

03099

1



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**MEDIDA DE SIMILITUD PARA OBJETOS
2D Y 3D A TRAVÉS DE UNA ENERGÍA DE
TRANSFORMACIÓN ÓPTIMA**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**DOCTOR EN CIENCIAS
(Computación)**

P R E S E N T A:

HERMILO SÁNCHEZ CRUZ

DIRECTOR DE LA TESIS: DR. ERNESTO BRIBIESCA CORREA

MÉXICO, D.F.

**TESIS CON
FALLA DE ORIGEN**

2002.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Deseo expresar mi agradecimiento al Consejo Nacional de Ciencia y Tecnología, a la Universidad Nacional Autónoma de México, a través de la Dirección General de Estudios de Posgrado, así como al Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas-UNAM, por el apoyo y facilidades que me brindaron para llevar a cabo esta tesis doctoral.

Agradezco a mi asesor, el doctor Ernesto Bribiesca Correa, el constante apoyo para la realización y buen término de esta tesis de investigación. Agradezco a mis sinodales, los doctores: Fernando Arámbula Cosío, Gabriel Corkidi Blanco, Boris Escalante Ramírez, Fabián García Nocetti, Jorge Márquez Flores y Humberto Sossa Azuela, por sus importantes críticas y sugerencias para el buen avance de las investigaciones realizadas durante este trabajo. A las coordinadoras del posgrado, las doctoras María Garza Virgil y Cristina Verde Rodarte, por los apoyos académicos otorgados. A la licenciada María Ochoa Macedo por sus sugerencias en la estructura de la tesis

A mis padres y hermanos, gracias por su amor.

Índice

Lista de figuras	v
Lista de tablas	ix
Resumen	1
Capítulo 1	2
<i>Introducción</i>	
1.1 ¿Qué es la <i>forma</i> ?	2
1.2 El problema a resolver en esta tesis	3
1.2.1 Restricciones	4
1.3 Aportaciones de la tesis	5
1.4 Representación de objetos 2D	5
1.4.1 Curvas parametrizadas	5
1.4.2 Arreglos de celdas	6
1.5 Representación de objetos 3D	7
1.5.1 Superficies envolventes	7
1.5.2 Representación por cortes	8
1.5.3 Arreglos de ocupación espacial	9
1.5.4 Primitivas	11
1.6 Organización de la tesis	12
Capítulo 2	14
<i>Antecedentes</i>	

2.1	Conceptos y definiciones	14
2.2	Trabajos previos	17
2.2.1	Medidas de similitud y reconocimiento de objetos	17
2.2.2	Números de forma	21
2.2.3	Registro de objetos	21
2.2.3.1	Mapeos conformes	21
2.2.3.2	Campos de distancia	23
2.2.4	La geometría fractal	24
2.3	Justificación de la tesis	26
2.3.1	La máxima correlación como medida débil de similitud	29
Capítulo 3		36
<i>Los invariantes</i>		
3.1	Normalización de la escala	37
3.2	Invariancia bajo traslación	39
3.3	Invariancia bajo rotación	41
3.3.1	Los ejes principales	43
3.3.2	Los ángulos de Euler	49
3.4	Invariantes basados en transformadas integrales	50
Capítulo 4		52
<i>¿Cómo transformar un objeto en otro?</i>		
4.1	Pasos para transformar a los objetos	52
4.2	Método de Optimización	55
4.2.1	Conceptos básicos de la teoría de gráficas	56
4.2.2	Gráficas bipartitas	56
4.2.3	Camino alternante y camino extendido	56
4.2.4	Problemas de asignación óptima	57
4.3	Problemas de transporte	58

4.3.1 Problema dual	61
4.4 Algoritmo húngaro	63
4.4.1 Algoritmo húngaro según Gould[60](1988)	65
4.4.2 Árbol húngaro	67
4.4.3 Limitaciones del método de Gould	69
4.5 El trabajo como medida de similitud	70
Capítulo 5	73
<i>Propiedades importantes de la medida propuesta</i>	
5.1 La compacidad discreta en 2D	73
5.2 Relación entre la compacidad discreta, el trabajo de transformación y el número de <i>pixels</i> positivos	74
5.3 ¿Cómo normalizar la medida cuando el conjunto de objetos esta completo?	77
5.4 Normalizando la medida de similitud sin el conjunto de objetos completo	80
5.5 La orientación, crucial para el desempeño del trabajo ..	81
5.6 Contradicciones entre el método optimizado y el método de Bribiesca[1](1996)	86
5.7 Desventajas del método propuesto	87
Capítulo 6	88
<i>Transformación de objetos 3D</i>	
6.1 Los objetos 3D a comparar	88
6.2 Los objetos 3D se normalizan en volumen	90

6.3 Transformando los volcanes	96
6.4 Resultados de las transformaciones entre los cinco objetos	102
6.5 La medida de compacidad discreta en 3D	103
6.6 El parecido entre dos objetos, no implica la misma semejanza a un tercero	106
Capítulo 7	110
<i>Conclusiones</i>	
Capítulo 8	113
<i>Trabajo futuro</i>	
8.1 Teoría del Espacio-Escala	114
8.1.1 Principio de incertidumbre	114
8.1.2 Método de multi-resolución	118
8.2 ¿Cómo aplicar algoritmos genéticos al problema de optimización?	119
8.2.1 Restricciones sobre los AG'a basadas en el método húngaro	123
8.2.2 Eficiencia de un AG para el cálculo de la distancia mínima en la transformación de dos objetos	126
Apéndice A	128
<i>Algunos objetos 3D regulares</i>	
Apéndice B	132
<i>Objetos simétricos</i>	
Apéndice C	134
<i>Artículos derivados de esta tesis</i>	
Referencias	135

Lista de Figuras

Figura 1.1. Superficie de un objeto 3D representada por una malla poligonal.	8
Figura 1.2. Un objeto con tres representaciones diferentes; a) Por cortes o cilindros generalizados; b) Un corte que muestra la representación por <i>octrees</i> ; c) representación de los objetos a través de <i>voxels</i>	10
Figura 2.1. Cuatro puntos en el plano. Se pretende calcular la distancia mínima entre el conjunto {A,B} y el conjunto {C,D}.....	27
Figura 2.2. Dos objetos a transformar: a) Dos barras paralelas unidas, b) una sola barra. Los dos objetos tienen el mismo número de <i>pixels</i> : 196	28
Figura 2.3. Cuando los rectángulos son iguales y están alineados respecto a sus ejes principales y centros de masa, el área común es máxima. Pero al hacer un pequeño giro el área común empieza a disminuir	30
Figura 2.4. Comparación de un rectángulo más alargado que otro.	31
Figura 2.5. Hay un rectángulo con dimensiones a partir de las cuales, al empezar a girarlo alrededor del centro de masas, el área común empieza a aumentar	32
Figura 2.6. Figuras de 144 <i>pixels</i> cada una. a) cuadrado, b) rectángulo c) rectángulo anterior rotado 45°.....	33
Figura 2.7. Se transformará a) cuadro en rectángulo horizontal, b) cuadro en rectángulo inclinado	34
Figura 2.8. Dos objetos a comparar; a) una elipse; b) un marco c) superposición de ambos objetos con sus ejes principales mayores a 90° entre sí; d) máxima área de intersección entre ambos objetos; e) el ángulo entre los ejes mayores es de 30°; e) los ejes principales están alineados completamente.	35
Figura 3.1. Doce objetos bidimensionales, normalizados. Cada uno está compuesto por 674 <i>pixels</i>	39

Figura 3.2 Objeto cualquiera girando con velocidad angular ω alrededor de un eje.	45
Figura 3.3 Los tres ejes principales de un elipsoide para ilustrar los eigenvectores de una figura irregular. Dados los eigenvalores $\lambda_3, \lambda_2, \lambda_1$, éstos corresponden a los eigenvectores cuyas magnitudes van del mayor al menor respectivamente.	48
Figura 3.4. Ángulos de Euler.	50
Figura 4.1. Dos objetos dados por a) y b) están alineados; c) superposición de ambos objetos; d) la intersección de los objetos está dada por las líneas en zigzag, la parte en negro con signo más corresponde a los <i>spels</i> positivos, mientras que la parte en blanco con signo menos corresponde a los <i>spels</i> negativos.	54
Figura 4.2. Gráfica pesada completa con bipartición (I_P, I_N), (<i>spels</i> positivos, <i>spels</i> negativos).	55
Figura 4.3. Camino extendido dado por $v_1, v_2, v_6, v_4, v_5, v_3$	57
Figura 4.4. Una gráfica bipartita con un apareo óptimo.	58
Figura 4.5. Representación matricial de la gráfica de la figura 4.4, sección 4.2.	66
Figura 4.6. a) Apareo, b) apareo perfecto arbitrario, dado por las líneas gruesas.	68
Figura 4.7. Caminos extendidos: x_1, y_4, x_2, y_2 y x_3, y_3	68
Figura 4.8. Apareo óptimo.	68
Figura 4.9. Matriz con todos sus ceros marcados con el mínimo número (diez) de renglones y/o columnas.	70
Figura 4.10. Matriz en donde sus ceros son marcados con 11 renglones y/o columnas, permitiendo que quede una submatriz.	71
Figura 5.1. Doce objetos normalizados y alineados. El eje vertical indica	

la dirección del eje principal mayor, mientras que el horizontal el eje principal menor.....	75
Figura 5.2. Trabajo realizado para transformar las figuras en términos del número de <i>pixels</i> positivos.....	76
Figura 5.3. Trabajo realizado con el algoritmo húngaro en términos de los <i>pixels</i> a mover, para transformar un cuadrado en una barra a diferentes resoluciones.....	79
Figura 5.4. Por comparar la araña y la serpiente; a) ambos objetos tienen sus ejes mayores principales en un ángulo de 169.15° ; b) los ejes principales mayores están orientados a 90° uno respecto del otro. El número de <i>pixels</i> comunes es de 229; c) los objetos están orientados conforme a sus ejes principales con 215 <i>pixels</i> comunes.....	86
Figura 6.1. Modelo digital de elevación de la Cuenca de México. Se muestran tres volcanes importantes: La Malinche, el Popocatépetl y el Iztaccíhuatl.....	89
Figura 6.2. Dos automóviles cuyas superficies están dadas en mallas poligonales triangulares; a) un Porsche, b) un Camaro.....	90
Figura 6.3. Los tres volcanes cercanos a la Cuenca de México, representados por <i>voxels</i> . a) El volcán La Malinche, compuesto por 19,154 <i>voxels</i> ; b) el volcán Popocatépetl, compuesto por 20,769 <i>voxels</i> y c) el volcán Iztaccíhuatl, compuesto por 19,776 <i>voxels</i>	91
Figura 6.4. La superficie que representa al volcán Popocatépetl se sumergirá en un arreglo 3D; a) Superficie que representa al Popocatépetl b) el arreglo de $50 \times 50 \times 50$ puntos.....	93
Figura 6.5. La superficie del objeto a escalar sumergida en el arreglo 3D de puntos.....	94
Figura 6.6. Se eliminan los puntos y parte de los planos de la superficie al borrar sobre el plano y-z, en la dirección positiva del eje x; b) conjunto de puntos que toma la forma del objeto original.....	95
Figura 6.7. Dos autos <i>voxelizados</i> a partir de una malla poligonal 3D;	

ambos compuestos de 31, 020 <i>voxels</i>	97
Figura 6.8. Los tres diferentes volcanes ya normalizados y orientados conforme a sus ejes principales. Están compuestos de 16,228 <i>voxels</i> a) La Malinche, b) el Popocatépetl y c) el Iztaccíhuatl.....	98
Figura 6.9. Transformación de los objetos: a) el objeto O_1 ; b) el objeto O_2 ; c) la superposición del objeto O_1 sobre el objeto O_2 ; d) los <i>voxels</i> comunes entre O_1 y O_2 ; e) <i>voxels</i> a mover (<i>voxels</i> positivos); f) lugares a donde se moverán los <i>voxels</i> positivos (<i>voxels</i> negativos).....	99
Figura 6.10. Los diferentes estados de la transformación del volcán Iztaccíhuatl en el volcán La Malinche en pasos de 25 <i>voxels</i> : a) el volcán Iztaccíhuatl; b) 25 <i>voxels</i> son desplazados; c) 50 <i>voxels</i> son desplazados; etcétera.....	100
Figura 6.11. Los diferentes estados de transformación del volcán Popocatépetl en el volcán La Malinche en pasos de 25 <i>voxels</i> : a) el volcán Popocatépetl; b) 25 <i>voxels</i> desplazados; c) 50 <i>voxels</i> desplazados; así sucesivamente.....	101
Figura 6.12. El trabajo realizado al transformar B en C está dado por los valores bajo la curva.....	107
Figura 8.1. Un elefante y un automóvil a transformarse. Los objetos se degradan en resolución con filtros gaussianos, desde d) hasta a); a) se transforman primero los que tengan menor número de <i>pixels</i> . Posteriormente, se va transformando el segundo objeto a ser comparado (el auto), entre sus versiones a diferentes resoluciones (en la dirección de la flecha).....	119
Figura 8.2. Apareo óptimo. Todos los vértices están saturados y la suma del subconjunto de aristas E es la menor.....	120
Figura A1. Siete objetos sintéticos empleados para transformarse entre algunos de ellos, usando el algoritmo planteado por Gould de la sección 4.3.....	128
Figura A2. Transformación $A \rightarrow B$	129

Figura A3. Transformaciones progresivas: a) $D \rightarrow E$ y b) $F \rightarrow G$	131
Figura B1. Para transformar A en B no es necesario mover los 1,916 <i>voxels</i> no comunes, sino sólo una cuarta parte de ellos (479 <i>voxels</i>), pues las otras tres cuartas partes son idénticas entre sí.	132

Lista de Tablas

Tabla 2.1. Transformación del objeto A en B y C.	32
Tabla 2.2. Información de la transformación de los objetos de la figura 2.8a) y b)33	
Tabla 5.1. Los <i>pixels</i> comunes, los <i>pixels</i> a mover, la compacidad discreta y los trabajos para transformar los 11 objetos en el círculo.	76
Tabla 5.2. Trabajo realizado para transformar un cuadrado en una barra de 676 <i>pixels</i> cada uno.	79
Tabla 5.3. Número de <i>pixels</i> comunes a ambos objetos y <i>pixels</i> positivos a mover en cada transformación.	81
Tabla 5.4. Se hace una comparación entre a) el algoritmo optimizado y b) el algoritmo EMC; c) <i>pixels</i> comunes entre cada par de objetos 2D a comparar.	83
Tabla 5.5. Medida de similitud normalizada.	84
Tabla 5.6. Número de <i>pixels</i> comunes a ambos objetos y <i>pixels</i> positivos a mover en cada transformación.	85
Tabla 6.1. <i>Voxels</i> comunes.	103
Tabla 6.2. <i>Voxels</i> positivos.	103
Tabla 6.3. Medida de similitud de la forma entre los objetos 3D, según el algoritmo húngaro.	103
Tabla 6.4. Medida de similitud de la forma entre los volcanes según el	

algoritmo <i>EMC</i>	104
Tabla 6.5. Medida de similitud normalizada y acorde al algoritmo húngaro.	104
Tabla 6.6. La compacidad discreta para los cinco objetos 3D.	105
Tabla A1. Número de <i>voxels</i> comunes a ambos objetos y <i>voxels</i> positivos a mover en cada transformación.	129
Tabla A2. Comparación en la medida de semejanza según el método aplicado para transformar a los objetos.	130
Tabla B1. Resumen de la transformación hecha para los objetos originales A y B.	133

Resumen

En este trabajo se presenta la optimización de un método, introducido por Bribiesca [1](1996), para transformar un objeto O_1 en un objeto O_2 . La contribución de esta tesis es que, con el fin de optimizar dicho método, se utilizan los ejes principales para orientar los objetos y se calcula la energía, o distancia mínima, para llegar de un objeto a otro. Además, se propone un método para normalizar esta medida. Se presentan ejemplos de objetos en 2D para mostrar las propiedades más importantes del método y, finalmente, aplicarlo a objetos en 3D del mundo real, tales como representaciones de la forma de diferentes volcanes cercanos a la cuenca de México, así como de automóviles.

El cálculo de esta transformación se usa para medir la similitud de la forma de los objetos. Los objetos considerados están compuestos de *spels* (de *voxels* para objetos 3D y de *pixels* para objetos 2D). De esta manera, la diferencia en la forma de dos objetos se logra al contabilizar el número de *spels* a mover y qué tan lejos se mueven, para transformar un objeto en otro.

Abstract

A method of transforming an object O_1 into another object O_2 is presented in this work. The computation of this transformation is performed to measure shape similarity of objects. The considered objects are composed of *spels* (*voxels* for 3D objects and *pixels* for 2D objects). The difference in the shape of objects is performed by computing the number of *spels* to be moved, and by the distance they have to cover in transforming one object into another.

This work is based on a method of transformation of objects, presented by Bribiesca [1](1996). The contribution of this work is to optimise the Bribiesca's method by using principal axes to orientate objects, and by computing minimum energy or distance, to go from an object into another. To show the most important properties of our method, a set of 2D objects are analysed, and finally, this method is applied to 3D real world-objects, like those of some cars and some volcanoes near to the basin of Mexico.

Capítulo 1

Introducción

1.1 ¿Qué es la *forma*?

El sistema visual de diversos seres vivos emplea la *forma* de los objetos del mundo como una característica muy importante para reconocerlos; además, es capaz de reconocer y asociar formas. Esto se debe, posiblemente, a la evolución biológica que han tenido las especies, entre ellas la humana, donde se ha empleado el sentido de la visión para reconocer los objetos de manera cotidiana. Los seres humanos hacen una descripción puramente cualitativa de las formas de manera natural y, probablemente por ello, no han construido aún un alfabeto simbólico o un vocabulario rico que permita emular este reconocimiento y poder describir, cuantificar o diferenciar formas para que las computadoras puedan hacerlo tan fácilmente como los seres humanos. El desarrollo de métodos cuantitativos para el análisis y comparación de la forma está limitado por la poca capacidad del ser humano para cuantificar y comparar diferencias en formas.

De acuerdo con el diccionario *The shorter Oxford English Dictionary* “la *forma* es el contorno o la parte externa que tiene un objeto material”, o, más precisamente, “aquella cualidad de un objeto material o figura geométrica que depende de relaciones constantes de posición y proporcionan relaciones entre todos los puntos que conforman la superficie externa o contorno del objeto”.

Generalmente, cuando se estudia la forma de un objeto, se trabaja con sus pro-

piedades geométricas.

Cuando se hace este análisis, con el fin de mapear un objeto en otro a través de las llamadas medidas de similitud, se pretende tener los objetos invariantes bajo transformaciones de rotación, escala y traslación, lo que permite establecer métodos de reconocimiento de formas.

1.2 El problema a resolver en esta tesis

Uno de los problemas actuales dentro de la visión por computadora es el reconocimiento de la forma de los objetos o figuras [2](Ballard, 1982). En este trabajo se propone la optimización de un método para lograr este objetivo. Para ello se utilizan algunas propiedades como la normalización de la escala, orientación de los objetos y la teoría de gráficas que permiten llevar a cabo la transformación de *objetos irregulares* de representación binaria para medir su semejanza. Para obtener la medida de similitud se normaliza la escala (área o volumen) de los objetos, se hacen coincidir sus centros de masa, se alinean conforme a sus ejes principales y se obtiene el trabajo mínimo en su transformación al usar la teoría de gráficas. En la parte de asignación óptima y, como parte importante del reconocimiento después de la alineación y normalización, se presenta el *algoritmo húngaro*, que permite hacer dicha transformación de manera óptima y obtener así una medida de semejanza para *objetos irregulares*.

El objetivo del presente trabajo es optimizar una medida de semejanza, propuesta por Bribiesca en 1996 [1], de objetos o figuras tridimensionales irregulares. Para medir la semejanza de las figuras, el trabajo realizado debe ser el menor posible.

Para representar a las figuras tridimensionales, se emplea un método llamado *voxelización*. Como parte del proceso de reconocimiento de la forma, a través de la

medida de similitud, se propone hacer comunes los centros de masa de los objetos, normalizar la escala y utilizar los ejes principales. Para esto último se hace un estudio de las propiedades de inercia de los objetos sólidos.

1.2.1 Restricciones

¿Qué tan acordes son las medidas de similitud construidas en forma cuantitativa con la descripción cualitativa que hacen las personas? Cuando se establecen medidas cuantitativas, éstas no son suficientes para dar un grado de parecido entre los objetos, pues deben estar acompañados de estimaciones subjetivas debido a que la percepción de la figura no está determinada solamente por sus propiedades geométricas, sino también por el contenido de su semántica, resultado de todo un bagaje cultural y social y en general, todo el contexto en el que se encuentra la figura.

La elección de una medida de similitud particular estará determinada por la naturaleza del problema, el ruido y los medios computacionales disponibles.

No es posible hacer una medida de similitud general, pues una medida puede resultar muy buena para reconocimiento de caracteres, la cual no necesariamente trabaja bien en la inspección automática industrial por ejemplo. Diferentes aplicaciones requerirán sus propias aproximaciones.

Una de las restricciones cuantitativas de la medida estudiada y optimizada en esta tesis, es que el tiempo de procesamiento es muy grande, pues su complejidad es de orden cúbico. Sin embargo, la medida, por si sola, da buenos resultados para obtener el grado de parecido de dos o más objetos, teniendo la ventaja, a diferencia de otras medidas, de poder aplicarse a objetos *irregulares*.

1.3 Aportaciones de la tesis

- Se logra optimizar un método para medir la similitud de dos o más objetos *irregulares* a partir del uso de los ejes principales para orientarlos, así como del cálculo de la energía mínima para llegar de un objeto a otro.
- Se demuestra que un método dado en la literatura es inconsistente para la búsqueda de distancias mínimas.
- Se propone un método para normalizar la medida de similitud optimizada.
- Se presentan ejemplos de objetos en 2D para mostrar las propiedades más importantes del método.
- Se aplica el método a objetos que representan las formas de dos automóviles y tres volcanes cercanos a la cuenca de México.

A continuación, se hará un bosquejo de las diferentes representaciones de los objetos, exponiendo la necesidad de emplear los *voxels* para la transformación de los mismos.

1.4 Representación de objetos 2D

Las formas más comunes para representar objetos 2D las constituyen las curvas parametrizadas y los arreglos de celdas llamados *pixels*.

1.4.1 Curvas parametrizadas

Las curvas parametrizadas son el resultado de graficar en el plano cartesiano funciones de la forma $s = s(x(t), y(t))$, donde t es el parámetro a utilizar para graficar la curva.

Cadenas. Las cadenas se pueden considerar como curvas parametrizadas de cierto comportamiento. Se trata de segmentos de línea recta, todos iguales. Cuando se recorre el contorno del objeto 2D en dirección contraria a las manecillas del reloj, los segmentos se orientan en las cuatro direcciones cardinales: N, S, E y O. Dos tipos de cadena conocidos son las llamadas cadenas de Freeman [3](Freeman, 1961) y las de Bribiesca [4](Bribiesca, 1999). Las cadenas de Freeman emplean cuatro *bits* para discretizar el contorno de una figura, mientras que Bribiesca emplea sólo tres *bits*.

Polilíneas. En este caso, segmentos de líneas rectas concatenados recorren el contorno de un objeto 2D.

1.4.2 Arreglos de celdas

Quad-trees. Los *quad-trees* conforman un arreglo jerárquico de celdas cuadrangulares, generalmente de diferentes tamaños. Cuando un *quad-tree* se emplea para representar un objeto 2D, cada cuadrante puede estar lleno, parcialmente lleno, o bien, vacío, dependiendo de la cantidad de área que contenga cada uno. Un cuadrante parcialmente lleno es subdividido de manera recursiva en subcuadrantes. La subdivisión continúa hasta que todos los cuadrantes son homogéneos, esto es, que cada uno de ellos esté, ya sea, lleno o vacío.

Pixels. Las celdas de este arreglo son polígonos, todos del mismo tamaño y, en principio, de cualquier número de lados cada uno, aunque frecuentemente se emplea el cuadrado, por el fácil manejo para la representación binaria.

1.5 Representación de objetos 3D

A continuación se mencionan algunas técnicas para representar formas tridimensionales. En principio se desearía emplear la infinidad de puntos de la superficie de un objeto tridimensional cualquiera, pero esto es imposible por la limitación computacional que ello representa. Así que una forma de representarlos es a través de una descripción matemática relativamente fácil de implementar, tal como el uso de trozos de planos, esferas u otras superficies geométricas.

Las representaciones más comunes de los objetos 3D se hacen a través de superficies envolventes, por cortes y por arreglos de ocupación espacial.

1.5.1 Superficies envolventes

Mallas poligonales. Una malla poligonal es un conjunto de aristas, vértices y polígonos conectados de manera que cada arista es adyacente a dos polígonos a lo más y cada vértice es adyacente a dos aristas a lo más. Una arista conecta dos vértices y un polígono es una sucesión cerrada de aristas. La representación de un polígono está dada por una lista de coordenadas de vértices:

$$P = ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n))$$

Los vértices se guardan con un orden preestablecido para después poder localizarse al recorrer el polígono. La Figura 1.1 muestra un ejemplo de malla 3D.

Las superficies envolventes son ampliamente utilizadas para representar los objetos 3D. Sin embargo, una desventaja de emplearlas es que no representan adecuadamente objetos con huecos, pues para ello se requeriría de otras superficies envolventes que representaran lo huecos, ello implicaría una mayor complejidad para transformar los

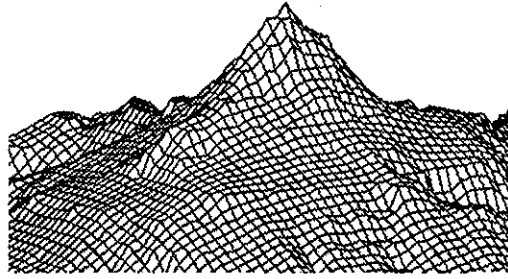


Figura 1.1. Superficie de un objeto 3D representada por una malla poligonal.

objetos.

1.5.2 Representación por cortes

Cilindros generalizados. Las formas 3D de muchos cuerpos ya sea biológicos, o bien, manufacturados, pueden describirse como “volúmenes de barrido” de un conjunto en 2D a lo largo de una curva en 3D [5](Besl, 1985).

Los cilindros se pueden modelar estableciendo curvas paramétricas que se recorren a lo largo del eje de un cilindro y, con ello definir una sección transversal que sufre rotaciones conforme se mueve a lo largo del eje. La curva en 2D puede ir cambiando de forma conforme se mueve a lo largo del eje del cilindro generalizado. La Figura 1.2a) muestra un ejemplo.

Cadenas. El método de código de cadena [6](Bribiesca, 2000) permite representar la superficie de cualquier sólido, ya sea regular o irregular. Se basa en un recorrido de la superficie de sólido (por capas por ejemplo) con movimientos discretos y, dependiendo de la dirección en que se recorra la superficie del objeto corresponderá una etiqueta, por ejemplo los símbolos 0, 1, 2, 3 y 4, a partir de un punto de apoyo. Conformando así una cadena compuesta de una combinación de cinco símbolos diferentes, cuya

longitud dependerá de la resolución o tamaño del objeto.

Se ha utilizado la representación por cortes principalmente para objetos manufacturados y muy extendidamente en imágenes biomédicas, tales como imágenes de resonancia magnética (MRI) y tomografías por emisión de positrones (PET).

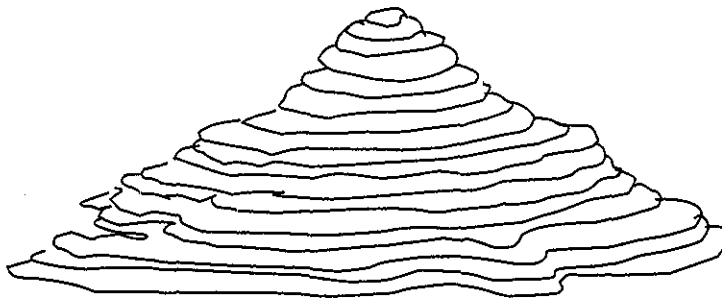
1.5.3. Arreglos de ocupación espacial

Otra forma de representar los objetos 3D es a través de los llamados *octrees* o de los *voxels* [7](Foley, 1987).

