1

Fig. 6.21 Comparación de la distancia de Hausdorff de Nut



Aciertos:12
Errores:1

Fig. 6.22 Comparación de la distancia de Hausdorff de Horse

Los totales se muestran en Tabla 6.8, cabe mencionar, que la suma total debe dividirse entre dos, ya que de no hacerlo, se estarían contado doble los aciertos y errores; y se puede observar que los resultados son bastante aceptable ya que se tiene un 94.51% de aciertos.

Tabla 6.8. Total de errores en la distancia de Hausdorff

	Total	Porcentaje
Aciertos	86	94.51
Error	5	5.4

Se hizo una combinación entre las características propias de cada figura y las distancias de Hausdorff, tratando de obtener mejores resultados al juntar la distancia de Hausdorff con la compacidad discreta de cada objeto, dando como resultado la siguiente relación:

$$\left|C_{D_{obj\,1}} - C_{D_{obj\,2}}\right| * \mathrm{DH}(obj\,1, obj\,2), \tag{20}$$

donde $obj\,1$,$obj\,2$ son los dos objetos que se están comparando, $\mathrm{DH}()$ es la distancia de Hausdorff. Los resultados son presentados en la Tabla 6.9. Los resultados no fueron los esperados, dando mejores resultados únicamente la distancia de Hausdorff entre objetos, debido a que los datos están más dispersos y no fue posible determinar un umbral. Las gráficas de comparación de entre cada objeto se pueden encontrar en los anexos del presente documento.

Tabla 6.9. Razón de la compacidad discreta y distancia de Hausdorff en (9).

	Brain	Bunny	Hippo	Lion	Pear	Santa	Cow	Torus	orange	banana	apple	pumpkin	nut	horse
Brain	0.0000	0.1249	0.0539	1.2994	0.5772	1.7084	0.5080	1.0755	0.5449	0.3946	0.5317	0.3429	0.1374	0.9393
Bunny	0.1249	0.0000	0.2428	1.2673	0.3276	3.2866	0.7026	1.2187	0.3889	0.0702	0.3367	0.1710	0.0867	1.2101
Hippo	0.0539	0.2428	0.0000	0.9276	0.5072	1.6718	0.3307	0.4380	0.5173	0.2312	0.5458	0.3205	0.0553	1.8322
Lion	1.2994	1.2673	0.9276	0.0000	1.7297	0.0433	0.9606	0.7576	1.6656	1.7572	1.5448	1.7459	1.0644	0.5250
Pear	0.5772	0.3276	0.5072	1.7297	0.0000	2.5577	1.2515	1.4702	0.0657	0.3054	0.0640	0.2370	0.3267	1.6359
Santa	1.7084	3.2866	1.6718	0.0433	2.5577	0.0000	2.3969	1.9489	2.1622	3.1474	2.0768	2.2585	1.9276	1.1224
Cow	0.5080	0.7026	0.3307	0.9606	1.2515	2.3969	0.0000	0.1657	0.7766	0.5129	0.7245	0.6179	0.2975	1.0874
Torus	1.0755	1.2187	0.4380	0.7576	1.4702	1.9489	0.1657	0.0000	0.9784	0.8942	0.9092	1.0958	0.3102	0.6026
orange	0.5449	0.3889	0.5173	1.6656	0.0657	2.1622	0.7766	0.9784	0.0000	0.6641	0.0005	0.3477	0.6977	1.5181
banana	0.3946	0.0702	0.2312	1.7572	0.3054	3.1474	0.5129	0.8942	0.6641	0.0000	0.2902	0.1264	0.1036	1.8858
apple	0.5317	0.3367	0.5458	1.5448	0.0640	2.0768	0.7245	0.9092	0.0005	0.2902	0.0000	0.3283	0.6942	1.4458
pumpkin	0.3429	0.1710	0.3205	1.7459	0.2370	2.2585	0.6179	1.0958	0.3477	0.1264	0.3283	0.0000	0.1993	1.8453
nut	0.1374	0.0867	0.0553	1.0644	0.3267	1.9276	0.2975	0.3102	0.6977	0.1036	0.6942	0.1993	0.0000	0.9461
horse	0.9393	1.2101	1.8322	0.5250	1.6359	1.1224	1.0874	0.6026	1.5181	1.8858	1.4458	1.8453	0.9461	0.0000

7. Conclusiones

Se presenta un método para la representación de objetos 3D a través de un poliedro, con la peculiaridad de representar el objeto 3D como un conjunto de capas de voxeles, brindando grandes ventajas de ver cada corte como un objeto 2D, y así implementar un método existente para encontrar los puntos dominantes.

Tiene las contribuciones de:

- crear un poliedro a partir de una nube de puntos,
- la nueva adaptación de ISE y, por lo tanto, CR y FOM,
- definición de un criterio de error: $n\mathrm{F}$, y
- permite hacer agrupación y distinguir un objeto a partir del poliedro creado.

