



**UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES**

CENTRO DE CIENCIAS BÁSICAS

DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL

TESIS

**COMPARACIÓN DE ESQUELETOS DE IMÁGENES DE ANIMALES EN 3D, A
TRAVÉS DEL PROBLEMA DEL COSTO MÍNIMO**

PRESENTA

José Alfredo Alonso Esquivel

**PARA OBTENER EL GRADO DE MAestrÍA EN CIENCIAS, CON OPCIÓN A LA
COMPUTACIÓN, MATEMÁTICAS APLICADAS**

TUTOR(ES)

Dr. Hermilo Sánchez Cruz: Director

Dr. Ángel Eduardo Muñoz Zavala: Director

COMITE TUTORAL

Dra. Eunice Esther Ponce de León: Asesor

Dr. Francisco Álvarez Rodríguez: Asesor

Aguascalientes, Ags., 6 Febrero del 2014



UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES



ANIVERSARIO
UAA

Centro de Ciencias Básicas

L.I. JOSÉ ALFREDO ALONSO ESQUIVEL
ALUMNO (A) DE LA MAESTRÍA EN CIENCIAS
CON OPCIÓN A LA COMPUTACIÓN,
P R E S E N T E .

Estimado (a) alumno (a) Alonso:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido el voto aprobatorio de su tutor de tesis titulada: **"COMPARACIÓN DE ESQUELETOS DE IMÁGENES DE ANIMALES EN 3D, A TRAVÉS DEL PROBLEMA DEL COSTO MÍNIMO"**, hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

ATENTAMENTE
Aguascalientes, Ags., 8 de enero de 2014
"SE LUMEN PROFERRE"
EL DECANO

M. en C. JOSÉ DE JESÚS RUIZ GALLEGOS



- c.c.cp.- Interesado
- c.c.p.- Secretaría de Investigación y Postgrado
- c.c.p.- Jefatura del Depto. de Apoyo al Posgrado
- c.c.p.- Jefatura del Depto. de Ciencias Computacionales
- c.c.p.- Consejero Académico
- c.c.p.- Minuta Secretario Técnico
- JJRG,mjda



Aguascalientes, Ags, Enero de 2014

M. en C. José de Jesús Ruíz Gallegos.
Decano del Centro de Ciencias Básicas.
Universidad Autónoma de Aguascalientes.

PRESENTE.

Por este conducto hago de su conocimiento que el **L.I. José Alfredo Alonso Esquivel**, egresado de la Maestría en Ciencias con opción a la computación, Matemáticas aplicadas de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: "**COMPARACIÓN DE ESQUELETOS DE IMÁGENES DE ANIMALES EN 3D, A TRAVÉS DEL PROBLEMA DEL COSTO MÍNIMO**", por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

ATENTAMENTE



Dr. Hermilo Sánchez Cruz
Director de Tesis

Aguascalientes, Ags, Enero de 2014

M. en C. José de Jesús Ruíz Gallegos.
Decano del Centro de Ciencias Básicas.
Universidad Autónoma de Aguascalientes.

PRESENTE.

Por este conducto hago de su conocimiento que el **L.I. José Alfredo Alonso Esquivel**, egresado de la Maestría en Ciencias con opción a la computación, Matemáticas aplicadas de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: **“COMPARACIÓN DE ESQUELETOS DE IMÁGENES DE ANIMALES EN 3D, A TRAVÉS DEL PROBLEMA DEL COSTO MÍNIMO”**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

ATENTAMENTE



Dr. Ángel Eduardo Muñoz Zavala
Director de Tesis

Aguascalientes, Ags, Enero de 2014.

M. en C. José de Jesús Ruíz Gallegos.
Decano del Centro de Ciencias Básicas.
Universidad Autónoma de Aguascalientes.

PRESENTE.

Por este conducto hago de su conocimiento que el **L.I. José Alfredo Alonso Esquivel**, egresado de la Maestría en Ciencias de la Computación del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: "**COMPARACIÓN DE IMÁGENES EN 3D DE ANIMALES A TRAVÉS DEL PROBLEMA DE COSTO MÍNIMO EN GRAFOS BIPARTITOS**", por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

ATENTAMENTE



Dra. Eunice Esther Ponce de León
Asesor de Tesis

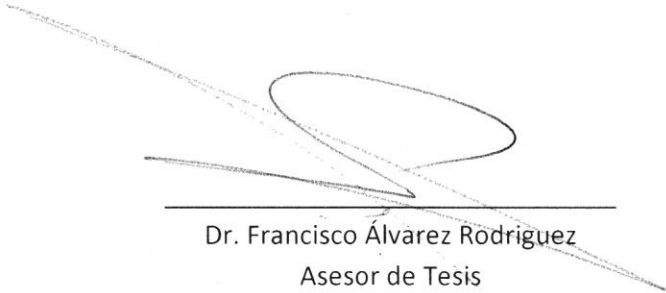
Aguascalientes, Ags, Enero de 2014

M. en C. José de Jesús Ruíz Gallegos.
Decano del Centro de Ciencias Básicas.
Universidad Autónoma de Aguascalientes.

PRESENTE.

Por este conducto hago de su conocimiento que el **L.I. José Alfredo Alonso Esquivel**, egresado de la Maestría en Ciencias con opción a la computación, Matemáticas aplicadas de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: "**COMPARACIÓN DE ESQUELETOS DE IMÁGENES DE ANIMALES EN 3D, A TRAVÉS DEL PROBLEMA DEL COSTO MÍNIMO**", por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

ATENTAMENTE

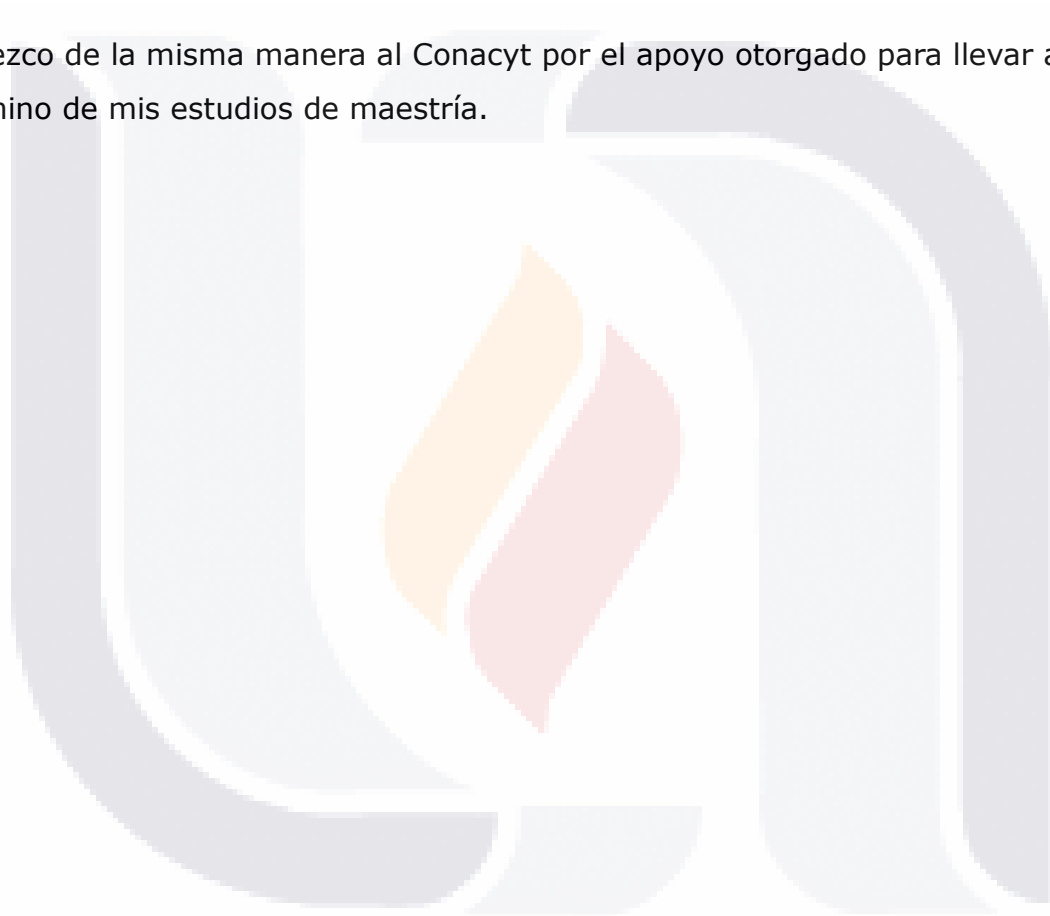


Dr. Francisco Álvarez Rodríguez
Asesor de Tesis

AGRADECIMIENTOS

Quisiera agradecer a mis directores de Tesis, Dr. Hermilo Sánchez Cruz y al Dr. Angel Muñoz por su apoyo y el tiempo que fue dedicado para la elaboración de este documento, y en general por su guía durante todo el tiempo que duro mi estudio. Agradezco también al departamento de posgrados por su apoyo, orientación y seguimiento a través de mi proceso de titulación.

Agradezco de la misma manera al Conacyt por el apoyo otorgado para llevar a cabo el termino de mis estudios de maestría.

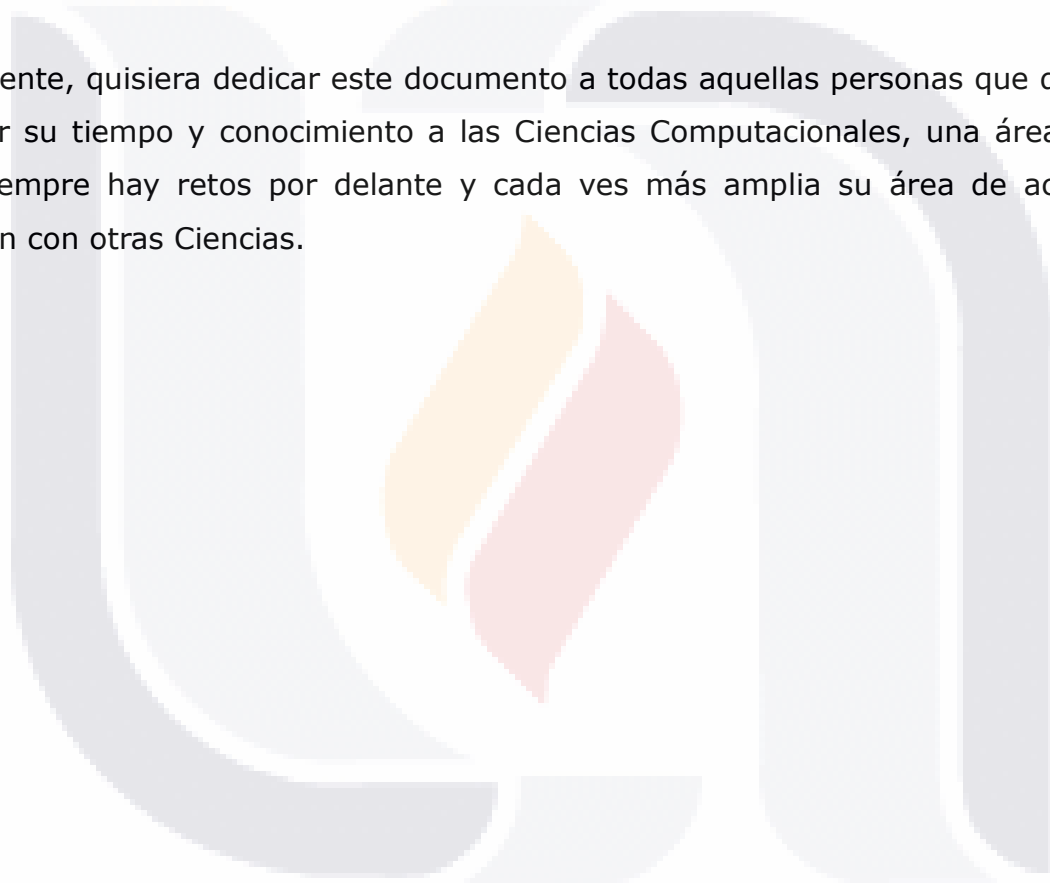


DEDICATORIAS

Este documento esta dedicado a mi esposa, que me apoyo durante esas largas noches. Sus palabras fueron de gran ayuda en los momentos difíciles... gracias Nanis =)

La culminación de la maestría se la dedico también a mi padre y a mi madre... Espero lograr todo lo que todos ellos siempre quisieron para mi.

Finalmente, quisiera dedicar este documento a todas aquellas personas que quieran aportar su tiempo y conocimiento a las Ciencias Computacionales, una área en la que siempre hay retos por delante y cada ves más amplia su área de acción y relación con otras Ciencias.



ÍNDICE GENERAL

ÍNDICE GENERAL 1

ÍNDICE DE TABLAS..... 3

ÍNDICE DE FIGURAS..... 4

RESUMEN 5

ABSTRACT 6

INTRODUCCION 7

1. FORMULACIÓN DE LA INVESTIGACIÓN 11

 1.1 PROBLEMA DE INVESTIGACIÓN 11

 1.2 PREGUNTA Y OBJETIVO DE INVESTIGACIÓN 11

 1.3 HIPÓTESIS DE INVESTIGACIÓN..... 13

2. MARCO TEÓRICO FUNDAMENTAL 15

 2.1 FUNDAMENTOS TEÓRICOS BASE 15

 2.2 DESCRIPCIÓN DE PRINCIPALES ESTUDIOS RELACIONADOS 21

 2.3 ANÁLISIS DE CONTRIBUCIONES Y LIMITACIONES DE PRINCIPALES ESTUDIOS RELACIONADOS 22

 2.4 MODELO DE INVESTIGACIÓN 23

 2.5 DESCRIPCIÓN DE CONSTRUCTOS, VARIABLES OPERACIONALES Y ESCALAS 23

3. METODOLOGÍA DE INVESTIGACIÓN 26

 3.1 METODOLOGÍA Y TIPO DE DISEÑO DE INVESTIGACIÓN 26

 3.2 POBLACIÓN, MUESTRA Y UNIDAD DE ESTUDIO 26

 3.3 PLAN DE RECOLECCIÓN DE DATOS 31

3.4 PROCEDIMIENTOS DE ANÁLISIS DE DATOS Y VALIDACIÓN32

3.5 EJEMPLO DE APLICACIÓN DE LA METODOLOGIA PROPUESTA48

4. EXPERIMENTOS Y RESULTADOS53

4.1 CONTRIBUCIONES AL CONOCIMIENTO53

4.2 CONTRIBUCIONES A LA PRÁCTICA.....53

4.3 LIMITACIONES CLAVES.....54

4.4 EXPERIMENTOS56

5. DISCUSION DE RESULTADOS59

5.1 RESULTADOS.....59

5.2 MEDIDA DE SIMILITUD.....62

5.3 DEFINICIÓN DE UMBRAL PARA IMÁGENES CANINAS62

CONCLUSIONES67

GLOSARIO70

BIBLIOGRAFÍA72

ANEXOS76

ÍNDICE DE TABLAS

Tabla 1	Tabla de masas	18
Tabla 2	Conos: voxeles, esqueletos, masa y centro de masa	28
Tabla 3	Figuras caninas: voxeles, esqueletos, masa y centro de masa	30
Tabla 4	Figuras diversas: poliedros, voxeles y esqueletos	31
Tabla 5	Voxeles positivos y negativos	57
Tabla 6	Costo del algoritmo Húngaro y tiempos de proceso en figuras caninas	58
Tabla 7	Similitud entre imágenes caninas	61
Tabla 8	Índice CM en figuras caninas	64
Tabla 9	Índice CM en figuras diversas	65



ÍNDICE DE FIGURAS

Figura 1	Grafo Bipartito de conjuntos U, V	21
Figura 2	Imagen de un cono	32
Figura 3	Imagen voxelizada de un cono	34
Figura 4	Imagen esqueletonizada de un cono	34
Figura 5	Vectores propios/ejes principales	40
Figura 6	Ejes principales de dos imágenes	40
Figura 7	Rotaciones con ángulos de Euler	41
Figura 8	Ángulos de Euler	42
Figura 9	Distancia Euclidiana entre dos puntos	45
Figura 10	Grafo Bipartito	47
Figura 11	Estructura de directorios	48
Figura 12	Cono poliédrico y voxelizado	50
Figura 13	Cono en volumen y esqueleto	51

RESUMEN

Las áreas de clasificación de especímenes, basados en imágenes, es un área poco explotada en el México actual. La utilización de herramientas de Inteligencia Artificial en estas áreas ha venido a maximizar el avance en tema de ciencias computacionales para bien. En este documento, se explora la idea de convertir las imágenes tridimensionales en datos discretos, específicamente en grafos. Con esta información, y al sobreponer dos imágenes, se pueden generar matrices de distancias entre cada uno de los puntos que no tienen en común las dos imágenes.

Nuestra adopción de las diferencias de esqueletos de imágenes en tres dimensiones, convertidas en grafos bipartitos, permite utilizar métodos de optimización combinatoria; es el caso del algoritmo Húngaro o Kuhn-Munkres. Se propone utilizar este algoritmo para encontrar el costo mínimo a problemas de asignación, podemos aplicar esta metodología y a través de la definición de un umbral de similitud confiable para poder decidir si las dos imágenes se parecen entre sí.

La implementación de la metodología propuesta se puede expandir a cualquier otra área que requiera la comparación y procesamiento de imágenes en dos y tres dimensiones.

ABSTRACT

Specimens classification is a barely explored area in actual Mexico. Usage of artificial intelligence methodologies have helped these areas in the last years. On this document, the idea to treat three dimensional images is researched. To convert images into distance data, specifically into graphs. With this information, and by overlapping two images, we can generate distance matrices between each set of points on a three dimensional euclidian space between the two images.

Our adoption of skeleton differences on three dimensional images, verted into weighed bipartite graphs, allows the usage of optimizational methodologies: in this special case, the Kuhn-Munkres algorithm. We propose to use this algorithm to find the minimum value to assignation problems. We can apply this methodology beyond and find a threshold value, by calculating a measurement value to tell if two images belong to the same category.

This implementation, can be extrapolated to other areas where three dimensional (and even two dimensional) image processing is required.