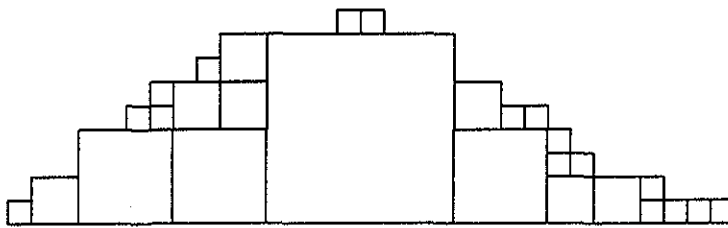
Los *octrees*. Los *octrees* conforman un arreglo jerárquico de cubos, generalmente de diferentes tamaños. Cuando un octree se emplea para representar una región en Z^3 , cada octante puede estar lleno, parcialmente lleno, o bien, vacío, dependiendo de la cantidad de volumen que contenga cada uno. Un octante parcialmente lleno es subdividido de manera recursiva en suboctantes. La subdivisión continúa hasta que todos los octantes son homogéneos, esto es, que cada uno de ellos esté, ya sea, lleno o vacío. Por claridad para su presentación, la Figura 1.2b) muestra un corte de un objeto 3D con una representación en *octree*.

Los *voxels*. Los *voxels* son elementos de volumen, a diferencia de los *octrees*, los *voxels* son todos del mismo tamaño, representados cada uno por un vector de coordenadas y propiedades: $[(x,y,z), \text{propiedades}]$. La Figura 1.2c) muestra un ejemplo.

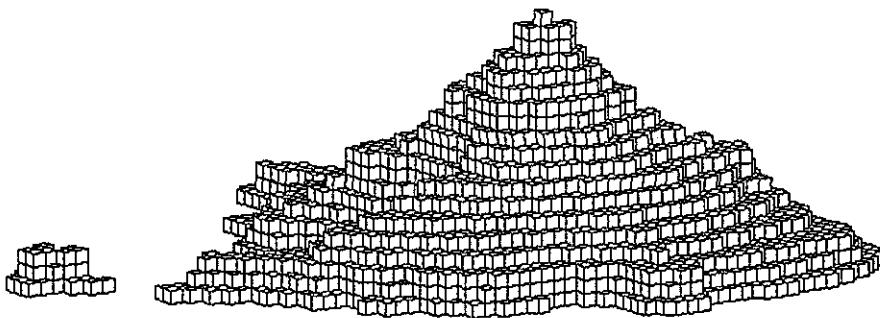
A diferencia de los objetos representados por superficie y por cilindros generalizados, una de las ventajas principales de representar objetos 3D por medio de arreglos de ocupación espacial es que, se puede representar casi cualquier tipo de objetos 3D irregulares, sin importar el número de huecos que tenga.



a)



b)



c)

Figura 1.2. Un objeto con tres representaciones diferentes; a) Por cortes o cilindros generalizados; b) Un corte que muestra la representación por *octrees*; c) representación de los objetos a través de *voxels*.

Con la representación por arreglos de ocupación espacial se tiene una manera idónea para poder transformar los objetos de manera natural, ello permite “desbaratar” los objetos para transformarlos encontrando el parecido y, además, llevar a cabo algo semejante a una metamorfosis. Asimismo, la representación por *voxels* permite procesar los objetos considerando las vecindades de todos ellos de manera idéntica. Permite, de manera natural, aplicar operadores morfológicos, o bien, apareo de *templates* [8](Russ, 1992). Como se verá, es muy conveniente emplear esta representación en la superposición de objetos, pues a los *voxels* se les puede ver como los átomos del objeto, de manera que cuando se intersectan, queda una cantidad exacta de elementos que forman la intersección de dos objetos, algo que no podría realizarse con *quad-trees* por ejemplo.

Una de las desventajas de emplear esta representación es que ocupa mucho espacio de memoria, ya que se incrementa de manera proporcional al cubo de la resolución del objeto. Sin embargo, el rápido desarrollo de las computadoras y su incremento en la capacidad de almacenar información, sugieren que podrá aprovecharse este método de representación. Es por ello que la representación 3D que se usa en este trabajo es a través de los *voxels*. Esto permite transformar los objetos a través de un método consistente que se explica a lo largo de esta tesis.

1.5.4. Primitivas

Las primitivas conforman un conjunto de Figuras 3D elementales, todas de diferentes formas, usadas para generar los objetos 3D. Las primitivas están dadas por elementos 3D sólidos [9](Dickinson *et al.*, 1992) o bien, por curvas paramétricas o elementos 3D de superficies, tales como las *cuádricas*, las cuales se explican a continuación.

Curvas paramétricas polinomiales. Estas curvas definen puntos en 3D al emplear tres polinomios (uno para cada coordenada), los cuales se ajustan a la superficie del objeto 3D al manipular adecuadamente sus coeficientes. Los polinomios que definen una curva 3D $Q(t) = Q[x(t), y(t), z(t)]$ son de la forma

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x,$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y,$$

$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z,$$

donde $0 \leq t \leq 1$.

Cuádricas. Se llama cuádrica a una superficie definida por una ecuación algebraica de grado 2 en la geometría tridimensional. Las esferas, cilindros circulares y conos circulares son cuádricas. Aunque no se utilizan para representar explícitamente cualquier tipo de objetos 3D, las cuádricas podrían modelar muy bien algunos de ellos, como los elipsoides, conos, sillas de montar, etcétera. Generalmente se usan para la reconstrucción de superficies en imágenes 2D.

1.6 Organización de la tesis

En el Capítulo 2 se dan algunos conceptos y definiciones. Se analiza la literatura estándar en cuanto a las medidas de similitud existentes, a los trabajos de alineación, invariancia y transformación y se da una justificación de la tesis.

En el Capítulo 3 se hace una revisión de los conceptos básicos de la teoría de invariantes, ya que se pretende argumentar que el hecho de orientar, escalar y trasladar los objetos, no modifica (salvo por errores de discretización) ciertas cantidades (invariantes) basadas en los momentos centrales. En este capítulo se consideran a los

ejes principales como el medio adecuado para alinear los objetos antes de llevar a cabo sus transformaciones.

Una vez que los objetos se han alineado a través de los ejes principales, en el capítulo 4 se explica la forma de optimizar el algoritmo introducido por Bribiesca [1](1996), el cual está basado en transformar los objetos. Se encuentran las limitaciones de uno de los algoritmos que aparece en la literatura para optimizar las transformaciones a base de matrices. Dichas limitaciones ya fueron reportadas por el autor de la tesis [66](Sánchez, 2000)

En el Capítulo 5 se dan a conocer las propiedades más importantes encontradas hasta ahora, que relacionan la forma de los objetos y sus propiedades geométricas con su parecido. En este capítulo se llega a saber experimentalmente que un aspecto muy importante de la forma y que interviene en el parecido, es la compacidad de los objetos. Dichas propiedades se analizan para el caso de objetos 2D, los cuales también son comparados para dar su grado de parecido.

En el Capítulo 6 se aplica el método optimizado propuesto para objetos 3D. Se muestra que el método no sólo estima el parecido entre objetos, sino que también permite realizar un cambio gradual y suave de un objeto en otro, parecido al llamado *morphing* [19](Cohen-Or, 1998), técnica que hasta ahora ha tenido como propósito fundamental, el análisis o animación de las imágenes

En el capítulo 7 se dan las conclusiones y en el Capítulo 8 se muestran los trabajos futuros derivados de esta tesis. Al final hay un apéndice, en el que se muestran algunas propiedades de objetos 3D y se dan a conocer los artículos derivados de esta tesis.

Capítulo 2

Antecedentes

A lo largo de esta tesis se hará referencia a los siguientes conceptos y definiciones, las cuales se han preferido mencionar en esta sección para comodidad del lector. En la siguiente sección se analizarán algunos trabajos previos a éste.

2.1 Conceptos y definiciones

Figura. Es una función binaria cuyo dominio es algún subconjunto A de coordenadas espaciales de la imagen total y cuyo rango es el conjunto $\{0,1\}$.

Imagen digital. Es un arreglo de números reales o complejos representados por un número finito de *bits*

En vez de usar la frase: *dos dimensiones* se usará la notación **2D**.

En vez de usar la frase: *tres dimensiones* se usará la notación **3D**.

Pixel. Elemento de área representado por un vector de coordenadas y propiedades: $(x,y, \text{propiedades})$. Geométricamente está representado por una celda, localizado en las coordenadas discretas espaciales (renglón, columna).

Voxel. Elemento de volumen representado por un vector de coordenadas y propiedades: $(x,y,z, \text{propiedades})$. Geométricamente está representado por un cubo, localizado en las coordenadas discretas espaciales (renglón, columna, capa).

Spel. Elemento de espacio n -dimensional, representado por un vector de n coordenadas junto con propiedades: $(x_1, x_2, \dots, x_n, \text{propiedades})$. En un espacio n -dimen-

sional, se puede definir al *spel* como un elemento del arreglo de imagen [10](Lohmann, 1998). El arreglo tiene un tamaño $m_1 \times m_2 \times \dots \times m_n$. En el caso de $n=2$, el tamaño del arreglo será $m_1 \times m_2$ y a cada elemento de ese arreglo se llamará *pixel*. Para $n=3$, al elemento se llamará *voxel*. Cuando se refiera indistintamente al *pixel* o *voxel*, se hablará de *spel*, sin mostrar que todas las aseveraciones que se hagan sobre estos dos elementos de imagen, sirvan para $n > 3$.

Objeto binario o simplemente *objeto*. Representación, a partir de *spels*, de un cuerpo material sólido de densidad constante. Está representado por tuplas de n elementos (x_1, x_2, \dots, x_n) . Cada tupla, representa un *spel* que puede tener el valor de cero o uno, indicando virtualmente la presencia de material del objeto.

Objeto binario 2D, o simplemente, *objeto 2D*. Es una representación espacial de un sólido o una escena bidimensional, en la que cada *pixel* toma el valor de cero o uno. El valor de la imagen localizada en las coordenadas espaciales (r, c) se denota por $I(r, c)$.

Objeto binario 3D, o simplemente *objeto 3D*. Es una representación espacial de un sólido o una escena tridimensional, en la que cada *voxel* toma el valor de cero o uno. El valor de la imagen localizada en las coordenadas espaciales (r, c, s) se denota por $I(r, c, s)$.

Objeto irregular. Es aquel objeto que no tiene ejes de simetría.

El *Centro de masa* $(\bar{r}, \bar{c}, \bar{s})$ de un objeto O (de densidad constante) está dado por:

$$\bar{r} = \frac{1}{n} \sum_{(r,c,s) \in V} r, \quad \bar{c} = \frac{1}{n} \sum_{(r,c,s) \in V} c, \quad \bar{s} = \frac{1}{n} \sum_{(r,c,s) \in V} s.$$

donde el punto $(\bar{r}, \bar{c}, \bar{s})$ son las coordenadas espaciales (dadas en renglones, columnas y capas) de los *voxels* y n es el número de *spels* que conforman al objeto. Para el caso

bidimensional, solo se considerarán las coordenadas \bar{r} y \bar{c} .

Normalizar áreas o volúmenes de objetos. Aplicar una transformación de escala a dos o más objetos para que todos tengan el mismo número de *spels*, es decir, que tengan la misma área o volumen

Spels positivos y spels negativos. Los *spels positivos* son los *spels* del objeto O_1 que no forman parte de la intersección, cuando éste se superpone con un objeto O_2 a comparar; mientras que los *spels negativos* son los *spels* del objeto O_2 que no forman parte de esa intersección.

La *Distancia Euclidiana* entre dos *spels* $a = (a_1, a_2, \dots, a_n)$ y $b = (b_1, b_2, \dots, b_n)$ es:

$$d(a, b) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2}$$

Algoritmo por elementos más cercanos (EMC). Se refiere al algoritmo propuesto por Bribiesca en 1996 [1](Bribiesca, 1996), en el que, una vez alineados (de cierta manera) dos objetos a comparar, para transformar los objetos, se forman dos conjuntos de *spels*, (definidos como *spels* positivos y *spels* negativos). Se mueve primero el *spel* positivo más cercano del conjunto de *spels* negativos, luego el siguiente más cercano y así sucesivamente, hasta desplazar el más lejano, del conjunto de *spels* positivos hacia el conjunto de *spels* negativos.

Trabajo o energía realizada en la transformación. Es el valor numérico (en unidades de trabajo) de la suma total de distancias entre los dos conjuntos de *spels* que intervienen en la transformación [1](Bribiesca, 1996).

Transformación de un objeto en otro. Es el algoritmo que permite asociar (mapeo) cada *spel* positivo, con un *spel* negativo, de manera de encontrar su diferencia numérica y, posteriormente, sumar todas las diferencias. En este sentido la transformación de

un objeto a otro es una función biunívoca, en donde a cada *spel* positivo le corresponde uno y sólo un *spel* negativo. Esta definición es equivalente al pensar en los *spels* como “ladrillos”, los cuales pueden trasladarse de un lugar a otro (de un conjunto positivo a uno negativo). Entre mayor sea la distancia recorrida por éstos más diferentes serán los objetos comparados.

2.2 Trabajos previos

2.2.1 Medidas de similitud y reconocimiento de objetos

Aunque la percepción visual humana toma en cuenta los aspectos de color y forma para comparar los objetos, la *forma* por sí sola (y sus propiedades geométricas) es una característica muy importante para llevar a cabo el reconocimiento, con el cual se han intentado establecer *medidas de similitud de la forma*, siendo un campo activo dentro del *Reconocimiento de Patrones*.

Se ha escrito mucho sobre cómo comparar, registrar y reconocer objetos bidimensionales. A veces se consideran los contornos de las figuras y sus propiedades geométricas [11](Henk, 1998),[12](Otterloo, 1991),[13](Arkin *et al.*, 1991), o bien, se emplean distancias [14](Jain, 1989) para establecer su parecido. En estos trabajos, se asume que generalmente se tiene información bidimensional disponible para reconocer los objetos, incluso de escenas tridimensionales. Se maneja también, el hecho de que las propiedades de la forma deben ser generalmente invariantes en transformaciones de rotación, traslación y escala. Existen técnicas de apareo para comparar las formas. Por ejemplo, Arkin *et al.* [13](Arkin, 1991) comparan varias formas 2D sintéticas a través de funciones de distancia. La ventaja de este método es que es de tiempo logarítmico y, la desventaja principal es que funciona para formas poligonales tales

como cuadrados y triángulos. Arkin *et al.* [13](1991) sólo probaron su método con varios polígonos de a lo más 21 vértices.

Danielsson [15](1978) establece un *factor de forma* basado en la distancia promedio entre un punto interior de una imagen 2D con el punto más cercano del contorno. Danielsson analiza las propiedades y compara este factor con otro factor de forma que relaciona el perímetro del objeto con su área. Encuentra mayores ventajas a su factor de forma, ya que este factor no diverge cuando el contorno del objeto se incrementa. Aunque Danielsson [15](Danielsson, 1978) aplica su factor de forma a algunos polígonos y dos formas arbitrarias, no lo aplica para objetos reales en 2D.

Hasta ahora hay algunos autores que han aplicado diferentes técnicas para intentar el reconocimiento de la forma de objetos 3D, Boyse [16](1979), Brooks [17](Brooks, 1983) y Dickinson, Pentland y Rosenfeld [9](1992), quienes basan sus métodos de reconocimiento en el uso de primitivas, es decir, ciertas figuras geométricas que sirven como unidades para conformar los objetos, tales como conos, conos truncados, cilindros, elipsoides, etcétera y de elementos de curva descritos por un alfabeto simbólico [14](Jain, 1988). La ventaja de estos métodos es que generalmente describen con buena aproximación objetos manufacturados, pero son difíciles de aplicar a objetos irregulares, sobre todo por la dificultad en la extracción de primitivas.

Besl y Jain [5](1985) describen algunas técnicas para obtener, procesar y caracterizar datos de profundidad. Estos autores buscan primitivas para reconocer objetos, haciendo una descripción detallada de cómo manejar los datos de imágenes para su representación computacional. Jain y Hoffman [18](1988) definen una medida de similitud de entre un conjunto de características observadas para un objeto dado en una base de datos, es decir, consideran los atributos y primitivas (superficies, contornos, aristas e intensidades de gris) más sobresalientes de los objetos, dando buenos

resultados para su identificación. Sin embargo, este trabajo está basado en extraer los datos de imágenes en 2D. Cohen-Or *et al.* [19](1998) construyen objetos 3D intermedios por medio de una metamorfosis por campos de distancia. La ventaja de este método permitió lograr el *morphing* o la transformación de objetos de diferente *genus* topológico. Adan *et al.* [20](2001) proponen un método que usa un nuevo conjunto de características globales como parámetros discriminantes. Estos parámetros están basados en la normalización de los objetos (los cuales están representados por mallas poligonales en 3D) a través de una esfera dada en malla poligonal y en el uso de los ejes principales. Con este método logran reducir de manera eficiente el tiempo de búsqueda de objetos en una base de datos. Holden *et al.* [21](2000) evaluaron ocho diferentes medidas de similitud aplicadas a imágenes de resonancia magnética y llegaron a la conclusión de que uno de esos métodos, basado en *entropía conjunta*, era más eficiente, ya que con él lograron identificar cambios anatómicos del cerebro con mayor certeza que con los otros siete. Mokhtarian *et al.* [22](2001) propusieron un método para reconocer objetos 3D representados por mallas poligonales bajo diferentes puntos de vista, basados en algoritmos de *hashing* y verificación global por medio de curvas gaussianas. Con ellas lograron detectar debidamente atributos de los objetos, tales como sus esquinas, permitiéndoles detectar correctamente 20 diferentes objetos 3D con distintas rotaciones.

Zhang *et al.* [23](1993) definen una construcción automática de un modelo relacional independiente del ángulo de visión. Para reconocer objetos 3D de escenas desconocidas, su sistema inspecciona un objeto 3D a partir de diferentes ángulos de observación para, posteriormente, aplicar inferencias estadísticas e identificar relaciones entre las diferentes partes del objeto. Con este método identifican debidamente un objeto 3D con varias orientaciones dentro de una imagen 2D.

Lohmann por ejemplo [10](1998), considera que la similitud se debe hacer usando el cociente de los volúmenes de los objetos a comparar, respecto a un cuerpo geométrico bien conocido, tales como una caja y una esfera *mínimos* que sirven como envolventes de los objetos a comparar. Si V_o es el volumen del objeto y V_c el volumen de la caja, la medida de similitud es de la forma: $S_{caja} = \frac{V_o}{V_c}$ donde la caja envuelve al objeto totalmente

Y para la esfera, la medida es simplemente: $S_{esfera} = \frac{3V_o}{4\pi r^3}$ donde r es el radio de la esfera mínima.

Aunque este tipo de media es de orden lineal, su desventaja es que arroja muchas inconsistencias en la manera de clasificar los objetos

Cuando se establece una medida de similitud, tradicionalmente existe una propiedad que se desea optimizar. Una medida de similitud propuesta recientemente por Bribiesca y Wilson [24](1997) se basa en encontrar el área máxima común de dos objetos, es decir, cuando a dos objetos se les aplica invariancia en escala, se pretende hacer traslaciones y rotaciones hasta encontrar la configuración de ambos objetos en donde el área común sea máxima. En esta tesis, se verá que esto no necesariamente es cierto cuando se desea establecer una medida de similitud, en términos de transformaciones. Lo que se desea optimizar es la energía de transformación

El método planteado en este trabajo se basa en las ideas iniciadas por Bribiesca [1](1996), quien definió una métrica y una medida de similitud, susceptibles de aplicarse para la transformación de figuras tridimensionales irregulares y no sólo figuras regulares, como lo han venido haciendo los autores mencionados. En este trabajo se logra optimizar el método de Bribiesca.

Los métodos de medidas de similitud basados en cantidades generadas por transformaciones globales son obviamente de una naturaleza global inherente, ya que las

propiedades locales de la forma no pueden tomarse en cuenta con estos métodos, algo que sí sucede con la medida de similitud que se establece en este trabajo.

2.2.2 Números de forma

Para encontrar el parecido entre dos objetos 2D, uno de los primeros intentos formales de establecer la relación entre la forma y el grado de similitud ha sido el propuesto por Bribiesca y Guzmán [25](1979). En este trabajo se presenta un método a partir de cadenas, las cuales están formadas por combinaciones de tres números (1, 2 y 3 por ejemplo). Cada cadena que conforma un número en la base 3 sirve como descriptor de cada forma. Por ello a estas cadenas se les llama *números de forma*. Cuando se pretende comparar dos objetos 2D, estos objetos se discretizan y se obtiene su número de forma. Conforme la resolución del objeto se incrementa, se forma un árbol de resolución, hasta que los números de forma de ambos objetos puedan llegar a ser diferentes. Con ello, existirá una resolución, para la cual los números de forma ya serán diferentes. Entre mayor sea la resolución (mayor la profundidad del árbol multiresolución) con iguales números de forma, más parecidos serán los objetos a comparar.

2.2.3 Registro de objetos

2.2.3.1 Mapeos conformes

El problema de registrar o alinear de manera óptima los objetos ha sido un campo de investigación activo en los últimos años. Un caso que presenta el problema de alineación es cuando se trata del mismo objeto que ha sido registrado con diferentes vistas, o simplemente cuando se quieren correlacionar dos objetos diferentes.

En la actualidad hay diferentes técnicas para alinear los objetos 3D, a través de representaciones por superficie. Como se verá a lo largo de este trabajo, una de ellas es utilizando los llamados ejes principales de los objetos.

Existe otro método para estimar la distancia mínima entre dos superficies a comparar; por ejemplo, Besl *et al.* [26](1992) encontraron la menor distancia entre las superficies a alinear al minimizar globalmente la distancia entre las superficie por el método de distancia cuadrática media.

También, otra forma de encontrar la distancia mínima es a través de los mapeos conformes, esto es, transformar una superficie en otra de manera que se preserven ángulos, es decir, que se preserven los productos internos de los vectores tangentes locales a cada superficie. Haker y Tannenbaum [27](2000) utilizaron los mapeos conformes para aplicarlos a superficies con textura, aprovechando que dichas superficies son topológicamente iguales a una esfera, o sea, las superficies no tienen hoyos, asas o intersecciones.

Un trabajo muy interesante, es el publicado por Malandain y Rocchisani [28](1993), quienes basaron su método para encontrar la menor distancia entre superficies al calcular un potencial mínimo. Lo que hicieron fue emplear las leyes fundamentales de la dinámica, en la que calculan la torca producida al hacer girar una de las superficies, la cual está sumergida en un cierto potencial debido a la presencia de la otra superficie. Posteriormente, introdujeron un coeficiente de viscosidad con el fin de detener la superficie y permitir que ésta llegue poco a poco al potencial mínimo.

En todos estos trabajos, la idea no fue dar una medida de similitud entre los diferentes objetos, sino solamente orientarlos de manera óptima para su posterior análisis. Sin embargo, el hecho de calcular una distancia durante el proceso de registro puede implicar también dar un grado de parecido.

En esta tesis, se alinean los objetos considerando primero los sólidos completos y, posteriormente utilizando los ejes principales.

2.2.3.2 Campos de distancia

El método de interpolación por campos de distancia (ICD) se utiliza generalmente para la reconstrucción de un modelo n -dimensional a partir de una sucesión de sus cortes o secciones transversales de dimensión $(n - 1)$. El método ICD se comenzó a utilizar principalmente para la reconstrucción de los cortes cerebrales [29](Louis y Natter). En 1986, Levin [30] presentó métodos en los que las secciones transversales de los objetos comenzaban a ser más generales que los casos anteriores (objetos con hoyos, por ejemplo).

En 1990, Payne y Toga [31] desarrollaron un sistema gráfico para desplegar información funcional y anatómica de superficies del cerebro humano, usando para ello campos de distancia. Dos años después, en 1992, Payne y Toga emplearon también los campos de distancia para combinar, modificar y analizar superficies [32](Payne y Toga, 1992).

La reconstrucción de objetos por el método de interpolación en general, está basada en dos métodos. Un método consiste en considerar las superficies poligonales de los objetos de manera que sus secciones transversales sean contornos poligonales [33](Barequet, 1994). La otra clase de métodos consiste en considerar todo el interior del objeto, esto es, se considera una representación de *voxels*, de manera que el volumen completo del objeto se reconstruye a partir de interpolaciones en niveles de grises de sus secciones transversales.

Otra de las aplicaciones que se ha dado con el método ICD es en la transformación progresiva, o *metamorfosis* como se le conoce, entre diferentes objetos 3D. Cohen-

Or *et al.* [19] recientemente (1998) utilizaron el método ICD para llevar a cabo la metamorfosis de diferentes objetos 3D. Con el fin de que ésta se logre de una manera suave aplicaron interpolación entre objetos intermedios de la transformación.

En la literatura no se ha encontrado, hasta ahora, algún trabajo que utilice campos de distancia y que defina de manera precisa medidas de similitud, esto es, que de un *grado del parecido* entre dos o más objetos. Los campos de distancia se han utilizado principalmente para la reconstrucción y animación de objetos, tanto en 2D como en 3D. Sin embargo, podría ser posible una indagación al respecto para establecer medidas de similitud y compararlas con los resultados que aquí se obtengan en cuanto a la transformación de objetos.

Hasta ahora, ninguno de los trabajos anteriores establece el grado de parecido entre los objetos 3D. Algo que se pretende en este trabajo es precisamente dar un método consistente para, no solamente registrar los objetos 3D, sino también encontrar el grado de similitud entre ellos.

La percepción humana de la figura y su parecido a otra figura, no solamente está determinada por aspectos puramente geométricos, sino por un contenido semántico, resultado de todo un bagaje cultural y social. Así que, establecer una medida de similitud lo mejor aproximada posible a la “realidad”, significaría establecer un espacio de patrones o de características de muchas dimensiones. Sin embargo, en este trabajo se estudiarán algunas características que permitan decir cuándo dos objetos son parecidos y qué tan parecidos son en su forma.

2.2.4 La geometría fractal

En 1975 Beneto Mandelbrot [34] introdujo la geometría fractal. Mandelbrot afirmaba que, usando la geometría fractal, era posible crear modelos matemáticos para todos

los objetos complejos que aparecen en la naturaleza, tales como las nubes, montañas y árboles, incluso ellos mismos son ejemplos de conjuntos fractales.

Un aspecto muy importante de la geometría fractal es la de poder expresar conjuntos complejos con tan sólo unos cuantos parámetros.

Las aplicaciones más importantes que se han hecho al procesamiento de las imágenes son la reconstrucción y análisis de texturas y superficies, la compresión de imágenes y, recientemente, el reconocimiento de objetos.

Puesto que existen muchos problemas en el análisis de superficies naturales y en la construcción de mapas de terrenos para modelar las superficies, Arakawa y Krotkov [35](1996) modelaron superficies de terreno e hicieron una reconstrucción de superficie fractal. Jennane *et al.* hicieron un análisis fractal de huesos humanos para el diagnóstico de enfermedades. A pesar de que los objetos son complejos y difíciles de describir, Turiel y Pozo [36](2002) reconstruyeron imágenes a partir de fractales gracias a que también poseen un alto grado de redundancia. Mostraron cómo descomponer imágenes en sus componentes fractales. Belloulata y Konrad [37](2002) emplean codificación fractal para la compresión de imágenes.

En 1995 aparece el primer trabajo sobre reconocimiento de objetos usando la geometría fractal. Con ese trabajo, Neil y Curtis [38](1997) presentaron transformaciones fractales para el reconocimiento de la forma. Con la técnica que establecieron estos autores, la cual se basa en construir invariantes en rotación y escala, se hace el reconocimiento de objetos manufacturados en 2D, pudiendo identificar correctamente más objetos con este método que con el basado en los momentos centrales.

Los trabajos realizados para reconocimiento han sido específicamente en objetos 2D. Hasta ahora no se ha encontrado algún trabajo que emplee la geometría fractal para obtención del grado de similitud de objetos 3D.

2.3 Justificación de la tesis

En su artículo de 1997, Bribiesca [24] diseñó un método para medir el parecido de dos o más objetos bidimensionales diferentes, el cual consiste en superponer dos regiones discretas orientadas respecto a su máxima área de intersección, o máxima correlación. Bribiesca se propone, por primera vez, encontrar el parecido de las figuras de los objetos a partir de las distancias entre los conjuntos de *pixels* que no se aparean en la superposición, (a los *pixels* de uno de los objetos a comparar que no se aparean les llama *pixels* positivos y a los que no se aparean en el otro objeto, *pixels* negativos). Considerar por si sola la cantidad de *pixels* que no fueron apareados en la superposición arroja una información pobre acerca de la forma de los objetos.

En [1], Bribiesca (1996) aplicó algunas de las ideas de su artículo anterior extendiéndolas a tres dimensiones. Este método se basa en orientar los objetos a comparar a través de los ejes mayores, mostrando la necesidad de encontrar una máxima correlación para luego mover los *spels*. Una vez orientados de esta manera, *transforma* los objetos al mover los *spels* no comunes a ellos siguiendo un orden heurístico: mover primero los *voxels* más cercanos, al usar la distancia euclidiana menor, enseguida los siguientes más cercanos y así sucesivamente (algoritmo *EMC*). Sin embargo, este procedimiento no siempre permite encontrar la menor distancia a recorrer por todos los *spels*.