El método presentado permite ajustar el nivel de detalle que se requiere en el poliedro dependiendo del fin que se va a tener con el poliedro, es decir, una alta o baja concentración de puntos dominantes, únicamente manipulando la distancia entre capas.

La principal limitación del método es la pérdida de información que se genera al hacer los saltos entre capas, dando posibilidad de que información muy relevante del objeto original localizada en esas partes ya no se recuperen al crear el poliedro, es por eso que se identifican las siguientes áreas de oportunidad para mejorar el método:

- creación de capas, desarrollar un algoritmo que permita identificar las capas con mayor información, permitiendo una distancia entre capas N variable dentro del mismo objeto,
- unión de capas, implementar un técnica para mejorar la unión entre capas para dar solución al problema con las extremidades que no están alineadas, por ejemplo, con la triangulación de Delaunay.

La técnica empleada puede ser utilizada para trabajo futuro en reconocimiento por medio de aprendizaje supervisado dentro de una base de datos de objetos.

8. Anexos

Las gráficas de comparación de entre cada objeto con los resultados de la tabla 6.9 con la ec. (20) son presentados en las Fig. 8.1 – 8.14 en donde no es posible determinar un umbral de parecido entre los objetos.

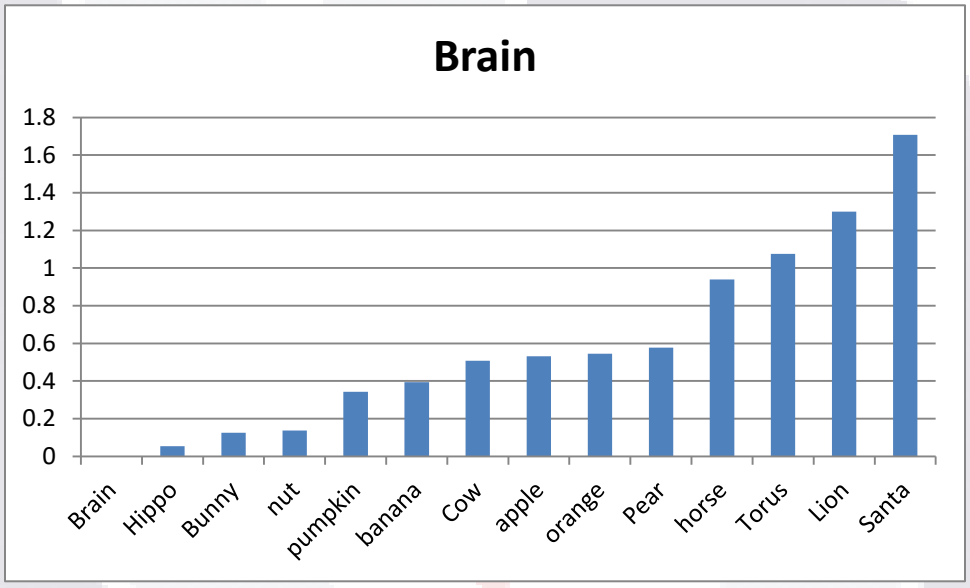