INTRODUCCIÓN

DESCRIPCIÓN DEL CONTEXTO DE LA SITUACIÓN PROBLEMÁTICA DE INVESTIGACIÓN

El área de reconocimiento de patrones ha avanzado ampliamente en los últimos años y sus aplicaciones son muy variadas, desde el reconocimiento de imágenes faciales en las cámaras fotográficas digitales, videojuegos, búsquedas de imágenes en internet, reconstrucción de entornos digitales en 3D hacia su contraparte en el mundo real, reconocimiento de imágenes en 2D y 3D; sin importar la posición, la distancia, el tamaño, la orientación, la calidad de la imagen y/o lo detallado que se vea el objeto de entre muchos otros que estén presentes en la imagen.

Aunque este tópico se está comenzando a estudiar cada vez más, en México no hay suficientes investigaciones enfocadas a la visión Computacional, mucho menos enfocadas a atacar el reconocimiento de imágenes en tres dimensiones.

Instituciones dedican departamentos enteros a la obtención de imágenes satelitales, pero muchas veces no se realiza ningún otro análisis a fondo de dichas imágenes; más que el de realizar mapas y mostrar información encima de ellos, como lo hace el INEGI.

Algunas de las investigaciones que se han realizado en los últimos años, han atacado el problema del reconocimiento y clasificación de diversos elementos de la naturaleza con métodos del área de Inteligencia Artificial como redes neuronales, para reconocer las hojas de las plantas [1]; utilizando un sistema experto para encontrar elementos clave [2], a través de taxónomos expertos entre otros.

El mayor problema radica en que los métodos que existen, no son sencillos de aplicar y generalmente requieren alto conocimiento en áreas taxonómicas o biológicas, equipo de cómputo especializado, así como gran espacio de

almacenamiento. En otras palabras, no hay mucha investigación realizada en torno al reconocimiento y clasificación de imágenes, basándose en técnicas obtenidas del área de Visión Computacional y/o utilizando métodos matemáticos y estadísticos.

Los esfuerzos que se han realizado en el área, al ser llevados a cabo por expertos de cada área, limita el campo de acción de otras disciplinas, por lo que se requiere implementar estas técnicas para poder ampliar el reconocimiento y clasificación de figuras en tres dimensiones; así como permitir que se puedan realizar análisis sin requerir de equipos profesionales.

RELEVANCIA Y JUSTIFICACIÓN DE LA INVESTIGACIÓN

Uno de los esfuerzos que se han hecho recientemente para abordar esta área es el de la metodología DELTA [2], la cual permite una clasificación de especies de plantas, identificando características claves y digitalizando toda esta información, para su posterior uso. Aún con esta investigación, no existe un canal intermedio para que esta información sea accesible a los investigadores del área.

La falta de la aplicación de métodos de visión computacional que permita realizar la correcta identificación y clasificación de imágenes en tres dimensiones, provoca que muchas especies permanezcan desconocidas en la actualidad, y los estudios que se realizan son en algunos casos inadecuados.

El uso de la inteligencia artificial enfocada en atacar este problema, provee una solución integral que permitiría el procesamiento de la información, en tiempo real, sin ser excesivamente costoso en tiempo de cómputo, ni en herramientas profesionales. Permite también mejorar la velocidad de procesamiento y comparación entre dos imágenes en tres dimensiones.

Al poder identificar y clasificar las imágenes en tres dimensiones, se tendrá la información accesible a más personas, no necesariamente expertos en la

materia, lo que contribuye a continuar con investigaciones que usen esta metodología como base para poder clasificar imágenes en tres dimensiones, con un bajo costo computacional.

De la misma manera, se expande el uso que se le puede dar a la información generada con esta investigación, y su campo de uso se puede extrapolar a más áreas sin necesidad de tanto re trabajo, ni de equipos profesionales o costosos.

DESCRIPCIÓN GENERAL DEL ENFOQUE DE INVESTIGACIÓN

La metodología de investigación que se usará en este documento es de tipo aplicada. Se intentará realizar el reconocimiento y comparación de imágenes, usando métodos heurísticos para la obtención del costo mínimo, para encontrar el trabajo o energía que se requiere para transformar una imagen en otra, basándonos en grafos bipartitos. Estos grafos serán construidos utilizando los esqueletos de imágenes de tres dimensiones. Estas imágenes, serán procesadas con diversas metodologías hasta obtener el formato requerido para poder aplicar el método del costo mínimo e identificar los porcentajes de similitud o diferencia de dos o más imágenes.

El algoritmo de optimización que utilizaremos es el algoritmo Húngaro; este se usará para encontrar el costo mínimo. Usando este algoritmo, se podría conseguir reducir costos, niveles de experiencia necesaria, y tiempo de cómputo al realizar cálculos con los esqueletos de las imágenes.

Es prácticamente imposible realizar un estudio con toda la población de imágenes existentes en la naturaleza; la inmensa variedad de imágenes que existen de una figura de animal es inmensa. Puede ser, por ejemplo, la imagen de un canino sentado, parado, parado sobre dos patas, corriendo, caminando, y un largo etcétera.

En este documento, se tomara solamente una muestra de imágenes de animales caninos y de figuras geométricas básicas (conos); se seleccionaron imágenes caninas, ya que es un animal muy recurrente en los modelos vectoriales previamente generados y disponibles en las bases de datos de imágenes que existen en la actualidad.



TESIS TESIS TESIS TESIS TESIS

1. FORMULACIÓN DE LA INVESTIGACIÓN

1.1 PROBLEMA DE INVESTIGACIÓN

Los esfuerzos que se han realizado en el área de visión computacional, aún siendo medianamente exitosos, no son nada óptimos y el trabajo es en su mayoría manual y por gente experta en materia; como ejemplo se tiene [1], en el cual la información fue obtenida y categorizada por expertos botánicos; aún cuando en ocasiones la clasificación se realice con la asistencia de computadoras, no existe una herramienta que permita categorizar la información. Si se descubre algún espécimen nuevo de árbol, es necesario realizar una búsqueda intensa manual o medianamente asistida por computadora, sobre la información previamente almacenada, haciendo este trabajo recurrente y exhaustivo.

Entrando más en materia, se tiene el problema de la comparación de imágenes 3D. Cuando las imágenes se tienen de un tamaño relativamente grande, al realizar una comparación el problema del grafo bipartito se vuelve intratable con los equipos computacionales actuales. Como ejemplo se puede observar una imagen de una resolución de $100*100*100$, esta imagen cuenta con 1,000,000 de voxels; y si se intenta comparar contra otra de la misma dimensión, al tratar de utilizar el algoritmo húngaro, se tendrían que hacer 1,000,000,000,000 de comparaciones.

1.2 PREGUNTA Y OBJETIVO DE INVESTIGACIÓN

Es factible diseñar y construir un artefacto conceptual y físico, para lograr:

- Procesar imágenes en 3D, y poder comparar dichas imágenes una contra otra, para poder obtener un nivel de similitud/diferencia.
- Utilizar métodos de Inteligencia Artificial para reducir costos y tiempos en operaciones de comparación.

- TESIS TESIS TESIS TESIS TESIS
- El uso de esqueletos para la comparación de dos imágenes, ayuda en el proceso.
 - Los ángulos de Euler ayudan en la orientación de los esqueletos.

Este documento pretende responder a estas interrogantes, con las siguientes restricciones:

- La población de imágenes que serán usadas como base, serán aquellas imágenes tridimensionales previamente digitalizadas, y obtenidas desde las bases de datos de imágenes 3D, como se discutirá en capítulos más adelante.
- Se usará el algoritmo Húngaro / Kuhn-Munkres para obtener el costo mínimo sobre la matriz de distancias que se tienen entre los voxeles positivos y negativos de los esqueletos, representados como grafos bipartitos.
- El método se validará utilizando los esqueletos de las imágenes, no se trabajara con objetos completos; el uso del algoritmo Húngaro nos limita en el tamaño de la matriz que podemos procesar.

Para procesar correctamente la información se necesitan establecer ciertos objetivos:

- Construir métodos de clasificación, tales como métodos heurísticos y de programación lineal, apoyándonos en obtención de características descriptoras de imágenes.
- Construir un grafo bipartito basado en las posiciones euclidianas en planos de cada voxel de la imagen tridimensional usada.
- Construir un espacio de patrones, con el fin de aplicar métodos de clasificación, en donde se tomen en cuenta los descriptores obtenidos, así como el tamaño, forma, color, textura y

características secundarias, como pueden ser localización geográfica, altitud y humedad.

- Utilizar los centros de masa de cada imagen, para orientar ambas sobre un solo punto central.
- Calcular los ejes principales de cada esqueleto, basándonos en los valores y vectores propios.
- Utilizar los ángulos de Euler para rotar las imágenes y sobreponer sus ejes principales.
- Realizar la comparación de dos o más imágenes, para encontrar el nivel de similitud que existe entre dichas imágenes, usando un grafo bipartito y los algoritmos o métodos mencionados anteriormente.

1.3 HIPÓTESIS DE INVESTIGACIÓN

Hipótesis 1: El uso del algoritmo Húngaro sobre grafos bipartitos (o bien, los voxeles positivos y negativos de dos imágenes), para resolver el problema del costo mínimo al comparar sus esqueletos de imágenes en 3D, ayuda a la obtención de un valor aproximado adecuado en la comparación de dos o más objetos.

Hipótesis 2: Los descriptores obtenidos de la imagen, ayudarán a la correcta clasificación de las imágenes.

Hipótesis 3: Las imágenes pueden ser orientadas usando los ángulos de Euler.

Hipótesis 4: Los tiempos de cómputo necesarios para la comparación de los esqueletos, son medibles y aceptables para procesar cualquier tipo de imágenes.

Hipótesis 5: El proceso de comparación de imágenes propuesto, se puede implementar en un programa computacional para su fácil ejecución.

Hipótesis 6: El método es invariante a rotación, escala y traslación de las imágenes.



2. MARCO TEÓRICO FUNDAMENTAL

2.1 FUNDAMENTOS TEÓRICOS BASE

En esta sección se describen conceptos básicos de inteligencia artificial y visión computacional, acerca del reconocimiento de patrones, meta heurísticas de optimización; morfología matemática de imágenes en tres dimensiones; las principales metodologías usadas en visión computacional, y las áreas en las que se aplican estos métodos.

Inicios del reconocimiento de patrones.

La Inteligencia Artificial es una ciencia e ingeniería que se encarga de transformar en código maquina el pensamiento humano racional; es decir, lograr que la maquina se comporte de manera parecida o casi idéntica a la racionalidad del ser humano [4].

Duda & Hart, hablaron por primera vez acerca del reconocimiento de patrones [35]. Mencionaron que el reconocimiento de patrones es que las maquinas puedan reconocer las regularidades que tengan algún significado preciso, dentro de los ambientes difusos.

Después de Duda & Hart, varios otros autores definieron el reconocimiento de patrones como una clasificación [36][37][38], categorización de problemas, estimación de densidades, medidas e inclusive como una herramienta para lograr toma de decisiones.

Clasificación de métodos para el reconocimiento de patrones

Los métodos que se han usado para el reconocimiento de patrones más importantes se clasifican en:

- a) Clustering de los datos

Mediante esta técnica, llamada Scale Invariant Feature Transform (SIFT), se transforman las imágenes en pequeños sub-vectores con características propias, que al unirlos, se logra identificar la imagen [39].

b) Lógica difusa

Esta área, se comenzó a usar a mediados de los 60's con la finalidad de utilizar la forma en la que la mente humana racionaliza la captura de imágenes, de manera "no exacta" [40].

c) Redes Neurales

Fue de las primeras áreas en emular o tratar de reproducir el modo de operar de la naturaleza. Aplica conceptos del área de biología. Es de las áreas más nobles, ya que no necesita de muchos datos para comenzar a operar [41].

d) Aprendizaje Maquina

El tema de aprendizaje maquina [42] aporta mucho a la escena, ya que permite ir entrenando el sistema experto en el reconocimiento de patrones para que las métricas puedan irse puliendo y especializando. Logrando una especificación que se acerca más con cada ciclo de aprendizaje, hasta lograr el reconocimiento perfecto de imágenes.

e) Análisis de imágenes a través de cámaras infrarrojas.

Esta área relativamente nueva, se ha impulsado gracias a los videojuegos. La tecnología actual utiliza cámaras y sensores infrarrojos para emitir y captar las emisiones infrarrojas, identificando la posición, distancia y tamaño de los objetos que se encuentren dentro del alcance de los sensores [32]. Un ejemplo de esta tecnología es el Kinect de Microsoft [57].

Este dispositivo es relativamente económico y de fácil configuración y adaptación para la captura y almacenamiento de imágenes del mundo real, en representaciones digitales con formatos estándar de dichas imágenes.

El sensor Kinect, se usa cada vez más en una amplia variedad de investigaciones [29] [30] [31].

En algunos de estos artículos, el Kinect es usado para maniobrar un quadrotor (helicóptero con 4 rotores para su sostén y propulsión) [32], en entornos cerrados, sin acceso a posicionamiento por GPS.

Áreas de aplicación

Las áreas en las que se pueden aplicar las técnicas de reconocimiento de patrones en visión computacional son bastante amplias, y algunos ejemplos son: Bioinformática; Astronomía; Reconocimiento de caracteres, sonidos, imágenes; Geografía; Visión Computacional; Carrera Espacial; Metodologías de Investigación; Videojuegos; y una creciente lista de disciplinas.

La investigación realizada en este documento, se podría aplicar en áreas de comparación de imágenes en 3D, como pueden ser tomografías de órganos, posicionamiento de escáneres en un espacio tridimensional, clasificación de imágenes, entre otros.

El método propuesto consta de varios apartados, cada módulo puede ser utilizado por separado e integrado en metodologías más complejas. Algunos de los módulos que se verán con cierto detalle son:

a) Esqueletonización de imágenes

La esqueletonización es utilizada como un método que nos ayuda en la obtención de los esqueletos de una imagen digitalizada. El esqueleto es un descriptor de una imagen en dos o tres dimensiones. En la actualidad se siguen estudiando diferentes descriptores de las imágenes, para poder realizar una comparación y clasificación más exacta y con menor costo computacional. Un efecto interesante es el de realizar un pre-procesamiento de las imágenes, usando el esqueleto como descriptor inicial, por el volumen de información que se reduce al 1% del volumen original de información.

La Tabla 1 muestra las masas de los objetos en su forma completa y su esqueleto. En la columna de la derecha, se observa el porcentaje de voxeles en total con los que se trabaja.

	Masa Imagen completa (voxeles)	Masa Imagen esqueleto (voxeles)	Porcentaje de reducción
Perro 1	53,824	305	99.43 %
Perro 2	29,641	267	99.10 %
Perro 3	15,192	237	98.44 %
Perro 4	41,431	283	99.32 %
Perro 5	26,950	259	99.04 %
Perro 6	31,798	244	99.23 %
Perro 7	45,886	283	99.38 %
Perro 8	54,231	348	99.36 %
Perro 9	50,706	265	99.48 %
Perro 10	34,322	293	99.15 %
Caballo	28,028	276	99.02 %
Gorila	14,880	210	98.59 %
Vaca	34,102	294	99.14 %
Gallo	66,592	333	99.50 %

Tabla 1 – Tabla de masas

b) Ángulos de Euler para la orientación de las imágenes.

Los ángulos de Euler, como se analizan a fondo en el **Anexo B**, nos ayudan en la orientación de los esqueletos de imágenes en tres dimensiones. Un paso intermedio para poder orientar las imágenes, es el cálculo de sus ejes principales y de su matriz de rotación. Estas operaciones se analizan en el **Anexo C**.

c) Voxelización de imágenes

Las imágenes digitales vienen en una gran variedad de formatos diferentes, para lograr tener un método congruente, es necesario estandarizar los formatos para que el método propuesto sea invariante a los formatos de los archivos. Una vez estandarizado el formato de las imágenes, se podrían analizar como información continua, ya que el objeto existe en un espacio tridimensional, y su volumen está dado en base a polígonos unidos en sus vértices uno tras de otro, formando un poliedro.

Mediante el procedimiento de voxelización, se logra dividir en una imagen representada en píxeles (o binaria), agregando capas de píxeles representando la tercera dimensión o profundidad de la imagen. Esta tercera dimensión se le conoce como "capas", guardando sus voxeles "prendidos" (activados) en un formato más genérico y sobre todo la información pasa a ser discreta. El formato binvox es una serie de puntos de coordenadas representadas con valores enteros sobre un espacio tridimensional, con ejes (X, Y, Z).

Con estos datos discretos, podemos realizar cálculos, ya que tenemos el sistema de coordenadas a disposición y el área de la figura está determinada por el número de voxeles; o bien, la suma de todas las coordenadas que se encuentran en estado "prendido". A esta suma le llamamos masa de objeto y es de gran ayuda para futuros cálculos sobre la imagen.

Con esta imagen en formato binvox, podemos realizar cálculos como son la obtención del centro de gravedad, el centro de masa, los ejes principales, rotación y escala de la imagen.

d) Algoritmo Húngaro

El algoritmo Húngaro, mejor conocido como Kunh-Munkres [34], por sus creadores, es de gran ayuda en la obtención del costo mínimo en problemas de asignación.

El algoritmo Húngaro trabaja con problemas de asignación en tiempo $O(n^3)$.

Para que este algoritmo funcione adecuadamente, se necesita una matriz de $n*m$, donde cada combinación de n y m , resulta en un costo o distancia de n , en función de m .

El problema se considera resuelto cuando se encuentra la mejor asignación de recursos (n) a cada tarea (m), en donde ningún recurso queda sin tarea asignada.

El procedimiento del algoritmo es el siguiente:

- 1) Se escoge el valor mínimo de cada fila y se le resta a todos los elementos de cada fila.
- 2) Se escoge el mínimo por columna, y se realiza la misma operación del punto anterior, pero ahora se usan las columnas.
- 3) Se trazan el menor número de líneas horizontales y verticales para cruzar a todos los ceros que existan en la matriz.
- 4) Si el número de líneas es igual a los valores n y m , entonces el algoritmo finaliza la búsqueda y comienza a asignar las combinaciones que tengan solo un cero.
- 5) En caso de que se tengan múltiples ceros en cada fila y columna, se realiza la asignación a aquellos que tengan un solo cero en la fila, aunque tengan más de uno en la columna y viceversa.
- 6) En caso de que el número de líneas trazadas no sea el mismo a n y m , entonces se realiza una búsqueda del número más pequeño que no haya sido cruzado por las líneas marcadas en el paso 3. Este valor se toma y se resta a todos los elementos que no estén cruzados por líneas, y a su vez, se le suma a aquellas locaciones en las que haya cruce de líneas.
- 7) Nuevamente se trazan líneas horizontales y verticales sobre los elementos que correspondan a la condición del paso 3.
- 8) Si el número de líneas aún no es igual a n y m , entonces se repite el paso 6 y 7, hasta que la condición se cumpla.
- 9) Una vez cumplida la condición, se procede a realizar las actividades del paso 4 y 5.

e) Grafos bipartitos

Un grafo bipartito, es una representación grafica en vértices y aristas, en las que se puede observar la división de dos grandes grupos de vértices, en los cuales, ningún vértice del mismo grupo tiene relación hacia elementos de su mismo grupo, solo hacia el "opuesto".