Para mostrar esto, considérese la siguiente configuración de puntos, los cuales se han puesto en el plano cartesiano de manera que los puntos A y B estén sobre el eje *y* y los puntos C y D sobre el eje *x*.

Las coordenadas de estos puntos son: $A(0,0)$, $B(0,d_1)$, $C(d_2,0)$ y $D(d_2,-d_3)$ (C y D están sobre una recta paralela al eje *y*), d_1 , d_2 y $d_3 \in \mathfrak{R}^+$. Supóngase que A y B

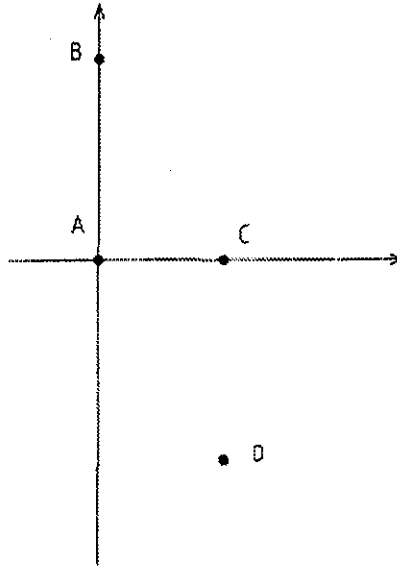


Figura 2.1. Cuatro puntos en el plano. Se pretende calcular la distancia mínima entre el conjunto $\{A,B\}$ y el conjunto $\{C,D\}$.

son las posiciones de dos *spels* a mover, es decir, dos *spels* positivos y C y D dos *spels* negativos. Según el algoritmo *EMC*, para transformar los objetos, hay que recorrer primero el *spel* que está en la posición A hasta alcanzar la posición C, pues son los más cercanos y, posteriormente, el *spel* de la posición B a la posición D (son los siguientes más cercanos). Supóngase que la distancia total recorrida para hacerlo es la menor de las posibilidades, es decir:

$$d(A, C) + d(B, D) \leq d(B, C) + d(A, D)$$

o sea,

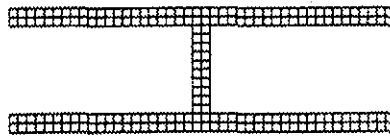
$$d_1 + [(d_2 + d_3)^2 + d_1^2]^{1/2} \leq (d_1^2 + d_2^2)^{1/2} + (d_1^2 + d_3^2)^{1/2}$$

Para ver que la desigualdad anterior es falsa, con un poco de álgebra se puede describir como:

$$4d_1^4[(d_2 + d_3)^2 + d_1^2] \leq d_2^2 d_3^2$$

Esta desigualdad es falsa para cualquier valor de $d_i \neq 0$. Con esto se observa claramente que el algoritmo *EMC* no siempre encuentra la distancia mínima al mover los *spels* de los objetos a transformar.

Esta forma de mover los *spels* se refleja al querer mover todo un conjunto de ellos. Considérese los siguientes objetos a transformar:



a)



b)

Figura 2.2. Dos objetos a transformar: a) Dos barras paralelas unidas, b) una sola barra. Los dos objetos tienen el mismo número de *pixels*: 196.

Comparando el trabajo realizado al seguir el algoritmo *EMC* con el trabajo al seguir el algoritmo *húngaro* que se presenta en el Capítulo 4, se tiene que con el primero se realiza un trabajo de $W_B = 3,345.48$, mientras que con el segundo, $W_H = 3,061.48$. Lo cual significa que hay una optimización en cuanto a la distancia empleada para transformar un objeto en otro del 8.48%. Es por ello que en esta tesis se emplea el algoritmo húngaro para encontrar las distancias entre los diferentes conjuntos de *spels* de los objetos a transformar. Por otro lado, la orientación que deben tener los objetos antes de transformarse es muy importante, ya que desempeña un papel crucial en la

eficiencia para el cálculo de la energía de transformación, como se verá en la siguiente sección.

2.3.1 La máxima correlación como medida débil de similitud

El hecho de alinear los objetos conforme a sus volúmenes de intersección máximos no garantiza una mejor alineación. Cuando se analizaron los trabajos previos en el primer capítulo, se observó que existen métodos para orientar las superficies de manera que se minimizara la distancia entre ellas. Por lo tanto, es de intuirse que lo mismo ocurre si en vez de considerar el contorno del objeto también se considera su interior.

Para registrar dos objetos se utilizará el centro de masa y los ejes principales. En 2D, intuitivamente se esperaría que los dos objetos registrados y, una vez superpuestos, tengan tantos *pixels* comunes como sea posible. Se verá que al encontrar *los ejes principales* y el centro de masa (los ejes principales en los rectángulos coinciden con los ejes de simetría), el área de intersección de dos objetos no necesariamente es el máximo. Se expondrán algunos ejemplos simples pero ilustrativos.

En la Figura 2.3 hay dos rectángulos iguales. Sus áreas están normalizadas, sus centros de masa son comunes y están alineados conforme a sus ejes principales. Si se empieza a rotar alrededor de un eje perpendicular al papel, claramente el área de intersección empieza a disminuir. Así que, en este caso la alineación de los ejes principales y centro de masa encontrados inicialmente coinciden con la idea del área máxima (o volumen para objetos 3D). Lo mismo pasa si se empieza a alargar uno de los rectángulos ligeramente: el área máxima corresponderá a la alineación de los ejes principales. Sin embargo, al seguir “alargando” uno de los rectángulos (ver Figura

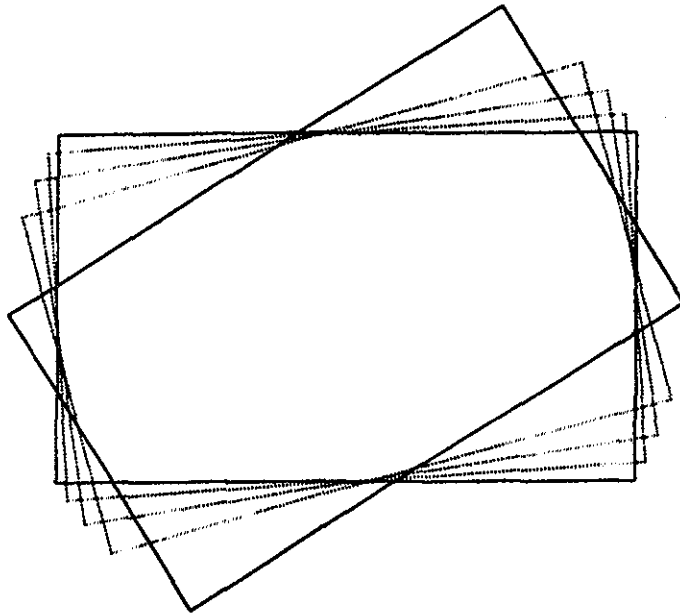


Figura 2.3. Cuando los rectángulos son iguales y están alineados respecto a sus ejes principales y centros de masa, el área común es máxima. Pero al hacer un pequeño giro el área común empieza a disminuir.

2.4), ambos serán cada vez más diferentes y el área máxima no corresponderá a la alineación de los ejes principales, el área común aumenta si se gira el rectángulo más alargado (ver Figura 2.5).

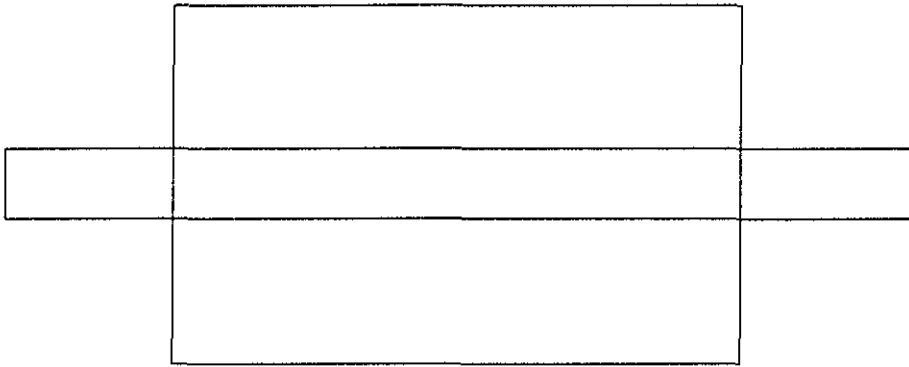


Figura 2.4. Comparación de un rectángulo más alargado que otro.

La alineación de los ejes principales en dos objetos muy diferentes da un área común pequeña en comparación con otro tipo de orientación.

Se hizo un experimento parecido al anterior y se calculó el trabajo al transformar los diferentes rectángulos.

Considérese el cuadrado y rectángulo de la Figura 2.6a) y 2.6b), respectivamente. Ambos tienen 144 *pixels*. La Figura 2.6c) muestra el rectángulo rotado -45° . Una vez que se superponen sus centros de masa se desea transformar el rectángulo en el cuadrado de dos maneras. En la primera el rectángulo está en forma horizontal y sus ejes principales coinciden con los del cuadrado (Figura 2.7a). En el segundo caso, el rectángulo está rotado -45° de manera que alcanza su máxima área de intersección con el cuadrado, pero los ejes principales están desalineados (Figura 2.7b).

La Tabla 2.1 muestra los datos de las características principales de los objetos a transformar.

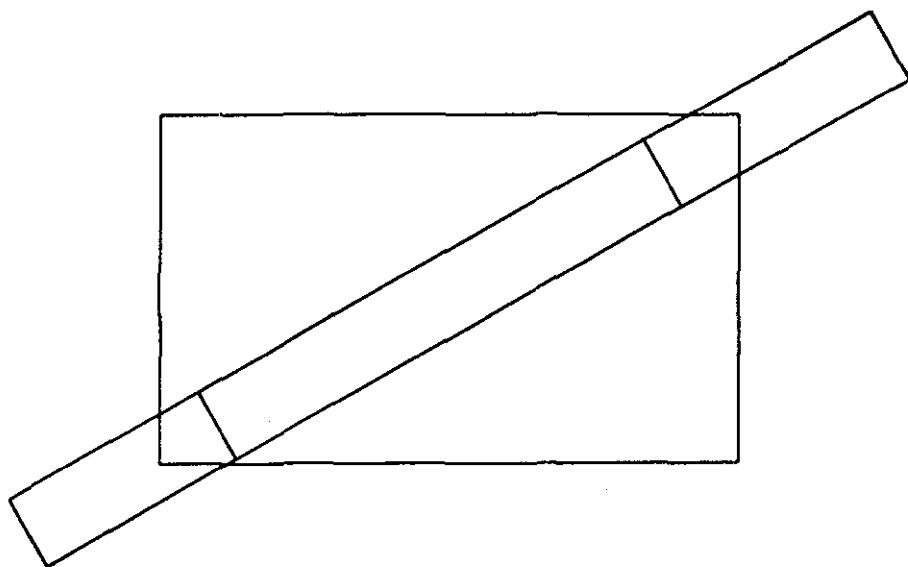


Figura 2.5. Hay un rectángulo con dimensiones a partir de las cuales, al empezar a girarlo alrededor del centro de masas, el área común empieza a aumentar.

Transformación	<i>Pixels</i> comunes	<i>Pixels</i> no comunes	Distancia
$a) \rightarrow b)$	24	120	2189.56
$a) \rightarrow c)$	31	113	2217.70

Tabla 2.1. Transformación del objeto A en B y C.

Pues bien, la distancia total recorrida para transformar el cuadrado en el rectángulo horizontal fue de 2,189.56 unidades y la distancia para transformarlo con el rectángulo girado fue de 2,217.70. Mayor cuando el área de intersección fue mayor.

Considérese la siguiente elipse y marco de la Figura 2.8a) y b) respectivamente. Ambos están alineados conforme a sus ejes principales. Están compuestos de 324 *pixels* cada uno. La Tabla 2.2 muestra el resultado de transformar ambos objetos: según la Figura 2.8c) los ejes principales mayores están a 90° entre ellos; según la Figura 2.8d) los objetos están a una máxima correlación entre ellos con 93 *pixels* en común; según

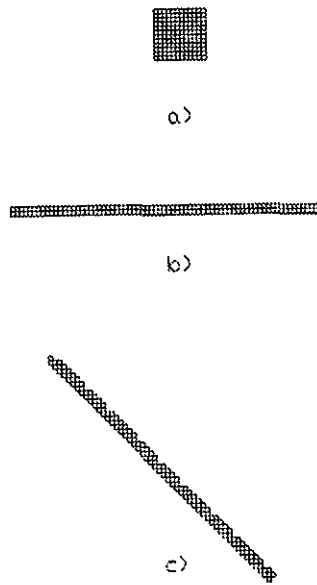


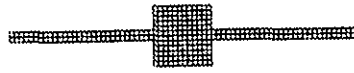
Figura 2.6. Figuras de 144 *pixels* cada una; a) cuadrado, b) rectángulo, c) rectángulo anterior rotado 45°.

la Figura 2.8e) sus ejes principales mayores están a 30°; en la Figura 2.8f) los ejes están alineados y es donde los objetos tienen el menor número de *pixels* en común y, sin embargo, la distancia de transformación es la menor de todos los casos.

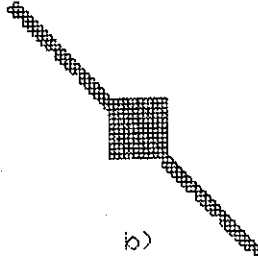
Caso de la Fig. 2.8	<i>Pixels</i> comunes	<i>Pixels</i> no comunes	Distancia
c)	59	265	2,945.17
d)	93	231	3,730.16
e)	12	312	2,738.61
f)	3	321	2,549.93

Tabla 2.2. Información de la transformación de los objetos de la Figura 2.8a) y b)

A diferencia de otros métodos, como los mapeos conformes y los campos de distancia, que trabajan preferentemente con las superficies o cortes transversales, en este trabajo se ha preferido emplear a los ejes principales para registrar los objetos.



a)



b)

Figura 2.7. Se transformará a) cuadro en rectángulo horizontal, b) cuadro en rectángulo inclinado.

Por lo tanto, sobre la base de la medida de similitud propuesta por Bribiesca [1](1996), lo que se propone en esta tesis es buscar la manera de mover todos los *spels* de un objeto en otro con la menor distancia posible. Para lograr esto, se propone el uso de los ejes principales de los objetos como mecanismo para orientarlos y, posteriormente, con el llamado algoritmo húngaro se procede a encontrar la distancia mínima para mover los *spels*. Se validará este procedimiento con diferentes objetos 2D y 3D y se propondrá una forma de normalizar el trabajo realizado en la transformación de los objetos, dando una medida de similitud en el intervalo de 0% al 100%.

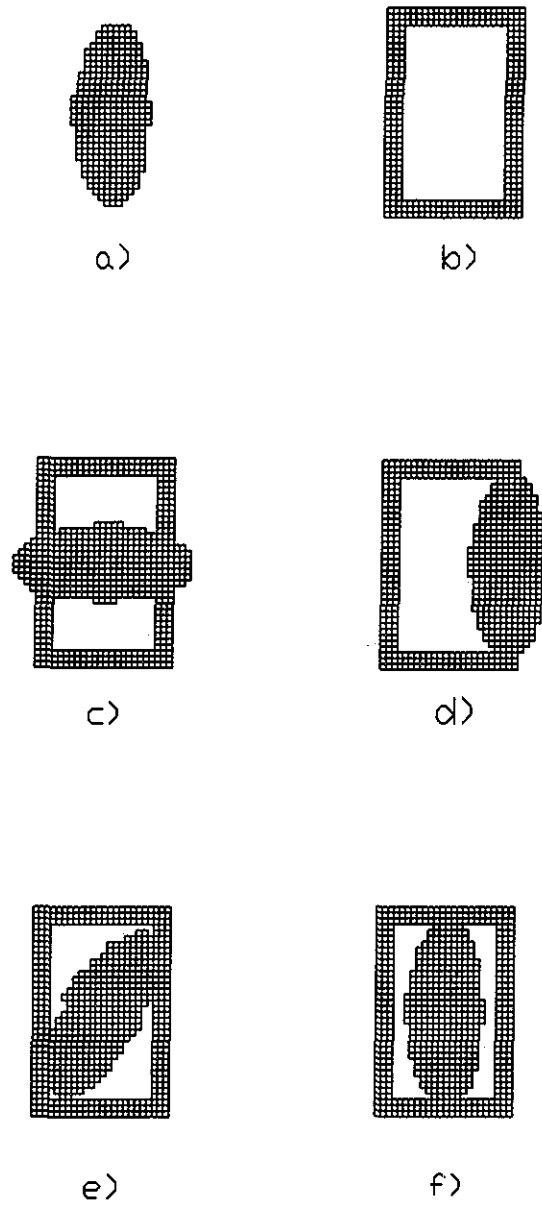


Figura 2.8. Dos objetos a comparar; a) Una elipse; b) un marco; c) superposición del marco con sus ejes principales mayores a 90° entre sí; d) máxima área de intersección entre ambos objetos; e) el ángulo entre los ejes mayores es de 30° ; f) los ejes principales están alineados completamente.

Capítulo 3

Los invariantes

El método seguido aquí para comparar dos objetos diferentes, consiste en llevar a cabo las siguientes transformaciones admisibles: 1) normalizar las áreas si se pretende comparar objetos 2D, o bien, normalizar los volúmenes si lo que se quiere es comparar objetos 3D; 2) hacer una traslación rígida de los objetos a comparar para que coincidan sus centros de masa; 3) realizar una transformación de rotación rígida de manera que los objetos a comparar estén orientados conforme a sus ejes principales; por último, 4) transformar un objeto en otro.

Ninguna de las transformaciones dadas por los tres primeros incisos anteriores modifica la forma de los objetos, esto es debido a que existen cantidades llamadas *invariantes*, es decir, cantidades que no cambian a pesar de hacer cualquiera de las transformaciones admisibles mencionadas en los tres primeros incisos del párrafo anterior.

El problema de encontrar invariantes para reconocimiento de objetos en 3D consiste en encontrar cantidades o propiedades de los objetos que no cambian, a pesar de que estos sean afectados por algún grupo de transformaciones admisibles en el espacio ocupado por estos objetos. Entre las categorías más importantes que se manejan en el estudio de invariantes están las basadas en los momentos centrales, las basadas en las transformadas integrales [39](Jeffrey, 1996) y, en este capítulo se verá que la magnitud de los ejes principales son invariantes en rotación.

El primer artículo sobre invariantes basados en los momentos centrales fue publi-

cado por Hu [40](1962), quien presentó una teoría basada en los momentos bidimensionales. Sea $\rho(x, y) \geq 0$ una función real y acotada, definida en una región R . Los momentos regulares bidimensionales de orden $(p + q)$ se definen como:

$$m_{pq} = \int_R x^p y^q \rho(x, y) dx dy, \quad (3.1)$$

donde R es la región (de la imagen) en la que se aplicará la doble integral y p y q son números enteros.

Los momentos centrales están dados por:

$$\mu_{pq} = \int_R (x - \xi)^p (y - \eta)^q \rho(x, y) dx dy, \quad (3.2)$$

donde $\xi = m_{10}/m_{00}$, $\eta = m_{01}/m_{00}$ y $p, q \in N$.

Las relaciones 3.1 y 3.2 pueden calcularse para el caso discreto, convirtiendo la integral \int_R en dos sumatorias: $\sum \sum_{x,y}$, considerando $\rho(x, y) = 1$ y el lado de cada *pixel* igual a uno.

3.1 Normalización de la escala

En el mundo real casi todos los objetos necesitan un número diferente de *spels* para describirlos. En este trabajo, cuando se quiera transformar un objeto en otro, es necesario que esos objetos tengan la misma cantidad de *spels*. Para lograr esto, se aplica una transformación de escala como la dada por la ecuación 3.4 para el caso bidimensional. Cuando esto pasa, la ecuación 3.3 se mantiene invariante (salvo por errores debido a la discretización) por cada objeto bidimensional escalado.

La ecuación (3.3) que aparece a continuación,

$$\phi = \frac{\mu_{pq}}{\mu^{\frac{(p+q)}{2}+1}}, \quad (3.3)$$

donde $\mu = m_{00}$, se mantiene invariante (no cambia) cuando un objeto sufre una transformación de escala (cuando se contrae o se dilata de manera uniforme) al aplicar la siguiente transformación:

$$\begin{aligned} x' &= \alpha x, \\ y' &= \alpha y, \end{aligned} \quad (3.4)$$

con $\alpha \in \mathbb{R}$.

Para el caso tridimensional, sólo hay que agregar una suma a las ecuaciones de momentos regulares y centrales (ecuaciones 3.1 y 3.2). Los momentos regulares quedan como:

$$m_{pqr} = \sum_x \sum_y \sum_z x^p y^q z^r. \quad (3.5)$$

Y los momentos centrales (caso discreto):

$$\mu_{pqr} = \sum_x \sum_y \sum_z (x - \xi)^p (y - \eta)^q (z - \zeta)^r. \quad (3.6)$$

Se puede probar que el siguiente cociente es invariante bajo transformaciones de escala:

$$\frac{\mu_{pqr}}{\mu^{(p+q+r)/3+1}}, \quad (3.7)$$

donde $\mu = m_{000}$.

Sin embargo, una vez que a los objetos se les ha aplicado una transformación admisible de escala, de manera que tengan el mismo número de *spels*, se dirá que ambos objetos están *normalizados*.

La Figura 3.1, muestra 12 objetos diferentes, todos ellos normalizados a 674 *pixels*.

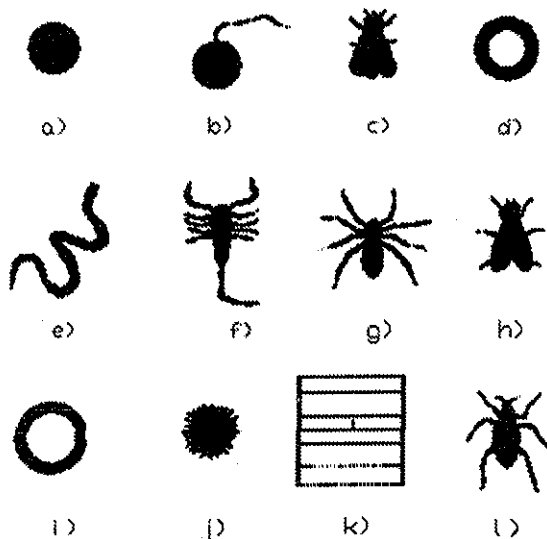


Figura 3.1. Doce objetos bidimensionales, normalizados. Cada uno está compuesto por 674 *pixels*.

3.2 Invariancia bajo traslación

Una *traslación de un objeto* es una transformación rígida en \mathbb{R}^n . Una traslación mueve cada *spel* del objeto en la misma dirección y por la misma cantidad [41](Karush, 1989), manteniendo constante las distancias entre cualquier par de *spels*.

En 2D los momentos dados por la ecuación 3.2 son invariantes bajo traslación. Puede hacerse una traslación de las coordenadas (x, y) de cada objeto a través de la siguiente transformación de coordenadas:

$$\begin{aligned}x' &= x + \alpha \\y' &= y + \beta\end{aligned}\tag{3.8}$$

con α y β constantes,

y la ecuación (3.2) sigue siendo la misma.

Para el caso tridimensional puede hacerse una traslación de las coordenadas (x, y, z) a través de la siguiente transformación de coordenadas:

$$\begin{aligned}x' &= x + \alpha \\y' &= y + \beta \\z' &= z + \gamma\end{aligned}\tag{3.9}$$

con α , β y γ constantes,

y la ecuación (3.6) se mantiene invariante.

Por otra parte, en este trabajo, cuando se compararen dos o más objetos se buscará superponerlos, lo que implica una traslación de cada uno en el espacio. Ya se mostró que al trasladar objetos existen propiedades invariantes (ecuaciones: 3.2 y 3.6). Lo que se hará aquí es trasladar los objetos, superponiéndolos, de manera que coincidan sus centros de masas.

3.3 Invariancia bajo rotación

Con el álgebra de invariantes, Hu [40](1962) llega a los siete invariantes ante transformaciones de rotación:

$$\begin{aligned}
 \phi_1 &= \mu_{20} + \mu_{02}, \\
 \phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2, \\
 \phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2, \\
 \phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2, \\
 \phi_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} - \mu_{03})(\mu_{21} - \\
 &\mu_{03})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2], \\
 \phi_6 &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}), \\
 \phi_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - (\mu_{30} - 3\mu_{12})(\mu_{21} + \\
 &\mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2], \tag{3.10}
 \end{aligned}$$

Hu [40](1962) aplicó estas ecuaciones por primera vez para reconocer caracteres alfabéticos, independientemente de su posición, tamaño y orientación. Una de las limitaciones de este método es que al aumentar la resolución de los objetos 2D, el número de invariantes debe crecer, de manera que si dos objetos tienen invariantes de segundo orden parecidos, las formas pueden ser muy diferentes. Este método es sensible al ruido, lo que constituye una desventaja.

Sadjadi y Hall [42](1980) utilizaron los momentos en 3D, al aumentar una integral a la ecuación 3.1. Encontraron tres invariantes de segundo orden (la suma de los subíndices de las ecuaciones, siempre es igual a dos):

$$\begin{aligned}
\phi_1 &= \mu_{200} + \mu_{020} + \mu_{002}, \\
\phi_2 &= \mu_{200}\mu_{020} + \mu_{200}\mu_{002} + \mu_{020}\mu_{002} - \mu_{101}^2 - \mu_{110}^2 - \mu_{011}^2, \\
\phi_3 &= \mu_{200}\mu_{020}\mu_{002} - \mu_{002}\mu_{110}^2 + 2\mu_{110}\mu_{101}\mu_{011} - \mu_{020}\mu_{101}^2 - \mu_{200}\mu_{011}^2.
\end{aligned} \tag{3.11}$$

los cuales son invariantes en rotación. Lo y Don [43](1989) utilizaron estas cantidades, así como los ejes principales para orientar objetos simétricos (cigarro, plato y esfera) y clasificarlos. Sin embargo, no probaron este método para reconocer objetos irregulares.

Otro trabajo interesante publicado en 1995 sobre invariantes en 3D, fue el realizado por Burel y Henocq [44](1995), quienes construyeron una serie de invariantes basados en resultados de la mecánica cuántica. Una rotación en el espacio (en \mathbb{R}^3), se considera como un operador lineal \mathfrak{R} (que es la conocida matriz de rotación). Este operador tiene una relación estrecha con el operador de rotación para momento angular que se usa en la mecánica cuántica. Con este principio y utilizando álgebra tensorial, Burel y Henocq encontraron una gran cantidad de invariantes que les permitió descomponer la forma de objetos 3D (vértebras del cuerpo humano) en una base ortonormal de eigenvectores del operador del momento angular. La ventaja de este método es que, entre más invariantes se calculen para clasificar los objetos, el orden de estos no aumenta. Este método es sensible a pequeñas diferencias en la similitud de los objetos 3D. Sin embargo, entre sus limitaciones, puede darse el caso de que para dos objetos irregulares existan los mismos invariantes, además no se ha mostrado si cualquier subconjunto de estos invariantes caracteriza sin ambigüedad la forma de un objeto a diferentes resoluciones.

Otros tipos de invariantes son los basados en los momentos de Zernike, obtenidos a partir de los polinomios (los cuales satisfacen la propiedad de ortogonalidad) de

Zernike [39](Jeffrey, 1996). El módulo de estos momentos es invariante ante transformaciones de rotación. Para encontrar invariantes en traslación y cambio de escala, Khotanzad y Hong [45](1990) hicieron una normalización de la función $f(x, y)$ en términos de los momentos regulares, utilizándolos ampliamente para reconocimiento de figuras 2D. Estos momentos son más robustos para imágenes con ruido que los momentos regulares. Se tienen, también, los momentos de Fourier-Mellin, estos momentos se derivan a partir de la llamada transformada Fourier-Mellin (que aparece en [39](Jeffrey, 1996). Li [46](1992) encontró una relación entre esta transformada y los momentos regulares, con lo que obtuvo tres invariantes en rotación y escala. La ventaja de estos invariantes es que no son sensibles al ruido. Se ha utilizado solamente para reconocer objetos en 2D. La desventaja es que no son invariantes en traslación.