Fig. 8.1 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Brain

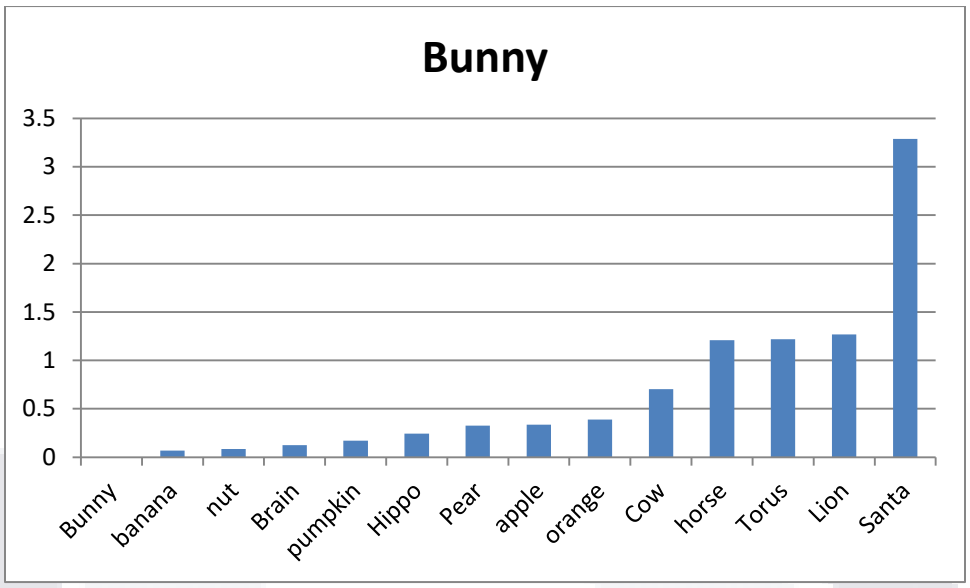


Fig. 8.2 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Bunny

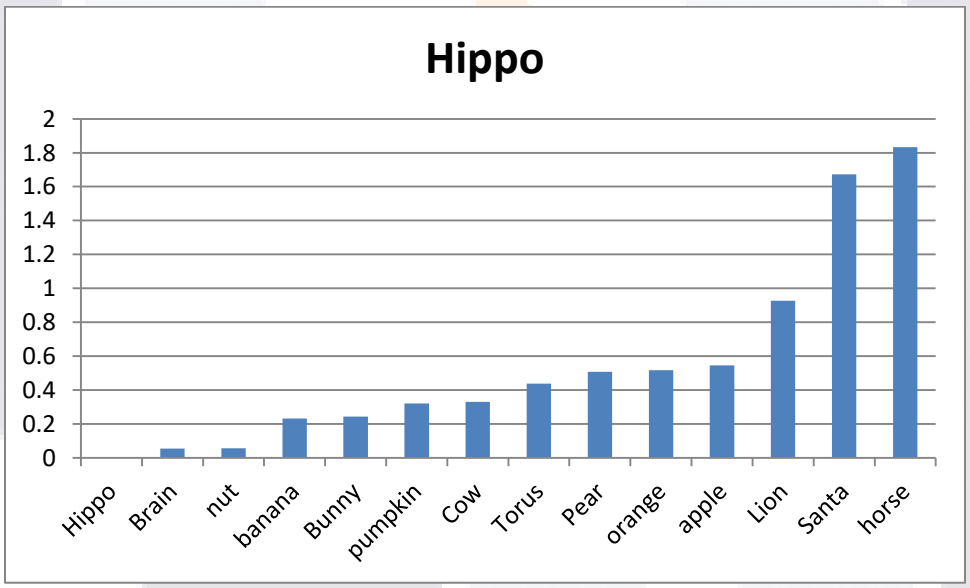


Fig. 8.3 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Hippo

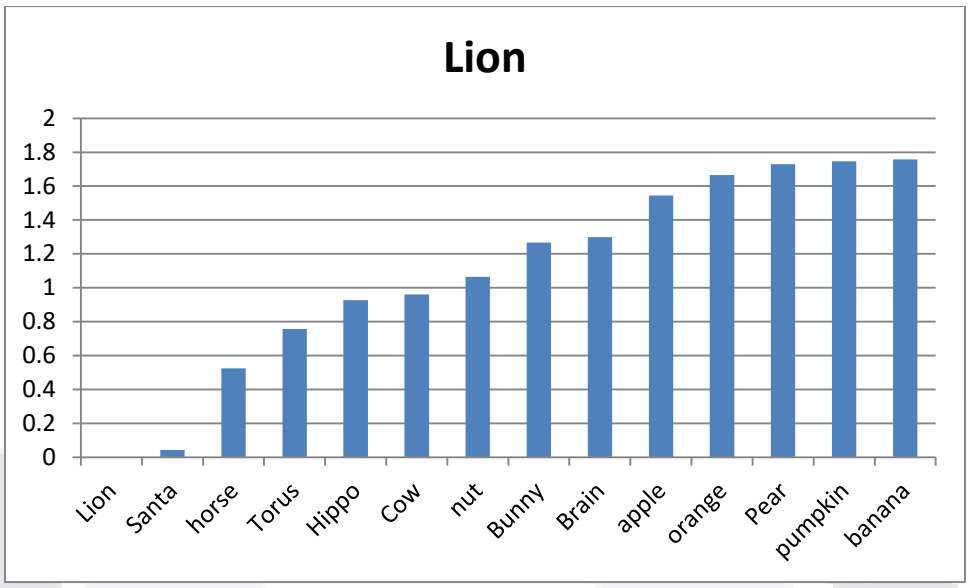


Fig. 8.4 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Lion

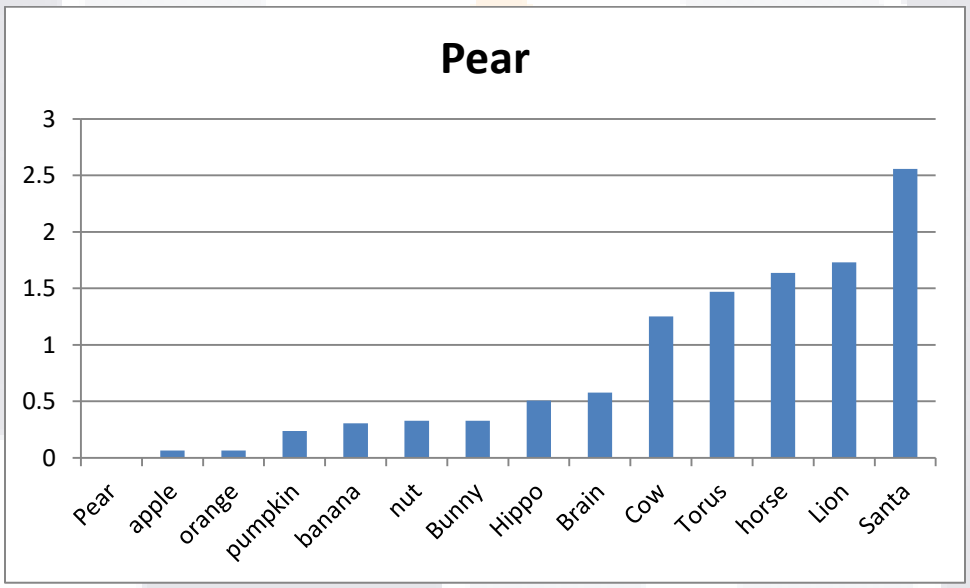


Fig. 8.5 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Pear

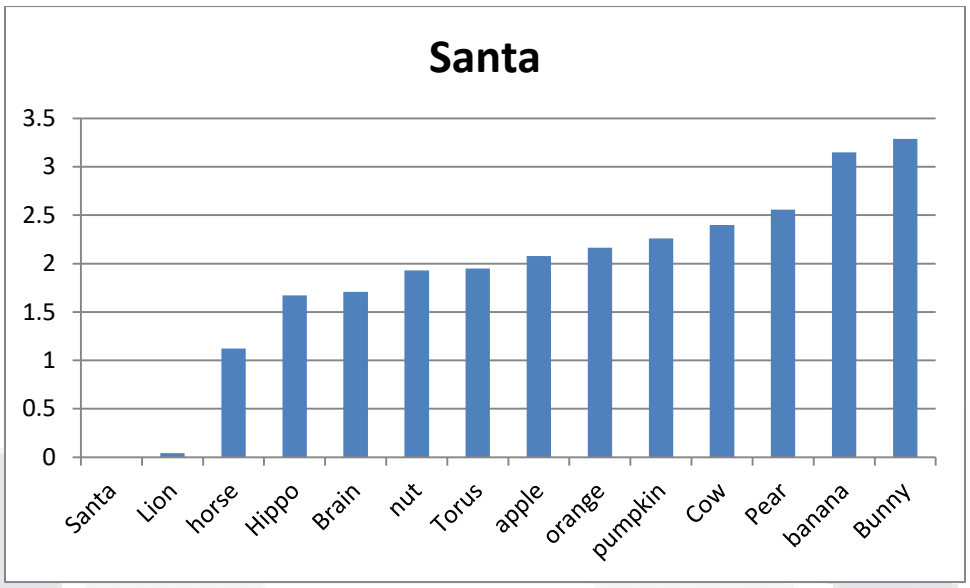


Fig. 8.6 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Santa

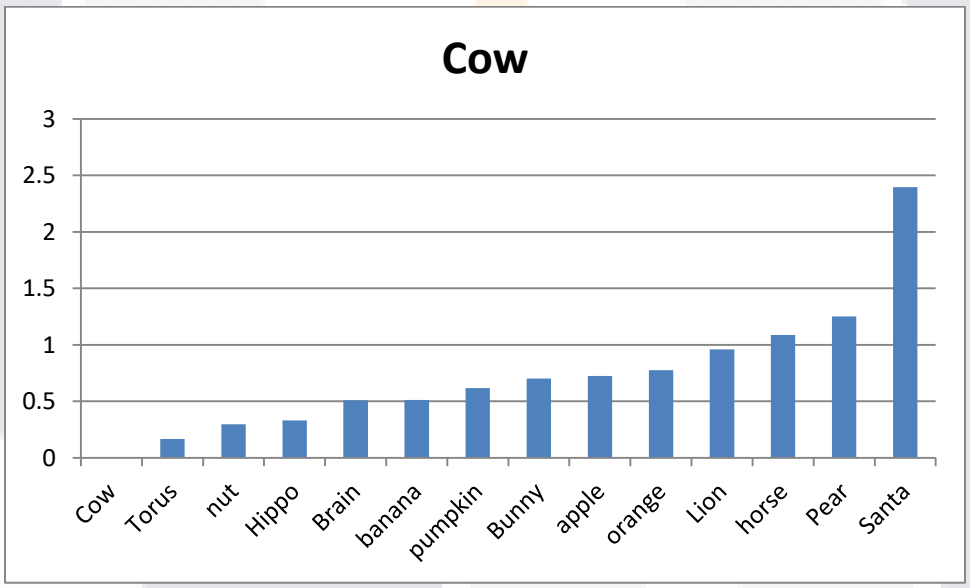


Fig. 8.7 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Cow