Aunque no es obligatoria la representación de un grafo bipartito como se puede observar en la Figura 1, esta forma de representación es la más adecuada para estos grafos.

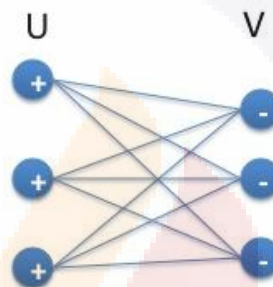


Figura 1 - Grafo Bipartito de conjuntos U, V

Este tipo de grafos, al agregarle el "peso" y convertirlo en un grafo bipartito pesado, se pueden representar en una matriz, como se ve en la Tabla 1. Los pesos del grafo se pueden representar como distancias, costo de moverse cada voxel de la primera imagen, hacia todas las posiciones de todos los voxeles de la segunda imagen.

2.2 DESCRIPCIÓN DE PRINCIPALES ESTUDIOS RELACIONADOS

Las investigaciones que tienen mayor peso sobre este documento, son las realizadas por Dr. Hermilo Sánchez Cruz y Dr. Ramón Rodríguez Dagnino en su artículo sobre la transformación de objetos voxelizados [17]. Estas son de vital importancia ya que en estas investigaciones se tocan temas específicos sobre la metodología para la obtención de ejes principales, invariantes de imágenes, voxelización y comparación de las áreas que ocupan ambas imágenes.

Un segundo artículo del Dr. Hermilo Sánchez y Dr. Ernesto Bribiesca [18], habla también acerca de cómo, utilizando métodos de voxelización, centro de masa e identificación de ejes principales; se puede transformar una imagen en otra. Dicha transformación se logra al encontrar la mínima energía posible, y mover cada uno de los voxeles hacia la segunda imagen.

Existe mas información en artículos como [14], en el que se discute acerca de la esqueletonización de imágenes en dos y tres dimensiones, así como sus aplicaciones.

El artículo en el cual se habla por primera vez del Algoritmo Húngaro o Kuhn-Munkres es [34], este algoritmo nos ayuda en localizar el costo mínimo de nuestra matriz de distancias. Dichas distancias las obtenemos al vaciar en una matriz el grafo bipartito pesado.

2.3 ANÁLISIS DE CONTRIBUCIONES Y LIMITACIONES DE PRINCIPALES ESTUDIOS RELACIONADOS

Hay varios esfuerzos en el área, entre los cuales se puede destacar un sistema experto que se dedicó a extraer elementos clave de cada espécimen encontrado en determinada región del país [1]. Esta base de datos es muy robusta y es piedra angular para otras investigaciones.

En relación a imágenes en 2D ó 3D, cada una de estas es toda un área de investigación y hay bastantes artículos escritos en dichas áreas por sí solas. Algunas de las investigaciones que proveen mayor información útil acerca del reconocimiento de imágenes en 2D se mencionan a continuación.

Para el caso de las plantas, han habido varios esfuerzos enfocados en la identificación de los tipos de hojas basándose en los contornos [7]; o bien usando el método de Fourier para encontrar un histograma de distribución de puntos [20].

Gran parte de los estudios en comparación de imágenes en 2D (y especialmente las de 3D) han recurrido a la eskeletonización de las mismas, para encontrar de manera más eficiente puntos de igualdad e identificación de componentes clave en la comparación de imágenes [3].

2.4 MODELO DE INVESTIGACIÓN

Desde el principio de implementar la metodología, se tiene conocimiento de la muestra de imágenes disponibles para procesar. Esto causa que el modelo de investigación sea cuasi-experimental, ya que no hay elementos sorpresa o datos que no se conozcan que pudieran causar efectos negativos o positivos en los resultados.

Se realizará una recolección de imágenes en dos dimensiones (para la validación de la hipótesis de invariancia en rotación) y tres dimensiones del sujeto de estudio en específico. Estas imágenes, representadas por poliedros, se delimitan en tamaño, al momento de realizar la voxelización; se usará la mayor resolución disponible siempre y cuando la metodología permita trabajar en un tiempo de cómputo aceptable. Para las imágenes en tres dimensiones, se deberá obtener sus descriptores y su esqueleto [3] [4].

2.5 DESCRIPCIÓN DE CONSTRUCTOS, VARIABLES OPERACIONALES Y ESCALAS

Los constructos iniciales para poder aplicar la metodología de este documento, son las diversas imágenes en tres dimensiones, así como todos los formatos que se crean en cada paso del método. Al momento de convertir un poliedro en una imagen voxelizada, damos dimensión digital a la imagen, lo cual nos permite aplicar métodos matemáticos y de mecánica en objetos estándares: dichos objetos son imágenes en tres dimensiones; el formato que nos permite aplicar métodos mencionados en este documento, es la representación de

voxels prendidos y apagados de una imagen en tres dimensiones. A cada voxel le corresponde un valor (X, Y, Z) , el cual se puede asociar con la posición de un punto en un sistema de coordenadas en tres dimensiones. La cantidad de voxels de la imagen es directamente proporcional al contorno máximo que se escoja al momento de generar dicha imagen. Para este documentos, el contorno máximo o caja limitante, es de $100*100*100$ voxels. Todas las imágenes usadas, serán escaladas a esta dimensión, para poder lograr también la invariancia a escala.

Cada voxel es un punto en las coordenadas (X, Y, Z) del espacio tridimensional. Dicho voxel puede tener solamente dos estados, lleno (prendido) o vacío (apagado).

Otro elemento importante es la forma del esqueleto de la imagen voxelizada. Este esqueleto nos da una idea más certera del tipo de imagen que se procesará; dado que la muestra de imágenes seleccionadas son imágenes caninas, podemos visualmente identificar visualmente que provienen de la misma familia de imágenes.

La justificación del porque se seleccionan solamente imágenes caninas, es porque se necesita construir un umbral construido con una base de datos de imágenes parecidas entre sí, para poder tener un nivel aceptable de similitud entre imágenes.

El esqueleto es una versión adelgazada del área voxelizada completa. Este esqueleto se obtiene con métodos de adelgazamiento de las imágenes, como se ve a detalle en el **Anexo C**.

Grafos bipartitos pesados, obtenidos en base a las distancias euclidianas que hay entre cada uno de los voxels de la imagen origen, contra todos los demás voxels de la imagen destino.

Así también, es de suma importancia que el método sea invariante a rotación, escala y translación. Las hipótesis que estaremos validando durante este documento se enfocan en la invariancia del método respecto a cualquier

imagen. Obteniendo esta invariancia, podemos lograr que la comparación y los resultados sean más asertivos.



3. METODOLOGÍA DE INVESTIGACIÓN

3.1 METODOLOGÍA Y TIPO DE DISEÑO DE INVESTIGACIÓN

La investigación es aplicada. Se aplicaran métodos de optimización, teoría de grafos y operadores morfológicos de imágenes en 3 dimensiones.

Para llevar a cabo la propuesta se cuentan con las imágenes tridimensionales virtuales de animales, las cuales fueron construidas manipulando figuras trigonométricas básicas (cubos, cilindros, esferas); se propone esta aseveración dado que las imágenes constan de varias partes o secciones, y un escaneo a través de dispositivos digitalizadores, resultarían en una imagen de una sola pieza. Las representaciones que se usan durante la metodología, se buscaron que fueran lo más cercano posible a la contraparte de un animal real.

La generación de las imágenes queda fuera del alcance del documento, por lo que solo se buscó bases de datos de imágenes 3D y se utilizaron aquellas imágenes que pudieran contribuir al desarrollo de la investigación.

3.2 POBLACIÓN, MUESTRA Y UNIDAD DE ESTUDIO

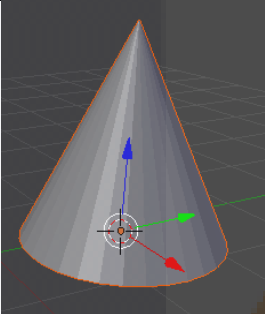

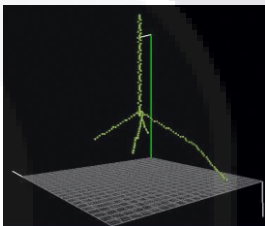
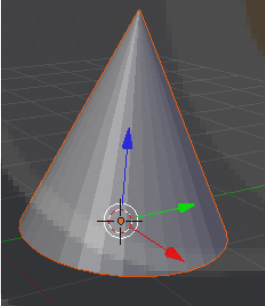
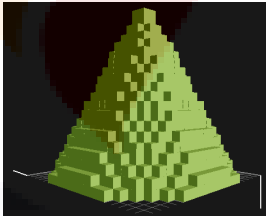
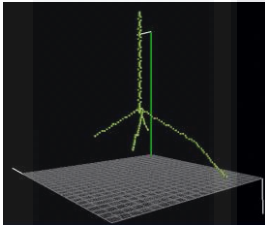
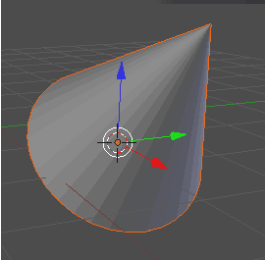
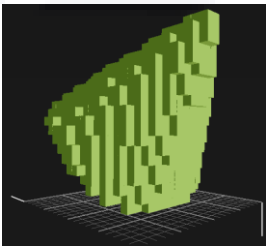
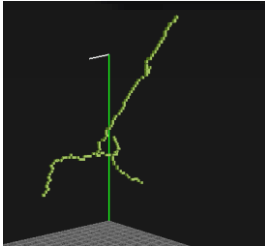
La población que tenemos es el conjunto de imágenes previamente generadas con algún software de creación de imágenes vectoriales usando primitivas y con programas computacionales como lo pueden ser Blender, AutoCad, entre otros. Estas imágenes se basan en representaciones del mundo real. En específico de animales. Las bases de datos de donde se obtuvieron las imágenes vectoriales originales son [47] [48] [49] [50] [51] [52] [53].

En concreto, en cuanto a figuras básicas tridimensionales, se utilizara un cono, ya que el cono es la figura más simple que nos arroja un esqueleto con el que podemos trabajar.

Si se realiza una eskeletonización de una esfera o un cubo, el resultado es un único punto (voxel) al centro de la imagen, por lo que no se pueden obtener mayores descriptores para efectos de la metodología de este documento.

En la categoría de imágenes de animales, tenemos un conjunto de figuras caninas y algunas imágenes individuales como se pueden observar a continuación.

Conos

	Poliedro	Voxelizada	Esqueleto	M y C
Cono sin rotación				173 [x] (42.0) [y] (55.0) [z] (47.0)
Cono grande sin rotación				173 [x] (42.0) [y] (55.0) [z] (47.0)
Cono 45 grados				136 [x] (54.0) [y] (54.0) [z] (44.0)

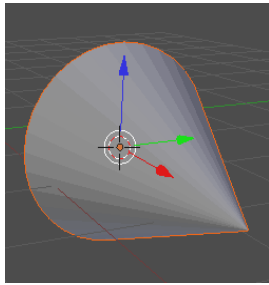
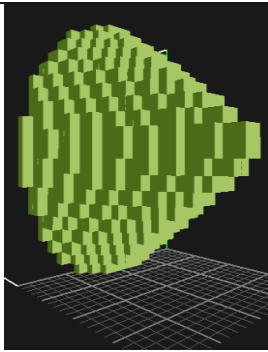
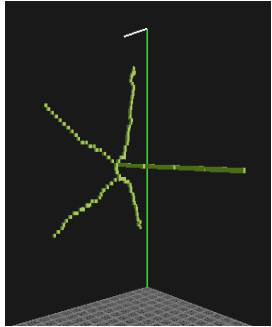
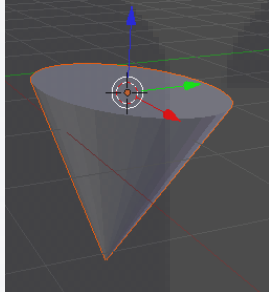
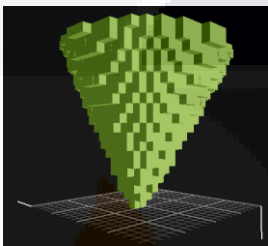
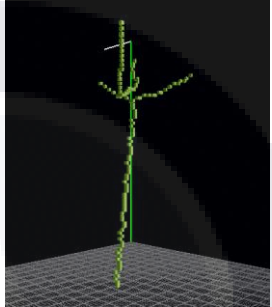
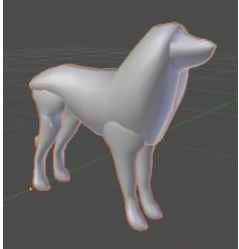
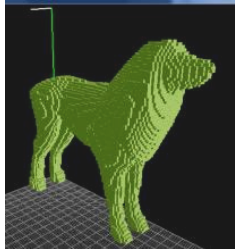
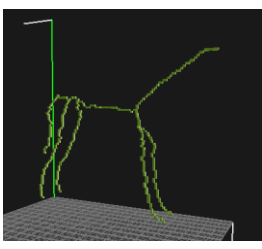
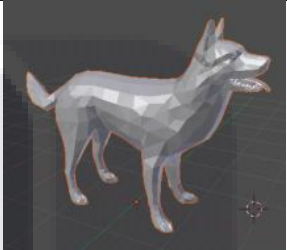
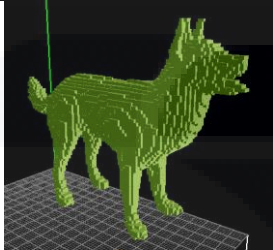
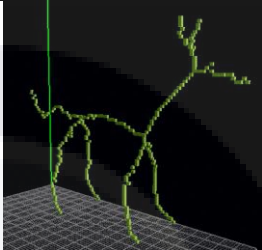

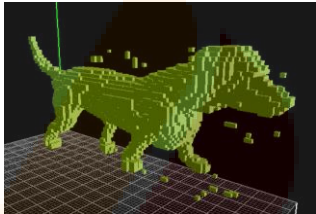
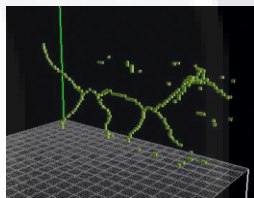

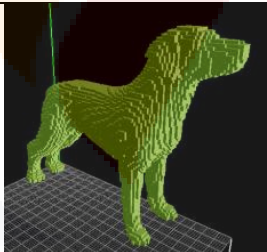
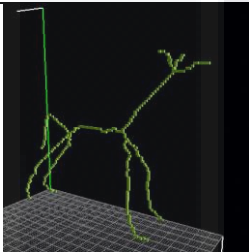

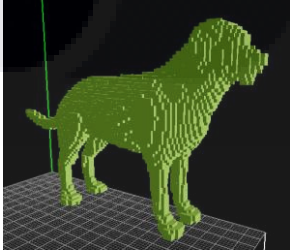
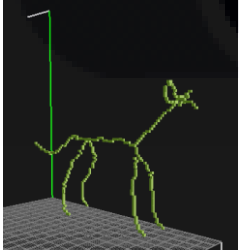
<p>Cono 90 grados</p>				<p>169 [x] (52.0) [y] (44.0) [z] (46.0)</p>
<p>Cono 180 grados</p>				<p>138 [x] (52.0) [y] (46.0) [z] (47.0)</p>

Tabla 2 – Conos: voxeles, esqueletos, masa y centro de masa

Donde Cono y Cono grande, es la misma imagen sin rotación pero con escala 2:1. Así también, **M** representa Masa y **C** representa centro de masa.

Caninos

El siguiente conjunto de imágenes caninas fue seleccionado para tener imágenes del mismo grupo, con ligeras diferencias. Este será nuestro grupo de control para conocer hasta qué punto los esqueletos se pueden considerar parecidos entre sí. En otras palabras este grupo de esqueletos nos ayudará a establecer el umbral que nos indicará el nivel de similitud entre una imagen y otra.