3.3.1 Los ejes principales.

Los ejes principales se han utilizado muchas veces para orientar objetos 2D. En 3D han habido algunos trabajos, por ejemplo el de Faber, *et al.* [47](1988), para quienes calcular ejes principales es una forma elegante de encontrar invariantes, pues a diferencia de los otros invariantes que ya se han mencionado, los ejes principales tienen un significado geométrico y analítico mucho más claro que los momentos y las transformadas. Es por ello que en este trabajo se consideran a las magnitudes de los ejes principales como invariantes en rotación. Entre las ventajas, Faber, *et al.* [47](1988) concluyeron que el uso de los ejes principales, como invariantes, fueron robustos independientemente de la resolución de los objetos. Entre las limitaciones que encontraron, observaron que para orientar los objetos 3D es necesario conocer factores de escala y deformación de los objetos, pues existe una ambigüedad en la dirección de los ejes cuando estos se calculan. Otro trabajo interesante para invariancia en rotación

usando ejes principales, es el publicado por Galvez y Canton [48](1993), quienes los utilizaron como invariantes en rotación para reconocimiento de objetos 3D sintéticos regulares, cuyas superficies están representadas por polígonos. Pero, nuevamente, observaron lo mismo que Faber, hay una ambigüedad en la dirección de los ejes principales, por lo que tuvieron que resolverla de manera heurística, es decir, ver cuál es la mejor orientación de estos ejes, al probar las ocho combinaciones diferentes que aparecen como resultado de los signos de los eigenvectores que representan a los ejes principales. Otra publicación reciente en el uso de los ejes principales, se aplicó a imágenes biomédicas [49](Bulow *et al*, 2000); en este caso, el trabajo se hizo para identificar correctamente el movimiento de algunos huesos de la mano de tres personas diferentes. Una de las limitaciones que se encuentran al utilizar este método, es que los ejes principales de los huesos de la mano pueden no estar determinados de manera única, es decir, pueden existir los mismos ejes (en orientación y tamaño) a pesar de que dos huesos sean un poco diferentes en forma, corriendo el riesgo de *registrarlos* u orientarlos de manera equivocada. En esta tesis se utilizan los ejes principales como la mejor forma de registrar los objetos antes de transformarlos.

Una rotación es una transformación rígida en \mathbb{R}^3 llevada a cabo alrededor de un eje de rotación en la que, físicamente, cada partícula de un objeto dado se mueve alrededor de un eje fijo, conservando siempre una distancia constante con él y en la que las distancias entre todas las partículas del objeto también permanecen constantes. Para el caso 3D, ver Figura 3.2.

Se analizará el caso 3D, pudiéndose reducir los resultados a 2D. El momento angular L de n partículas respecto al origen de un sistema de coordenadas dado es:

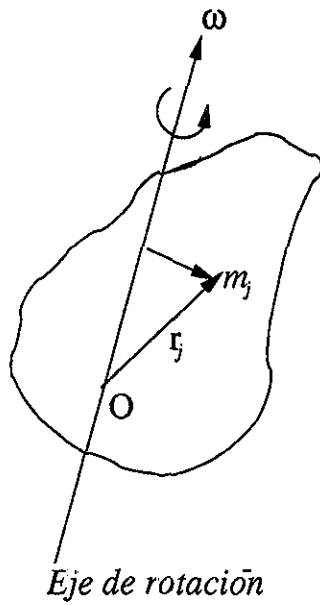


Figura 3.2. Objeto cualquiera girando con velocidad angular ω alrededor de un eje.

$$\bar{L} = \sum_{j=1}^n m_j (\bar{r}_j \times \bar{v}_j), \quad (3.12)$$

donde: m_j es la masa del j -ésimo *voxel*, r_j es el radio vector cuya magnitud es la distancia de una partícula al eje de rotación y v_j es la velocidad. En este caso los objetos se consideran como sólidos rígidos. La velocidad puede escribirse como:

$$\bar{v}_j = \bar{\omega}_j \times \bar{r}_j,$$

donde $\bar{\omega}_j$ es la velocidad angular. Considerando $\omega_i = \delta_{ik} \omega_k$, donde δ_{ik} es la delta de Kronecker (si $i = k$, entonces $\delta_{ik} = 1$; si $i \neq k$, entonces $\delta_{ik} = 0$). Las componentes del momento angular $\bar{L} = (L_1, L_2, L_3)$ son:

$$L_i = \omega_k \sum_{j=1}^n m_j (\delta_{ik} x_l^{(j)} x_l^{(j)} - x_i^{(j)} x_k^{(j)}), \quad (3.13)$$

donde: $i, k = [1, 2, 3]$, hay una suma sobre l , mientras que j se refiere a la j -ésima partícula (en este trabajo las partículas se consideran como *voxels* o *pixels*). La suma de la ecuación (3.13) se define como el *tensor de momento-inercia* T_{ik} :

$$T_{ik} = \sum_{j=1}^n m_j (\delta_{ik} x_l^{(j)} x_l^{(j)} - x_i^{(j)} x_k^{(j)}). \quad (3.14)$$

Usando nueve combinaciones diferentes de esta cantidad (que forma un tensor de segundo orden [50](Borisenko, 1979), la bien conocida expresión de *momentos regulares* para el caso discreto:

$$M_{ijk} = \sum x_1^p x_2^q x_3^r f(x, y, z), \quad (3.15)$$

y considerando $m_j = 1$ y $f(x, y, z) = 1$ (debido al hecho de que los objetos considerados son sólidos de densidad constante), se obtiene la *matriz de inercia* dada por:

$$T = \begin{pmatrix} M_{020} + M_{002} & -M_{110} & -M_{101} \\ -M_{110} & M_{002} + M_{200} & -M_{011} \\ -M_{101} & -M_{011} & M_{200} + M_{020} \end{pmatrix}. \quad (3.16)$$

Si se forma el producto interno entre T_{ik} y cualquier otro vector: $T_{ik} A_k = B_i$, el nuevo vector B con componentes B_i es diferente a A_i . En general, la operación $T_{ik} A_k$ puede rotar y cambiar la magnitud de A . Si se quiere encontrar todos los vectores que no estén rotados debido al producto interno, entonces se tiene que resolver la ecuación $T_{ik} A_k = \lambda A_i$, donde λ es un escalar. Si los vectores arriba mencionados

existen, estos constituyen los llamados *eigenvectores* o *ejes principales* del tensor T_{ik} . Para registrar u orientar los objetos debidamente, en este trabajo se utilizan los ejes principales. La igualdad anterior puede escribirse como:

$$T_{ik}A_k - \lambda A_i = (T_{ik} - \lambda\delta_{ik})A_i = 0,$$

6

$$\begin{aligned} (T_{11} - \lambda)A_1 + T_{12}A_2 + T_{13}A_3 &= 0 \\ T_{21}A_1 + (T_{22} - \lambda)A_2 + T_{23}A_3 &= 0 \\ T_{31}A_1 + T_{32}A_2 + (T_{33} - \lambda)A_3 &= 0. \end{aligned} \tag{3.17}$$

Este sistema tiene una solución no trivial, si y sólo si su determinante se anula. Cuando se resuelve esta ecuación, se obtiene un polinomio de tercer orden, con tres raíces: λ_1 , λ_2 y λ_3 . Se resolvió la ecuación al encontrar los eigenvectores como se presenta en [50](Borisenko, 1979). Para llevar a cabo los cálculos se utilizó el programa de aplicación *Mathematica* [51](Wolfram, 1991).

Ahora bien, para objetos irregulares y un eigenvector de tamaño dado, las tres λ s pueden ser diferentes. La Figura 3.3 muestra el comportamiento de los eigenvalores para un elipsoide.

En general, la λ menor corresponde a la dirección más elongada del objeto y la λ mayor a la dirección donde el objeto se alarga menos. Claramente, otros valores de λ corresponderán a situaciones intermedias. Por lo tanto, las λ s dan un cierto grado de información sobre la forma del objeto.

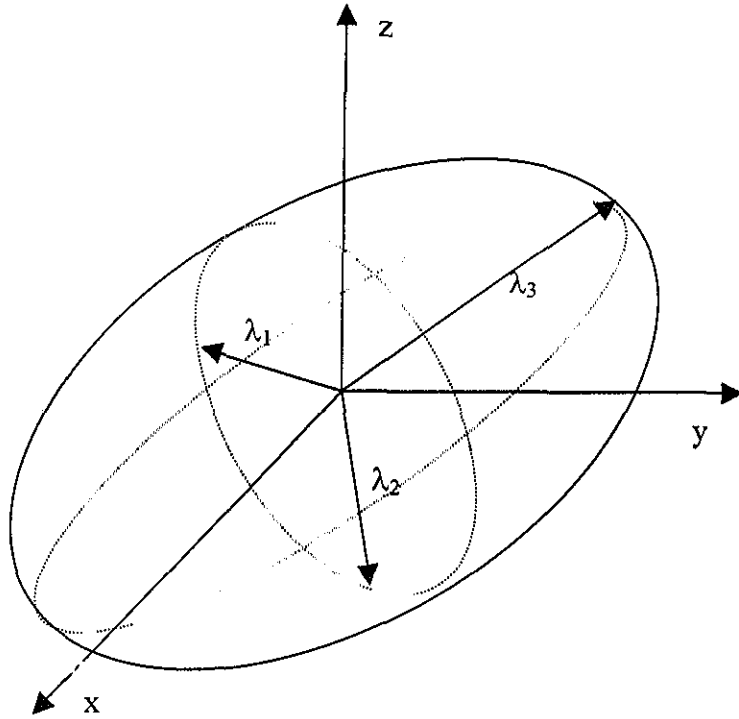


Figura 3 3. Los tres ejes principales de un elipsoide para ilustrar los eigenvectores de una figura irregular. Dados los eigenvalores $\lambda_3 \leq \lambda_2 \leq \lambda_1$, éstos corresponden a los eigenvectores cuyas magnitudes van del mayor al menor respectivamente.

3.3.2 Los Ángulos de Euler

Los ángulos de Euler $\{\theta, \phi, \psi\}$ son conocidos como una representación de una rotación en 3D, definidos como se muestra en la Figura 3.4; donde se ha considerado el sistema cartesiano x'_i a partir de una rotación 3D realizada sobre el sistema x_i . Los ángulos se encuentran en los siguientes intervalos: $\theta \in [0, \pi]$ y $[\phi, \psi] \in [0, 2\pi]$, y hay una intersección del plano $x_1 - x_2$ con el plano $x'_1 - x'_2$.

La rotación 3D representada en términos de cuaternios es:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & -2(q_0q_3 - q_1q_2) & 2(q_0q_2 + q_1q_3) \\ 2(q_0q_3 + q_1q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & -2(q_0q_1 - q_2q_3) \\ -2(q_0q_2 - q_1q_3) & -2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.18)$$

Donde $q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$, los cuales satisfacen $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$ e $\mathbf{i}, \mathbf{j}, \mathbf{k}$ están sujetas a las reglas:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1, \mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \mathbf{ki} = -\mathbf{ik} = \mathbf{j},$$

para una explicación más detallada ver [52](Kanatani, 1990).

En términos de los ángulos de Euler se tiene que [53]:

$$\begin{aligned} q_0 &= \cos(\theta/2)\cos[(\psi + \phi)/2], \\ q_1 &= \sin(\theta/2)\sin[(\psi - \phi)/2], \\ q_2 &= \sin(\theta/2)\cos[(\psi - \phi)/2], \\ q_3 &= \cos(\theta/2)\sin[(\psi + \phi)/2]. \end{aligned} \quad (3.19)$$

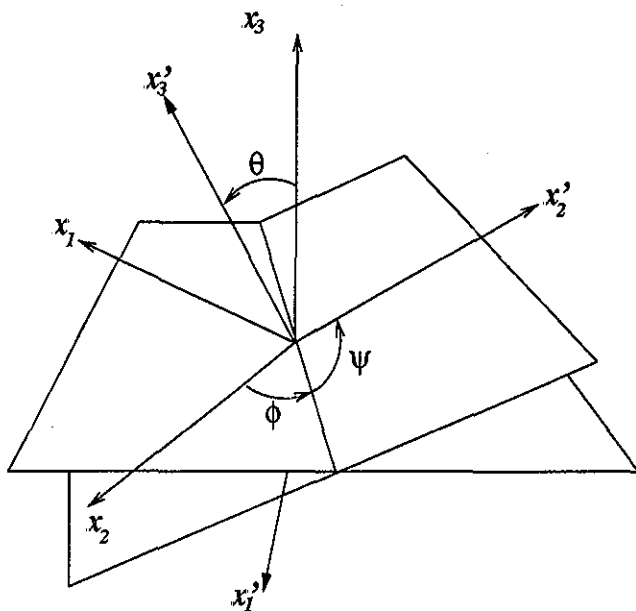


Figura 3.4. Ángulos de Euler.

3.4 Invariantes basados en transformadas integrales

En general, las transformadas integrales tienen la siguiente forma. Sean X y W dos conjuntos, sea f una función compleja valuada en el conjunto X , k una función compleja valuada en el conjunto $X \times W$. Entonces, la transformada integral de la función f sobre el conjunto X está dada por:

$$z(w) = \int_X f(x)k(w, x)dx, \quad (3.20)$$

donde z es una función compleja valuada en el conjunto W . Por las propiedades que tienen las transformadas (son invertibles y sus módulos son invariantes bajo la acción de transformaciones), se han utilizado diferentes versiones para reconocimiento de objetos, tales como la transformada de Fourier [54](Poularikas, 2000), la transfor-

mada de Mellin, la de Mellin-Fourier, la transformada Radon [55](Helgason, 1980), las cuales se han utilizado ampliamente como invariantes en rotación, traslación y escala principalmente, para reconocimiento de objetos en 2D. A diferencia de los momentos, una ventaja en el uso de las transformadas es que son más eficientes al programarlas, no son sensibles al ruido, pero hasta el momento no se sabe de aplicaciones hechas a imágenes binarias en 3D. Existen otras transformadas que se han utilizado para el manejo de las imágenes, nuevamente limitadas al caso bidimensional, o bien, para reconocimiento tridimensional a partir de imágenes 2D. Otras transformadas utilizadas en el análisis de imágenes son: la transformada Wavelet [54](Poulikaris, 2000), la de Gabor [54], la transformada Cortex [56] y, recientemente, Martens [57](1997) ha propuesto la transformada de Hermite. La principal aplicación de estas transformadas es la de codificar, comprimir, segmentar o extraer contornos de imágenes bidimensionales.

Los objetos están ya orientados por medio del método de los ejes principales y compuestos del mismo número de *spels*. En esta etapa se han concluido los pasos de normalización de escala, coincidencia del centro de masa y alineación por ejes principales. Los objetos pueden ahora ser transformados entre sí.

Capítulo 4

¿Cómo transformar un objeto en otro?

En este capítulo se explicará la forma de llevar a cabo la transformación de los objetos, con el fin de compararlos entre sí. Cuando dos objetos están alineados, normalizados y con los mismos centros de masa, el método aquí propuesto transforma el primer objeto en el segundo. Lo anterior constituye una ventaja debido a que, cuando esta transformación se lleva a cabo, se calcula la diferencia de las formas de los objetos. Entonces, cuando los *spels* se distribuyen de manera similar en los dos objetos, estos pueden verse más parecidos entre sí. Lo contrario sucede cuando la diferencia en la distribución sea muy marcada, en tal caso parecerán ser más diferentes entre sí. En el primer caso la distancia total recorrida por los *spels* será menor al transformar los objetos.

4.1 Pasos para transformar los objetos

Los pasos para transformar el objeto O_1 en el objeto O_2 son como sigue:

1. *Encontrar los spels comunes y dejarlos sin cambiar.* Cuando los objetos O_1 y O_2 tienen el mismo centro de masa, los mismos ejes principales y la misma escala (la cual se encuentra al normalizar el área o volumen), se lleva a cabo la superposición de los objetos mencionados, lo que define los *spels* comunes. Así, sea la imagen binaria 3D de O_1 como I_{O_1} y la imagen binaria 3D de O_2 como I_{O_2} .

El conjunto de *spels* comunes I_C se define como:

$$I_C = I_{O_1} \cap I_{O_2}. \quad (4.1)$$

2. *Cómo encontrar los spels positivos.* Los *spels* positivos corresponden a los *spels* a moverse, representados por la imagen binaria 3D, se trata del conjunto I_P , i.e.

$$I_P = I_{O_1} \setminus I_{O_2}. \quad (4.2)$$

3. *Cómo encontrar los spels negativos.* El conjunto de *spels* negativos corresponde a la imagen binaria 3D I_N , definido por:

$$I_N = I_{O_2} \setminus I_{O_1}, \quad (4.3)$$

i.e., la imagen binaria I_N representa a los *spels* donde serán colocados los *spels* positivos (véase la Figura 4.1).

4. *Cómo desplazar los spels.* Hay muchas formas de desplazar los *spels*. Si k es el número de *spels* a mover, entonces $k!$ es el número total de formas diferentes para desplazarlos de I_P a I_N . Las imágenes binarias y las distancias entre los *spels* pueden representarse como una gráfica bipartita pesada [58](Bondy, 1976) con bipartición (I_P, I_N) , donde:

$$I_P = \{p_i : i \leq k\}, \quad (4.4)$$

$$I_N = \{n_j : j \leq k\} \quad (4.5)$$

y la arista $p_i n_j$ tiene un peso w_{ij} (cada peso w_{ij} corresponde a la distancia Euclidiana entre los *spels* p_i y n_j). Así, el *problema de asignación*

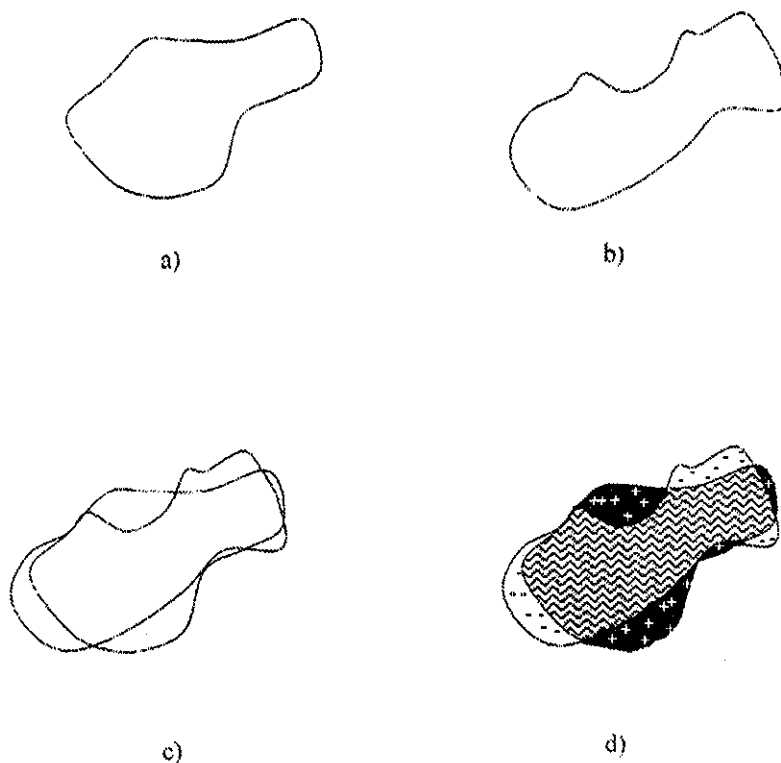


Figura 4.1. Dos objetos dados por a) y b) están alineados; c) superposición de ambos objetos; d) la intersección de los objetos está dada por las líneas en zigzag, la parte en negro con signo más corresponde a los *spels* positivos, mientras que la parte en blanco con signo menos corresponde a los *spels* negativos.

óptima consiste en encontrar un apareo perfecto de peso mínimo en la gráfica pesada, lo que se conoce como *apareo óptimo*.

La Figura 4.2 ilustra la gráfica completa pesada con bipartición (I_P, I_N) , donde las líneas en negro representan al apareo óptimo.

Para encontrar un apareo óptimo en una gráfica bipartita pesada se puede utilizar el *algoritmo húngaro* [58](Bondy, 1976). Así, usando este algoritmo, se minimiza la distancia total recorrida por los *spels* a desplazarse (*spels* positivos), lo que produce

una transformación óptima de los objetos.

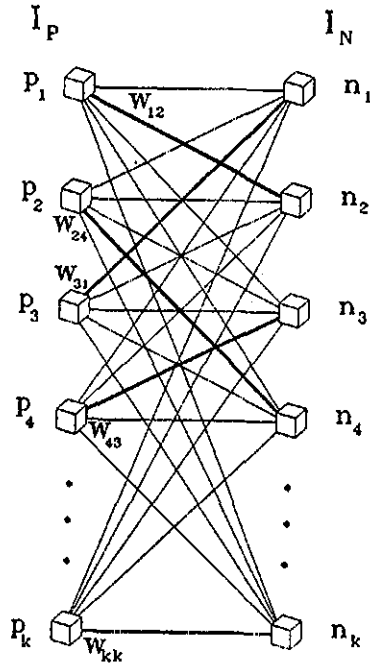


Figura 4.2. Gráfica pesada completa con bipartición (I_P, I_N) ,
(*spels positivos, spels negativos*).

4.2 Método de Optimización

En esta sección se explicará el método seguido para encontrar la distancia mínima total que emplearán los *spels* en la transformación de los objetos.

A continuación se presenta una introducción a la teoría de gráficas para posteriormente estudiar el algoritmo húngaro, el cual permitirá encontrar la distancia mínima para desplazar todos los *spels* positivos en negativos.

4.2.1 Conceptos básicos de la teoría de gráficas

Una gráfica es un conjunto finito, no vacío, compuesto de un conjunto de vértices $V = \{v_1, v_2, \dots, v_n\}$ y de un conjunto de aristas E , formado por parejas de la forma $\{v_i, v_j\}$, con $v_i, v_j \in V$. Sea E el conjunto de parejas (aristas) de la forma $\{v_i, v_j\}$. Una gráfica puede definirse simplemente a través de los conjuntos V y E como $G = G < V, E >$ [59](Chartrand, 1985).

A cada arista de una gráfica G se asocia un número real $w(e)$, al cual se llama su *peso*. En este sentido a G se le llama *gráfica pesada*. Si H es una subgráfica de G , el peso $w(H)$ es la suma de los pesos de las aristas de H , $\sum w(e)$. Cuando se desea que la suma de los pesos de las aristas que conforman una subgráfica H sea la menor, esta subgráfica debe ser *conectada* (es decir, que cada par de vértices puedan ser alcanzados por un camino dentro de la gráfica) y *acíclica* (que no haya un camino que conecte a un mismo vértice). En este caso se obtiene un *spanning tree* de *peso mínimo*. El peso mínimo de una gráfica es aquel tal que $w(H) = w(e)$ sea el menor de todas las subgráficas conectadas de G .

4.2.2 Gráficas bipartitas

Una gráfica $G < V, E >$ es bipartita si existe una partición de los vértices en dos conjuntos U y V , tales que si (u, v) está en E implica que $u \in U$ y $v \in V$ (o al revés).

4.2.3 Camino alternante y camino extendido

Sea $M(E)$ un subconjunto de aristas de la gráfica G . Se dice que M es un *apareo* (en G) si sus vértices son adyacentes a, a lo más, una arista de M . Se dice que un apareo M satura a un vértice v , cuando es extremo de una arista en E , es decir v está

saturado si alguna arista de M es adyacente a v , de otra forma, v no está saturado. Si todos los vértices de G son saturados, el apareo M es *perfecto*. Sea M un apareo en G , un *camino alternante* en G es un camino cuyas aristas están alternativamente en $E \setminus M$ y en M . Un *camino extendido* es un camino alternante cuyo inicio y final no están saturados (Figura 4.3).

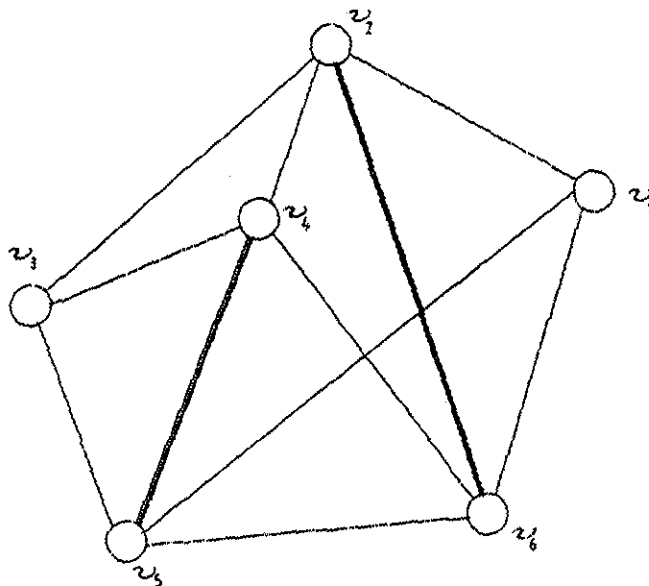


Figura 4.3. Camino extendido dado por $v_1, v_2, v_6, v_4, v_5, v_3$.

4.2.4 Problemas de asignación óptima

El problema de *asignación óptima* que se considerará consiste en que la suma de las distancias que los *spels* que conforman a un objeto tienen que recorrer, para transformarlo en otro objeto, sea la menor de todas las posibilidades. Sea una gráfica bipartita pesada completa con bipartición (X, Y) , donde $X = \{x_1, x_2, \dots, x_n\}$, $Y = \{y_1, y_2, \dots, y_n\}$. El conjunto X puede representar al conjunto de *spels* de un objeto a transformar y

el conjunto Y a los *spels* del objeto transformado. Las aristas (x_i, y_j) tienen un peso $w_{ij} = w(x_i, y_j)$. En esta gráfica pesada se debe encontrar un apareo perfecto de peso mínimo, es decir, un *apareo óptimo*. El algoritmo húngaro [58](Bondy, 1976) ya mencionado, puede encontrar *apareos óptimos*.

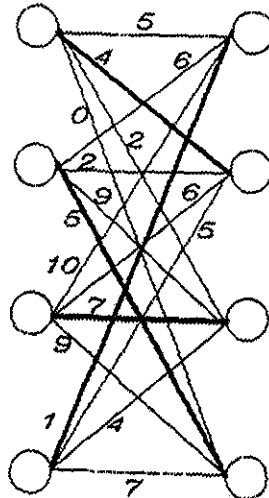


Figura 4.4. Una gráfica bipartita con un apareo óptimo.

La Figura 4.4 muestra un apareo óptimo, dado por las aristas remarcadas en negro. Bondy y Murty (1976) [58] presentan un algoritmo que resuelve el problema de asignación al encontrar un apareo óptimo de peso máximo. Gould [60](1988) encuentra el apareo óptimo de peso mínimo basado en el algoritmo húngaro.

El problema de encontrar el apareo óptimo de una gráfica bipartita pesada, es equivalente a resolver lo que en la programación lineal se conoce como *problemas de transporte*, que a continuación se explica.

4.3 Problemas de transporte

Se desea determinar el costo total mínimo para trasladar m unidades de producto

material a n lugares específicos, donde cada unidad de producto tiene cierto costo de envío (para una explicación detallada de esta sección se sugiere ver las referencias [61](Gass, 1958) y [62](Hadley, 1962)).

Sea a_i la cantidad del producto disponible en el origen i y b_j la cantidad de producto requerido en el destino j . Sea c_{ij} el costo para enviar una unidad de producto del origen i al destino j . Se supondrá que:

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j, \quad (4.6)$$

de manera que la cantidad total requerida para los destinos sea precisamente la misma que la disponible en los orígenes. Entonces, si x_{ij} es la cantidad trasladada del origen i al destino j , se desea encontrar $x_{ij} \geq 0$ que satisfaga las $m + n$ restricciones siguientes:

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= a_i, \quad a_i > 0 \quad i = 1, \dots, m, \\ \sum_{i=1}^m x_{ij} &= b_j, \quad b_j > 0 \quad j = 1, \dots, n, \end{aligned} \quad (4.7)$$

y que minimice:

$$z = \sum_{i,j}^{m,n} c_{ij} x_{ij}. \quad (4.8)$$

La cantidad total disponible en los orígenes pudo haber sido mayor que la cantidad total disponible en los destinos. Sin embargo, cualquier problema de transporte puede reducirse a las igualdades anteriores, de manera que no hay pérdida de generalidad.