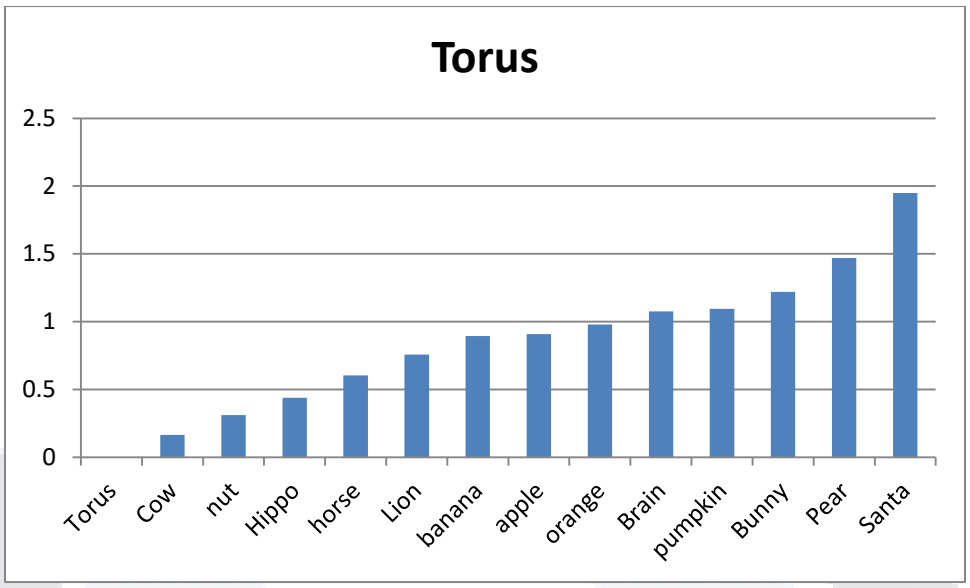


Fig. 8.8 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Torus

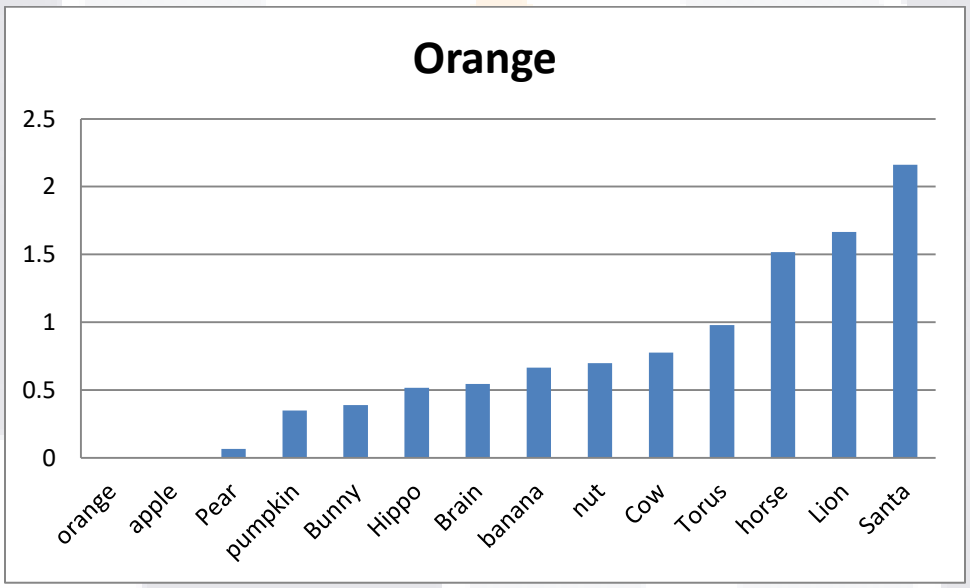


Fig. 8.9 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Orange

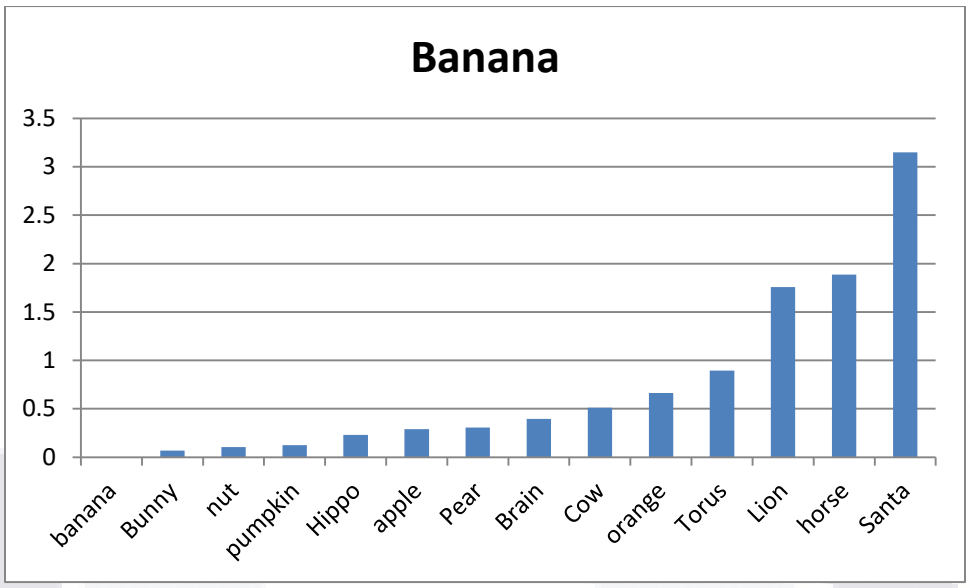


Fig. 8.10 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Banana

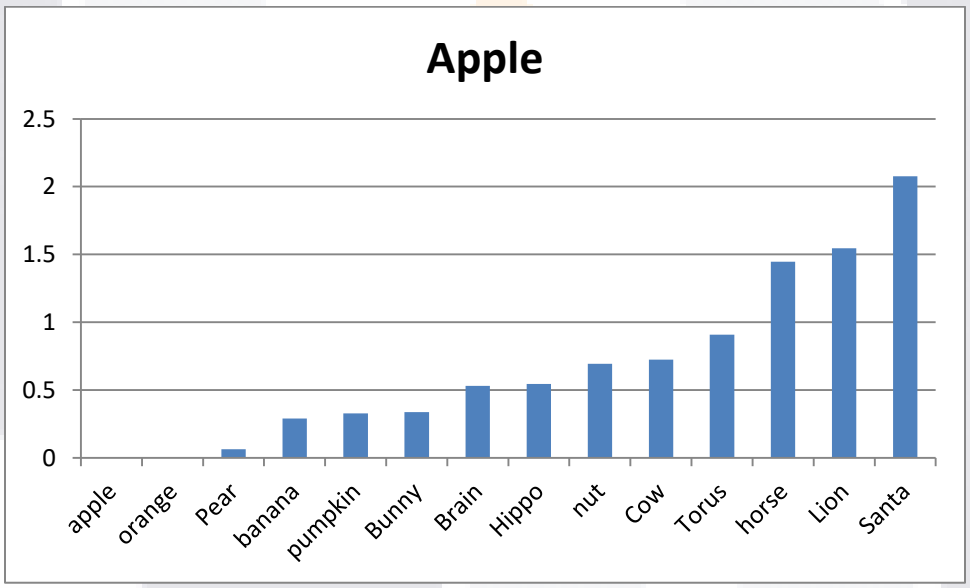


Fig. 8.11 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Apple

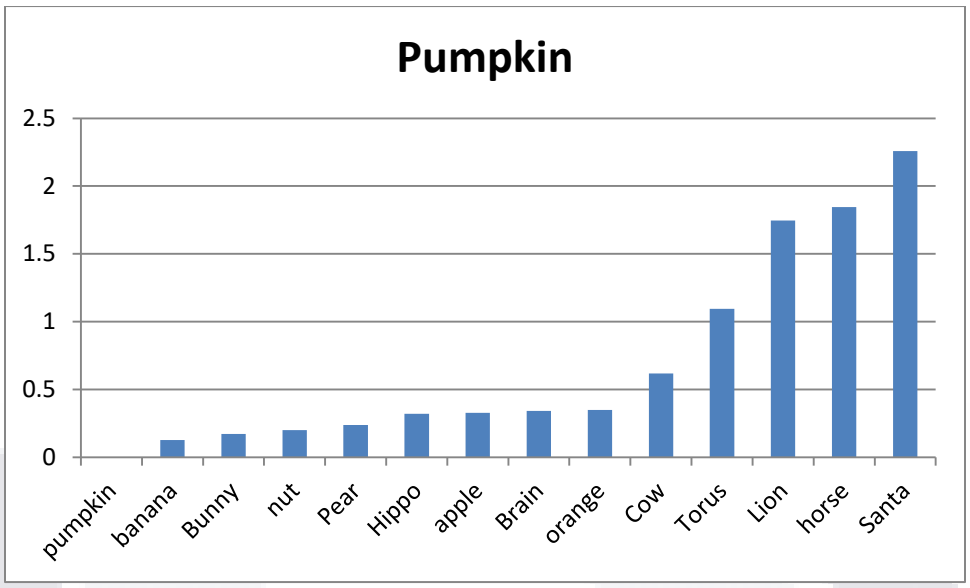


Fig. 8.12 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Pumpkin

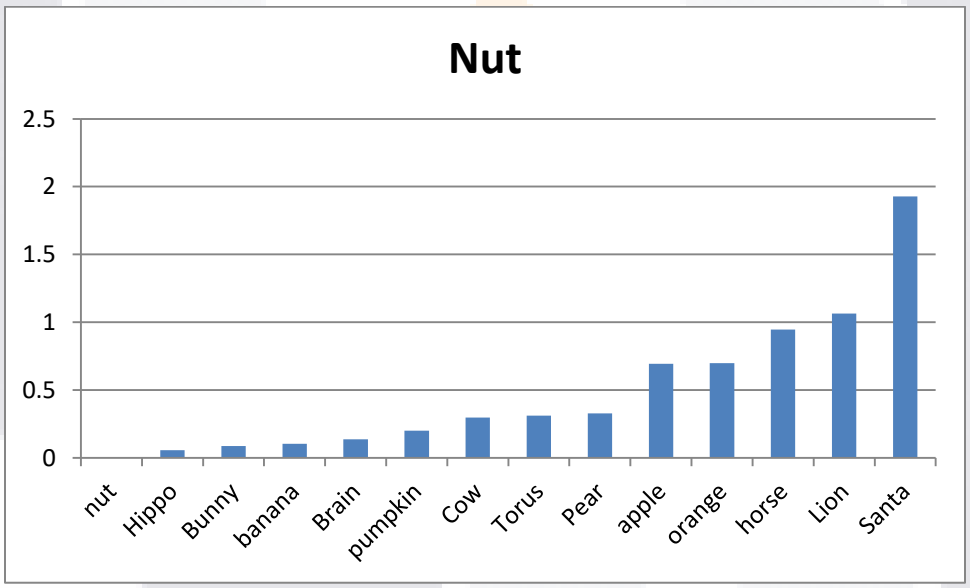


Fig. 8.13 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Nut

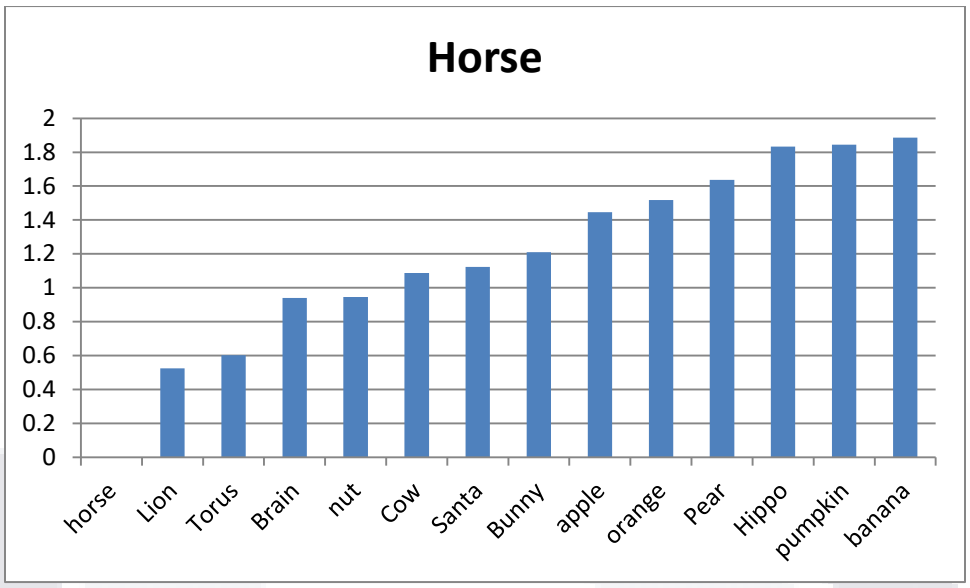