	Poliedro	Voxelizada	Esqueleto	M y C
P_1				305 [x] (37.0) [y] (41.0) [z] (14.0)
P_2				267 [x] (32.0) [y] (52.0) [z] (13.0)
P_3				237 [x] (23.0) [y] (56.0) [z] (12.0)
P_4				283 [x] (34.0) [y] (47.0) [z] (14.0)
P_5				259 [x] (29.0) [y] (56.0) [z] (12.0)


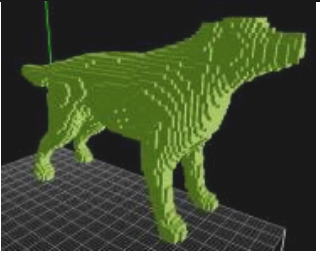
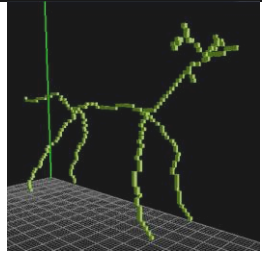

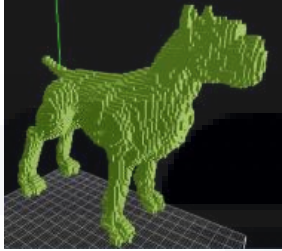
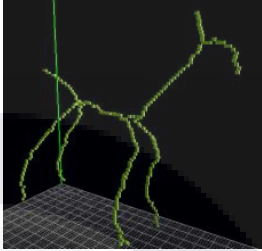

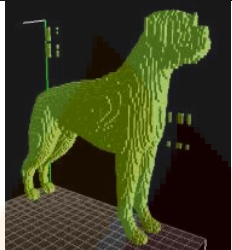
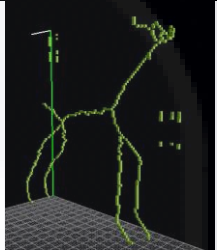

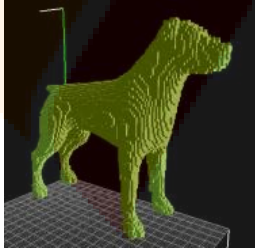
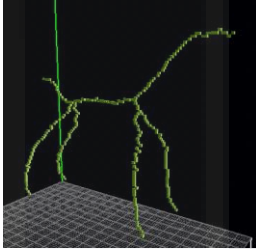

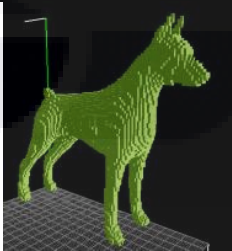
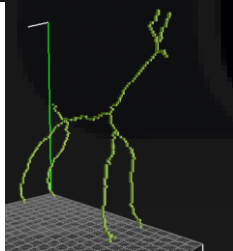
P_6				<p>244</p> <p>[x] (29.0)</p> <p>[y] (53.0)</p> <p>[z] (13.0)</p>
P_7				<p>283</p> <p>[x] (38.0)</p> <p>[y] (44.0)</p> <p>[z] (19.0)</p>
P_8				<p>348</p> <p>[x] (45.0)</p> <p>[y] (51.0)</p> <p>[z] (14.0)</p>
P_9				<p>265</p> <p>[x] (37.0)</p> <p>[y] (43.0)</p> <p>[z] (17.0)</p>
P_{10}				<p>293</p> <p>[x] (37.0)</p> <p>[y] (48.0)</p> <p>[z] (14.0)</p>

Tabla 3 – Figuras caninas: voxeles, esqueletos, masa y centro de masa

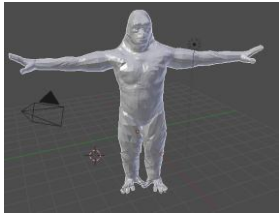
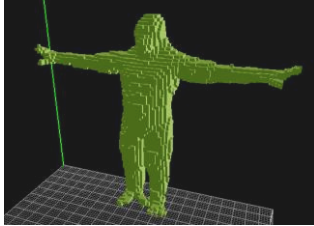
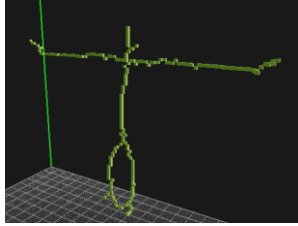
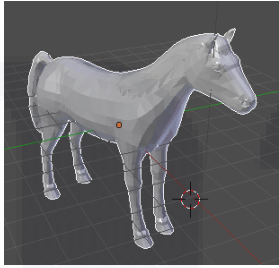
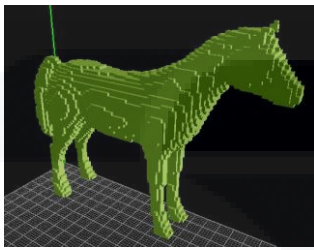
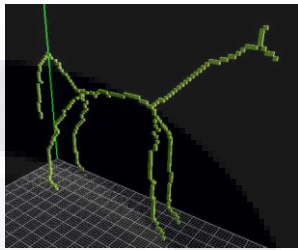

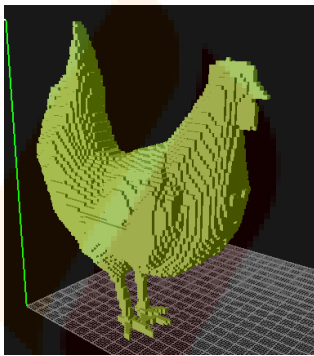
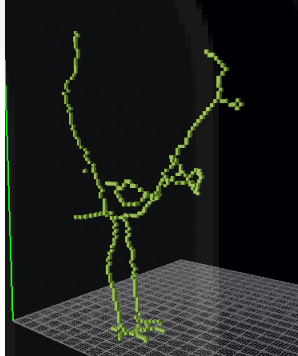
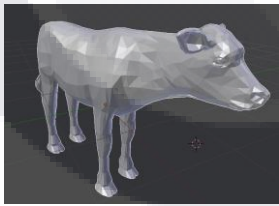
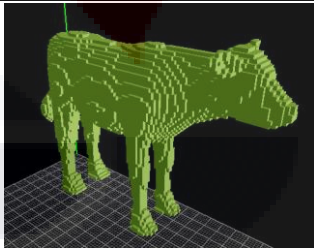
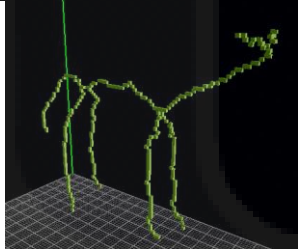
	Poliedro	Voxelizada	Esqueleto	
Gorila				210 [x] (45) [y] (50) [z] (10)
Caballo				276 [x] (32) [y] (42) [z] (11)
Gallo				333 [x] (43) [y] (20) [z] (40)
Vaca				294 [x] (33) [y] (43) [z] (13)

Tabla 4 – Figuras diversas: poliedros, voxeles y esqueletos

3.3 PLAN DE RECOLECCIÓN DE DATOS

Las imágenes fueron filtradas utilizando el programa de edición de gráficos vectoriales Blender (<http://www.blender.org/>). Como se observó anteriormente, los archivos que se obtuvieron están en diferentes formatos, y se realizó un pre-

procesamiento para asegurarse que los formatos que se utilizarán, sean compatibles con el método usado.

3.4 PROCEDIMIENTOS DE ANÁLISIS DE DATOS Y VALIDACIÓN

Método principal

Los cálculos necesarios incluyen la discretización de la imagen original en voxeles; obtención de datos discretos relacionados con la voxelización, específicamente la obtención de coordenadas en plano euclidiano de cada voxel, en un espacio tridimensional dividido por capas, renglones y columnas, o bien, coordenadas (X, Y, Z) , como se puede observar en la Figura 2.

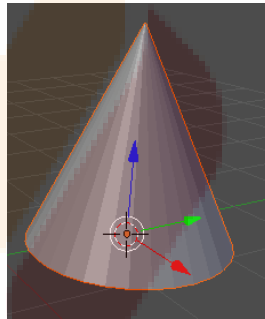


Figura 2 – Imagen de un cono

Resumiendo el método, y como una introducción, los pasos a seguir de la metodología propuesta son:

1. Obtención y filtrado de las muestras a usar.
2. Voxelización.
3. Esqueletonización.
4. Cálculo de centro de masa.
5. Cálculo de momentos geométricos.

6. Cálculo de matriz de momentos.
7. Valores y vectores propios (ejes principales).
8. Comparación de ejes principales de dos imágenes a comparar.
9. Alineación de imágenes.
10. Rotación de imagen original, en base a los resultados de la comparación y alineación del paso anterior.
11. Obtención de voxeles positivos y negativos (grafo bipartito).
12. Cálculo de la matriz de distancias euclidianas.
13. Cálculo del costo mínimo con algoritmo húngaro.
14. Análisis de los resultados y comparación contra umbral, para definir el grado de similitud entre ambas imágenes.

Esta serie de pasos, describe a grandes rasgos la metodología a usar, y durante esta sección del documento, se describirá cada paso independiente y a detalle.

Voxelización

Se opto por el método más usado de esqueletonización para dar rapidez en la obtención de resultados. Dicho método es el creado por Cornea et al [13]. Este método se basa en la metodología de ray stabbing y parity count, como se puede observar en [19].

La mejor manera de comprobar la metodología es aplicarla sobre imágenes sencillas y simples. Como primera opción se tomo un cubo, pero al momento de esqueletonizar esta imagen, obtenemos un voxel, sin ramificación ni volumen por lo cual esta imagen queda fuera de nuestro universo de imágenes disponibles.

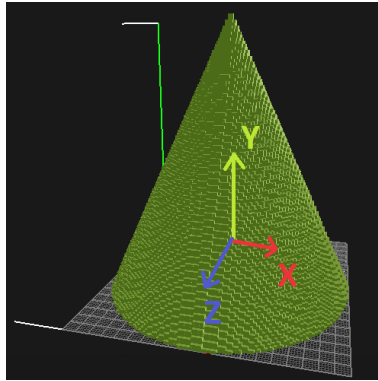


Figura 3 – Imagen voxelizada de un cono

La siguiente imagen que se propone para obtener un esqueleto y trabajar es un cono. El esqueleto de un cono contiene al menos 3 ramificaciones, por lo cual nos permite poder realizar los cálculos. El esqueleto de un cono se puede observar en la Figura 4. El cono se utilizara en una orientación "natural" sin rotación. Nos referimos a orientación natural a la manera en la que se representa normalmente un cono en libros de geometría. La segunda imagen, será el mismo cono, rotado 90 grados sobre el eje X.

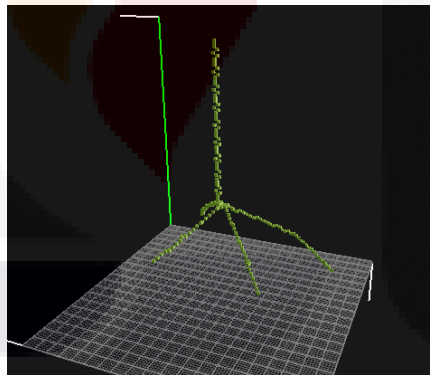


Figura 4 – Imagen esqueletonizada de un cono

Es importante destacar la resolución con la cual se estarán utilizando las imágenes, ya que la resolución de nuestra imagen es directamente proporcional al tiempo de procesamiento de la computadora y a la precisión con la cual obtenemos resultados de similitud entre esqueletos de imágenes. Los resultados de aplicar el método de voxelización a ambas imágenes se

pueden observar en la Tabla 2, 3 y 4, junto con algunos datos que nos arroja el método y que nos servirán más adelante, como lo son la masa total del objeto y resolución máxima, tomando en cuenta la "caja" de voxeles apagados que envuelven a la imagen.

Con la imagen voxelizada, contamos con dos tipos de voxeles: Los voxeles "prendidos" son aquellos voxeles que son parte de la imagen, y los voxeles "apagados", son aquellos que pertenecen al fondo de la imagen y no son parte de la misma. Se tiene una limitante en cuanto al tamaño máximo disponible para el software que se utiliza al voxelizar las imágenes, denominado binvox y thinvox [13]. El límite es de 255 voxeles por eje, por lo que la imagen más grande que podremos manejar es de $255 \times 255 \times 255$ voxeles, dando una área total de 16, 581,375 voxeles posibles.

Esqueletonización

Con la imagen voxelizada, se puede proceder a la esqueletonización de ambas imágenes. Como se vio anteriormente, este paso se realiza aplicando el programa desarrollado por Patrick Min, usando una mezcla de los métodos de conteo de paridad y apuñalamiento en rayos, previamente codificados y listos para ser usados.

Ambos esqueletos, junto con información relacionada, se pueden observar en la Tabla 2, 3 y 4.

El esqueleto de la imagen nos dará una idea muy clara de la orientación. Al revisar los datos que nos arrojan los esqueletos de la imagen, nos damos cuenta que en la misma imagen, con una rotación de 90 grados, el esqueleto obtenido no tienen la misma masa, por lo que nos indica claramente que el método es sensible a rotación.

Una vez que tenemos ambos esqueletos y la información de la Tabla 3, se procede al cálculo de su centro de masa.

Centro de masa

Para poder calcular el centro de masa, en base al esqueleto de la imagen, necesitamos un archivo intermedio. Este archivo, con extensión `.raw`, es generado de la misma manera en la cual se genera el archivo esqueletonizado. Solo se necesita procesar el método que obtiene el esqueleto pero se le manda un parámetro especial. Con esta operación, se obtiene el archivo `.raw` o bien llamado unos y ceros.

El valor del centro de masa no necesariamente tiene que caer dentro de alguno de los voxeles prendidos de nuestra imagen. El centro de masa es el punto en el cual se podría ejercer una fuerza de empuje y el objeto se desplazaría si ninguna rotación sobre ningún de sus ejes.

El archivo obtenido, es una cadena de n posiciones, donde n es igual a la resolución de la imagen que estamos procesando (si la imagen tiene una resolución de $100*100*100$, n valdrá 1, 000,000). Este archivo debe ser procesado carácter por carácter, para poder dividirlo en capas, renglones y columnas. Si dividimos el archivo `.raw` original, tendremos un nuevo archivo, que le llamaremos *archivo de capas*. Este archivo de capas, contiene m numero de renglones, dados por la resolución que utilizamos al principio de nuestro método (si la imagen tiene una resolución de $100*100*100$, el valor de m será 100).

Ahora bien, con el archivo de capas calculado, si lo procesamos carácter por carácter, podemos obtener la posición euclidiana de cada uno de los voxeles prendidos, en relación a un centro imaginario $(0, 0, 0)$, que se encuentra en el principio del procesamiento del archivo de capas.

Obteniendo las posiciones euclidianas de cada voxel, podemos entonces proceder a calcular el centro de masa, con la ecuación (1):

$$C_k = \frac{\sum_{i=1}^n V_i(k)}{N} \quad (1)$$

Donde:

C_k = Centro de masa en el eje k – ésimo

$k = \{x, y, z\}$

$V_i(k)$ = coordenada k del i – ésimo voxel prendido

N = total de voxeles prendidos

La obtención del centro de masa es una de las partes más importantes que necesitamos realizar, ya que será nuestra información primordial para poder lograr la orientación de los voxeles.

Nuestro método es invariante a translación y tamaño, el único descriptor por el cual nos debemos de preocupar es el de rotación. La siguiente parte de la metodología se encarga de eso.

Para poder asegurar una invariancia en rotación, necesitamos ejecutar un pre-procesamiento de las imágenes, para esto necesitamos conocer los momentos geométricos y realizar una rotación de las imágenes para poder alinearlas.

Momentos geométricos

Los momentos geométricos de la imagen [20], se calculan aplicando la ecuaciones (2)-(8):

$$M_{pqr} = \sum_{x=1}^N \sum_{y=1}^M \sum_{z=1}^P x^p y^q z^r I(x, y, z) \quad (2)$$

$$M_{020} = \sum_{x=1}^N \sum_{y=1}^M \sum_{z=1}^O y^2 I(x, y, z) \quad (3)$$

$$M_{002} = \sum_{x=1}^N \sum_{y=1}^M \sum_{z=1}^O z^2 I(x, y, z) \quad (4)$$

$$M_{110} = \sum_{x=1}^N \sum_{y=1}^M \sum_{z=1}^O x^1 y^1 I(x, y, z) \tag{5}$$

$$M_{101} = \sum_{x=1}^N \sum_{y=1}^M \sum_{z=1}^O x^1 z^1 I(x, y, z) \tag{6}$$

$$M_{200} = \sum_{x=1}^N \sum_{y=1}^M \sum_{z=1}^O x^2 I(x, y, z) \tag{7}$$

$$M_{011} = \sum_{x=1}^N \sum_{y=1}^M \sum_{z=1}^O y^1 z^1 I(x, y, z) \tag{8}$$

Con los momentos geométricos calculados, podemos proceder al cálculo de la matriz de momentos, la cual se verá en la próxima sección.

Matriz de momentos

La matriz de momentos se calcula con la ecuación (9). Se realiza una adición o resta correspondiente, para terminar con 9 elementos.

$$M = \begin{bmatrix} M_{020} + M_{002} & -M_{110} & -M_{101} \\ -M_{110} & M_{200} + M_{002} & -M_{011} \\ -M_{101} & -M_{011} & M_{200} + M_{020} \end{bmatrix} \tag{9}$$

Esta matriz es la base para poder realizar el cálculo siguiente.

Valores y vectores propios

Los valores y vectores propios los obtenemos de la matriz de momentos mostrada en la sección anterior. El cálculo de los vectores y valores se realizan utilizando las ecuaciones (10)-(14).

$$\cos(\alpha) = \frac{(M' \cdot N)}{\|M'\| \|N\|} \tag{ 10 }$$

$$\cos(\beta) = \frac{(m' \cdot m)}{\|m'\| \|m\|} \tag{ 11 }$$

$$\cos(\gamma) = \frac{(N \cdot M)}{\|N\| \|M\|} \tag{ 12 }$$

$$N = m' \times m \tag{ 13 }$$

$$\|M'\| = \sqrt{M_x^2 + M_y^2 + M_z^2} \tag{ 14 }$$

Los vectores y los valores propios obtenidos están ordenados. El primer valor obtenido corresponde al primer vector, y así sucesivamente. Adicionalmente los valores propios nos ayudan a encontrar los ejes principales de la imagen. El valor propio mayor se relaciona con el eje mayor, el valor propio menor se relaciona con el eje menor y el valor propio restante, se relaciona con el eje mediano.

Ordenados los valores propios y sus correspondientes vectores propios, obtenemos los 3 ejes principales de la imagen. De este momento en adelante, llamaremos ejes principales a los vectores propios ordenados.

Adicionalmente, y como un apoyo visual, podemos trazar los ejes principales sobre un espacio tridimensional, esto nos ayudará a darle dimensión y sentido a los ejes, sobre todo al momento de comparar los ejes principales de una imagen, contra los de una segunda imagen. Los ejes principales de la imagen los podemos observar en la Figura 5. En la imagen se observa numerados, los 3 ejes que calculamos, y en color negro se observan los ejes fijos (X, Y, Z) de nuestro sistema de coordenadas.

Los ejes calculados corresponden a la figura del cono sin rotación, y a cono rotado 90 grados alrededor del eje mayor, o eje 1.

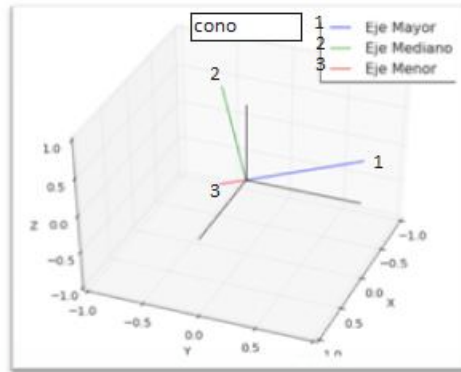


Figura 5 – Vectores propios/ejes principales

Si trazamos los dos sistemas de ejes principales sobre un mismo sistema de coordenadas, tenemos la representación grafica de la Figura 6. En dicha figura podemos observar la rotación que se requiere realizar para que ambos sistemas estén alineados. Para poder realizar la alineación de un sistema de coordenadas sobre el otro, se utiliza la teoría de los Ángulos de Euler [21].