Las restricciones del problema de transporte se pueden escribir en forma matricial, viéndolo como un problema de programación lineal, $\mathbf{Ax} = \mathbf{b}$. Escríbase:

$$\mathbf{x} = [x_{11}, \dots, x_{1n}, x_{21}, \dots, x_{2n}, \dots, x_{m1}, \dots, x_{mn}], \quad \mathbf{b} = [a_1, \dots, a_m, b_1, \dots, b_n].$$

Si las restricciones se escriben como:

$$\begin{array}{rcl}
 x_{11} + x_{12} + \dots + x_{1n} & & = a_1, \\
 & x_{21} + \dots + x_{2n} & = a_2, \\
 & & \vdots \\
 & & x_{m1} + \dots + x_{mn} = a_m, \\
 x_{11} & + x_{21} + \dots + & x_{m1} = b_1, \\
 & & \vdots \\
 & x_{1n} + & x_{2n} + & \dots + x_{mn} = b_n,
 \end{array}$$

estas ecuaciones toman la forma $\mathbf{Ax} = \mathbf{b}$, donde:

$$\mathbf{A} = \begin{bmatrix}
 1_n & 0 & 0 & \dots & 0 \\
 0 & 1_n & 0 & \dots & 0 \\
 0 & 0 & 1_n & \dots & 0 \\
 \vdots & & & \ddots & \\
 0 & 0 & 0 & \dots & 1_n \\
 I_n & I_n & I_n & \dots & I_n
 \end{bmatrix} \quad (4.9)$$

\mathbf{A} es una matriz de $(m+n) \times (mn)$. $\mathbf{1}$ es el vector de unos y el subíndice n significa que hay n componentes.

La ecuación $\mathbf{Ax} = \mathbf{b}$ puede constituir una definición rigurosa de lo que se entiende por problema de transporte. Cualquier problema de programación lineal $\mathbf{Ax} = \mathbf{b}$,

$x \geq 0$, \max ó $\min z = cx$ se llamará problema de transporte si se puede poner como en la ecuación anterior. Para cualquier problema de optimización en la programación lineal que pretenda maximizar una cantidad, existe un problema dual, en el que puede reformularse la cantidad a maximizar por otra cantidad a minimizar. A continuación se explica el problema dual.

4.3.1 Problema dual

Sea:

$$\sum_{j=1}^r d_{ij}x_j \geq d_i, i = 1, \dots, m + n,$$

En forma matricial la ecuación anterior se puede escribir como:

$$Dx \geq d,$$

donde d puede ser positivo, negativo o cero.

Dado cualquier problema de programación lineal:

$$Dx \geq d, \quad x \geq 0, \quad \min z = cx,$$

existe otro problema de programación lineal:

$$D'w \leq c', \quad w \geq 0, \quad \max Z = d'x,$$

con w como el vector de *variables duales*.

Considérese un ejemplo de tres orígenes y tres destinos. Se tiene que $Dx \geq d$ toma el caso particular:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{31} \\ x_{32} \\ x_{33} \end{bmatrix} \geq \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}, \quad (4.10)$$

y

$$\text{minimizar } z = \sum_{i,j}^{3,3} c_{ij} x_{ij}. \quad (4.11)$$

El problema dual a las ecuaciones (4.10) y (4.11) es el siguiente:

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \leq \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{21} \\ c_{22} \\ c_{23} \\ c_{31} \\ c_{32} \\ c_{33} \end{bmatrix}. \quad (4.12)$$

O bien $u_i + v_j \leq c_{ij}$

y

$$\text{maximizar } Z = \sum_i u_i + \sum_j v_j. \quad (4.13)$$

Donde u_i y v_j son las variables duales.

De esta manera, buscar una cantidad mínima (en este caso dada por una distancia) de una gráfica bipartita pesada, es equivalente a buscar una cantidad máxima dada por variables duales, como se explica a continuación.

4.4 Algoritmo húngaro

Si, por simplicidad, las cantidades de productos disponibles en los orígenes y destinos se consideran como unitarios, las constricciones a la ecuación 4.7 pueden escribirse como:

$$\begin{aligned} \sum_{j=1}^n x_{ij} &\geq 1, i = 1, \dots, m, \\ \sum_{i=1}^m x_{ij} &\geq 1, j = 1, \dots, n, \\ x_{ij} &\geq 0. \end{aligned} \tag{4.14}$$

Si se considera una gráfica bipartita con bipartición (S, T) , entonces $x_{ij} = 1$ si y sólo si la arista de $i \in S$ a $j \in T$ está incluida en un apareo óptimo. Las constricciones garantizan que a lo más una de las aristas en cualquier solución posible sea incidente a cualquier vértice (es decir, que un vértice sea saturado por, a lo más, una arista).

El algoritmo húngaro, desarrollado por Kuhn [63](1955), comienza por considerar lo que en programación lineal se llaman variables duales. Dadas u_i con vértices $i \in S$ y v_j con vértices $j \in T$, se tiene que:

$$u_i + v_j \leq c_{ij} \quad \forall i, j$$

$$u_i \geq 0 \text{ y } v_j \leq 0 \quad \forall i, j. \quad (4.15)$$

donde el problema se convierte en encontrar el

$$\text{máx}(\sum_{i=1}^n u_i + \sum_{j=1}^n v_j).$$

La *complementariedad débil* [63] implica que:

- (a) si $x_{ij} = 1$ entonces $u_i + v_j = c_{ij}$,
 - (b) si $u_i > 0$ entonces $\sum_j x_{ij} = 1$,
 - (c) si $v_j < 0$ entonces $\sum_i x_{ij} = 1$.
- (4.16)

El algoritmo húngaro trata de encontrar un camino extendido, de manera que la subgráfica obtenida esté conformada por aristas tales que $u_i + v_j = c_{ij}$ y que no se violen las condiciones (a) y (c). Si este camino no se encuentra, enseguida se ajustan las variables duales, de manera que se añada al menos una arista más a la subgráfica. Para encontrar el apareo óptimo de peso mínimo, a continuación se escribe el algoritmo húngaro basado en la versión de Evans [64](1992) (quien encuentra un apareo óptimo de peso máximo, pero con las condiciones 4.14 a 4.16 se ha adaptado por el autor de esta tesis para encontrar el peso mínimo).

- *Inicio.* Usando las variables duales u_i y v_j , e introduciendo una nueva variable, π_j , encuéntrese $u_i = \min_j [c_{ij}]$ y hágase $\pi_j = \infty$, $j \in T$. Construir una subgráfica que consiste de las aristas tales que $u_i + v_j = c_{ij}$. Para cada $i \in S$, elegir la primera arista (i, j) tal que j no esté saturada y ponerla en el apareo inicial M . En este momento ningún vértice ha sido visitado ni etiquetado.

- *Paso 1* Etiquetar cada vértice no saturado $i \in S$ con $p(i) = 0$.
- *Paso 2*. Seleccionar cualquier vértice $i \in S$ que no haya sido visitado pero que esté etiquetado, o bien, elegir $j \in T$ tal que $\pi_j = 0$. Si ninguno de ellos existe ir al paso 5.
- *Paso 3* Si el vértice elegido en el paso 2 es $i \in S$, entonces para cada arista (i, j) que no esté en M , y si $-u_i + v_j + c_{ij} < \pi_j$, etiquetar el vértice $j \in T$ con $p(j) = i$, reemplazando cualquier etiqueta y hacer $\pi_j = -u_i + v_j + c_{ij}$. Si el vértice elegido en el paso 2 es $j \in T$, entonces si j no está saturado, ir al paso 4. Si no, entonces hay una arista $(i, j) \in M$; etiquetar el vértice $i \in S$ con $p(i) = j$. Regresar al paso 2.
- *Paso 4*. Se ha encontrado un camino extendido que termina en el vértice $i \in S$ ó $j \in T$ (dependiendo de cual se haya elegido en el paso 3). Construir este camino usando la función predecesora $p(\cdot)$. Aumentar el apareo M al añadir todas las aristas del camino extendido que no estén en M y quitar de M las que ya estaban. Hacer $\pi_j = \infty \forall j \in T$, borrar todas las etiquetas y regresar al paso 1.
- *Paso 5*. Calcular $\delta_1 = \max \{u_i \in S\}$, $\delta_2 = \min \{\pi_j > 0, j \in T\}$, y $\delta = \min \{\delta_1, \delta_2\}$. Hacer $u_i = u_i - \delta$ para cada vértice etiquetado $i \in S$. Hacer $v_j = v_j + \delta, \forall j \in T$ con $\pi_j = 0$. Hacer $\pi_j = \pi_j - \delta$ para cada vértice $j \in T$ etiquetado con $\pi_j > 0$. Si $\delta = \delta_2$, ir al paso 2. De otra forma se ha encontrado el apareo óptimo de peso máximo.

4.4.1 Algoritmo húngaro según Gould[60](1988)

El algoritmo húngaro dado en la sección anterior tiene una complejidad de $O(n^3)$. Un método reportado que intenta mejorar el tiempo de ejecución del algoritmo húngaro, es el dado por Jonker y Volgenant [65](1986). Este método está basado en el siguiente algoritmo [60](Gould, 1988), el cual tiene las limitaciones que ya fueron reportadas por el autor de esta tesis [66](Sánchez, 2000).

Sea el peso de un apareo M dado por $W(M) = \sum w(e), e \in M$. Una solución

óptima es un apareo perfecto con $W(M)$ como mínimo.

- 1. Representar la gráfica bipartita en forma de matriz de tamaño $n \times n$, $U = [w_{ij}]$.

Ver Figura 4.5.

- 2. Minimizar la matriz de la siguiente manera:
 - Elegir el menor valor de cada renglón y restarlo a ese renglón.
 - Elegir el menor valor de cada columna y restarlo a esa columna.
- 3. Encontrar n ceros *independientes*, esto es, se etiquetan n ceros, tales que no haya otro cero (etiquetado) en el renglón y columna. Se seleccionan los números correspondientes a estos ceros en la matriz original como solución.

U	y_1	y_2	y_3	y_4
x_1	5	4	2	0
x_2	6	2	9	5
x_3	10	6	7	9
x_4	1	5	4	7

Figura 4.5. Representación matricial de la gráfica de la Figura 4.4, sección 4.2.

- 4. Sin embargo, la gráfica inicial puede ser tal que al minimizarla por primera vez no se puedan encontrar n ceros independientes, en este caso se marcan los renglones y/o columnas que contengan ceros. La idea es “cubrir” todos los ceros (se utilizará el menor número de líneas posibles para marcarlos).
- 5. Encontrar el número menor, **min**, que esté en la submatriz, es decir que no esté en ninguno de los renglones y/o columnas marcados.
- 6. Restar **min** a las columnas y renglones de la submatriz y sumar **min** a los pesos que están doblemente marcados.

- 7. Enseguida una función inspecciona si se pueden encontrar n ceros independientes. Si es así, el algoritmo termina satisfactoriamente y se llega a un apareo óptimo. Si no, se va otra vez al paso 4.
- 8. Se suman los valores que estaban en la matriz original en la posición de los ceros independientes de la última matriz modificada, obteniendo el valor buscado $W(M)$.

Una forma de localizar los llamados ceros independientes de manera eficiente es utilizando el llamado *Árbol húngaro*.

4.4.2 Árbol húngaro.

Para ejecutar el paso 7 se considera una matriz con posibles ceros independientes, para buscarlos se procede a construir al llamado *árbol húngaro*, y posteriormente se busca el llamado *camino húngaro* dado por un camino extendido.

- 1. Para construir el árbol húngaro se visitan los ceros de la matriz y se elige un apareo cualquiera. La Figura 4.6a) muestra el apareo obtenido a partir de los ceros encontrados y la Figura 4.6b) la elección de un apareo arbitrario.
- 2. Se toma un vértice raíz x de la gráfica bipartita.
- 3. A partir de este vértice raíz se van buscando todos los caminos alternantes posibles.
- 4. Se repite el paso anterior con los vértices x 's para los que las y 's están saturadas.
- 5. Si alguna y no encuentra su respectiva x para ser saturada, entonces se ha encontrado un camino extendido.
- 6. Si hay otra x que no esté saturada (raíz) regresar a 3. De lo contrario se detiene el programa y se construye un nuevo apareo M' al usar el camino extendido, en el que se intercambian las x adyacentes a las y 's que no estaban en el apareo M anterior

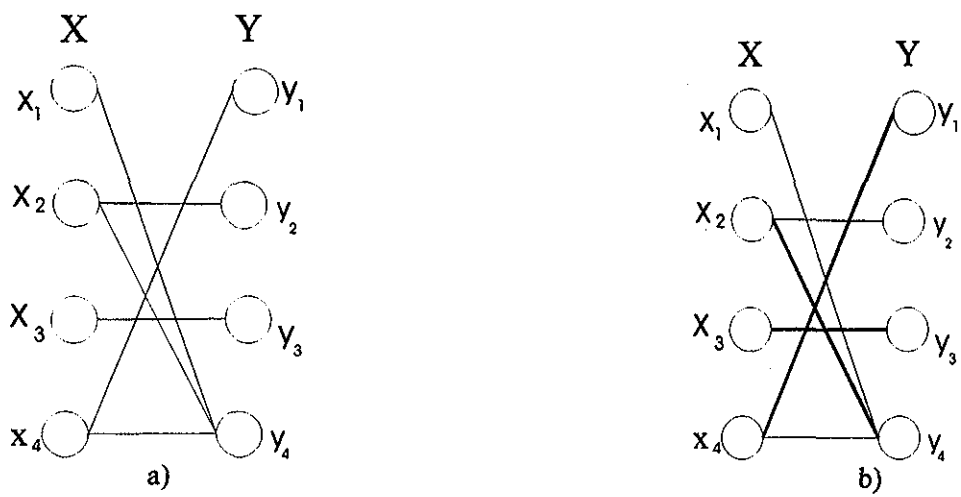


Figura 4.6. a) Apareo, b) apareo perfecto arbitrario, dado por las líneas gruesas.

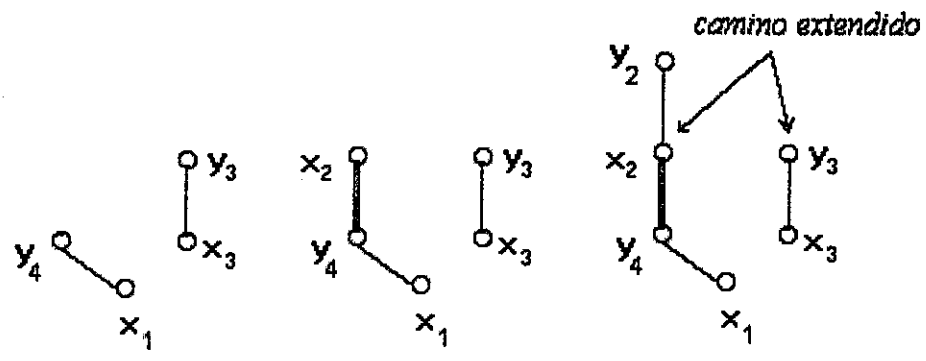


Figura 4.7. Caminos extendidos: x_1, y_4, x_2, y_2 y x_3, y_3 .

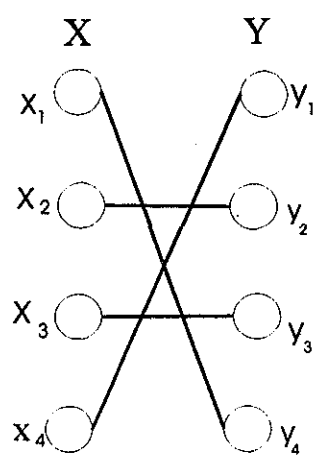


Figura 4.8. Apareo óptimo.

TESIS CON
FALLA DE ORIGEN

por un apareo nuevo M' , y viceversa, se sacan del apareo anterior la (x,y) que se encontraban bajo M . De esta manera se ha encontrado un apareo perfecto. Puesto que los pesos son todos iguales a cero, se trata de un apareo óptimo.

- 7. Si nunca se encontró el camino extendido regresar al paso 1. En la Figura 4.7 se muestra que el camino dado por x_1, y_4, x_2, y_2 es el camino extendido buscado

Así, el nuevo apareo queda como en la Figura 4.8.

Jonker y Volgenant [65](1986) sugieren encontrar el mínimo de un sólo renglón no marcado, restarlo a la submatriz no marcada y sumarlo a los elementos doblemente marcados.

En ambos casos, la de Gould y la de Jonker y Volgenant, las limitaciones que pueden aparecer al aplicar estos métodos se explican a continuación

4.4.3 Limitaciones del método de Gould

En el transcurso de la búsqueda de los ceros independientes se puede llegar a una etapa de la matriz como la presentada en la Figura 4.9.

A esta matriz no se le puede extraer sus 10 ceros independientes. A través del método empleado aquí para encontrar los ceros independientes, los renglones y columnas a marcar quedan como se muestra en la Figura 4.9. Como se puede observar, ya no es posible trabajar con una submatriz que permita buscar los 10 ceros independientes. Sin embargo, al hacerlo manualmente se cubrirían los ceros de la manera en que se muestra en la Figura 4.10. En ella se puede observar que se ha empleado una línea más para poder marcar todos los ceros y así seguir buscando los ceros independientes. En el apéndice A se muestran algunos objetos de baja resolución que fueron transformados con el algoritmo de Gould [60](1988). Sin embargo, para continuar con este trabajo, se seguirá el algoritmo húngaro basado en la versión de Evans [64](1992)

a través de los cinco pasos que establece para encontrar el llamado camino extendido de una gráfica bipartita, el cual permite encontrar su peso mínimo y explicado al principio de esta sección.

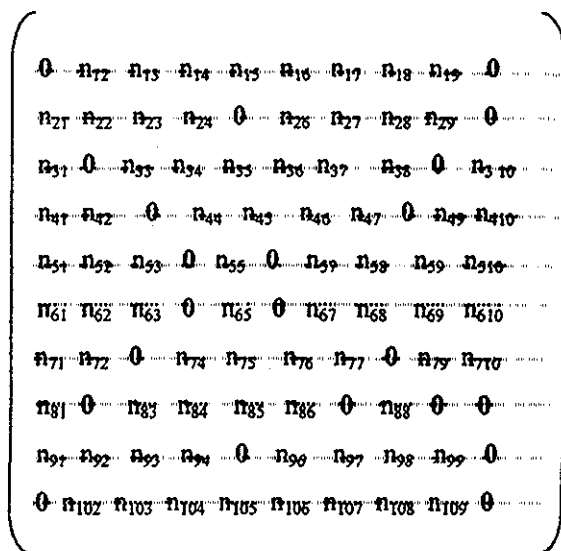


Figura 4.9. Matriz con todos sus ceros marcados con el mínimo número (diez) de renglones y/o columnas.

4.5 El trabajo como medida de similitud

Esta sección establece un procedimiento para medir el grado de parecido de formas entre dos objetos, basado en la transformación de uno en otro [1](Bribiesca, 1996). Objetos diferentes requerirán una mayor energía en la transformación de uno a otro, mientras que objetos similares requerirán muy poca. Cuando dos objetos son idénticos la energía de transformación será igual a cero. Intuitivamente se esperaría que los objetos a transformar tengan el máximo número de *spels* comunes como sea posible. Se hicieron algunos experimentos (como en el capítulo 2) y se observó que esto no

necesariamente es cierto. Un resultado importante de este trabajo es haber estudiado que los ejes principales, como método de registro de objetos, logran prescindir de la idea de correlación máxima (máxima área o volumen entre los objetos a transformar), lo que permite encontrar la energía mínima de transformación.

0	n ₁₂	n ₁₃	n ₁₄	n ₁₅	n ₁₆	n ₁₇	n ₁₈	n ₁₉	0
n ₂₁	n ₂₂	n ₂₃	n ₂₄	0	n ₂₆	n ₂₇	n ₂₈	n ₂₉	0
n ₃₁	0	n ₃₃	n ₃₄	n ₃₅	n ₃₆	n ₃₇	n ₃₈	0	n ₃₁₀
n ₄₁	n ₄₂	0	n ₄₄	n ₄₅	n ₄₆	n ₄₇	0	n ₄₉	n ₄₁₀
n ₅₁	n ₅₂	n ₅₃	0	n ₅₅	0	n ₅₇	n ₅₈	n ₅₉	n ₅₁₀
n ₆₁	n ₆₂	n ₆₃	0	n ₆₅	0	n ₆₇	n ₆₈	n ₆₉	n ₆₁₀
n ₇₁	n ₇₂	0	n ₇₄	n ₇₅	n ₇₆	n ₇₇	0	n ₇₉	n ₇₁₀
n ₈₁	0	n ₈₃	n ₈₄	n ₈₅	n ₈₆	0	n ₈₈	0	0
n ₉₁	n ₉₂	n ₉₃	n ₉₄	0	n ₉₆	n ₉₇	n ₉₈	n ₉₉	0
0	n ₁₀₂	n ₁₀₃	n ₁₀₄	n ₁₀₅	n ₁₀₆	n ₁₀₇	n ₁₀₈	n ₁₀₉	0

Figura 4.10. Matriz en donde sus ceros son marcados con 11 renglones y/o columnas, permitiendo que quede una submatriz.

En la transformación cada *spel* se mueve de la posición *X* a la posición *Y*, lo que implica de un trabajo realizado. De la física clásica el trabajo *dW* se define como

$$dW = F ds. \tag{4.17}$$

Así, la energía para transformar los objetos está dada por el trabajo total:

$$W_H = \sum_{i=1}^n F_i s_i, \tag{4.18}$$

TESIS CON
FALLA DE ORIGEN

donde F_i es la fuerza aplicada a cada *spel* para moverlo una distancia s_i . El subíndice H es para recordar que se ha empleado el algoritmo húngaro. Se puede construir una métrica para transformar un objeto A en un objeto B ($W(A \rightarrow B)$):

$$\begin{aligned}
 W(A \rightarrow B) &= W(B \rightarrow A), \\
 W(A \rightarrow B) &\geq 0, \\
 W(A \rightarrow B) &= 0 \Leftrightarrow A = B, \text{ y} \\
 W(A \rightarrow C) &\leq W(A \rightarrow B) + W(B \rightarrow C).
 \end{aligned}
 \tag{4.19}$$

Sea F_i (la fuerza para mover los *spels*) constante e igual a 1 (un Newton o una dina, por ejemplo). Por el momento no importará la masa de los *spels* o la fricción que podría “obstaculizar” su movimiento. De manera que la energía para transformar a los dos objetos será:

$$W = \sum_{i=1}^n s_i
 \tag{4.20}$$

Donde la ecuación anterior está dada en unidades de trabajo. Por tanto, el valor numérico de la energía será igual a la distancia total recorrida, y cuyo valor es igual al peso total de un apareo óptimo.

Así, la pregunta ¿cuál es el parecido entre objetos? se logra contestar a través de calcular la *medida de similitud de la forma* entre dos objetos. En cierto sentido, esta pregunta se puede plantear como: “¿qué tan lejos se mueven todos los *spels* para transformar los objetos?”, en vez de (como podría suceder con un método de correlación) “¿cuántos *spels* hay que mover para transformar los objetos?”.

Capítulo 5

Propiedades importantes de la medida propuesta

Cuando se pretende hacer reconocimiento de patrones se constituyen vectores, cuyas componentes serán los *atributos* de esos patrones [67](Meisel, 1974). A continuación se verá el comportamiento de algunos atributos de diferentes patrones, que en este caso se trata de objetos 2D. El estudio se hará en términos de tres atributos, a saber: número de *pixels* positivos, la *compacidad* de cada objeto binario y el trabajo realizado al transformar cada uno de los objetos en el círculo. Posteriormente, se harán las comparaciones entre todos los objetos.

5.1 La compacidad discreta en 2D

En 1997 Bribiesca [68] estableció una medida de compacidad discreta (cuyos valores están en el intervalo $[0,1]$) para objetos 2D compuestos por *pixels*. Esta medida relaciona el perímetro de contorno con el perímetro de contacto, esto es, con la longitud total de los lados de los *pixels* que se tocan (conformando un perímetro interior al objeto 2D).

Dado un objeto 2D compuesto por n *pixels*, sea P_c el perímetro total de los lados de contacto de todos *pixels* que conforman el objeto. La compacidad discreta está definida por la expresión:

$$C_D = \frac{P_c - P_{cmin}}{P_{cmax} - P_{cmin}}, \quad (5.1)$$

donde:

$$P_{cmin} = n - 1, \quad (5.2)$$

y, con el objeto de normalizar la compacidad discreta, P_{cmax} se define como:

$$P_{cmax} = 2(n - n^{1/2}). \quad (5.3)$$

5.2 Relación entre la compacidad discreta, el trabajo de transformación y el número de *pixels* positivos

Considérense nuevamente los 12 objetos introducidos en el Capítulo 3, pero ahora normalizados y alineados. La Figura 5.1 muestra los objetos (las formas de algunos de ellos corresponden a insectos conocidos) ya alineados conforme a sus ejes principales.

En esta sección se considerará al círculo como objeto prueba para ser comparado por los 11 objetos restantes debido a que tiene una alta compacidad (muy cercana al 1), es decir, al calcularse con la ecuación 5.1 su compacidad es de 0.9954,

La Tabla 5.1 y la Figura 5.2 muestran el trabajo realizado para transformar los objetos en el círculo a través de los dos algoritmos para el cálculo del trabajo total: el húngaro, propuesto aquí (W_H) y el propuesto por Bribiesca (EMC) [1](Bribiesca, 1996), al cual se le ha llamado algoritmo por elementos más cercanos. La Tabla 5.1 contiene, además, el número de *pixels* positivos y la compacidad discreta de cada objeto.

Como se puede observar, el objeto más parecido al círculo es el círculo con ruido (circur), seguido del círculo con un cabello (cirpel). El más diferente al círculo es

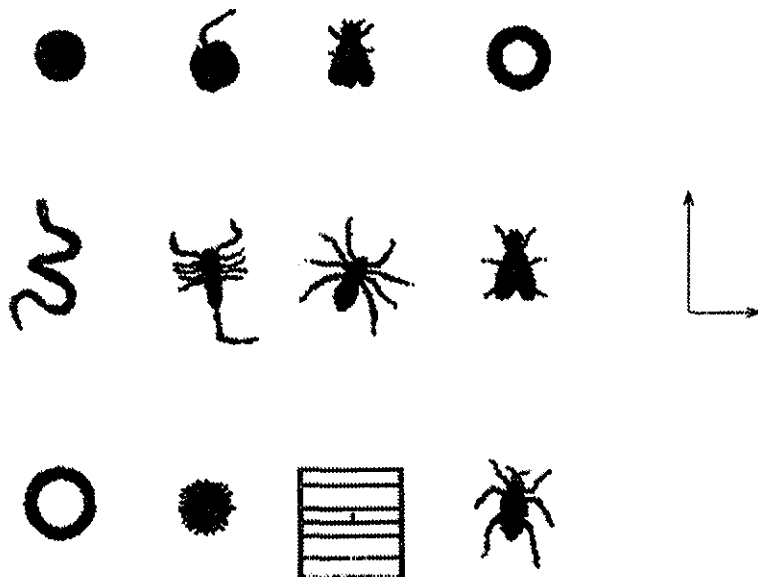


Figura 5.1. Doce objetos normalizados y alineados. El eje vertical indica la dirección del eje principal mayor, mientras que el horizontal el eje principal menor.

el marco, seguido de la serpiente. La Tabla 5.1 muestra que existe una pequeña diferencia entre los trabajos realizados por los diferentes algoritmos (W_H y EMC). Esta diferencia va de cerca del 1%, para el caso del anillo2 (el cual es muy compacto), hasta cerca del 7%, para el caso de circur (también con una alta compacidad).

Cabe aclarar que esto no significa que la diferencia de errores entre los métodos empleados esté también en ese intervalo. Solamente se han mostrado los trabajos realizados para transformarlos, sin embargo, los objetos ya se alinearon conforme a los ejes principales, a diferencia del método propuesto por Bribiesca, quien prefería buscar una correlación máxima. Sólo se pretende poner a prueba el algoritmo húngaro y confirmar, en todos los casos, que la distancia total de transformación es menor o

OBJETOS	<i>Pixels</i> coms	<i>Pixels</i> (+)	C_D	W_H	EMC
cirpel	574	100	0,90358	1702	1747,6
abeja	516	158	0,876296	1785,16	1834,8
mosca	483	191	0,816914	2083,78	2139,99
escarabajo	433	241	0,669259	2893,49	3028,17
araña	392	282	0,539259	3359,12	3489,79
escorpion	379	295	0,574568	4994,21	5197,49
anillo	291	383	0,893951	3531,62	3570,5
serpiente	229	445	0,751111	6920,96	7267,49
marco	85	589	0,211852	12903,1	13160,62
anillo2	24	650	0,853827	5483,75	5541,21
circur	622	52	0,873086	334,77	358,72

Tabla 5 1 Los *pixels* comunes, los *pixels* a mover, la compacidad discreta y los trabajos para transformar los 11 objetos en el círculo.