Fig. 8.14 Comparación de la razón de la compacidad discreta y distancia de Hausdorff de Horse

Referencias

[1] H. Blum, A transformation for extracting new descriptors of the shape, in: W. Whaten-Dunn (Ed.), Models of the Perception of Speech and Visual Forms, MIT Press, Cambridge, 1967, pp. 362–380.

[2] J.J. Koenderink, A.J. van Doorn, The internal representation of solid shape with respect to vision, Biological Cybernetics 32 (1979) 211–216.

[3] B. Kartikeyan, A. Sarkar, Shape description by time series, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (1989) 977–984.

[4] R. Kashyap, R. Dhellapa, Stochastic models for closed boundary analysis: representation and reconstruction, IEEE Transactions on Information Theory 27 (1981) 627–637.

[5] T. Kaneko, M. Okudaira, Encoding of arbitrary curves based on the chain code representation, IEEE Transactions on Communications 33 (1985) 697–707.

[6] J. Koplowitz, On the performance of chain codes for quantization of the line drawings, IEEE Transactions on Pattern Analysis and Machine Intelligence 3 (1981) 180–185.

[7] D. Neuhoff, K. Castor, A rate and distortion analysis of chain codes for line drawings, IEEE Transactions on Information Theory 31 (1985) 53–68.

[8] J. Saghri, H. Freeman, Analysis of the precision of generalized chain codes for the representation of planar curves, IEEE Transactions on Pattern Analysis and Machine Intelligence 3 (1981) 533–539.