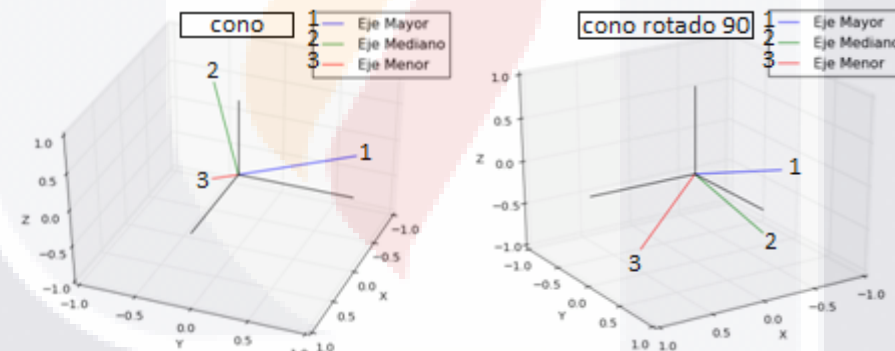


Figura 6 – Ejes principales de dos imágenes

Ángulos de Euler

La teoría detrás de los ángulos de Euler **Anexo B**, nos indican que se realiza una rotación sobre 3 ejes, de esta manera se logra obtener un sistema rotado tomando como base un segundo sistema. En nuestro caso, el primer sistema (ejes principales de la primera imagen) será nuestro sistema base, sobre el cual el segundo sistema será rotado (orientado).

El orden en el cual se realiza la rotación de las imágenes, afecta en el resultado final, por lo que es muy importante escoger correctamente los vectores. Los ángulos con los que se trabajan se representan con las siguientes letras: α , β , γ . El orden en el que se realizan las rotaciones es:

1. Rotación alrededor del eje Z
2. Rotación alrededor del eje X'
3. Rotación alrededor del eje Z'

Estas tres rotaciones, como se observan en la Figura 7, nos arrojan un sistema rotado en base a un primer sistema. Consecuentemente podemos realizar la orientación en sentido opuesto, eso es, tomar el segundo sistema (sistema rotado) y calculando los 3 ángulos de Euler, podemos invertir la rotación, para alinear los dos sistemas. En la Figura 7 se observa una representación grafica de cada una de las 3 rotaciones necesarias. La primera parte de la imagen, representa una rotación en α grados sobre el eje z, la segunda imagen representa la rotación realizada en β grados sobre el eje x', y la tercera representa una rotación en γ grados sobre el eje z'.

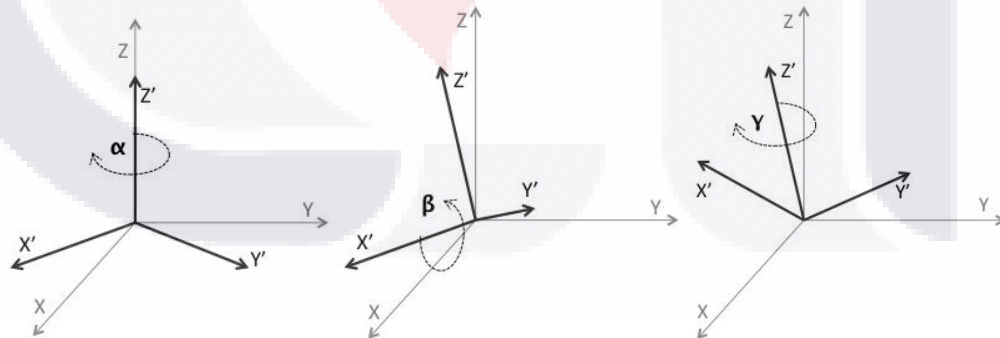


Figura 7 – Rotaciones con ángulos de Euler

Las formulas que se utilizan para poder realizar la rotación “inversa” se observan en la ecuaciones (11)-(14). Se presentan todas las rotaciones resumidas en una sola matriz de rotación. Al aplicar esta matriz de rotación sobre el segundo sistema, logramos realizar la rotación en un solo paso.

$$\cos(\alpha) = \frac{(M' \cdot n)}{\|M'\| \|n\|} \tag{ 15 }$$

$$\cos(\beta) = \frac{(m' \cdot n)}{\|m'\| \|n\|} \tag{ 16 }$$

$$\cos(\gamma) = \frac{(N \cdot M)}{\|N\| \|M\|} \tag{ 17 }$$

$$R = \begin{bmatrix} \cos \beta \cos \alpha & \sin \gamma \sin \beta \cos \alpha - \cos \gamma \sin \alpha & \cos \gamma \sin \beta \cos \alpha + \sin \gamma \sin \alpha \\ \cos \beta \sin \alpha & \sin \gamma \sin \beta \sin \alpha + \cos \gamma \cos \alpha & \cos \gamma \sin \beta \sin \alpha - \sin \gamma \cos \alpha \\ -\sin \alpha & \sin \gamma \cos \beta & \cos \gamma \cos \beta \end{bmatrix} \tag{ 18 }$$

Donde:

M = eje mayor del primer sistema

M' = eje mayor del sistema primado

n = eje mediano del primer sistema

n' = eje mediano del sistema primado

m = eje menor del primer sistema

m' = eje menor del sistema primado

Podemos también realizar un cálculo de los ángulos por separado, con las formulas (15)-(18). El sistema de ejes y la línea de nodos, representa la línea por la cual los dos planos se interceptan, se pueden observar en la Figura 8.

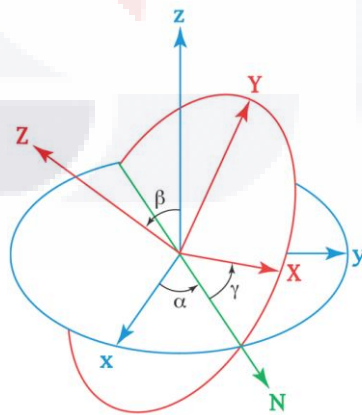


Figura 8 – Ángulos de Euler

Cuando conocemos los ángulos de cada parte, podemos realizar la rotación en la imagen original, en este caso utilizando un programa de edición de imágenes digitales en tres dimensiones.

Rotación de imagen original

Utilizando los ángulos de Euler podemos rotar la imagen poliédrica original para lograr la alineación con la otra imagen. Los detalles de la rotación se pueden ver en el **Anexo E**.

Al tener las imágenes orientadas, podemos proceder con la comparación de ambas imágenes. Para esto, necesitamos nuestros archivos base (thinvox), recalculados con las nuevas posiciones de las imágenes, para lo cual necesitamos volver a calcular el archivo .raw.

Una vez que ambas imágenes has ido orientadas y rotadas correctamente, podemos proceder con la siguiente parte de la metodología, la cual es la translación de las posiciones de cada voxel hacia una nueva posición, relativa al centro de masa de cada imagen.

En este paso, tenemos que trasladar cada uno de los voxeles prendidos, utilizando el centro de masa como modificador de las posiciones de cada voxeles. Para orientar los voxeles, se requiere tomar cada elemento del componente y restarle su correspondiente parte del centro de masa. Esta operación se observa en (12).

$$V'_{xyz} = \sum_{p=1}^m \sum_{q=1}^n \sum_{r=1}^r (x_p - x_c, y_q - y_c, z_r - z_c) \tag{ 19 }$$

Donde:

V'_{xyz} = nueva posición del voxel prendido

x_c, y_c, z_c = componente x, y, z del centro de masa

m_i = número total de voxeles prendidos

Reemplazamos cada valor de nuestros voxeles prendidos con los nuevos valores, y resulta una matriz que llamaremos, matriz orientada. Esta matriz contiene las nuevas posiciones orientadas relativas al centro de masa.

Procedemos a obtener una tercera matriz, la cual es la diferencia entre las matrices orientadas de la primera y la segunda imagen.

Voxeles positivos y negativos

El resultado de comparar las dos matrices orientadas, nos da como resultado dos nuevas matrices que discutiremos a continuación.

Al tomar cada uno de los elementos de la primera matriz, y buscarlo en la segunda matriz, tres categorías de voxeles aparecen:

1. Aquellos que se encuentran repetidos en ambas matrices se les etiqueta como *voxeles descartados*, y se eliminan de las matrices; no se consideran para las siguientes partes de la metodología.
2. Aquellos voxeles que existen en la matriz de la primera imagen, y no existen en la matriz orientada de la segunda imagen, se etiquetan como *voxeles positivos*.
3. De la misma manera, aquellos voxeles que existen en la segunda imagen, pero no en la primera, se les conoce como *voxeles negativos*.

Esta definición es válida, siempre y cuando, se mantenga la selección de cuál es la imagen original y cuál es la rotada de la sección anterior.

Los voxeles descartados, se les etiqueta así, ya que no tienen ningún sentido en buscar las diferencias entre dos voxeles que se encuentran en el mismo

espacio. Más aún, el resultado de buscar su comparación es 0, indicando que no hay costo en mover un voxel hacia su misma posición.

Con estas dos matrices calculadas, procedemos a calcular la distancia que existe entre los voxeles positivos y los negativos.

Calculo de la matriz de distancias

Una vez conociendo los voxeles que no se encuentran en la segunda imagen, pero en la primera sí, podemos obtener una tercera matriz de valores. Esta matriz nos calculará las distancias que existen de cada uno de los voxeles positivos, hacia los voxeles negativos. El resultado es una matriz n*m, donde n es el numero de voxeles positivos y m el numero de voxeles negativos.

Existen múltiples maneras de obtener las distancias entre dos puntos de un espacio tridimensional, entre algunas de las más usadas son la distancia de Tanimoto [44], distancia de Manhattan [45] o distancia euclidiana [46] (Figura 9).

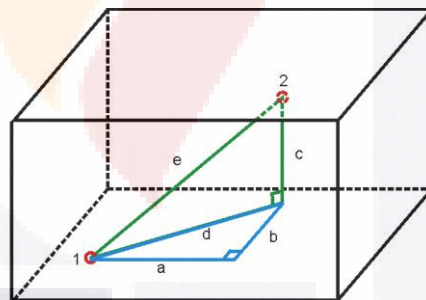


Figura 9 – Distancia euclidiana entre dos puntos

Se utilizara la distancia euclidiana para calcular las distancias mínimas entre voxeles. Para este cálculo usamos la ecuación (20)

$$d_E(x, y, z) = \sqrt{(x - x') + (y - y') + (z - z')}$$
(20)

Donde:

x = componente x de voxel positivo
 x' = componente x' de voxel negativo
 y = componente y de voxel positivo
 y' = componente x de voxel negativo
 z = componente z de voxel positivo
 z' = componente z' de voxel negativo

De la misma manera hacemos un cálculo de las distancias de los voxeles negativos hacia los positivos. Este cálculo simplemente se realiza como validación del primer cálculo, ya que al ejecutar el algoritmo Húngaro sobre ambas matrices los resultados que obtendremos, deberán ser idénticos.

Dado que el algoritmo Húngaro tiene elementos en los que se requiere tomar decisiones que afectan el resultado final, el tiempo de procesamiento y en costo mínimo obtenido no los mismos si se procesa repetidamente. Vale la pena resaltar este punto ya que si el método se realiza en matrices muy grandes, se pueden llegar a obtener dos resultados diferentes cuando se procesan las matrices. Para las dimensiones que se están usando en este documento, no se identificaron diferencias en los resultados al procesar la misma comparación en tres ocasiones diferentes. Sin embargo, es un punto que debe considerarse si se desea utilizar este método para matrices e imágenes más grandes.

Esta manera de representar las distancias de voxeles positivos hacia negativos, puede ser analizada como si fuera un grafo bipartito pesado. Si lo representamos en una imagen, como se observa en la Figura 10, el grafo izquierdo (A) corresponde a los voxeles positivos, y el grafo derecho (B) corresponde a los voxeles negativos. Cuando se realiza el análisis de validación, el orden de los voxeles cambia, pero sigue representando un grafo bipartito pesado. Los pesos de los grafos están representados por las distancias euclidianas que calculamos en este paso de la metodología. Así pues, tenemos la representación completa con todos los elementos que se necesitan para nuestro grafo.



Figura 10 – Grafo Bipartito

Con nuestras dos matrices calculadas, podemos continuar con la última parte de la metodología, pero antes necesitamos asegurarnos que las matrices tienen la dimensión adecuada para proceder con el cálculo del algoritmo Húngaro.

Calculo del costo mínimo con algoritmo Húngaro

El algoritmo Húngaro es un algoritmo de optimización para la resolución de problemas de asignación. El resultado obtenido es el costo mínimo de asignar todos los elementos de los componentes X de una matriz hacia su correspondiente componente Y.

Enfocado en nuestro problema, los renglones representan nuestros voxeles positivos, y las columnas corresponden a los voxeles positivos (y viceversa cuando ejecutemos nuestro método propuesto).

El algoritmo Húngaro funciona con matrices cuadradas, o bien cuando se tienen más renglones que columnas. Si como resultado de nuestras operaciones anteriores, se obtiene alguna matriz que no cumpla con la restricción anterior, es necesario que se rellene con valores por default (1), para que el método pueda funcionar y el problema pueda ser resuelto.

3.5 EJEMPLO DE APLICACIÓN DE LA METODOLOGÍA PROPUESTA

El proceso de generar los esqueletos se puede resumir en el siguiente listado de actividades.

Se aclara primeramente, que se necesita una estructura de directorios como se observa en la imagen 4.10.

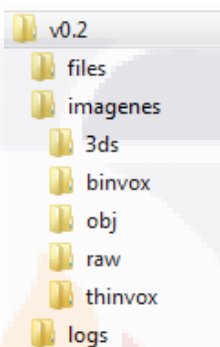


Figura 11 – Estructura de directorios

Dentro del directorio v0.2 (de aquí en adelante, se puede referir también como directorio padre), se tienen los archivos que se encarga de generar la información para la siguiente parte de la metodología.

Los archivos que necesitamos dentro del directorio padre son:

- binvox.exe (bv.exe)
- thinvox.exe (tv.exe)
- viewvox.exe (vv.exe)
- glut32.dll

El archivo .dll es necesario para que bv.exe y tv.exe funcionen correctamente. Dentro del folder de imágenes, dividiremos los diferentes formatos obtenidos durante el proceso completo.

1) Obtención de imágenes 3DS

Como se discutió previamente, la obtención de imágenes se realiza en las bases de datos que se pueden encontrar en internet [47] [48] [49] [50] [51] [52] [53].

Cuando se tienen las imágenes, se almacenan en el directorio "3ds".

2) Conversión a formato .obj

Utilizando el SW Blender, se importa la imagen 3DS original, a través del menú: **Archivo -> Importar -> 3D Studio (.3ds)**.

Una vez la imagen importada, si se requiere se puede manipular con el mismo software, rotándola. La escala y la posición no es indispensable que se acomoden, ya que en el siguiente paso estableceremos el tamaño máximo de la caja que envolverá a nuestra imagen.

La exportación se logra a través del menú: **Archivo -> Exportar -> Wavefront (.obj)**.

Almacenamos la imagen exportada en su directorio correspondiente (imágenes/obj).

3) Generación de archivo binvox

La siguiente parte de la metodología es obtener la imagen voxelizada. En la imagen 4.11, se puede observar lo que se logrará en este paso.

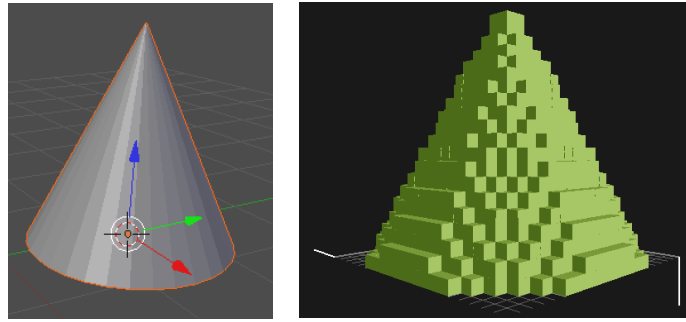


Figura 12 – Cono poliédrico y voxelizado

La voxelización la logramos con el programa desarrollado por Patrick Min basándose en [13]. El código es código abierto y de libre distribución; el cual se puede descargar en [54]. La sintaxis que se debe seguir, para el caso de nuestra figura de ejemplo es la siguiente:

```
C:\v0.2>binvox.exe -d 100 imagenes/obj/cono.obj
```

-d 100 nos indica la resolución que se usara para envolver a la imagen que se está generando, esto nos garantiza la robustez en escala y traslación.

El archivo resultante tendrá la extensión `.binvox` (el nombre completo es `cone.binvox`). Este archivo solamente puede ser visualizado a través del programa `viewvox.exe` [56]. La segunda parte de la imagen 4.10 muestra el resultado de ejecutar este programa.

Vale la pena mencionar que para mantener ordenado nuestras imágenes y archivos resultantes. La imagen resultante de este proceso debe moverse a su folder correspondiente (imágenes/binvox)

Para poder visualizarla, se necesita ejecutar el comando `viewvox.exe` en la consola y mandarle como parámetro el nombre del archivo `binvox` deseado. Recordando que movimos el archivo `binvox` a su directorio correspondiente:

```
C:\v0.2>viewvox.exe imagenes/binvox/cone.binvox_
```

4) Generación de archivo thinvox

El archivo generado anteriormente además de un apoyo visual, es necesario para continuar con la metodología. El esqueleto se obtiene ejecutando el siguiente comando:

```
C:\v0.2>thinvox.exe imagenes/binvox/cone.binvox
```

El único parámetro que se requiere, es el nombre (y directorio) de las imágenes generada con el paso anterior.

En la consola, se podrán leer los resultados del programa. Dentro de los resultados que se obtienen, se puede leer el nombre del archivo generado (en este caso, el archivo generado se llama thinned_3.binvox):

```
[thinned.binvox] already exists  
[thinned_1.binvox] already exists  
[thinned_2.binvox] already exists  
VoxelFile::write_file<thinned_3.binvox>
```

Podemos ejecutar nuevamente el comando viewvox.exe para visualizar el esqueleto resultante. En la figura 13 se puede observar tanto la imagen voxelizada como el esqueleto resultante.