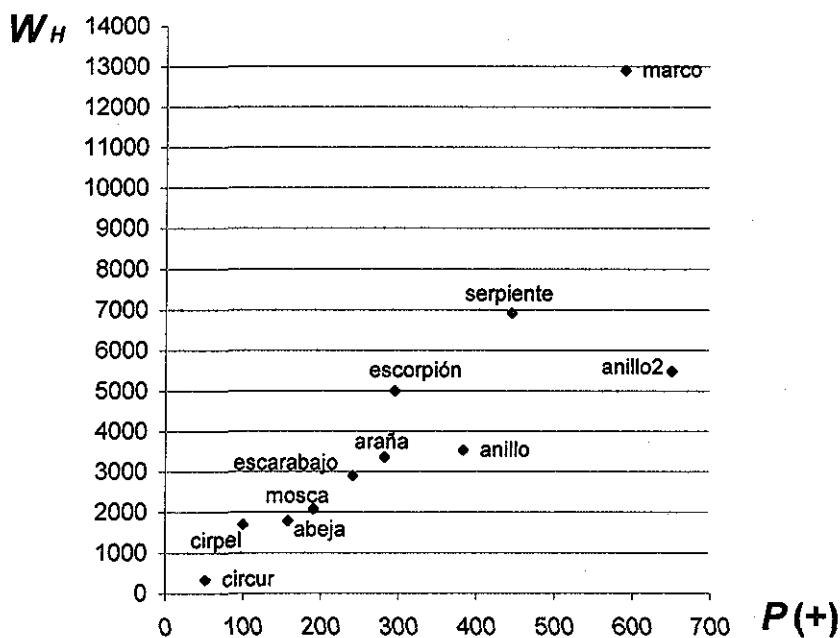


Figura 5.2. Trabajo realizado para transformar las figuras en términos del número de *pixels* positivos.

igual que al usar el algoritmo *EMC*.

Obsérvese que el marco tiene la menor compacidad: 0.212. A pesar de no tener el máximo número de *pixels* a mover, el trabajo se dispara a 12,903.10. Pero, para el caso del anillo2, cuya compacidad es mucho mayor que la del marco (0.854), difícilmente el trabajo para transformarlo en el círculo se incrementa, llegando muy por abajo de lo sucedido con el marco: 5,541.25. Por poner otro ejemplo, obsérvese también lo sucedido entre el escorpión y el anillo. El primero tiene una compacidad 0.32 veces menos que el segundo. A pesar de que el escorpión tiene menor número de *pixels* a mover que el anillo, el trabajo en transformarlo al círculo es mayor.

En conclusión, de esta muestra comparativa, los objetos menos compactos se alejan en parecido más rápidamente del círculo que los más compactos, a pesar de que algunas veces éstos últimos tengan más *pixels* a mover. Se observa también que cuando se desplazan algunos *pixels* como el caso del círculo con un cabello, su parecido no cambia demasiado. También, a pesar del ruido, el parecido con el círculo con ruido es muy grande, mostrándose que la medida es robusta respecto al ruido.

5.3 ¿Cómo normalizar la medida cuando el conjunto de objetos está completo?

Un método de normalización se logra cuando se toma en cuenta los siguientes aspectos. Como ya se mencionó al analizar la gráfica de la Figura 5.2, los objetos menos compactos tienden a ser *más rápido* diferentes al círculo (claro que, la compacidad no es determinante para calcular el parecido, recuérdese que el número de *pixels* positivos juega también un papel determinante). Entonces, ¿cuál es el objeto más compacto y cual el menos compacto de todo el universo de formas? Si se considera el conjunto completo de todas las formas, compuestas de n *pixels*, no es necesario calcular el

trabajo realizado al comparar cada pareja de objetos, hasta encontrar los dos más diferentes. De los resultados de la sección anterior, se deduce que una forma de saber cuáles son los objetos menos parecidos de todo el universo de formas compuestas de n *pixels*, es comparar el objeto más compacto con el menos compacto de entre ellos. Estos corresponden a un cuadrado de $n^{1/2}$ de lado, que tiene la compacidad máxima de 1, y a una varilla delgada cuyo largo es, precisamente, n , la cual tiene una compacidad de 0. Claramente, el conjunto de objetos con todas las formas existentes a base de n *pixels* es muy grande, en comparación con el conjunto de 12 objetos que se presenta en este trabajo. Para darse idea de que la serpiente o el marco no son tan diferentes al círculo cuando se toma en cuenta un conjunto completo con todas las formas, hay que considerar que hay muchos más objetos que tienen unas formas más caprichosas que la serpiente o el marco. Puede intuirse que visualmente deben de existir miles y miles de ellos mucho menos parecidos que los objetos expuestos aquí, de manera que si se consideran a todos los demás, entonces, hay que normalizar respecto al trabajo que se hace para transformar un cuadrado en una varilla larga, ambos de 676 *pixels*.

Una forma de saber cuál es el trabajo realizado entre un cuadrado y una varilla de n *pixels*, se logra al calcularlo primero para n 's pequeñas, por ejemplo a unas cinco diferentes resoluciones, y posteriormente se ajusta una curva polinomial a los datos, con la cual se obtiene el valor esperado para la n tan grande como se quiera (es decir, con la resolución que se desee). La Tabla 5.2 muestra el trabajo realizado entre el cuadrado y la varilla para cinco diferentes valores de n .

Una curva aproximada que ajusta a estos puntos, está dada por un polinomio cuadrático. Así, para $x = 674$ *pixels*, aproximadamente se tiene que el trabajo realizado debe ser de $W_H = 454,276$. Esto significa que el círculo y la serpiente se parecen

Total de <i>pixels</i>	$P(+)$	EMC	W_H
9	6	15,54	15,28
25	20	131,14	129,6
49	42	527,06	522,82
81	72	1479,6	1471,02
121	110	3257,46	3346,14

Tabla 5.2 Trabajo realizado para transformar un cuadrado en una barra de 676 *pixels* cada uno

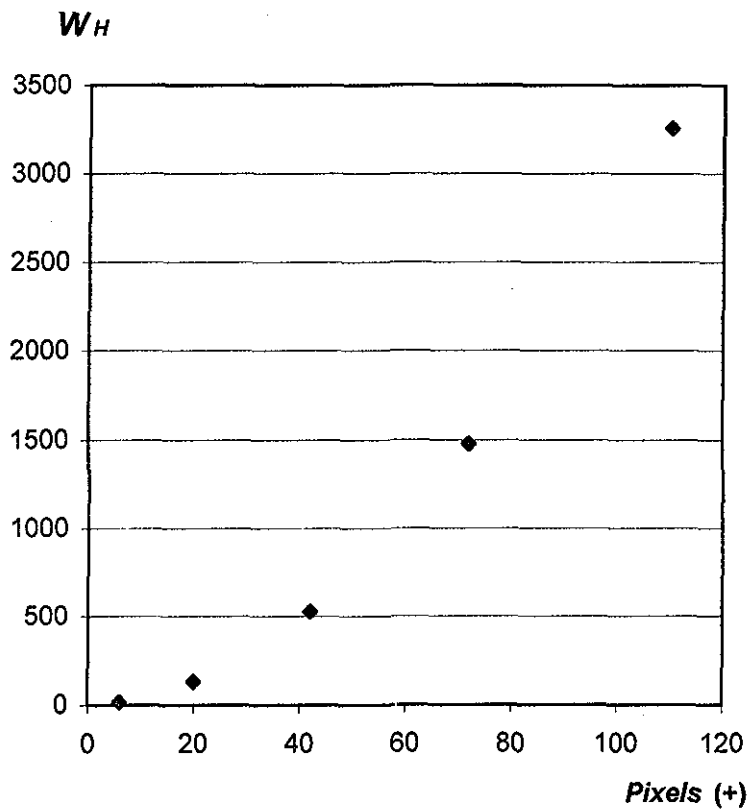


Figura 5.3 Trabajo realizado con el algoritmo húngaro en términos de los *pixels* a mover, para transformar un cuadrado en una barra a diferentes resoluciones.

en un 98.47%, mientras que el círculo y el marco se parecen en un 97.16%. Claramente estos valores no corresponden a los observados, debido al resto de objetos compuestos por 674 *pixels* que no aparecen en el conjunto presentado de 12 objetos. Por ello, es apropiado calcular la normalización por este método si se tuviera un conjunto muy grande de objetos, con un amplio rango de formas. De otra manera, con una muestra como la dada en este capítulo para 12 objetos, la normalización de la medida de similitud se propone hacer como se explica en la siguiente sección.

5.4 Normalizando la medida de similitud sin el conjunto de objetos completo

Dado un universo de N objetos normalizados en escala, supóngase que todos ellos están formados con una resolución de n *pixels*. Una forma de normalizar la medida de similitud, es buscar, de entre los N objetos, cuáles son los dos menos parecidos. Con ello se tendría el trabajo máximo en la transformación de todos los objetos con dicha resolución y, además, se podría tener una medida normalizada, que iría de 0 a 100%, por ejemplo. A continuación, ésta será precisamente la manera en que se normalizará la medida de similitud.

De las observaciones hechas en la sección anterior, se puede saber cómo encontrar los dos objetos menos parecidos. Considerando el conjunto de 12 objetos se observa que los más diferentes entre sí son el marco y el círculo, esto es, el más compacto y el menos compacto de todos los que hay en dicho conjunto. Así que, asígnese al trabajo W_H realizado para transformar el marco en el círculo un parecido igual 0% y, al trabajo realizado para transformar al círculo en él mismo (el cual es $W_H = 0$) un parecido del 100%. De esta manera, en términos del trabajo y con las asignaciones dadas, la medida de similitud puede escribirse como:

Transformación	<i>Pixels</i> comunes	<i>Pixels</i> a mover	Parecido con el círculo (%)
cirpel	574	100	86.81
abeja	516	158	86.16
mosca	483	191	83.85
escarabajo	433	241	77.58
araña	392	282	73.97
escorpión	379	295	61.29
anillo	291	383	72.63
serpiente	229	445	46.36
marco	85	589	0
anillo2	24	650	57.5
circur	622	52	97.4

Tabla 5.3. Número de *pixels* comunes al círculo, *pixels* positivos a mover en cada transformación y la medida normalizada de 0% a 100%.

$$S_{1,2} = \left(1 - \frac{W_{1,2}}{W_{m\acute{a}x}}\right) * 100\%, \quad (5.4)$$

donde los subíndices indican los objetos 1 y 2 a comparar, y $W_{m\acute{a}x}$ se refiere al máximo trabajo realizado al comparar N objetos entre sí.

Así, se puede dar una tabla con el trabajo normalizado para transformar todos los objetos en el círculo (Tabla 5.3).

Es claro que hasta aquí sólo se han dado algunos ejemplos de un subconjunto del conjunto total de formas que se pueden obtener con 674 *pixels*, pues el conjunto completo es demasiado grande (arriba de 2^{674} formas diferentes).

5.5 La orientación, crucial para el desempeño del trabajo

De la Tabla 5.3 se puede observar cuales son los objetos más parecidos al círculo, pero

además, siguiendo una de las propiedades básicas en reconocimiento de patrones, se establece que dos puntos muy cercanos en el espacio de atributos (en este caso este espacio está compuesto de dos atributos, a saber: el trabajo realizado para transformar objetos en una circunferencia y el número de *pixels* positivos), corresponden a objetos muy similares. Así que, por ejemplo, se puede decir que la araña se parece más al escarabajo que a la mosca, por estar los dos primeros más cercanos en el espacio de atributos, el círculo con cabello se parece más al escarabajo que al escorpión, etcétera. Para dar una medida precisa en el parecido de cada par de objetos hay que hacer lo mismo que se hizo para todas las figuras con el círculo, esto es, transformar por parejas.

La Tabla 5.4 muestra los valores de los trabajos sin normalizar, realizados por parejas, mientras que la Tabla 5.5 los valores ya normalizados. Se observan algunos aspectos interesantes que a continuación se analizarán.

Se usará la notación W_B para indicar el trabajo realizado al transformar los objetos cuando sus ejes no están alineados, de manera que en los casos presentados exista un menor número de *pixels* a mover y cuando se use el algoritmo *EMC* para mover los *pixels*.

Por ejemplo, de la Tabla 5.5 se observa que la araña se parece más al escarabajo que al círculo con cabello.

Experimentalmente, (véanse los dos últimos renglones de la Tabla 5.6) se encuentra que al transformar la araña en la mosca sin alinearlos conforme a sus ejes principales, haciendo un ángulo de 28.6° entre sus ejes mayores y con 355 *pixels* no comunes, se mide que el trabajo realizado es de $W_B = 3,642.76$, mientras que, ya alineados conforme a sus ejes principales y con 342 *pixels* no comunes, el trabajo empleado en la transformación es de $W_H = 3,219.17$, disminuyendo el trabajo en una

Objetos	círculo	circur	cirpel	abeja	mosca	escarabajo	araña	anillo	escorpión	anillo2	serpiente
círculo	0										
circur	334,77	0									
cirpel	1702	1818,89	0								
abeja	1785,16	1689,01	1607,14	0							
mosca	2083,78	2042,55	1124,29	723,65	0						
escarabajo	2893,49	2770,64	1351,85	1747,92	1762,42	0					
araña	3359,12	2195	3348,87	3212,7	3219,17	2570,85	0				
anillo	3531,62	3456,6	4061,87	3487,34	3535,87	3739,13	3763,97	0			
escorpión	4994,21	4913,96	4932,39	4627,46	4062,13	3421,89	3822,41	5181,84	0		
anillo2	5483,75	5410,92	5649,88	5013,2	4885,95	4818,14	4633,84	1962,94	5897,01	0	
serpiente	6920,96	6823,44	6292,82	7401,57	7162,41	6755,82	5212,06	6047,34	4750,16	5816,78	0
marco	12903,1	12821,66	12538,4	12537,46	12226,81	11241,52	10004,18	9947,5	10452,46	8605,07	8672,84

a) Trabajo realizado en la transformación de los objetos acorde al algoritmo húngaro

Objetos	círculo	circur	cirpel	abeja	mosca	escarabajo	araña	anillo	escorpión	anillo2	serpiente
círculo	0										
circur	359	0									
cirpel	1748	1868,06	0								
abeja	1835	1728,99	1701,87	0							
mosca	2140	2179,59	1279,68	849,87	0						
escarabajo	3028	2880,01	1561,45	1931,31	2025,85	0					
araña	3490	2561,65	3695,63	3363,18	3473,26	2876,25	0				
anillo	3571	3510,23	4338,11	3487,34	3760,11	4159,79	4037,7	0			
escorpión	5197	5134,98	5264,62	5041,07	5073,04	3964,33	4230,81	5831,97	0		
anillo2	5541	5497,55	5922,29	5096,07	4984,61	5003,16	4794,84	2082,9	6505,98	0	
serpiente	7267,49	7298,72	6713,88	7706,99	7439,4	7194,42	5787,63	6568,13	5301,17	6409,41	0
marco	13160,62	13114,98	12871,7	13018,06	12767,53	11978,89	10636,23	10484,6	11194,8	9150,5	9301,76

b) Trabajo realizado en la transformación de los objetos acorde al algoritmo por elementos más cercanos.

Objetos	círculo	circur	cirpel	abeja	mosca	escarabajo	araña	anillo	escorpión	anillo2	serpiente
círculo	674										
circur	622	674									
cirpel	574	553	674								
abeja	516	511	514	674							
mosca	483	476	479	559	674						
escarabajo	433	240	265	468	444	674					
araña	392	391	367	369	332	351	674				
anillo	291	291	235	268	239	213	204	674			
escorpión	379	371	357	353	318	352	276	204	674		
anillo2	24	57	72	149	162	131	132	378	116	674	
serpiente	229	233	228	262	277	222	215	86	179	110	674
marco	85	78	81	65	52	69	111	52	58	46	80

c) *Pixels* comunes

Tabla 5 4 Se hace una comparación entre a) el algoritmo optimizado y b) el algoritmo *EMC*; c) *pixels* comunes entre cada par de objetos 2D a comparar

Objetos	círculo	circur	cirpel	abeja	mosca	escarabajo	araña	anillo	escorpión	anillo2	serpiente
círculo	100										
circur	97,41	100									
cirpel	86,81	85,90	100								
abeja	86,16	86,91	87,54	100							
mosca	83,85	84,17	91,29	94,39	100						
escarabajo	77,58	78,53	89,52	86,45	86,34	100					
araña	73,97	82,99	74,05	75,10	75,05	80,08	100				
anillo	72,63	73,21	68,52	72,97	72,60	71,02	70,83	100			
escorpión	61,29	61,92	61,77	64,14	68,52	73,48	70,38	59,84	100		
anillo2	57,50	58,06	56,21	61,15	62,13	62,66	64,09	84,79	54,30	100	
serpiente	46,36	47,12	51,23	42,64	44,49	47,64	59,61	53,13	63,19	54,92	100
marco	0	0,63	2,83	2,83	5,24	12,88	22,47	22,91	18,99	33,31	32,78

a) Grado de parecido entre los diferentes objetos acorde con el algoritmo húngaro. La medida esta normalizada de 0 a 100%

Objetos	círculo	circur	cirpel	abeja	mosca	escarabajo	araña	anillo	escorpión	anillo2	serpiente
círculo	100										
circur	97,27	100									
cirpel	86,72	85,81	100								
abeja	86,06	86,86	87,07	100							
mosca	83,74	83,44	90,28	93,54	100						
escarabajo	76,99	78,12	88,14	85,33	84,61	100					
araña	73,48	80,54	71,92	74,45	73,61	78,15	100				
anillo	72,87	73,33	67,04	73,50	71,43	68,39	68,70752	100			
escorpión	60,51	60,98	60,00	60,9313	61,45	69,88	67,2109	54,80179	100		
anillo2	57,90	58,23	54,1018	60,5051	61,3689	61,98	62,83963	83,85737	49,578	100	
serpiente	44,78	43,43	47,96692	40,2702	43,47	44,242701	55,14543	49,0965	58,916	50,3266	100
marco	0,00	0,35	2,20	1,09	2,98908	7,1626973	17,56841	18,74371	13,239	29,0829	27,9106571

b) Grado de parecido acorde con el algoritmo EMC

Tabla 5.5. Medida de similitud normalizada

Objeto a Transformar	Objeto Transformado	Ángulo entre ejes mayores	P(+)	W_H	W_B
araña	serpiente	169.15°	445	5,685.12	6,499.19
araña	serpiente	90°	459	5,598.48	6,323.44
araña	serpiente	1°	496	5,212.06	5,787.63
araña	mosca	28.6°	355	3,270.25	3,642.76
araña	mosca	0.47°	342	3,219.17	3,473.26

Tabla 5.6. Número de *pixels* comunes a ambos objetos y *pixels* positivos a mover en cada transformación.

cantidad dada por la ecuación 5.5.

$$\frac{W_B - W_H}{W_B} * 100\% = 11.6\% \quad (5.5)$$

Al hacer algo similar entre la araña y la serpiente, obsérvese tres casos de orientación diferentes mostrados en la Tabla 5.6. En el primer renglón se presenta el menor número de *pixels* a mover de los tres casos. Siguiendo con el algoritmo *EMC*, se considera W_B (método de Bribiesca [1](Bribiesca, 1996) en el que se busca mayor cantidad de *pixels* en común) y se encuentra que, al superponerlos sin alinearlos conforme a sus ejes principales (ver Figura 5.4a), el trabajo es de $W_B = 6,499.19$, con una cantidad de 445 *pixels* a mover; mientras que alineados conforme a sus ejes principales (Figura 5.4c) el trabajo realizado es de $W_H = 5,212.06$. Así que hay un error del 19.81% al comparar el método de Bribiesca con el optimizado. Un caso intermedio se observa en el segundo renglón. Nótese cómo a pesar de tener más *pixels* a mover entre dos objetos, el trabajo fue menor con el método optimizado.

5.6 Contradicciones entre el método optimizado y el método de Bribiesca[1](1996)

De la Tabla 5.5 se observa que, mientras el algoritmo de Bribiesca muestra que la araña se parece más al círculo que a cirpel, el optimizado muestra que la araña se parece más al cirpel que al círculo. Pero un argumento como el siguiente podría favorecer al hecho de que la araña se parece “realmente” más al cirpel que al círculo.

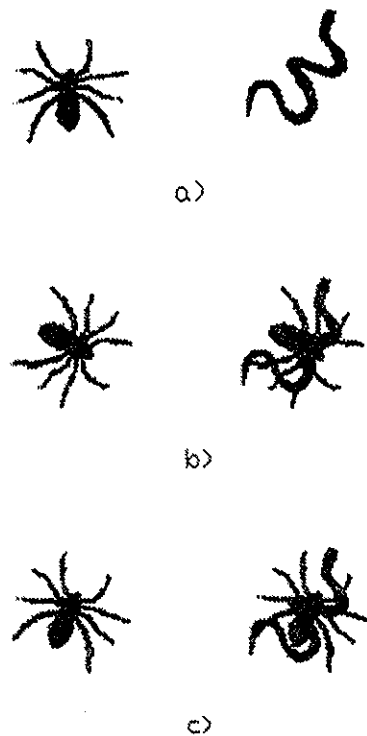


Figura 5.4. Por comparar la araña y la serpiente; a) ambos objetos tienen sus ejes mayores principales en un ángulo de 169.15° ; b) los ejes principales mayores están orientados en 90° uno respecto del otro. El número de *pixels* comunes es de 229; c) los objetos están orientados conforme a sus ejes principales con 215 *pixels* comunes.

Por un lado, la araña tiene más *pixels* en común (392) con el círculo que con el cirpel (367).

Pero ya se estudió que la máxima correlación no implica mayor parecido entre los objetos. Así que este argumento no es adecuado para decir que las araña debiera parecerse más al círculo que al cirpel.

Sin embargo, la diferencia en las compacidades entre la araña y el círculo es mayor que entre ésta y el cirpel. Además de lo que visualmente podría parecer una “pata” que sale del círculo para conformar el cirpel, uno podría decir con toda confianza que el parecido de la araña es mayor al cirpel que al círculo. Por dar otro ejemplo, la Tabla 5.5a) muestra que la serpiente se parece más al círculo con ruido, mientras que en la Tabla 5.5b) se ve que ésta se parece más al círculo.

5.7 Desventajas del método propuesto

Para clasificar N objetos, una desventaja de compararlos por parejas es que se requiere una tabla de $N(N-1)/2$ comparaciones. A diferencia de lo que podría hacerse con un modelo de clasificación como el de Lohmann [10](1998), quien establece una medida de similitud al comparar los objetos con una esfera o con una caja. Algo parecido se hizo en este capítulo cuando se compararon todos los objetos con el círculo. Otra desventaja es que el tiempo de procesamiento es muy grande, pues la complejidad del algoritmo húngaro para encontrar la distancia mínima es de orden cúbico. Así que la desventaja principal de este método es que es muy lento. Sin embargo, es muy consistente para especificar el grado de parecido entre dos objetos cualesquiera.

Capítulo 6

Transformación de objetos 3D

En este capítulo se presentan las transformaciones realizadas a cinco objetos 3D a partir de datos del mundo real. Los objetos se normalizaron en la escala, se hicieron coincidir sus centros de masa y se alinearon conforme a sus ejes principales, ello permitió obtener la mejor orientación para minimizar el trabajo de transformación. Una vez hecho lo anterior, se procederá a transformar los objetos entre sí.

Aplicando el algoritmo húngaro en la versión de Evans [64](1992) explicada en el Capítulo 4, la cual a su vez está basada en la forma original de la versión del algoritmo húngaro dada por Egerváry [69](1931), se encontró el apareo óptimo de la gráfica bipartita que surge de querer mover el conjunto de *spels* positivos al conjunto de *spels* negativos.

6.1 Los objetos 3D a comparar

En esta sección se presentan las superficies de tres volcanes cercanos a la cuenca de México: La Malinche, el Popocatepetl y el Iztaccíhuatl; así como de dos automóviles: *Camaro* y *Porsche*. Los tres volcanes fueron obtenidos a partir de un Modelo Digital de Elevación (MDE).

Los archivos MDE son representaciones digitales de la superficie de la Tierra, los cuales se generan como una malla uniforme rectangular organizada por capas. La digitalización de estos modelos está basada en contornos con escala de 1:250,000.

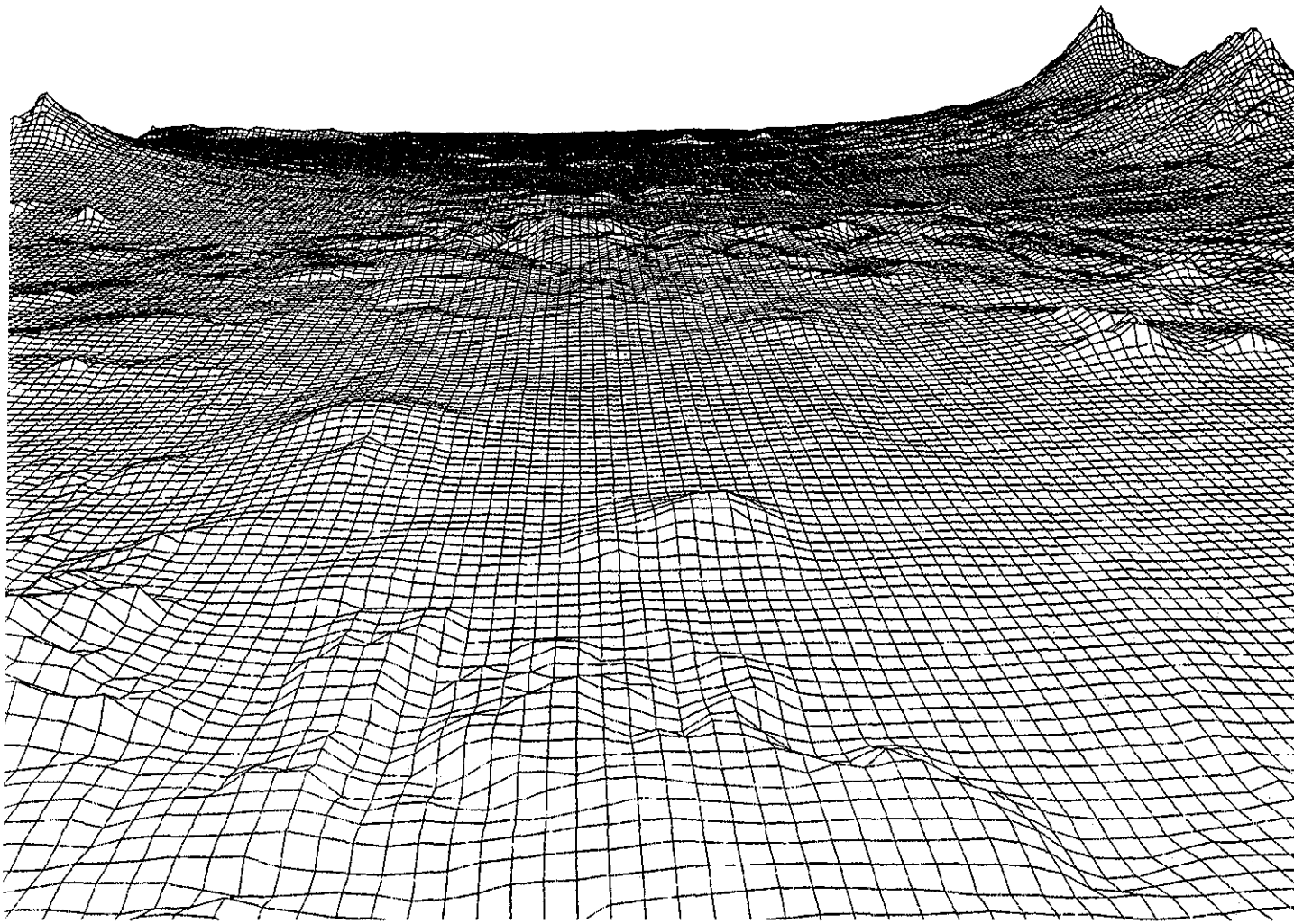


Figura 6.1. Modelo digital de elevación de la cuenca de México. Se muestran tres volcanes importantes: La Malinche, el Popocatépetl y el Iztaccíhuatl.

La Figura 6.1 muestra un modelo digital de elevación cercano a la cuenca de México. Este modelo está representado por una malla poligonal 3D y despliega tres volcanes. En la parte superior izquierda se encuentra el volcán La Malinche, en la parte superior derecha se encuentran los otros dos volcanes: el Popocatépetl y, ligeramente hacia abajo, el Iztaccíhuatl.

Por otro lado, se presentan las superficies de mallas poligonales 3D de dos automóviles: un Porsche y un Camaro, los cuales se muestran en la Figura 6.2. El

formato original de estos objetos está dado por archivos DXF.

La información de los archivos MDE y DXF se utilizaron para representarlos como sólidos compuestos de *voxels*.