[9] C.P. Chau, W.C. Siu, New nonparametric dominant point detection algorithm, Vision, Image and Signal Processing 148 (5) (2001) 363–374

[10] T.M. Cronin, A boundary concavity code to support dominant points detection, Pattern Recognition Letters 20 (1999) 617–634.

[11] M. Marji, P. Siy, A new algorithm for dominant points detection and polygonization of digital curves, Pattern Recognition 36 (2003) 2239–2251.

[12] H. Park, J.H. Lee, Error-bounded B-spline curve approximation based on dominant point selection, IEEE International Conference on Computer, Imaging and Vision: New Trends (2005) 437–446.

[13] F. Attneave, Some informational aspects of visual perception, Psychological Review 61 (1954) 183–193.

[14] A. Masood, Shaiq A. Haq, A novel approach to polygonal approximation of digital curves, Journal of Visual Communication and Image Representation, Volume 18, Issue 3, (2007), 264-274.

[15] B.K. Ray, K.S. Ray, Determination of optimal polygon from digital curve using L1 norm, Pattern Recognition 26 (1993) 505–509.

[16] Y. Kurozumi, W.A. Davis, Polygonal approximation by the minimax method, Computer Graphics and Image Processing 19 (1982) 248–264.

[17] C. Teh, R. Chin, On the detection of dominant points on digital curves, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1989) 859–873.

[18] U. Ramer, An iterative procedure for the polygonal approximation of plane curves, Computer Graphics and Image Processing 1 (1972) 244–256.

[19] A. Held, K. Abe, C. Arcelli, Towards a hierarchical contour description via dominant point detection, IEEE Transactions on Systems, Man and Cybernetics 24 (6) (1994) 942–949.

[20] J.G. Dunham, Optimum uniform piecewise linear approximation of planar curves, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986) 67–75.

[21] Y. Sato, Piecewise linear approximation of plane curves by perimeter optimization, Pattern Recognition 25 (1992) 1535–1543.

[22] E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, Addison, Reading, MA, 1989.

[23] .C. Huang, Y.N. Sun, Polygonal approximation using genetic algorithms, Pattern Recognition 32 (1999) 1409–1420.

[24] P.Y. Yin, Genetic algorithms for polygonal approximation of digital curves, International Journal of Pattern Recognition and Artificial Intelligence 13 (1999) 1–22.

[25] N.L. Fernández-García, L. Del-Moral Martínez, A. Carmona-Poyato, F.J. Madrid Cuevas, R. Medina-Carnicer, A new thresholding approach for automatic generation of polygonal approximations, Journal of Visual Communication and Image Representation, Volume 35, (2016), 155-168.

[26] Lafarge, T., Pateiro-López, B., Possolo, A., Dunkers, Joy P. (2014). R implementation of a polyhedral approximation to a 3D set of points using the a-shape. Journal of Statistical Software, 56, 4.

[27] Sánchez-Cruz H., Tapia-Dueñas O.A., Cuevas F., Polygonal Approximation Using a Multiresolution Method and a Context-free Grammar. In: Carrasco-Ochoa J., Mart´ınez-Trinidad J., Olvera-López J., Salas J. (eds) Pattern Recognition. MCPR 2019. Lecture Notes in Computer Science, (2019), vol 11524. Springer, Cham.

[28] Turk, G. (1994) The PLY Polygon File Format. The Board of Trustees of The Leland Stanford Junior University

[29] Paulbourke.net. 2021. Object Files (.obj)

[30] tecnonautas.net. 2018. ¿Qué es un archivo STP y cómo se abre uno?

[31] Festa. P. Borland, J. (2005) Is a 3D web more than just empty promises?, CNET.

[32] Noguera, J., Rueda A., (2008) Voxelización de sólidos mediante instanciación de geometría, Barcelona, The Eurographics Association

[33] Magaña Z., J. B.; Atoche E., J. R.; Molina C., J. C; Blanco V., M.; Pérez C., E. (2017) Estimación de la Distancia a un Objeto con Visión Computacional Ingeniería, vol. 21, núm. 2, pp. 31-40 Universidad Autónoma de Yucatán, Mérida, México.

[34] Ochoa Castillo, C., Forero Vega, L. (2016) Metrica de Hausdorff en el Ambiente Difuso, INGENIERIA, vol. 21, no. 3, pp. 346-359.

[35] Jordan, M.I., Mitchell, T.M., (2015). Machine learning: Trends, perspectives, and prospects. Science

[36] Martin, John C. (2010). Introduction to Languages and the Theory of Computation. 4th ed. Mc Graw Gill.

[37] D. Sarkar, A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves, Pattern Recognition. Letters 14 (1993) 959–964.

[38] Montero, R. S., & Bribiesca, E. (2009). State of the art of compactness and circularity measures. In International mathematical forum (Vol. 4, No. 27, pp. 1305-1335).