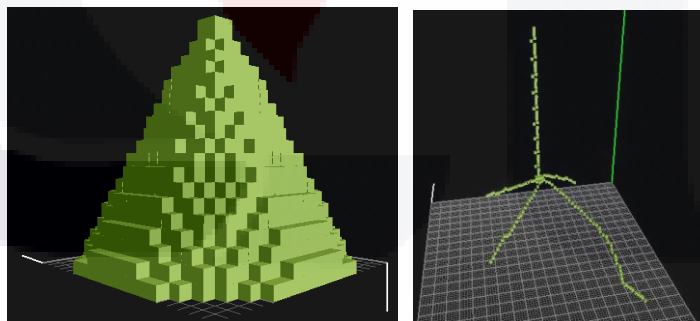


Figura 13 – Cono en volumen y esqueleto

De esta manera, logramos obtener el esqueleto de la imagen vectorial original. Teniendo cuidado con revisar los nombres de los archivos generados y almacenándolos en sus directorios correspondientes, se pueden identificar mejor todos los archivos obtenidos durante cada paso de la metodología y

podemos regresar a alguna sección en específico, para realizar un análisis a fondo en caso de ser necesario.



4. EXPERIMENTOS Y RESULTADOS

4.1 CONTRIBUCIONES AL CONOCIMIENTO

Se pretende que el método propuesto sea utilizado como una herramienta para encontrar la similitud entre dos esqueletos de imágenes en tres dimensiones.

La aplicación del algoritmo Húngaro para encontrar el costo mínimo entre los voxeles positivos y negativos de dos esqueletos, apoya en el estudio de la identificación de características de esqueletos, que nos puedan permitir la correcta clasificación y los resultados de los costos mínimos, mas no es suficiente este valor únicamente. La unión de este valor con la suma de voxeles positivos y negativos permite encontrar un umbral con el cual se pueda clasificar la igualdad o diferencia entre esqueletos de imágenes.

Trabajar con esqueletos, reduce el volumen de datos en un 99% en promedio, así pues, trabajamos con aproximadamente el 1% de los voxeles, que son los voxeles restantes en un esqueleto (ver Tabla 1).

Se pretende también encontrar si, trabajando con esqueletos en lugar de imágenes completas, es posible conocer la similitud entre dos imágenes. Por último, se propone un índice de proporción de distancia, para la comparación de los esqueletos de dos imágenes en 3D; el cual puede ser aplicado en términos de la distancia (costo mínimo) y masa de las imágenes (voxeles positivos y negativos).

4.2 CONTRIBUCIONES A LA PRÁCTICA

Se propone una herramienta que permite agrupar todo el proceso de comparación de esqueletos de imágenes en tres dimensiones en un solo proceso que integre los diversos pasos involucrados.

Aún cuando la metodología utiliza programas previamente desarrollados, se ha logrado crear un sistema único con el cual se pueden comparar dos imágenes

vectoriales y se obtiene un Índice de similitud que puede ser utilizado como criterio para decidir si dos imágenes son similares.

La implementación del método, se desarrolló en un intérprete de medio nivel, el cual es portable a cualquier plataforma y sistema operativo que existen al momento de escribir este documento.

Se logro demostrar que el algoritmo Húngaro puede trabajar hasta cierto nivel, con los voxes positivos y negativos de las imágenes, aún cuando su capacidad es limitada, dado que es un algoritmo de optimización en tiempo $O(n^3)$. El método nos permite conocer un resultado de similitud relativamente rápido cuando el esqueleto resultante es pequeño.

Finalmente, el índice de verosimilitud propuesto puede ser aplicado a cualquier otro conjunto de imágenes para la obtención de un umbral que proporcione un criterio estadístico sobre la similitud de un conjunto de imágenes del mismo tópico.

4.3 LIMITACIONES CLAVE

Se tiene una cantidad relativamente limitada de imágenes, ya que se requiere contar con imágenes previamente generadas por algún software de creación de imágenes vectoriales en 3D; que representen fielmente a algún animal del mundo real.

Para el alcance de esta investigación, no hay forma de realizar una captura automática de imágenes del mundo real.

Dado que se trabaja con esqueletos, si una imagen relativamente igual, tiene una extremidad o rama orientada diferente, el resultado final puede ser muy diferente a lo que se espera. Si se cuenta con un cono, y se compara contra otro muy parecido, pero con la parte superior erosionada, el esqueleto final no se parece lo suficiente para poder decidir que ambas imágenes son iguales. Por lo que es muy importante establecer el umbral con el cual podemos aceptar la similitud de un esqueleto con otro.

Otra limitante es que el método de voxelización, permite escribir solamente imágenes de una resolución máxima de $255*255*255$. Esta resolución máxima limita el tamaño final del esqueleto y puede presentar un problema al momento de buscar un umbral y resolución óptima para trabajar con los esqueletos. Con las computadoras actuales una resolución mas grande de la mencionada arriba puede causar que ciertas partes de la metodología se vuelvan relativamente imposibles de procesar. Por esta limitante en poder de procesamiento, los cálculos se limitaron a imágenes de 100 voxeles por dimensión. Con esta limitante, logramos la robustez en escala.

El algoritmo húngaro, al ser un método de optimización en tiempo $O(n^3)$, tiene una limitante en cuanto al tamaño de la matriz que se puede utilizar, para que el problema no se convierta en un problema intratable (para el poder de procesamiento de los procesadores actuales).

La computadora en la cual se realizaron las pruebas, tiene las siguientes especificaciones de Hardware:

- Windows 7 home Basic (64 bits)
- Procesador: Intel Core i3 2.10 GHz
- Memoria RAM: 4.00 GB
- Tarjeta de Video: GeForce GT 520M. 1030 MHz reloj de procesador

En relación al software usado para la obtención de resultados, se usaron las siguientes versiones:

- Blender 2.69
- Python 2.7.6
- Munkres 1.0.6
- PIL 1.1.7
- Numpy 1.8.0

- Scipy 0.13.1
- Matplotlib 1.3.1

La mayor limitante que se encontró durante el desarrollo del método, fue que los esqueletos son sensibles a la orientación, y hay pérdida o ganancia de información, tomando en cuenta que los esqueletos deben ser 8 conectados. Una prueba más detallada de esto, se encuentra en el **Anexo F**. Aunque el análisis se realizó en 2D, los resultados obtenidos nos dan una idea muy clara de lo que pasaría al usar métodos en imágenes 3D.

Por último, la obtención del umbral propuesto en la sección 5.3 requiere la comparación de un subgrupo de imágenes del área o tema del cual se desee obtener dicho umbral. Por ejemplo si se desea obtener un umbral para decidir si la tomografía obtenida del cerebro de una persona, muestra un tumor; se debe tener una base de cerebros sanos para los cuales se haya obtenido el índice CM, y definir su umbral de similitud. Esto se considera una limitante debido a la dificultad de obtener dicho conjunto de imágenes para cada área.

4.4 EXPERIMENTOS

Parte de los experimentos realizados se enfocaron en la generación del umbral con base en los índices CM de nuestra base de datos de imágenes caninas. La Tabla 5 muestran los voxeles en común, positivos y negativos de cada uno de los esqueletos comparados contra todos los demás.

En la Tabla 6 tenemos los costos mínimos resultantes de la comparación de cada una de las imágenes caninas, una contra todas las demás. Adicionalmente se registraron los tiempos que le tomo al método completar el proceso entero.

		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Perro 1	=		2	1	3	2	2	3	1	1	0
	+		303	304	302	303	303	302	304	304	305
	-		265	236	280	257	242	280	347	264	293
Perro 2	=	2		1	3	10	10	0	2	0	1
	+	265		266	264	257	257	267	265	267	266
	-	303		236	280	249	234	283	346	265	292
Perro 3	=	1	1		0	0	0	0	0	0	0
	+	236	236		237	237	237	237	237	237	237
	-	304	266		283	259	244	283	348	265	293
Perro 4	=	3	3	0		8	16	4	15	0	2
	+	280	280	283		275	267	279	268	283	281
	-	302	264	237		251	228	279	333	265	291
Perro 5	=	2	10	0	8		12	1	5	2	0
	+	257	249	259	251		247	258	254	257	259
	-	303	257	237	275		232	282	343	263	293
Perro 6	=	2	10	0	16	12		0	3	0	5
	+	242	234	244	228	232		244	241	244	239
	-	303	257	237	267	247		283	345	265	288
Perro 7	=	3	0	0	4	1	0		4	15	0
	+	280	283	283	279	282	283		279	268	283
	-	302	267	237	279	258	244		344	250	293
Perro 8	=	1	2	0	15	5	3	4		1	1
	+	347	346	348	333	343	345	344		347	347
	-	304	265	237	268	254	241	279		264	292
Perro 9	=	1	0	0	0	2	0	15	1		14
	+	264	265	265	265	263	265	250	264		251
	-	304	267	237	283	257	244	268	347		279
Perro 10	=	0	1	0	2	0	5	0	1	14	
	+	293	292	293	291	293	288	293	292	279	
	-	305	266	237	281	259	239	283	347	251	

Tabla 5 -Voxeles positivos y negativos

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	
P_1		2663	2812	2167	2230	1793	2127	3144	1279	3107	costo mínimo
		204	480	310	359	370	310	439	276	503	tiempo (seg)
P_2	2663		2165	2213	1073	1126	2457	2709	2906	2996	costo mínimo
	473		198	498	254	210	390	514	482	502	tiempo (seg)
P_3	2812	2165		2905	2193	2078	3132	2885	3249	3397	costo mínimo
	613	335		475	313	260	520	738	441	580	tiempo (seg)
P_4	2167	2213	2905		1960	1406	1889	1217	1276	1450	costo mínimo
	372	266	434		267	276	243	197	270	231	tiempo (seg)
P_5	2230	1073	2193	1960		920	2428	2488	2703	2680	costo mínimo
	525	164	233	344		154	423	551	308	468	tiempo (seg)
P_6	1793	1126	2078	1406	920		1943	1996	2034	2227	costo mínimo
	529	197	199	232	132		377	531	305	388	tiempo (seg)
P_7	2127	2457	3132	1889	2428	1943		2083	1031	1752	costo mínimo
	352	310	373	191	303	264		390	237	314	tiempo (seg)
P_8	3144	2709	2885	1217	2488	1996	2083		1642	1315	costo mínimo
	322	458	690	373	180	301	334		231	182	tiempo (seg)
P_9	1279	2906	3249	1276	2703	2034	1031	1642		1220	costo mínimo
	278	214	421	354	412	225	192	203		163	tiempo (seg)
P_{10}	3107	2996	3397	1450	2680	2227	1752	1315	1220		costo mínimo
	312	207	286	202	361	190	182	211	149		tiempo (seg)

Tabla 6 - Costo del algoritmo Húngaro y tiempos de proceso en figuras caninas

5. DISCUSIÓN DE RESULTADOS

5.1 RESULTADOS

En esta sección se describen los principales resultados obtenidos de los índices CM calculados de cada imagen.

Las características de la computadora en la cual se registraron estos tiempos, se pueden ver en la sección de limitaciones clave.

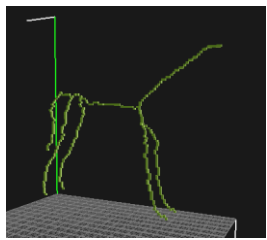
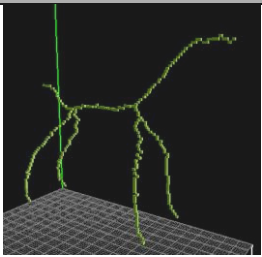
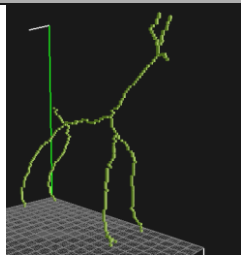
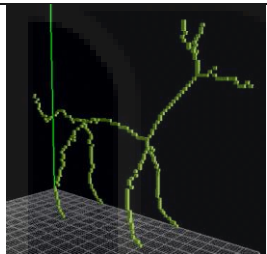
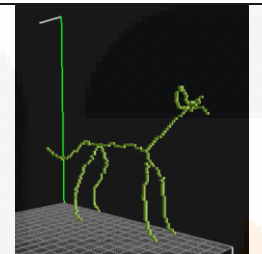
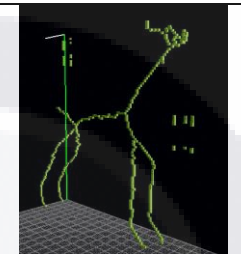
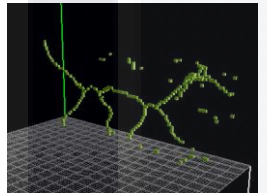
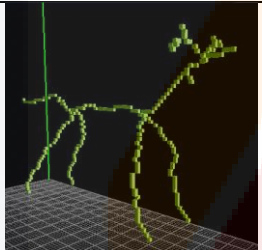
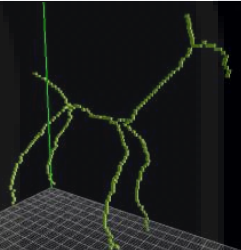
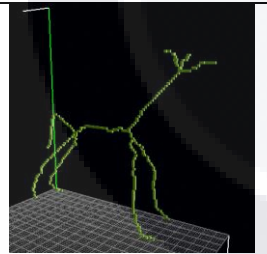
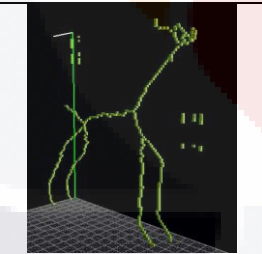
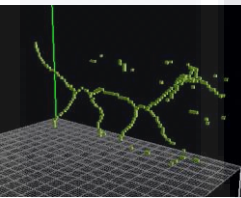
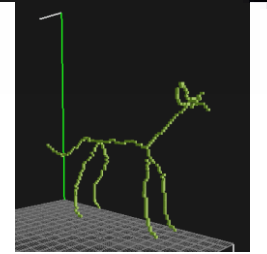
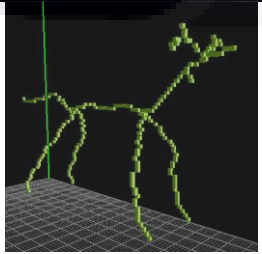
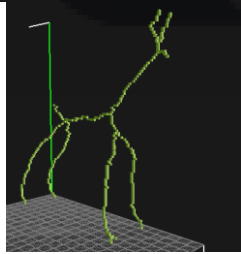
Los esqueletos de las imágenes que más se parecen entre sí (dado que el valor obtenido por el algoritmo Húngaro es el mínimo) son el esqueleto del Perro5 y el esqueleto del Perro6. Así también, las imágenes que menos se parecen son la del Perro3 y la del Perro10.

En la Tabla 7 se muestran todas las imágenes caninas, y a su derecha, la imagen que más se parece y la que menos se parece, de acuerdo al índice CM de cada comparación.

Un dato importante es que aún cuando la imagen Perro6 y la imagen Perro4 comparten más voxeles en común (16), no son las que más parecido tiene.

Tomando como ejemplo la imagen Perro1 y la información de la Tabla 7, el mayor y menor número de voxeles en común son con las imágenes Perro4 y Perro10 respectivamente. Sin embargo, el algoritmo Húngaro nos arroja el costo para poder comprobar que Perro1 tiene mayor similitud con la imagen Perro9 y menor similitud con la imagen Perro8.

Esta comparativa nos permite observar que el número de voxeles en común no es una característica que nos ayude a encontrar un par de imágenes que tenga mayor similitud.

Imagen	Mayor Similitud	Índice CM	Menor Similitud	Índice CM
 Perro1	 Perro9	2.53	 Perro10	6.26
 Perro2	 Perro5	2.19	 Perro8	5.66
 Perro3	 Perro6	3.85	 Perro7	5.94
 Perro4	 Perro8	1.95	 Perro3	5.58
 Perro5	 Perro6	1.83	 Perro10	5.17

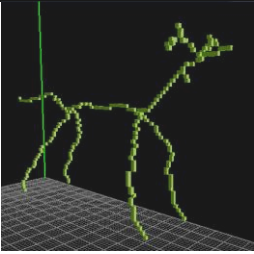
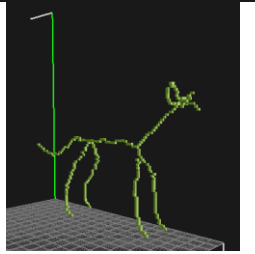
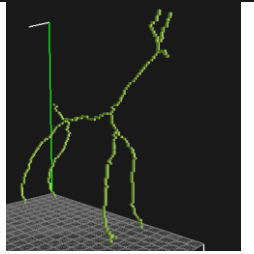
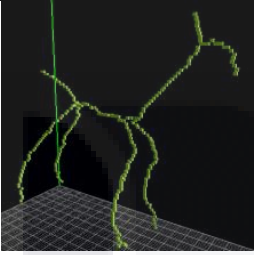
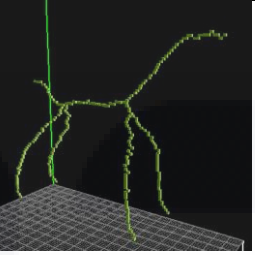
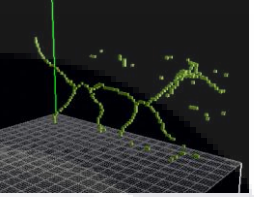
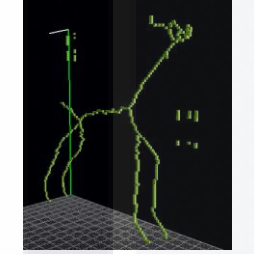
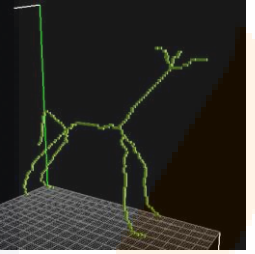
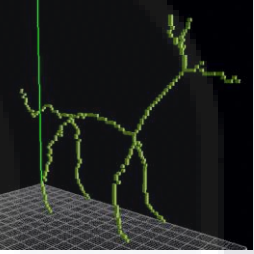
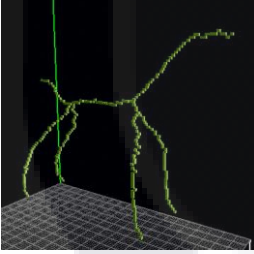
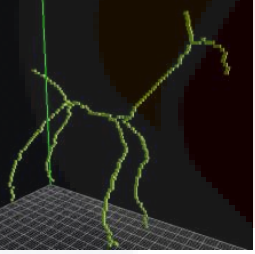
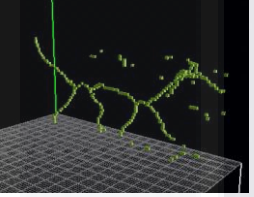
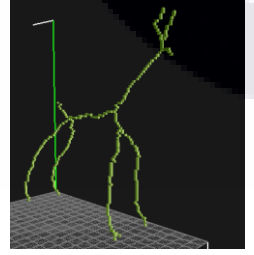
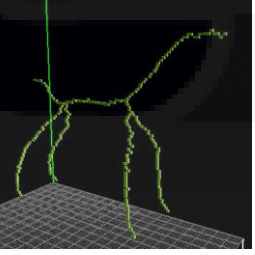
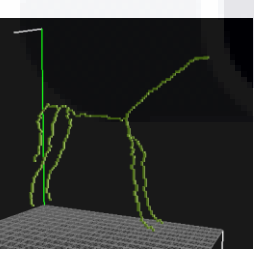
 Perro6	 Perro5	1.83	 Perro10	4.20
 Perro7	 Perro9	1.87	 Perro3	5.94
 Perro8	 Perro4	1.95	 Perro2	5.66
 Perro9	 Perro7	1.87	 Perro3	5.32
 Perro10	 Perro9	2.30	 Perro1	6.26

Tabla 7 - Similitud entre imágenes caninas

5.2 MEDIDA DE SIMILITUD

Analizando las imágenes caninas eskeletonizadas presentadas en la Tabla 3, se puede apreciar similitudes significativas en la estructura de los esqueletos. Sin embargo, los datos presentados en la Tabla 5 acerca de los voxeles en común de ambos esqueletos, tienden a 0; esto debido a las diferencias en los centros de masa de los esqueletos. Lo anterior ocasiona que el costo obtenido con el algoritmo Húngaro sea muy elevado, debido a la gran cantidad de voxeles positivos y negativos que se tienen al comparar las imágenes.