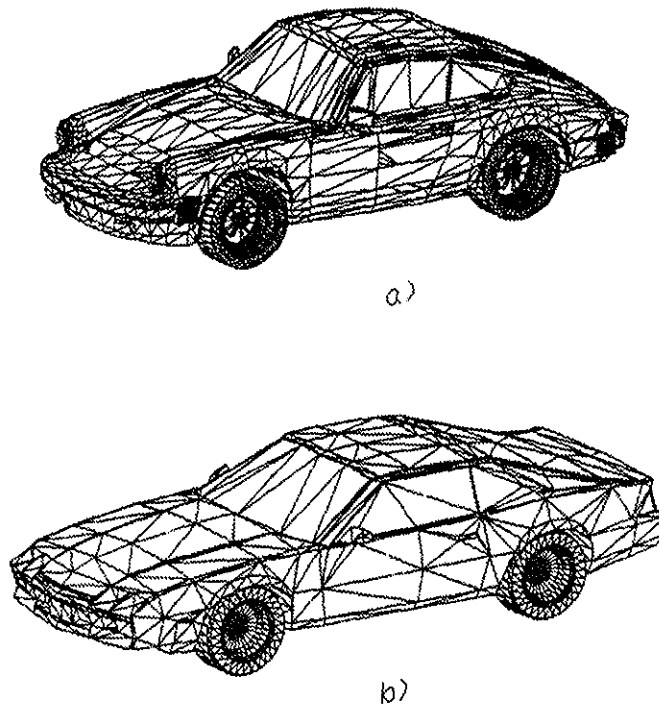


Figura 6.2. Dos automóviles cuyas superficies están dadas en mallas poligonales triangulares; a) un Porsche, b) un Camaro.

6.2 Los objetos 3D se normalizan en volumen

Como ya se mencionó al principio del Capítulo 3, el primer paso para comparar los objetos es normalizar sus áreas o volúmenes. Ello dependerá del número de puntos obtenidos de la imagen real y de su tamaño. El programa de aplicación AUTOCAD,

permite escalar casi cualquier forma cerrada en 3D.

Pueden usarse representaciones de objetos con mallas poligonales 3D como las mostradas en las Figuras 6.1 y 6.2, o bien, las superficies de figuras *voxelizadas* originalmente, como los volcanes de la Figura 6.3 los cuales originalmente son diferentes en escala y, además, no están alineados conforme a sus ejes principales.

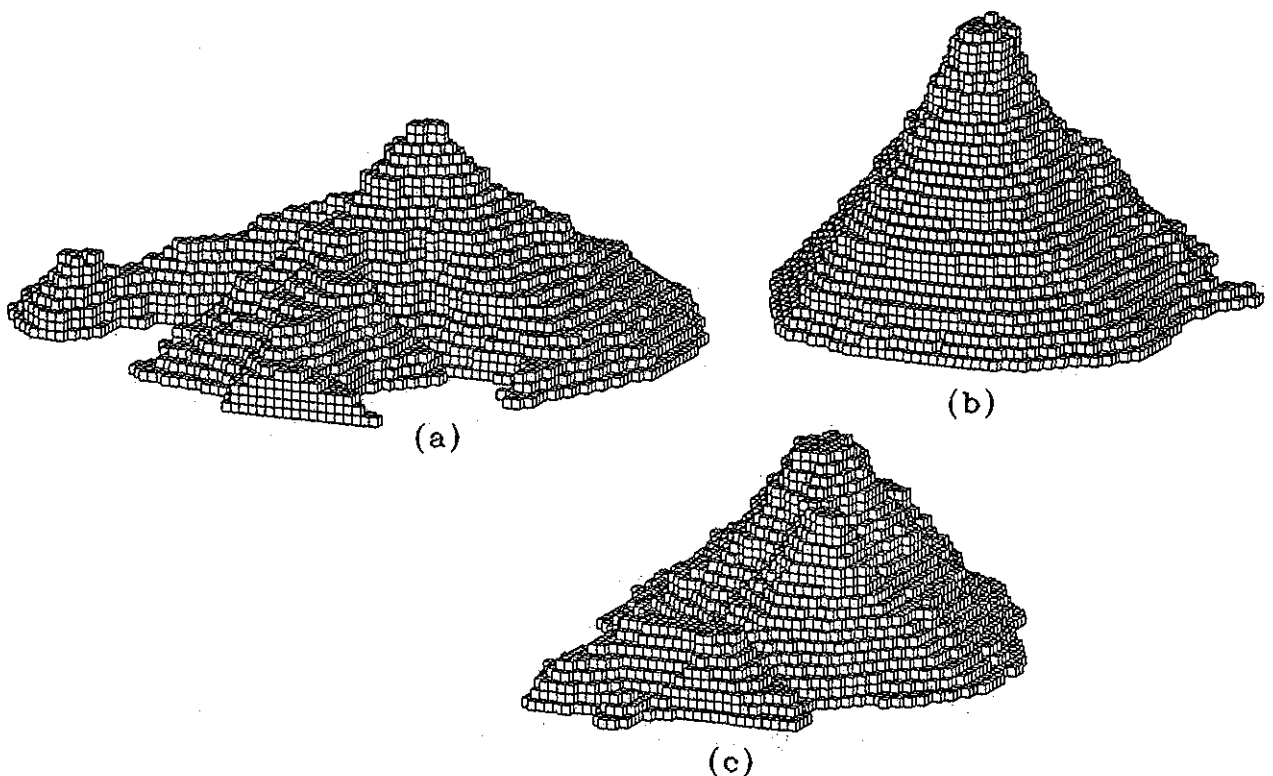


Figura 6.3. Los tres volcanes cercanos a la cuenca de México, representados por *voxels*. a) El volcán La Malinche, compuesto por 19,154 *voxels*; b) el volcán Popocatepetl, compuesto por 20,769 *voxels* y c) el volcán Iztaccíhuatl, compuesto por 19,776 *voxels*

Enseguida se crea una matriz tridimensional de puntos. La Figura 6.4 muestra la superficie del objeto 3D, que en este caso corresponde al volcán Popocatepetl (el

cual se reescalará) y de un arreglo tridimensional de puntos, de $50 \times 50 \times 50$. La superficie se “sumerge” totalmente en el arreglo tridimensional (véase la Figura 6.5), quedando una cantidad de puntos dentro de la superficie y otra cantidad fuera de ella. Se procede a ocultar los puntos interiores a la superficie del objeto, conformada en este ejemplo por las caras de los *voxels* que lo representan. En la Figura 6.6 se observa que el primer ocultamiento de puntos se hizo en la dirección positiva del eje *x* para, posteriormente, borrar los puntos externos (y parte de las caras de los *voxels* de la superficie) a la superficie de esa vista en particular (la que se encuentra en el plano *z-y*). Un segundo ocultamiento sobre la vista *y-z*, en la dirección negativa del eje *x*, y enseguida una eliminación de puntos, permite obtener todos los puntos interiores del objeto. Los planos que por alguna razón no terminaron de borrarse, debido al apantallamiento que pueda existir entre ellos, pueden borrarse manualmente con la ayuda del uso de *acetatos* que AUTOCAD maneja. Es recomendable tener al objeto y al arreglo de puntos en acetatos diferentes.

Recuérdese que al borrar los puntos que no están en el interior de la superficie, dada cualquier vista, también se pueden borrar muchos elementos o planos que conforman al objeto. Para borrar puntos paulatinamente, con la ayuda de AUTOCAD se puede rotar al objeto, auxiliándose de copias de la superficie, superponiéndolas en el mismo lugar, de manera de borrar los puntos que no pertenecen al interior de la superficie.

Si se pretende normalizar el volumen para el caso tridimensional, AUTOCAD permite cambiar el tamaño del objeto y repetir el procedimiento, hasta que se obtenga el número deseado de puntos para, posteriormente, convertirlos a *voxels*.

En el caso de los volcanes que se presentan en este trabajo, su superficie está compuesta por caras de cubos, pues se aplicó un programa que permite representar cualquier objeto binario en términos de *voxels*. Estas mismas caras pueden servir

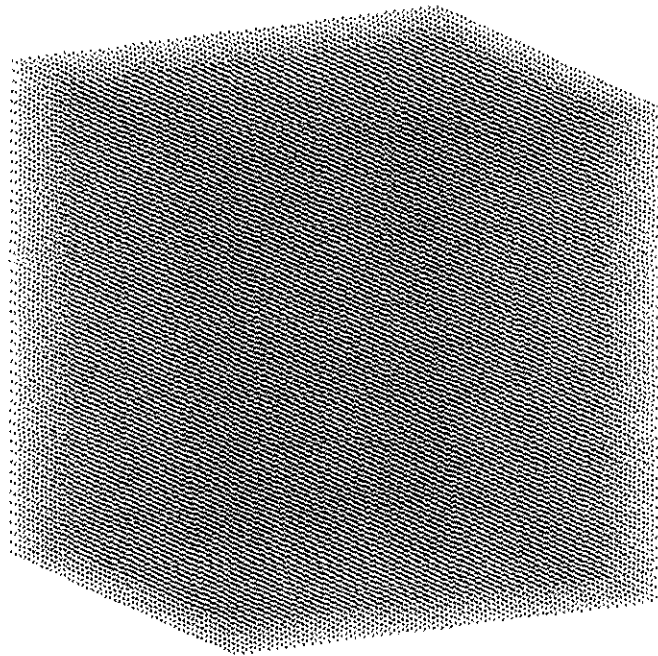
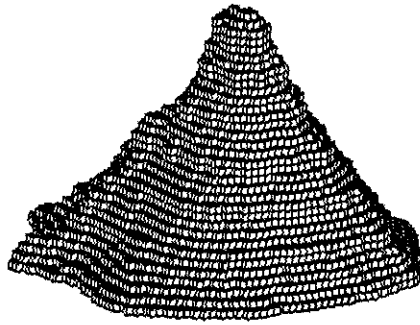


Figura 6.4 La superficie del a) volcán Popocatepetl se sumergirá en un arreglo 3D; b) el arreglo de 50 x 50 x 50 puntos

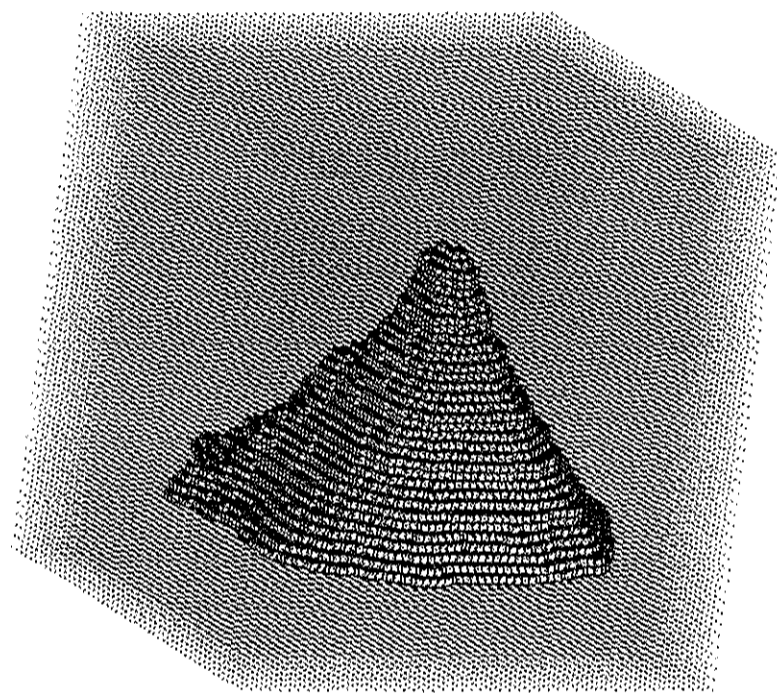


Figura 6.5. La superficie del objeto a escalas sumergida en el arreglo 3D de puntos.

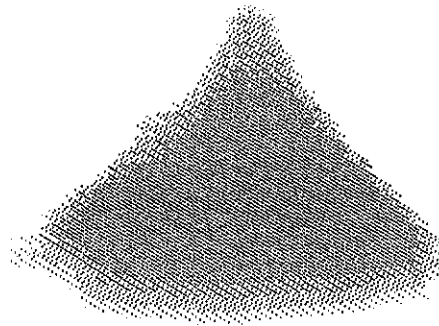
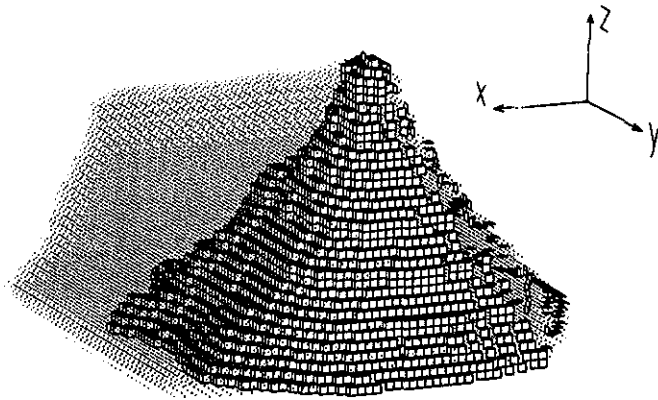


Figura 6.6. Se eliminan los puntos y parte de los planos de la superficie al borrar sobre el plano y-z. en la dirección positiva del eje x; b) conjunto de puntos que toma la forma del objeto original

como pantalla para ocultar los puntos interiores y seguir el procedimiento descrito para escalar.

AUTOCAD permite cambiar la escala de las figuras 3D. Permitted acercarse tanto como se quiera al número de puntos que debe tener una figura.

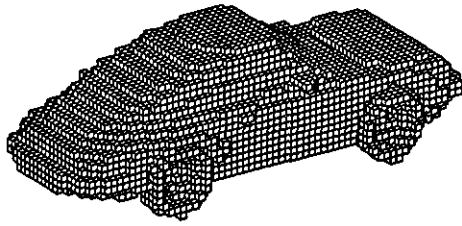
Una vez que el conjunto de puntos ya tiene la forma del objeto deseado, se procede a exportar los datos a formato DXF. Posteriormente, con un programa que lee este formato, se procede a *voxelizar* el conjunto de puntos, dando como resultado al objeto *voxelizado*.

Este procedimiento es válido para otro tipo de superficies, tal como las mallas poligonales 3D, las cuales emplean generalmente triángulos y rectángulos para representar los planos que conforman la superficie de los objetos. Como ejemplo de aplicación del procedimiento descrito, se muestran dos autos que originalmente estaban representados por mallas poligonales en 3D: un *Porsche* y un *Camaro*, obteniendo dos objetos normalizados a 31,020 *voxels*, como en la Figura 6.7.

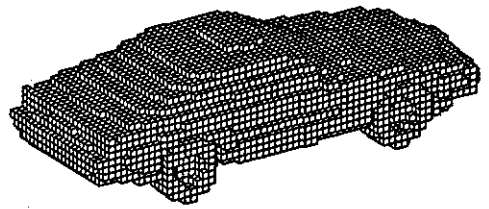
6.3 Transformando los volcanes

En esta sección, se utilizan los volcanes como objetos prueba para mostrar con detalle el procedimiento de llevar a cabo la transformación de un objeto a otro, y en la siguiente sección se dan a conocer los resultados de haber hecho las transformaciones entre los cinco objetos mencionados en este capítulo.

La Figura 6.8 muestra los volcanes ya orientados conforme a sus ejes principales. Se puede aplicar la técnica de la sección anterior para normalizar el volumen y, posteriormente, calcular los ejes principales para orientarlos (ver Capítulo 3). En este ejemplo, cada volcán está compuesto de 16,228 *voxels*. La Figura 6.8(a) muestra al



Porsche



Camaro

Figura 6.7 Dos autos *voxelizados* a partir de una malla poligonal 3D; ambos compuestos de 31,020 *voxels*

volcán La Malinche, la Figura 6.8(b) presenta al volcán Popocatépetl, y finalmente la Figura 6.8(c) muestra al volcán Iztaccíhuatl.

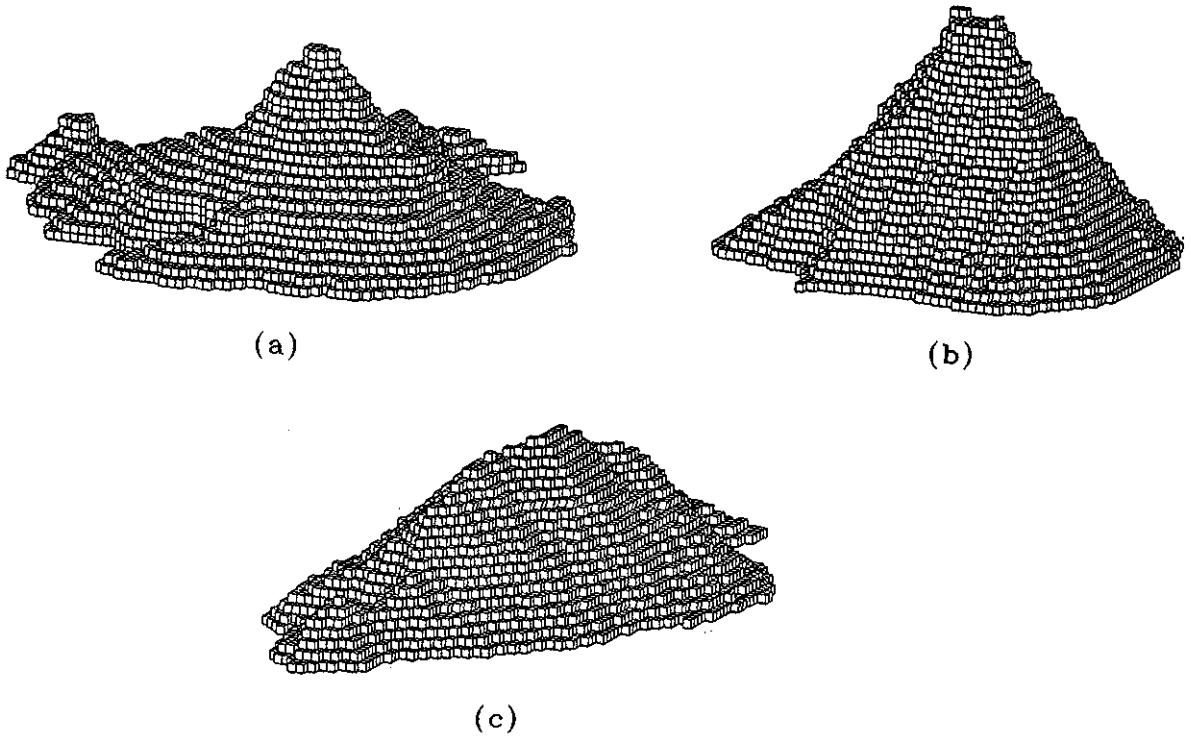


Figura 6.8. Los tres diferentes volcanes normalizados y orientados conforme a sus ejes principales. Están compuestos de 16,228 *voxels*. a) La Malinche, b) el Popocatépetl y c) el Iztaccíhuatl

La Figura 6.9 ilustra cómo transformar un volcán en otro. La Figura 6.9(a) muestra a la Malinche y la Figura 6.9(b) presenta al Popocatépetl. Ambos volcanes tienen el mismo número de *voxels*, es decir, tienen 16,228 cada uno. La Malinche será transformado en el Popocatépetl. Los ejemplos para ilustrar el método de transformación expuestos aquí, son objetos 3D binarios.

La Figura 6.9(c) ilustra la superposición de los objetos, y la Figura 6.9(d) a los

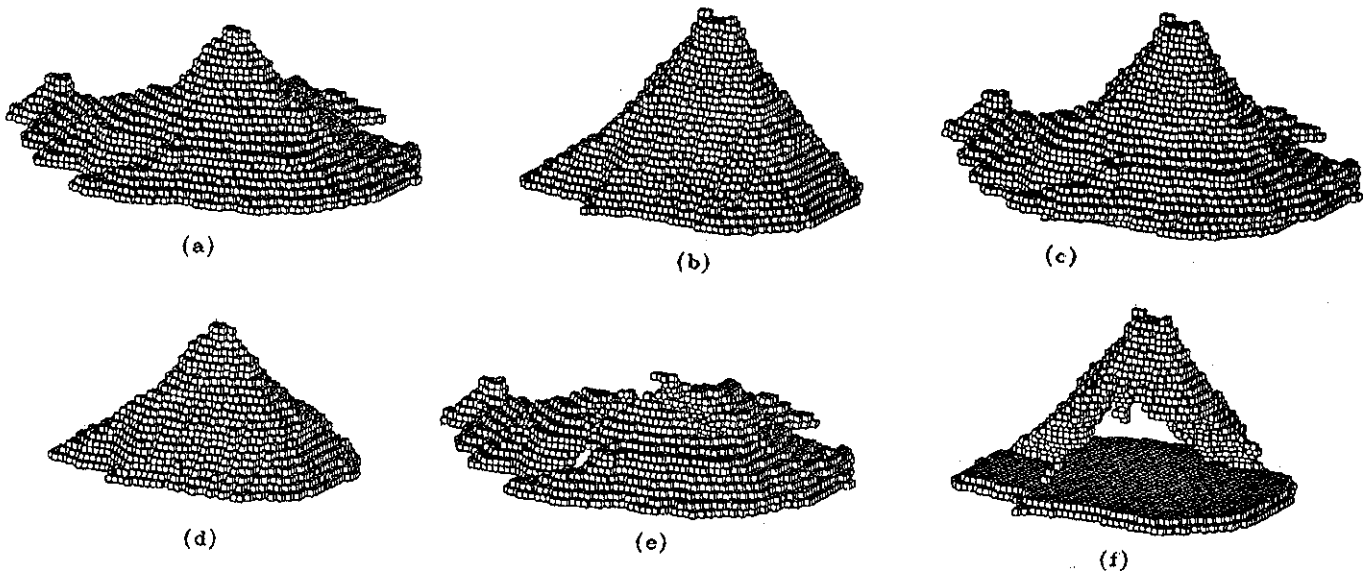


Figura 6.9. Transformación de los objetos: a) el objeto O_1 ; b) el objeto O_2 ; c) la superposición del objeto O_1 sobre el objeto O_2 ; d) los *voxels* comunes entre O_1 y O_2 ; e) *voxels* a mover (*voxels* positivos); f) lugares a donde se moverán los *voxels* positivos (*voxels* negativos).

voxels comunes. El número de *voxels* comunes es igual a 10,266, los cuales pueden verse en la Figura 6.9(d). La Figura 6.9(e) ilustra todos los *voxels* positivos, cuya cantidad es igual a 5,962. La Figura 6.9(f) muestra los *voxels* negativos, el número de éstos es igual a 5,962.

Por otro lado, la Figura 6.10 muestra los diferentes estados de la transformación del volcán Iztaccíhuatl (Figura 6.10(a)) en el volcán La Malinche (Figura 6.10(v)) en pasos de 25 *voxels*.

Los objetos mostrados en las Figuras 6.10(a) y 6.10(v) están normalizados en volumen (se representan usando la misma cantidad de volumen) y orientados conforme a sus ejes principales. Para simplificar las transformaciones progresivas, los cinco objetos fueron normalizados a 2,122 *voxels* cada uno.

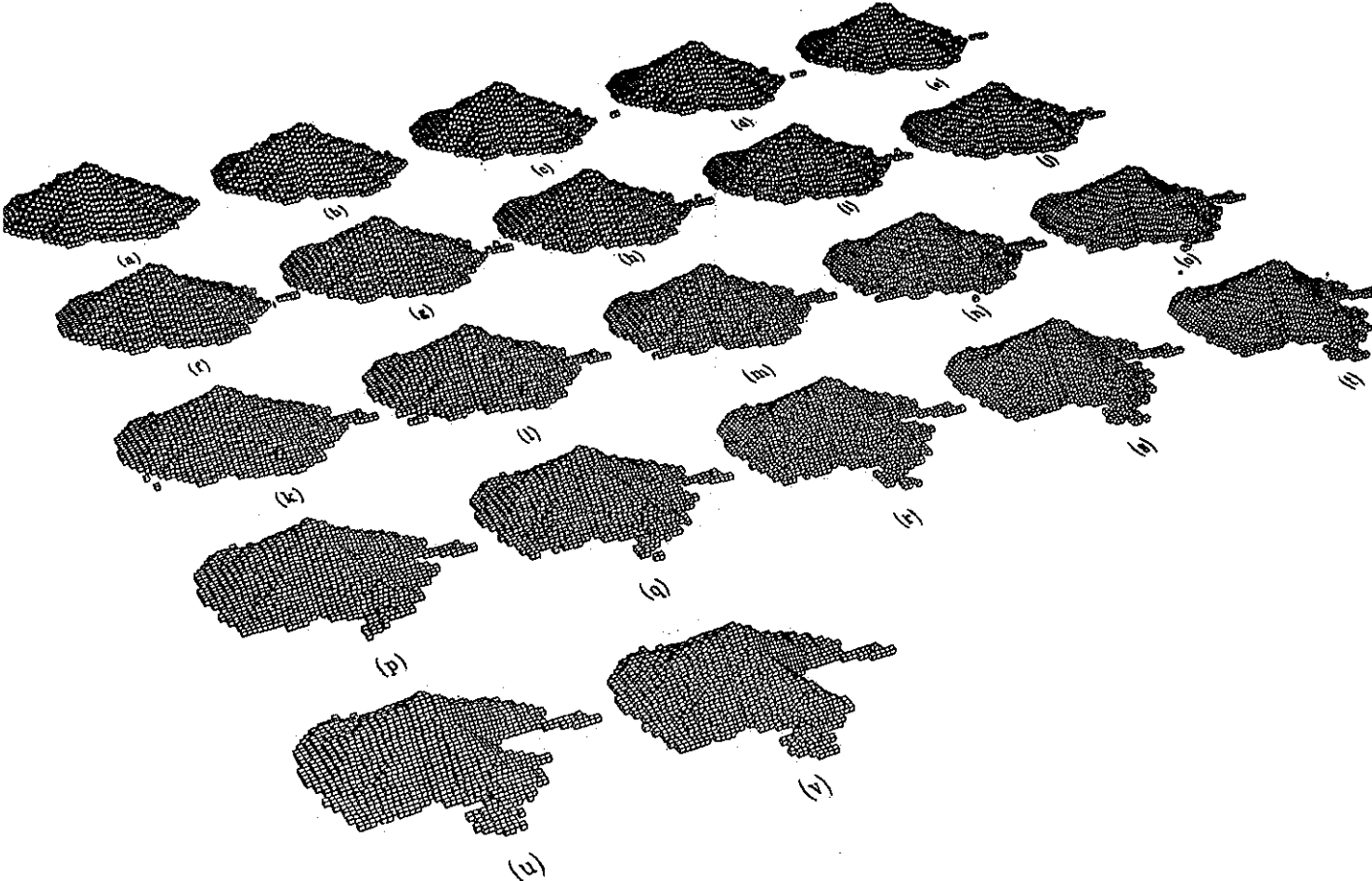


Figura 6.10. Los diferentes estados de la transformación del volcán Iztaccíhuatl en el volcán La Malinche en pasos de 25 *voxels*: a) el volcán Iztaccíhuatl; b) 25 *voxels* son desplazados; c) 50 *voxels* son desplazados; etcétera

Los *voxels* comunes a las figuras 6.10(a) y (v) son 1,609 y los *voxels* a mover son 513.

La Figura 6.10 presenta, entonces, la transformación del volcán Iztaccíhuatl en el volcán La Malinche. La Figura 6.10(b) presenta los primeros 25 *voxels* movidos. La Figura 6.10(c) muestra 50 *voxels* movidos, y así sucesivamente.

La Figura 6.11 presenta los diferentes estados de la transformación del volcán Popocatepetl (ver Figura 6.11(a)) en el volcán La Malinche (ver Figura 6.11(aa)).

La Figura 6.11 presenta los diferentes estados de la transformación del volcán Popocatepetl (ver Figura 6.11(a)) en el volcán La Malinche (ver Figura 6.11(aa)).