En base a este análisis, debemos proponer una medida de similitud basada en la cantidad de voxeles negativos y positivos al comparar dos imágenes; y el costo obtenido con el algoritmo Húngaro. La métrica que se propone se muestra en la ecuación (21):

$$CM_{i,j} = \frac{DM_{i,j}}{VNP_{i,j}} \quad (21)$$

Donde:

$CM_{i,j}$ = Índice de proporción distancia – masa de las imágenes i y j

$DM_{i,j}$ = costo mínimo de las distancias entre voxeles positivos y negativos de las imágenes i y j

$VNP_{i,j}$ = número de voxeles negativos y positivos de las imágenes i y j

5.3 DEFINICIÓN DE UMBRAL PARA IMÁGENES CANINAS

Para ejemplificar la utilización de la métrica propuesta, vamos a aplicarlo en las imágenes caninas mostradas en la sección anterior. La Tabla 9 muestra el índice CM para las combinaciones de las 10 imágenes caninas.

Aplicando la prueba de normalidad Shapiro-Wilk [43], se puede validar estadísticamente que los índices CM de las figuras caninas poseen una distribución normal con un p-valor de 0.1499. Esto nos permite proponer un umbral basado en los percentiles de la distribución normal en un nivel de confianza, típicamente de 95%, para definir un criterio de similitud entre imágenes.

Para este experimento el percentil 0.95% obtenido con la distribución normal nos indica que se pueden considerar similares aquellas imágenes cuyo CM sea menor o igual a 6 unidades. En este ejemplo las imágenes Perro1 y Perro10 no serian similares, debido a que su índice CM es de 6.26 (de acuerdo a la Tabla 8). El resto de las imágenes serian similares entre sí, ya que se encuentran dentro del 95% de confianza obtenido con la distribución del índice CM representado por el umbral de 6 unidades.

En la siguiente tabla, se realizaron experimentos con más imágenes: un gorila, un gallo, una vaca y un caballo.

Se realizaron las siguientes comparaciones, y se reportan los Índices CM, junto con algunos valores importantes en la Tabla que se muestra a continuación.

	Perro 1	Perro 2	Perro 3	Perro 4	Perro 5	Perro 6	Perro 7	Perro 8	Perro 9	Perro 10
Perro 1		4.69	5.21	4.32	3.83	3.30	4.09	5.61	2.53	6.26
Perro2	4.69		4.12	4.06	2.19	2.34	4.96	5.66	4.99	5.45
Perro3	5.21	4.12		5.58	3.93	3.85	5.94	4.43	5.32	5.81
Perro4	4.32	4.06	5.58		3.26	2.36	3.22	1.95	2.25	2.73
Perro5	3.83	2.19	3.93	3.26		1.83	4.43	4.78	5.31	5.17
Perro6	3.30	2.34	3.85	2.36	1.83		3.18	3.34	3.65	4.20
Perro7	4.09	4.96	5.94	3.22	4.43	3.18		3.64	1.87	3.32
Perro8	5.61	5.66	4.43	1.95	4.78	3.34	3.64		0	2.85
Perro9	2.53	4.99	5.32	2.25	5.31	3.65	1.87	2.85		0
Perro10	6.26	5.45	5.81	2.73	5.17	4.20	3.32	2.06	2.30	

Tabla 8 – Índice CM en figuras caninas

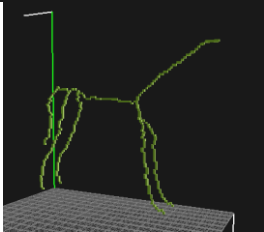
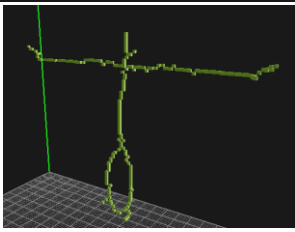
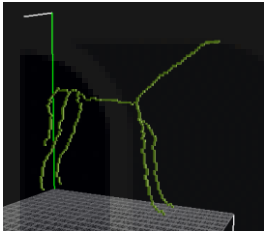
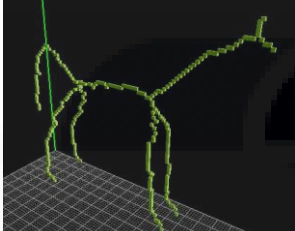
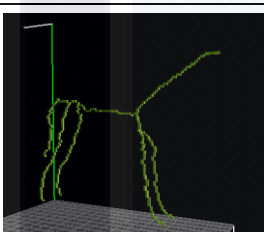
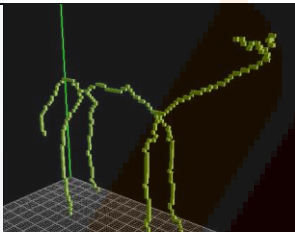
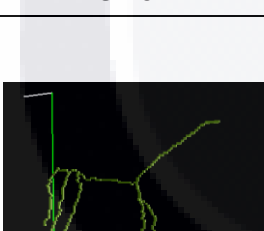

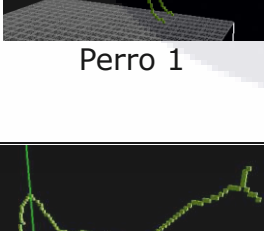
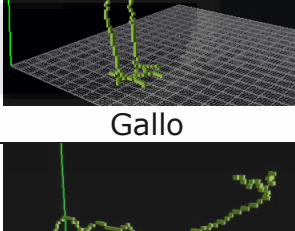
Imagen 1	Imagen 2	Costo mínimo	Voxeles positivos	Voxeles negativos	Índice CM
 Perro 1	 Gorila	3317	301	206	6.54
 Perro 1	 Caballo	1075	304	275	1.86
 Perro 1	 Vaca	2268	301	290	3.84
 Perro 1	 Gallo	7367	305	333	11.55
 Caballo	 Vaca	1216	290	276	2.16

Tabla 9 – Índices CM de imágenes diversas

Analizando los Índices resultantes de la tabla anterior, podemos observar que las imágenes Perro1 y caballo son las que tienen mayor similitud, dado a el bajo valor de su índice CM. De la misma manera, los esqueletos de las imágenes caballo y vaca.

El contraejemplo lo podemos encontrar al analizar el Índice CM de los esqueletos de las imágenes Perro1 y el gallo.



CONCLUSIONES

El objetivo principal de la presente investigación es saber si dos esqueletos de imágenes en tres dimensiones son parecidas, y a qué nivel.

Nuestra **primera hipótesis**: uso del algoritmo Húngaro ayuda a la obtención de un valor aproximado adecuado en la comparación de dos o más objetos, se rechaza, ya que como se puede observar en la Tabla 5, aún cuando las figuras caninas tienen bastante parecido, los voxeles en común que tienen tienden a 0, por esta razón el costo obtenido con el algoritmo Húngaro es muy elevado. Esto provoca que la primera hipótesis se rechace. El costo mínimo obtenido a través del algoritmo Húngaro, por su propia cuenta, no ayuda para la obtención de un umbral de similitud.

La **segunda hipótesis** no se rechaza. Los descriptores obtenidos de la imagen, (centro de masa, ejes principales), ayudarán a la correcta clasificación de las imágenes. Estos son piedra angular para la orientación de cada voxel sobre el centro de masa y el cálculo de distancias entre voxeles positivos y negativos. El uso de esqueletos reduce el volumen de los voxeles en un 99.9% por lo que los esqueletos, al mantener la topología de la imagen original, nos ayudan al reducir el volumen de información a ser procesada. Aún cuando no es definitiva, es un descriptor de la imagen que ayuda en gran manera al procesamiento de imágenes.

En relación a la **tercera hipótesis**: orientación. Efectivamente los ángulos de Euler nos ayudan en la orientación, así que esta hipótesis no se rechaza, sin embargo, durante la implementación de la metodología, nos dimos cuenta que el método es sensible a rotación. Dado que la imagen voxelizada siendo representación en elementos discretos, al rotar una imagen, y realizar una esqueletonización, el método usado tiene un orden de procesamiento. En otras palabras, el método comienza con el procesamiento de la parte superior, y los constructos usados eliminan voxeles en diferente orden, lo cual afecta el resultado final del esqueleto. Aún antes de obtener el esqueleto, se puede

TESIS TESIS TESIS TESIS TESIS

observar que una rotación afecta en el conteo de los voxeles, así se demuestra en múltiples ocurrencias que el método es sensible a rotación.

En la Tabla 8, se pueden observar los tiempos de cómputo que tarda nuestro proceso en terminar la evaluación de la comparación. Tomando en cuenta que el equipo con el que se realizaron los experimentos no es de última generación, los tiempos son bastante razonables, ya que el máximo tiempo registrado (entre las dos imágenes que mayor costo mínimo tienen) es de 738 segundos (12 minutos aproximadamente). Para efectos de los tamaños de nuestros esqueletos, todos los tiempos son medibles, por lo tanto nuestra **cuarta hipótesis** no se rechaza.

Al unir todas las partes de la metodología en un solo programa, se redujo el tiempo de procesamiento y la interacción humana necesaria para llevar cada imagen o archivo resultante hacia el siguiente paso. El programa computacional requiere que se alimente con dos imágenes en formato .obj, y al término del mismo, obtenemos el Índice CM: nuestro indicador de similitud entre las dos imágenes procesadas. El programa funciona sin interacción más que al inicio y al final para leer el Índice resultante; con esta evidencia, podemos decir que nuestra **quinta hipótesis** no se rechaza.

En relación a la **sexta hipótesis**, esta es rechazada parcialmente, por la parte de la rotación, ya que el método es sensible a rotación, como se pudo demostrar en el caso de dos dimensiones (ver **anexo D**). En el apartado de invariancia en escala, dado que nuestras imágenes originales son vectoriales, el escalamiento de las imágenes no representa ganancia ni pérdida de información. Con esta afirmación, y estableciendo el tamaño estándar de 100*100*100 para la voxelización y la esqueletonización de las imágenes, logramos la invariancia en escala.

Para el caso de la traslación, el método de voxelización y esqueletonización automáticamente hace la traslación del objeto .obj hacia el centro de coordenadas (X, Y, Z), por lo que no importa en donde se encuentre la imagen

posicionada, al momento de generar los archivos necesario, el objeto es trasladado automáticamente; logrando así la invariancia en escala y traslación.

Finalmente, uno de los objetivos de esta Tesis, era proponer un umbral para decidir de forma automatizada si dos imágenes 3D son similares entre sí. El índice CM propuesto en este documento, trata de relacionar el costo mínimo obtenido por el Algoritmo Húngaro y la cantidad de voxeles positivos y negativos para obtener una métrica confiable para poder comparar los esqueletos de ambas imágenes. Dicho índice nos permite calcular un umbral de similitud basado en un conjunto de imágenes provenientes de la misma área que las imágenes que se deseen comparar. De esta manera, la métrica propuesta se puede aplicar a cualquier conjunto de esqueletos de la misma área, con la limitante de la escasa cantidad de imágenes 3D disponibles para cada área (flores, animales, cerebros, objetos).

GLOSARIO

Algoritmo: Procedimiento ordenado para la resolución de problemas. El método se define en un número finito de instrucciones bien definidas para la obtención de un resultado.

Algoritmo Húngaro: Algoritmo de combinatoria que soluciona problemas de asignación en tiempo polinomial. El resultado de este algoritmo es el costo mínimo de un sistema de optimización.

Blender: Programa gratuito de edición y creación de imágenes en tres dimensiones

Esqueleto: se logra al aplicar métodos de adelgazamiento a una imagen en tres dimensiones. Es un descriptor de dicha imagen que nos permite trabajar con diversas heurísticas y meta heurísticas ya que el volumen de información de un esqueleto es de aproximadamente del 0.1% en relación al volumen de la imagen original.

Heurística: Técnica de resolución de problemas basados cuyos resultados no garantizan ser óptimos, sin embargo permiten encontrar problemas con un nivel de aceptación bastante alto cuando el procesamiento exhaustivo no es viable.

Kinect: Dispositivo de interfaz natural de usuario que permite interactuar a las personas con dispositivos electrónicos sin la necesidad de algún dispositivo de entrada conocido, como lo son el teclado y el ratón electrónico.

Meta heurísticas: Heurística o procedimiento de alto nivel diseñada para encontrar una solución aceptable a un problema de asignación; una de sus características es que puede trabajar sin problema aún en equipos de limitado poder computacional.

Raw: Archivo con $n*m*r$ posiciones, en donde las variables n , m y r representan las dimensiones del cubo que encierra a una imagen de tres

dimensiones. Este archivo generado, consiste en una cadena de caracteres de distancia $n*m*r$, que consta de dos posibles valores, 0 para los voxeles apagados y 1 para todos aquellos voxeles prendidos.

Thinvox: tipo de archivo generado por el programa de adelgazamiento de imágenes en tres dimensiones. Este archivo .thinvox, contiene todas las posiciones de los voxeles prendidos, medido en coordenadas de un sistema de tres dimensiones.

Valores propios: el valor no nulo por el cual se cambia un vector para lograr su múltiplo escalar, recibe el nombre de valor propio.

Vectores propios: son aquellos vectores no nulos, que cuando son transformados por un operador, dan como salida un nuevo vector que es múltiplo escalar del vector original.

Voxel: Elemento volumétrico que representa la unidad mínima discretizada en un espacio tridimensional. Es la extrapolación a tres dimensiones de un pixel.

BIBLIOGRAFÍA

- [1] M.C. Nisao Ogata Aguilar, Programa experto en identificación de árboles tropicales comunes es del Sureste de México, Belice y Guatemala, Informe final SNIB-CONABIO proyecto No. K018, 1999.
- [2] Dallwitz, Michael John; Toni Anne Paine; E. J. Zurcher, User's guide to the DELTA system: a general system for processing taxonomic descriptions, Vol. 136, Commonwealth and Industrial Research Organization, Australia, 1993.
- [3] Krinidis, Stelios; Vassilios Chatzis, A skeleton family generator via physics-based deformable models, Image Processing, IEEE Transactions on 18.1, 2009, 1-11.
- [4] Daliri, Mohammad Reza; Vincent Torre, Robust symbolic representation for shape recognition and retrieval, Pattern Recognition 41.5, 2008, 1782-1798.
- [5] Liu, Jie; Jigui Sun; Shengsheng Wang, Pattern recognition: An overview, IJCSNS International Journal of Computer Science and Network Security 6.6, 2006, 57-61.
- [6] Lowe, David G, Object recognition from local scale-invariant features, Computer vision, 1999, The proceedings of the seventh IEEE international conference on, Vol. 2. IEEE, 1999.
- [7] Adel, Abdelhadi; Mouss Leila-Hayet, Efficient Tool for the Recognition of the Leaves of Plants, IJSCI International Journal of Computer Science Issues 8.2, 2011.
- [8] Deselaers, Thomas; Daniel Keysers; Hermann Ney. Clustering visually similar images to improve image search engines, hist 1.1, 2003, 1.
- [9] Deselaers, Thomas, Features for image retrieval, Rheinisch-Westfälische Technische Hochschule, Technical Report, Aachen, 2003.
- [11] Gonzalez, Rafael C; Richard E. Woods; Steven L. Eddins, Digital image processing using MATLAB, Vol. 2, Knoxville, Gatesmark Publishing, 2009.
- [11] Nils J. Nilsson, Introduction to machine learning, Stanford online courses, 2011.
- [12] Andrew Ng, Conference on Stanford University, Machine Learning Talk, 2011.

- [13] Cornea, Nicu D; Deborah Silver; Patrick Min, Curve-skeleton properties, applications, and algorithms, Visualization and Computer Graphics, IEEE Transactions on 13.3, 2007, 530-548.
- [14] Wang, Yu-Shuen; Tong-Yee Lee, Curve-skeleton extraction using iterative least squares optimization, Visualization and Computer Graphics, IEEE Transactions on 14.4, 2008, 926-936.
- [15] Jonker, Roy; Anton Volgenant, A shortest augmenting path algorithm for dense and sparse linear assignment problems, Computing 38.4, 1987, 325-340.
- [16] Shu, Xin; Xiao-Jun Wu, A novel contour descriptor for 2D shape matching and its application to image retrieval, Image and vision Computing 29.4, 2011, 286-294.
- [17] Cruz, Hermilo Sánchez; Ramón M. Rodríguez Dagnino, Normalization of a 3D-Shape Similarity Measure with Voxel Representation, Computación y Sistemas 10.4, 2007, 372-387.
- [18] Sánchez-Cruz, Hermilo; Ernesto Bribiesca, A method of optimum transformation of 3D objects used as a measure of shape dissimilarity, Image and Vision Computing 21.12, 2003, 1027-1036.
- [19] Nooruddin, Fakir S.; Greg Turk, Simplification and repair of polygonal models using volumetric techniques, Visualization and Computer Graphics, IEEE Transactions on 9.2, 2003, 191-205.
- [20] Xu, Dong; Hua Li, Geometric moment invariants, Pattern recognition 41.1, 2008, 240-249.
- [21] Shoemake, Ken, Animating rotation with quaternion curves, ACM SIGGRAPH computer graphics 19.3, 1985, 245-254.
- [22] Pang-Ning, Tan; Michael Steinbach; Vipin Kumar, Introduction to data mining, Library of Congress, 2006.
- [23] Krause, Eugene, Taxicab geometry: An adventure in non-Euclidean geometry, DoverPublications, com, 1987.
- [24] Danielsson, Per-Erik, Euclidean distance mapping, Computer Graphics and image processing 14.3, 1980, 227-248.
- [25] Smisek, Jan; Michal Jancosek; Tomas Pajdla, 3D with Kinect, Consumer Depth Cameras for Computer Vision, Springer London, 2013, 3-25.

[26] Sanna_, Andrea et al, A Kinect-based natural interface for quadrotor control, Entertainment Computing, 2013.

[27] Hernandez-Lopez Jose-Juan et al, Detecting objects using color and depth segmentation with Kinect sensor. Procedia Technology 3, 2012, 196-204.

[28] Slabaugh, Gregory G, Computing Euler angles from a rotation matrix, Retrieved on August 6 1999, 2000.

[29] Zhang, Zhengyou, Microsoft Kinect sensor and its effect, Multimedia, IEEE 19.2, 2012, 4-10.

[30] Shotton, Jamie et al, Real-time human pose recognition in parts from single depth images, Communications of the ACM 56.1, 2013, 116-124.

[31] Izadi, Shahram, et a, KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera, Proceedings of the 24th annual ACM symposium on User interface software and technology, ACM, 2011.

[32] Stowers, John; Michael Hayes; Andrew Bainbridge-Smith, Altitude control of a quadrotor helicopter using depth map from Microsoft Kinect sensor, Mechatronics (ICM), 2011 IEEE International Conference on. IEEE, 2011.

[33] Krause, Eugene F, Taxicab geometry, The Mathematics Teacher 66.8, 1973, 695-706.

[34] Kuhn, Harold W, The Hungarian method for the assignment problem, Naval research logistics quarterly 2.1-2, 1955, 83-97.

[35] Duda, Richard O.; Peter E. Hart; David G. Stork, Pattern classification, John Wiley & Sons, 2012.

[36] Pavlidis, Theodosios, Structural pattern recognition, Springer Series in Electrophysics, Berlin: Springer, 1977 1.

[37] Gonzalez, Rafael C.; Michael G. Thomason, Syntactic pattern recognition: An introduction, 1978.

[38] Watanabe, Satosi, Pattern recognition: human and mechanical, John Wiley & Sons, Inc., 1985.

[39] Jain, Anil K.; M. Narasimha Murty; Patrick J. Flynn, Data clustering: a review, ACM computing surveys (CSUR) 31.3, 1999, 264-323.

[40] Duda, Richard O.; Peter E. Hart; David G. Stork, Pattern classification, John Wiley & Sons, 2012.

- [41] Holland, John H, Adaption in natural and artificial systems, 1975.
- [42] Goldberg, David E.; and John H. Holland, Genetic algorithms and machine learning, Machine learning 3.2, 1988, 95-99.
- [43] Royston, J. P., An extension of Shapiro and Wilk's W test for normality to large samples, Applied Statistics, 1982, 115-124.
- [44] Real, Raimundo; Juan M. Vargas, The probabilistic basis of Jaccard's index of similarity, Systematic biology, 1996, 380-385.
- [45] Reinelt, Gerhard, TSPLIB – A traveling salesman problem library, ORSA journal on computing 3.4, 1991, 376-384.
- [46] Danielsson, Per-Erik, Euclidean distance mapping, Computer Graphics and image processing 14.3, 1980, 227-248.
- [47] Lv jianbiao. <http://www.3dmodelfree.com/>
- [48] Ivan Pisklyukov. <http://archive3d.net>
- [49] Registration Private. <http://www.3dxtras.com>
- [50] Registration Private. <http://www.blendswap.com/>
- [51] TurboSquid, Inc. <http://www.turbosquid.com>
- [52] Google Inc. <http://sketchup.google.com>
- [53] Registration Private. <http://artist-3d.com>
- [54] Princeton Consultants. <http://www.cs.princeton.edu/~min/binvox/>
- [55] Princeton Consultants. <http://www.cs.princeton.edu/~min/thinvoy/>
- [56] Princeton Consultants. <http://www.cs.princeton.edu/~min/viewvox/>
- [57] Microsoft Corp <http://www.microsoft.com/en-us/kinectforwindows/>

ANEXOS



ANEXO A: FORMATOS DE IMÁGENES

Los formatos que se mencionan a lo largo del documento son:

3DS	Asociación primaria: 3D Studio Compañía: Autodesk, Inc. Tipo Mime: application/x-3ds, image/x-3ds Caracteres hexadecimales identificados: 4D 4D , ASCII: MM Identificador de Programa: 3dsFile Ligas relacionadas: RapidForm Free Viewer
------------	--

El archivo de tipo 3DS es primordialmente asociado a el programa '3D Studio' desarrollado por Autodesk, Inc.

DAE	Asociación primaria: 3D Studio Compañía: Autodesk, Inc. Tipo Mime: application/x-3ds, image/x-3ds Caracteres hexadecimales identificados: 4D 4D , ASCII: MM Identificador de programa: 3dsFile Ligas relacionadas: RapidForm Free Viewer
------------	--

Caracteres hexadecimales reconocidos: 3C 3F 78 6D 6C 20 76 65 72 73 69 6F 6E 3D 22 31 2E 30 22 20 65 6E 63 6F 64 69 6E 67 3D 22 75 74 66 2D 38 22 ,
 ASCII: .?xml.version="1.0".encoding="utf-8"

El archivo del tipo DAE esta primordialmente relacionado con 'COLLADA (COLLABorative Design Activity)' del grupo Khronos. COLLADA ha sido establecido como un estándar abierto de un formato digital para aplicaciones interactivas en tres dimensiones. COLLADA actualmente es un estándar del grupo Group.

PLY	Asociación Primaria: Formato de Modelos Poligonal Compañía: Stanford University, Georgia Tech Clasificación de archivo: CAD/CAM Ligas relacionadas: Formato de archivo
------------	---

El archivo con formato PLY se asocia primordialmente asociado con otro formato llamado "formato del poliedro" creado en la universidad de Stanford, Georgia. Un archivo PLY, consiste en un encabezado seguido de una larga lista de vértices y después otra larga lista de polígonos. Muchos programas pueden manejar este archivo.

OBJ	Asociación primaria: Código de objeto relocalizable Tipo Mime: aplicación/octet-stream Caracteres hexadecimales identificados: 80 Ligas relacionadas: Specification , TrID
------------	---

El archivo OBJ se asocia primordialmente con el término "código de objeto relocalizable". Usa "TrID" para determinar cual compilador genera el archivo de salida.

RAW

Tipo de archivo propietario del método que se usa para obtener las imágenes voxelizadas y esqueletos de las imágenes 3D. Este archivo contiene un solo renglón de información, con solo dos valores posibles: 0 y 1.

Estos valores, representan los dos estados que pueden tener cada voxel de un espacio tridimensional. 1 para cuando el voxel se encuentra lleno y 0 para cuando el voxel es parte del fondo de la imagen.

Binvox & Thinvox

Tipos de archivo propietarios del método en uso. Estos archivos almacenan la posición de cada voxel, así como más información relevante de la posición de cada voxel.

Aún cuando el método usado es código abierto, el formato y orden de la información de estos archivos, no cuentan con una descripción suficiente ni un uso masivo en el área de visión computacional para tener mayor información o referencias.



ANEXO B: ÁNGULOS DE EULER

Los ángulos de Euler ayudan en la descripción de la orientación de un cuerpo rígido. Para describir dicha orientación en el espacio euclidiano tridimensional, se necesitan 3 parámetros denotados como alpha, beta y gamma.

Los ángulos de Euler también representan una secuencia de tres rotaciones elementales: rotaciones sobre los ejes de un sistema coordenado. Por ejemplo, una rotación sobre el eje Z, por un ángulo alpha.

Cualquier orientación puede ser lograda realizando una rotación elemental sobre cada uno de los ejes de coordenadas de un sistema fijo (rotación extrínseca), o bien, alrededor de los ejes de un sistema coordenado rotatorio; el cual originalmente se encuentra alineado con el sistema de coordenadas rotatorio primario, y después de realizar las rotaciones sobre cada eje, termina rotado en los 3 ejes del sistema (rotaciones intrínsecas). En algunos casos, esto es llamado un sistema de coordenadas locales. Sin considerar la posibilidad de que existan dos sistemas con diferentes convenciones acerca de las rotaciones sobre los ejes, existen doce posibles secuencias de rotaciones:

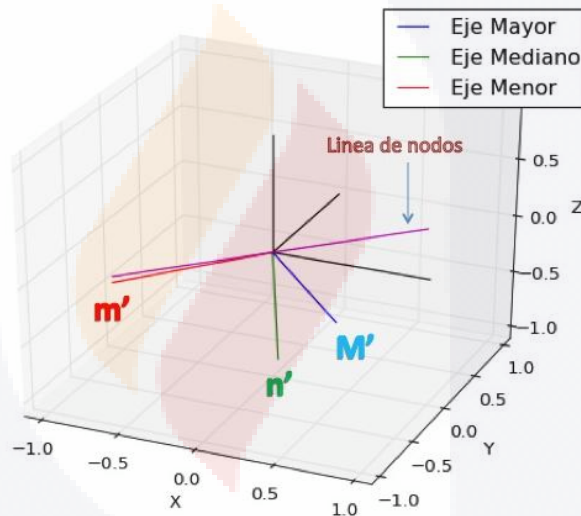
Rotaciones intrínsecas					
(Z, X, Z)	(X, Y, X)	(Y, Z, Y)	(Z, Y, Z)	(X, Z, X)	(Y, X, Y)
Rotaciones extrínsecas					
(X, Y, Z)	(Y, Z, X)	(Z, X, Y)	(X, Z, Y)	(Z, Y, X)	(Y, X, Z)

ANEXO C: EJES PRINCIPALES Y MATRIZ DE ROTACION

En este anexo, haremos un cálculo manual de los ángulos de Euler, usando los ejes principales de el cono en su posición natural y de los ejes X, Y y Z de un sistema coordenado normal. Estos ángulos se validarán contra su representación grafica.

Como punto de partida, tenemos nuestros dos sistemas: el coordenado y el que está representado por los vectores propios del cono.

En la Figura siguiente, se puede observar la representación grafica de ambos sistemas.



En la Figura se pueden observar ambos sistemas... Cada vector tiene su correspondiente valor numérico de la Tabla siguiente.

<p>M = [1,0,0] n = [0,1,0] m = [0,0,1]</p>	<p>M' = [0.7071, -0.7050, -0.0535] (eje mayor) n' = [0, 0.0757, -0.9971] (eje mediano) m' = [-0.7071, -0.7050, -0.0535] (eje menor)</p>
Sistema original	Sistema del cono

Teniendo ambos sistemas, podemos proceder a realizar los cálculos necesarios. Las formulas que se muestran a continuación, son aquellas que se necesitan para realizar el cálculo.

$$\cos(a) = \frac{(M' \cdot n)}{\|M'\| \|n\|}$$

$$\cos(b) = \frac{(m' \cdot m)}{\|m'\| \|m\|}$$

$$\cos(c) = \frac{(N \cdot M)}{\|N\| \|M\|}$$

$$N = m' * m$$

$$\|M\| = \sqrt{M_x^2 + M_y^2 + M_z^2}$$

Donde:

N = línea de nodos

$\|M\|$ = modulo de M

M, m, n = vectores del sistema original

M', m', n' = vectores del sistema del cono

ANEXO D: CONTEO DE PARIDAD y APUÑALAMIENTO EN RAYOS

Estos métodos ayudan en la obtención de un modelo voxelizado de un poliedro. El modelo voxelizado permite utilizar esta información en consecuentes procedimientos estadísticos, matemáticos y de optimización.

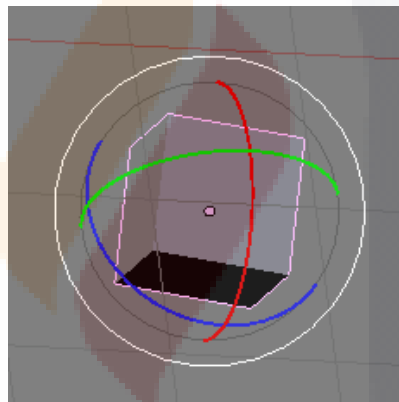
La metodología usará para lograr la voxelización de la imagen, está dividida principalmente en dos partes: conteo de paridad y apuñalamiento de rayos.

El método de conteo de paridad, consiste en trazar líneas desde el centro de la imagen hacia fuera de la misma, en diversas direcciones. Si la línea atraviesa un número par de polígonos, el voxel es clasificado como interno o prendido, si el número de ocurrencias es impar, el voxel se clasifica como parte del contorno de la imagen.

La diferencia principal con el método de apuñalamiento en rayos, es que en este último, las ocurrencias entre el primero polígono y el último son usadas como elementos delimitados del objeto y cualquier elemento que termine adentro de estos dos elementos colindantes, se considera como voxel interno.

ANEXO E: ROTACIÓN CON BLENDER

Podemos realizar una rotación con Blender, simplemente se selecciona la imagen y se puede realizar una rotación sobre cualquiera de sus tres ejes. Con la imagen seleccionada, se puede proceder de varias maneras, la primera y más fácil, es presionar la tecla R, para entrar en el modo de rotación en Blender. Este modo colocara 3 circunferencias alrededor de la imagen, cada una de estas circunferencias, al moverlas individualmente, nos permite rotar sobre cualquiera de sus 3 ejes principales de la imagen. Como se puede ver en la Figura siguiente, cada componente de los 3 ejes le corresponde un color. También podemos realizar una rotación específica en grados, Blender permite escribir el numero de grados que rotaremos la imagen.

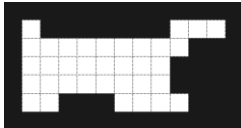
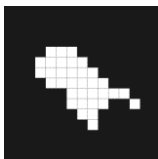
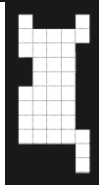




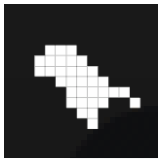
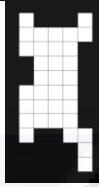
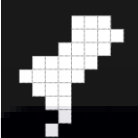
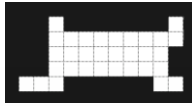
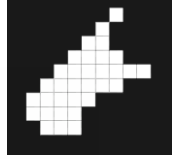
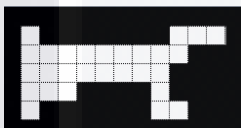
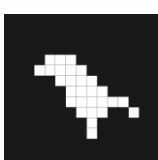

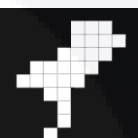

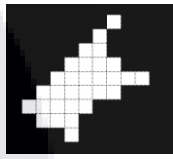

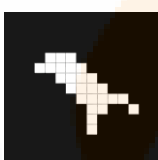


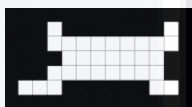
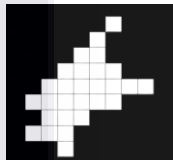



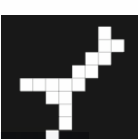
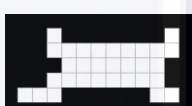
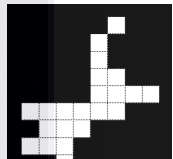


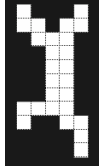





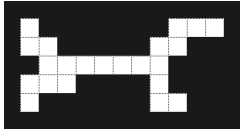
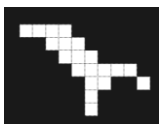


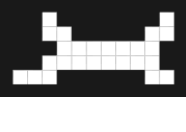

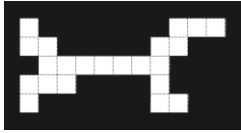



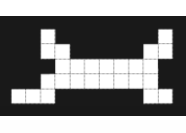

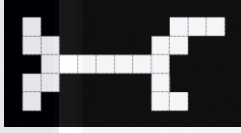


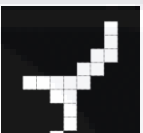


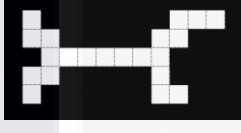



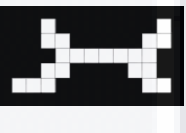





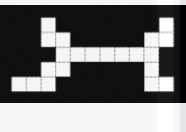

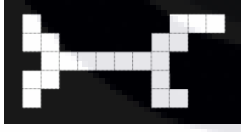



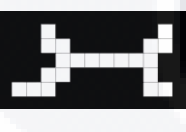

ANEXO F: ROTACIÓN EN 2D

Para complementar el análisis del problema de invariancia en rotación, realizaremos un análisis de esqueletonización y rotación en una imagen simple en dos dimensiones.

En la Tabla que se muestra a continuación, se pueden observar la rotación y la esqueletonización de la misma imagen, rotada a diferentes grados. Los renglones representan los diferentes momentos por los cuales los elementos estructurantes básicos. Al usar 8 elementos estructurantes, se necesitan solamente 2 ciclos para que el esqueleto final no se afecte.

	0	45	90	135	180	225
1ª vuelta						
E1						
E2						
E3						

E4						
E5						
E6						
E7						
E8						
2ª vuelta						
E1						

E2						
E3						
E4						
E5						
E6						
E7						
E8	