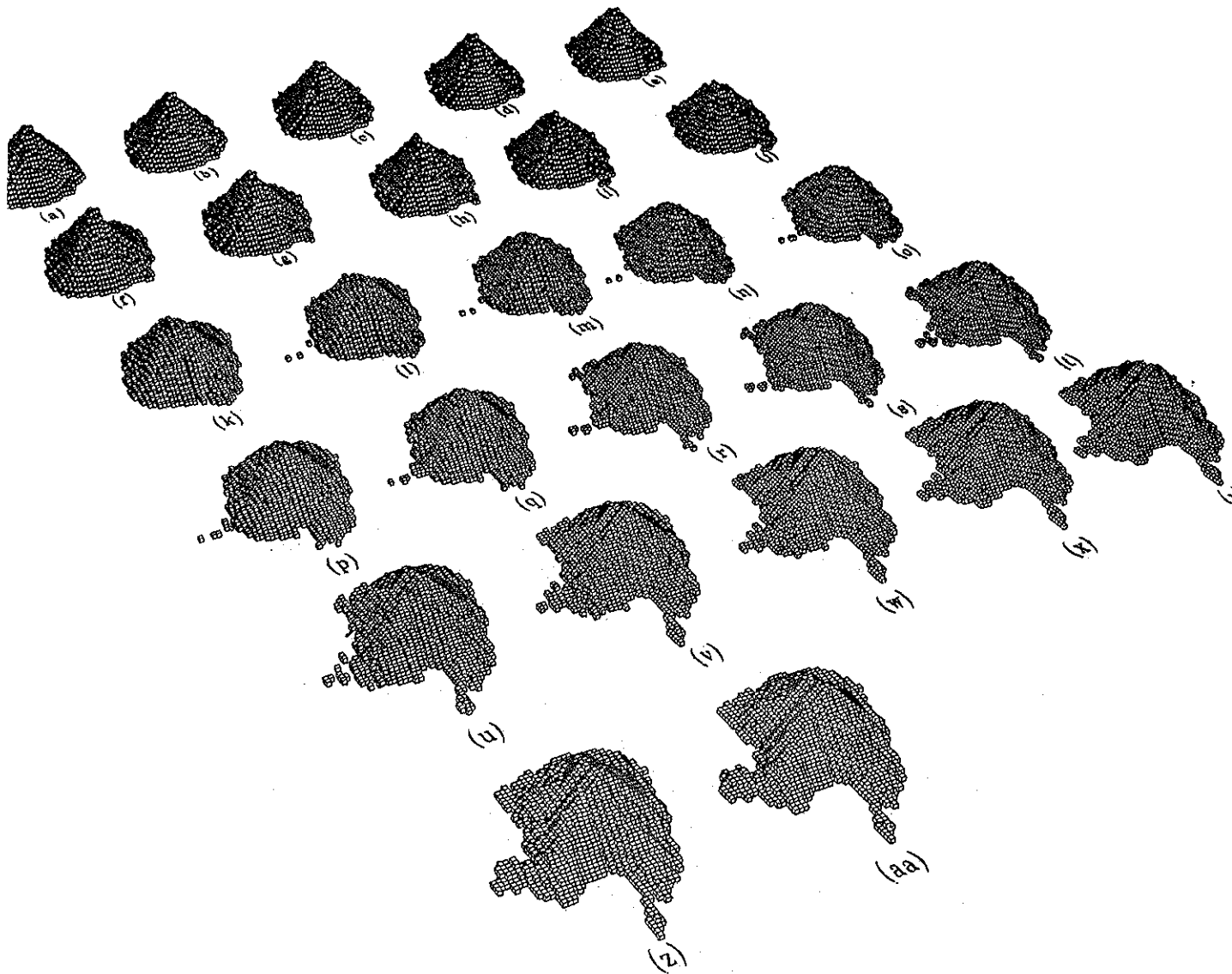


Figura 6.11. Los diferentes estados de transformación del volcán Popocatépetl en el volcán La Malinche en pasos de 25 *voxels*: a) el volcán Popocatépetl; b) 25 *voxels* desplazados; c) 50 *voxels* desplazados; así sucesivamente.

Se ha llevado a cabo una transformación óptima de los objetos. Cada uno de los volcanes está compuesto de 2,122 *voxels*, siendo ya normalizados en volumen y orientados conforme a sus ejes principales. La Figura 6.11 muestra las transformaciones progresivas en pasos de 25 *voxels*. Las figuras 6.11(a) y (aa) tienen 1,361 *voxels* en común y 761 *voxels* a mover. La Figura 6.11(b) muestra los primeros 25 *voxels* desplazados, la Figura 6.11(c) ilustra 50 *voxels* desplazados, etcétera. Nótese que las vistas de las figuras 6.11 y 6.10 se presentan en perspectiva.

Para mejorar la presentación de la transformación de los volcanes, el punto de vista de la Figura 6.11 se cambió en relación a la Figura 6.10.

6.4 Resultados de las transformaciones entre los cinco objetos

Usando el procedimiento de la sección anterior se procedió a transformar los cinco objetos por parejas. En la Tabla 6.1 aparecen los *voxels* comunes entre los cinco objetos. Nótese que el número máximo de *voxels* comunes entre diferentes parejas es de 1,765, correspondiente a los autos Camaro y Porsche. La Tabla 6.2 presenta los *voxels* positivos de los objetos.

La Tabla 6.3 muestra el trabajo W_H realizado para transformar los objetos mencionados usando el algoritmo húngaro. Mientras que la Tabla 6.4 muestra el trabajo W_{EMC} usando el algoritmo *EMC*. Como se observa, el algoritmo *EMC* siempre arroja valores mayores que el algoritmo húngaro en el trabajo realizado.

La tabla 6.5 muestra la medida de similitud normalizada a través de la ecuación 5.4, considerando que la similitud es igual a cero para el caso del Camaro e Iztaccíhuatl, los cuales arrojaron el mayor trabajo en su transformación.

De la tabla 6.5, se observa que los objetos *más parecidos* son los dos automóviles, el Camaro y el Porsche, los cuales se parecen en un 66.22%, mientras que los *más diferentes* son el Camaro y el Iztaccíhuatl, con un parecido del 0%.

Observando el caso de los volcanes solamente, los más parecidos son La Malinche e Iztaccíhuatl, por otro lado los más diferentes son el Popocatépetl e Iztaccíhuatl.

Objetos: $O_1 - O_2$	Iztaccíhuatl	La Malinche	Popocatépetl	Camaro	Porsche
Iztaccíhuatl	2,122	1,609	1,388	1,217	1,237
La Malinche	1,609	2,122	1,361	1,311	1,305
Popocatépetl	1,388	1,361	2,122	1,293	1,348
Camaro	1,217	1,311	1,293	2,122	1,765
Porsche	1,237	1,305	1,348	1,765	2,122

Tabla 6.1. *Voxels* comunes.

Objetos: $O_1 - O_2$	Iztaccíhuatl	La Malinche	Popocatépetl	Camaro	Porsche
Iztaccíhuatl	0	513	734	905	885
La Malinche	513	0	761	811	817
Popocatépetl	734	761	0	829	774
Camaro	905	811	829	0	357
Porsche	885	817	774	357	0

Tabla 6.2. *Voxels* positivos.

6.5 La medida de compacidad discreta en 3D

Recientemente (2000) Bribiesca [70] estableció una medida de compacidad discreta para objetos sólidos 3D compuestos por *voxels*, como una extensión de la medida propuesta para 2D [68](Bribiesca, 1997). Dicha medida relaciona la suma de las superficies de contacto de los *voxels* que conforman al objeto, con el área de la superficie que envuelve al objeto.

$W_H(O_1, O_2)$	La Malinche	Popocatépetl	Iztaccíhuatl	Camaro	Porsche
La Malinche	0	4,367.86	3,829.56	6,475.66	6,341.65
Popocatépetl	4,367.86	0	4,755.78	7,163.24	5,723.60
Iztaccíhuatl	3,829.56	4,755.78	0	7,908.27	7,271.72
Camaro	6,475.66	7,163.24	7,908.27	0	2,671.79
Porsche	6,341.65	5,723.60	7,271.72	2,671.79	0

Tabla 6.3. Medida de similitud de la forma entre los objetos 3D, según el algoritmo húngaro.

$W_{EMC}(O_1, O_2)$	La Malinche	Popocatépetl	Iztaccíhuatl	Camaro	Porsche
La Malinche	0	4,939.32	3,988.10	6,767.43	6,737.35
Popocatépetl	4,939.32	0	5,342.02	7,491.79	6,529.99
Iztaccíhuatl	3,988.10	5,342.02	0	8,210.97	7,526.59
Camaro	6,737.35	7,491.79	8,210.97	0	2,822.12
Porsche	6,737.35	6,529.99	7,526.59	2,822.12	0

Tabla 6.4. Medida de similitud de la forma entre los objetos 3D, según el algoritmo *EMC*.

$S(O_1, O_2)$	La Malinche	Popocatépetl	Iztaccíhuatl	Camaro	Porsche
La Malinche	100	44.77	51.58	18.12	19.81
Popocatépetl	44.77	100	39.86	9.42	27.63
Iztaccíhuatl	51.58	39.86	100	0	8.05
Camaro	18.12	9.42	0	100	66.22
Porsche	19.81	27.63	8.05	66.22	100

Tabla 6.5. Medida de similitud normalizada y acorde al algoritmo húngaro.

Dado un objeto 3D compuesto por n *voxels*, sea A_c el área total de las superficies de contacto de todos los *voxels* que conforman al objeto, la compacidad discreta está definida por la expresión:

$$C_D = \frac{A_c - A_{cmin}}{A_{cmax} - A_{cmin}} \quad (6.1)$$

Donde:

$$A_{cmin} = n - 1 \quad (6.2)$$

y

$$A_{cmax} = 3(n - n^{2/3}) \quad (6.3)$$

Haciendo un cálculo de estas cantidades sobre los objetos propuestos, se encuentra

	A_c	C_D
La Malinche	5,337	0.8577
Popocatépetl	5,504	0.9022
Iztaccíhuatl	5,338	0.8780
Camaro	5,539	0.9116
Porsche	5,631	0.9310

Tabla 6.6. La compacidad discreta para los cinco objetos 3D.

que el valor para A_{cmin} es de 2,121, y para A_{cmax} es de 5,870.6077. La Tabla 6.6 muestra los valores para el área total de la superficie de contacto A_c y la compacidad discreta de los objetos. Como se puede observar, los objetos más compactos son los autos, siendo el Porsche el más compacto. De los volcanes, el más compacto es el Popocatépetl, seguido del Iztaccíhuatl; el menos compacto de ellos es La Malinche.

De la Tabla 6.6, se observa que la diferencia de compacidades entre los volcanes que más se parecen (La Malinche y el Iztaccíhuatl) es muy pequeña (0.0203). Aunque la diferencia de compacidades entre el Popocatépetl y La Malinche es mayor que la diferencia entre el Popocatépetl y el Iztaccíhuatl (0.0445 y 0.0242 respectivamente) el parecido fue mayor entre los dos primeros, pues el número de *voxels* positivos (761) no fue decisivo para que Popocatépetl fuera más diferente a La Malinche que al Iztaccíhuatl. La diferencia de compacidades entre los autos (que son los objetos que más se parecen de todo el conjunto) es de 0.0194. Nuevamente se observa la tendencia como para el caso 2D: los objetos más compactos tienden a parecerse más al objeto más compacto (el Porsche) que a los menos compactos, aunque esta vez no se presentó ningún caso en el que, a más *voxels* a mover, mayor el parecido con el Porsche (recordar el caso del anillo2 con el círculo, Figura 5.2).

6.8 El parecido entre dos objetos no implica la misma semejanza a un tercero

Debido a que se está planteando una métrica con el uso de la medida propuesta, esto quiere decir, que si el objeto A requiere el mismo trabajo W_0 (lo cual arroja un parecido S_0) para transformarse en los objetos B y C , esto no significa que B y C siempre sean iguales en parecido. Debido a las propiedades de la métrica euclidiana, B y C sí pueden ser parecidos pero también un “poco” diferentes, ¿qué tan diferentes? La métrica establece (ecuación 4.19, sección 4.5) que el trabajo de B a C está acotado por la desigualdad:

$$0 \leq W_{BC} \leq 2W_0. \quad (6.4)$$

Para el caso extremo en el que $A = B$ y $A = C$, entonces $W_0 = 0$, debido a la desigualdad anterior es claro que $B = C$. Cuando A y B (o C) comienzan a ser un poco diferentes, también lo empiezan a ser B y C , pero dentro de un rango establecido por la desigualdad anterior.

La gráfica siguiente (Figura 6.12) pone de manera ideal, el trabajo realizado para transformar estos tres objetos en cuestión. El eje vertical contiene los valores que van de 0 a 100, indicando la *desemejanza* de los objetos: cero para los totalmente semejantes (el trabajo W_{BC} es cero cuando los objetos son iguales) y 100 para los “totalmente” diferentes; el eje horizontal representa al trabajo realizado en transformar los objetos A y B (o A y C). El valor de W_{BC} no es función de W_0 , corresponde a toda la región que se encuentra acotada entre las rectas horizontal ($W_{BC} = 0$), la recta que aparece inclinada con una pendiente igual a dos en la Figura 6.12 ($W_{BC} = 2W_0$)

y la recta $W_{BC} = W_{m\acute{a}x}$, en el intervalo $W_0 = [W_{m\acute{a}x}/2, W_{m\acute{a}x}]$.

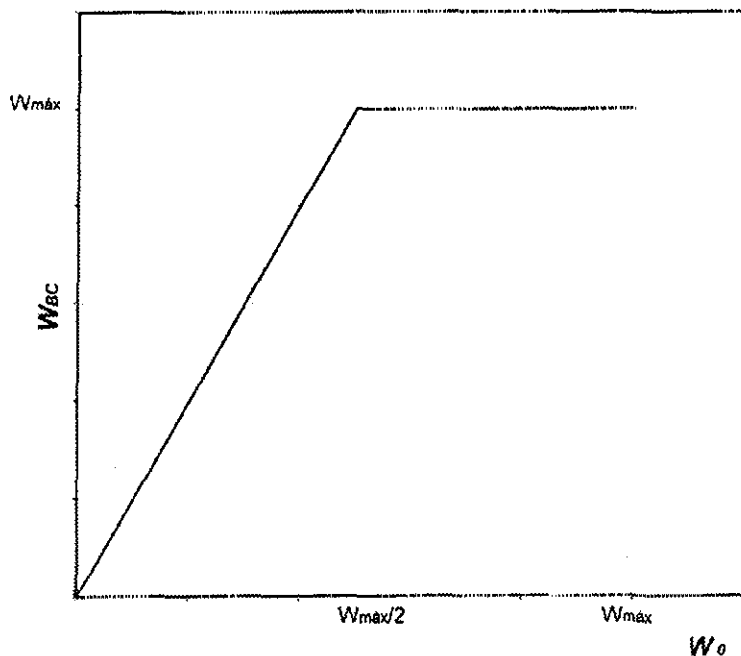


Figura 6.12. El trabajo realizado al transformar B en C está dado por los valores bajo la curva.

A continuación se verá la forma que toma ecuación 6.4, cuando se relaciona con la medida de similitud normalizada de 0 a 100%. Como se explicó en el capítulo anterior, para normalizar la medida de similitud se considera el trabajo máximo $W_{m\acute{a}x}$ realizado en las transformaciones de todos los objetos de un conjunto dado. La relación que existe entre el trabajo y la medida normalizada está dada por:

$$S_{1,2} = \left(1 - \frac{W_{1,2}}{W_{m\acute{a}x}}\right) * 100\%, \quad (6.5)$$

donde los subíndices indican los objetos 1 y 2 a comparar.

Considérense dos casos.

- Caso 1: $0 \leq W_{BC} \leq 2W_0$ si $0 < W_0 \leq \frac{W_{máx}}{2}$.

Combinando la primera desigualdad con la ecuación 6.5, se obtiene que:

$$2S_0 - 100 \leq S_{BC} \leq 100. \quad (6.6)$$

Se observa que cuando S_0 tiene valores cercanos a 100, también lo tiene S_{BC} . Véase el caso extremo: si los objetos A y B (ó A y C) se parecen en un 100% (son iguales), entonces $S_0 = 100\%$ y en la ecuación 6.6 S_{AB} también tiene que ser 100%, es decir, B y C también son los mismos. Si, por ejemplo $S_0 = 95\%$ en parecido, entonces el parecido entre B y C estará en el intervalo: $90\% \leq S_{BC} \leq 100$ según la ecuación 6.6. Si, por otro lado $S_0 = 90\%$ de parecido, entonces al parecido entre B y C estará en el intervalo: $80 \leq S_{AB} \leq 100\%$, si $S_0 = 80\%$ de parecido, entonces al parecido entre B y C estará en el intervalo: $60\% \leq S_{AB} \leq 100\%$, etc. Es decir, conforme A y B tiendan a ser muy diferentes, al intervalo de parecido que pueden tener B y C , se incrementa en la proporción dada por la desigualdad 6.6. El ejemplo límite en este caso, es cuando $S_0 = 50\%$ (cuando A y B “medio” se parecen), entonces B y C pueden estar en el rango de, o “totalmente” diferentes ($S_{BC} = 0\%$), o muy parecidos ($S_{BC} = 100\%$).

- Caso 2: $0 \leq W_{BC} \leq W_{máx}$ si $W_0 > \frac{W_{máx}}{2}$.

Nuevamente, de la ec. 6.5 se tiene que:

$$0 \leq S_{BC} \leq 100. \quad (6.7)$$

Para el caso 2, simplemente conforme S_0 toma valores menores a 50% S_{BC} toma valores dados por la ecuación 6.7: o B y C son totalmente diferentes entre sí o pueden llegar a ser tan similares hasta en un 100%.

Todos los experimentos dados en esta tesis no violan ninguno de los supuestos y ecuaciones anteriores.

Capítulo 7

Conclusiones

A pesar de que existen diferentes formas para alinear los objetos, en este trabajo se consideraron los ejes principales de objetos 2D y 3D irregulares así como sus centros de masa y, se empleó el algoritmo húngaro en la transformación de un objeto a otro.

Cuando dos objetos son irregulares, por ejemplo, piedras, cerros, etcétera, puede ser muy difícil la extracción de características o primitivas. Por ello se propuso emplear los *spels* para el análisis y comparación de los objetos.

El método propuesto es robusto para medir la similitud de la forma de dos objetos irregulares. Se presentaron diferentes objetos irregulares en 2D que mostraron las propiedades más importantes de este método.

Se mostró que hay tres características determinantes en el valor de la similitud al transformar los objetos, estas son: el *número de spels a mover*, su *distancia total recorrida* y la *compacidad* de cada objeto. Además, la medida propuesta no es sensible al ruido.

El método para lograr una medida de similitud, tal como la propuesta aquí, depende de dos factores: primero, hacer coincidir los centros de masa y los ejes principales de los objetos así como normalizarlos en escala; segundo, encontrar la distancia total menor para mover los *spels* en la transformación, a través del algoritmo húngaro. Este método produce una transformación óptima de los objetos. Se estudiaron las propiedades más importantes de este algoritmo, encontrando limitaciones en su de-

sempañó debido a una de sus versiones dada por uno de los autores de las referencias.

El estudio del comportamiento de la compacidad de los objetos, en relación con el número de *spels* positivos y el trabajo en la transformación de los objetos, permitió encontrar un procedimiento consistente para normalizar la medida de similitud.

Haber considerado una representación a partir de *spels* permitió transformar los objetos y, aunque la transformación pudo haberse hecho por sólo considerar las distancias entre ellos, también permitió desplazarlos de un objeto a otro. La manera en que se hizo fue gracias a que los *voxels* están organizados por capas, renglones y columnas, de manera que se movieron por capas. Pudo haberse realizado con otro orden, por ejemplo de manera aleatoria. Sin embargo, cuales sean las opciones en el orden, es claro que este tipo de “movimientos de *spels*” permite obtener objetos intermedios de manera tan suave como lo permita la resolución de los objetos, así como del número de *spels* a desplazar entre los objetos intermedios.

Un trabajo futuro puede consistir en minimizar el tiempo de proceso computacional para calcular la distancia total, ya que el algoritmo húngaro implementado aquí es de orden $O(n^3)$, lo que hace que cuando se incrementa la resolución de los objetos este método sea muy lento, (en el apéndice B, se dan algunos ejemplos de objetos simétricos con los que se puede disminuir el tiempo de ejecución del algoritmo).

En los dos primeros capítulos, se expuso la necesidad de emplear los *spels* para la transformación de los objetos y, se dio un resumen de la literatura actual sobre las medidas de similitud existentes, haciendo hincapié en que existen muchos trabajos sobre reconocimiento de objetos 3D, pero que trabajos explícitos en el grado de parecido de la forma, sobre todos de objetos irregulares (no manufacturados) no se han encontrado.

Existen trabajos de alineación, como los mapeos conformes. Pero no se eligió un

método como éstos debido a la necesidad inherente de representar al objeto a través de *spels*, considerando también la región envuelta por la superficie, a diferencia de los mapeos conformes que sólo emplean la superficie para el análisis de las imágenes.

Existen trabajos sobre transformación de objetos como la interpolación por campos de distancia o la geometría fractal. Sin embargo, está abierta la posibilidad para que en un trabajo futuro, se empleen una de estas técnicas que permita encontrar el grado de parecido de diferentes objetos. En este trabajo se ha empleado un método alternativo, también válido, que es precisamente el de mover los *spels* cuando se pretenda comparar los objetos.

Se ha propuesto emplear al objeto 3D como sólido *voxelizado*. Es una nueva manera de calcular la diferencia entre objetos, a diferencia de como venía haciéndose con los trabajos en campos de distancia, mapeos conformes, etcétera. Aquí, considerar los elementos de imagen representa una manera diferente de encontrar esas diferencias.

Respecto a las aplicaciones, éstas se pueden dar con imágenes *voxelizadas*, tales como los propios volcanes que se han exhibido, y cuya medida puede contribuir en la construcción de una característica más para clasificar a los volcanes. Un trabajo futuro podría analizar este tipo de experimentos ampliando el conjunto de volcanes para su apropiada clasificación.

El método puede darse también en el ámbito de la medicina, en especial con imágenes del cerebro, corazón y, además todos aquellos órganos que tengan huecos o que evolucionen con el tiempo, o bien, que evolucionen debido a alguna enfermedad que modifique sus formas. Aplicando el método se pueden generar tablas de correlación que permitan ir diagnosticando el avance de la enfermedad o crecimiento de algún órgano en particular.

Capítulo 8

Trabajo futuro

1. Queda pendiente la solución al problema planteado en la sección 4.4.3. Un trabajo posterior debe atacar el problema de marcar correctamente los ceros de las matrices. Tal trabajo debe encontrar un algoritmo que “sepa decidir” cómo marcar todos los ceros sin llegar a marcar todos los renglones (o columnas).
2. Otro trabajo futuro consiste en hacer un estudio de si, la manera en que se pueden obtener objetos intermedios para llegar de un objeto a otro al ser comparados, con el método propuesto, es equivalente a llevar a cabo lo que se conoce en la literatura como *morphing* [19](Cohen-Or, 1998).
3. En esta tesis se emplearon los ejes principales, pero un trabajo futuro podría consistir en comparar este método de alineación con otros métodos, tales como el uso de campos de distancia, o como el dado por Malandain y Roscchisani [28](1993) y ver si son equivalentes en el desempeño del trabajo realizado en la transformación de objetos.
4. El método propuesto consiste en normalizar la escala de los objetos. El estudio de la teoría del espacio-escala y sus propiedades podría contribuir a prescindir de este hecho. Una forma alternativa de transformar a los objetos, es a través de un método de multi-resolución, que

se estudia dentro de la teoría de espacio-escala, en el que no sea necesario mover de una sólo vez todos los *spels* de un objeto a otro sino ir transformando al objeto a diferentes resoluciones.

5. Otro de los trabajos pendientes, es poder llevar a un menor tiempo la ejecución del algoritmo húngaro. Los algoritmos genéticos, o bien, el cómputo paralelo podrían servir de apoyo, para de esta manera, poder *voxelizar* a los objetos con una mayor resolución y llevar a cabo sus transformaciones.

A continuación se hace un bosquejo de la *teoría del espacio-escala* (en la que se trata el principio de insertidumbre y el método de multi-resolución) y de *algoritmos genéticos*.

8.1 Teoría del espacio-escala

La teoría del *espacio-escala* conforma una técnica de multi-resolución, que ha ido ganando aceptación desde que empezó a ser desarrollada por Witkin [71](Babaud *et al.*, 1986) y Koenderink [72](1984) en la década de los ochenta. Este método consiste en considerar al hiperplano espacio-escala (x, σ) como el lugar donde se llevan a cabo los eventos para el análisis de las señales y en particular de las imágenes.

8.1.1 Principio de Incertidumbre

Este principio establece que la incertidumbre en el tiempo y en la frecuencia satisfacen la siguiente desigualdad [73](Christopher, 1992):

$$\omega t \geq \frac{1}{2} \quad (8.1)$$

En pocas palabras, la desigualdad 8.1 expresa que, mientras una de las cantidades se determina con mayor precisión, la otra se indetermina, de manera que el producto de ambos esté acotado por $1/2$. Se podría buscar algo similar con parámetros de escala y de forma. Sea S un parámetro de la forma (puede pensarse, para simplificar, en alguna cantidad proporcional al área o volumen de un objeto) y sea σ el parámetro de escala que es inversamente proporcional a la resolución del objeto. Esto quiere decir que cuando se toma la fotografía de un objeto representado por *pixels* o *voxels*, si $\sigma \rightarrow \infty$, o sea, si la resolución disminuye demasiado, se captará con gran incertidumbre la forma S del objeto, es decir, se sabrá poco sobre la forma del objeto.

Y al contrario, si $\sigma \rightarrow 0$ (la resolución se hace muy grande), se puede captar la forma del objeto con una gran precisión. Así que, la ecuación 8.1 puede escribirse ahora en términos del parámetro de la forma y del parámetro de escala como:

$$S\sigma \geq cte. \quad (8.2)$$

De manera que el producto de estas dos cantidades siempre está acotado por una constante (habría que estudiar el valor de esta constante).

Algo que pretende la teoría del espacio-escala es trabajar con objetos independientemente de su escala, pues trata de establecer mecanismos para emular el sistema visual humano, el cual puede trabajar o reconocer objetos a diferentes escalas o resoluciones (“¿se quiere ver las ramas o las hojas del árbol?”); para lograrlo, en matemáticas cualquier medida es adimensional, a diferencia de lo que sucede en la física, en donde toda medida tiene dimensiones. En física se establece la llamada *Ley*

de la invariancia en escala: Las leyes físicas deben ser independientes de la elección de parámetros fundamentales.

Cuando se trata de cumplir la ley de invariancia en escala, se pretende encontrar una relación que no dependa de ninguna dirección espacial, propiedad a la cual se llama *isotropía*. Para encontrar un *kernel* que cumpla con la propiedad de isotropía y con la ley anterior, se utiliza el análisis dimensional [74](Bridgman, 1931). El análisis dimensional trata la forma de combinar cantidades físicas de manera de encontrar relaciones independientes de las unidades (y por ello independientes de la escala). Considérese que la forma que percibe una cámara es S y que la forma del objeto real es S_0 , las cantidades adimensionales que describen este fenómeno son:

$$\frac{S}{S_0} = G(\omega, \sigma)^p, \quad (8.3)$$

donde σ es el parámetro de escala, ω la frecuencia (en el dominio de Fourier) y p es un entero que surge como parte del análisis dimensional.

De la relación anterior, G debe ser una función que cumpla con las siguientes características: si $\sigma \rightarrow \infty$ (es decir, si la resolución disminuye), la forma del objeto captada por la cámara será de menor precisión. S no deberá exceder la precisión de S_0 . Si $\sigma \rightarrow 0$, según la ecuación (8.2) S se determina mejor (permitiendo observar a S con mejor detalle). El límite superior de G es igual a uno (que es cuando S y S_0 son iguales). En resumen, se tiene que encontrar una función $G(\sigma, \omega)$ (en el dominio de Fourier) que cumpla con las siguientes características:

$$\lim_{\sigma\omega \rightarrow 0} G(\sigma, \omega) \rightarrow 1, \quad (8.4)$$

y

$$\lim_{\sigma\omega \rightarrow \infty} G(\sigma, \omega) \rightarrow 0. \quad (8.5)$$

La función de Gauss es el llamado *kernel* que cumple con tales características. En coordenadas espaciales de tres dimensiones (en el dominio espacial) el *kernel* queda como:

$$G(\mathbf{x}, \sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{3}{2}}} \exp\left(-\frac{x^2 + y^2 + z^2}{2\sigma^2}\right), \quad (8.6)$$

donde σ es una medida de la llamada escala interior o parámetro de escala, en el caso del presente trabajo tiene que ver con las dimensiones del *voxel*, y (x, y, z) son las coordenadas espaciales de los *voxels* del objeto.

Este *kernel* es invariante en escala, rotación (hay isotropía espacial) y traslación. Para utilizarlo en el problema de reconocimiento de objetos 3D *voxelizados*, se puede aplicar sobre dos objetos sólidos, aislados, que tienen la misma forma, pero cada uno con una orientación y escalas diferentes, para luego obtener una tabla de valores.

Dados dos objetos a comparar, se pueden aprovechar las propiedades de las derivadas gaussianas, calcular las direcciones normales a la superficie usando la curvatura gaussiana que permita orientar a los objetos; cuando éstos son diferentes, sus *kernels* pueden dar también valores diferentes.

Se podría establecer un valor de energía en términos de la diferencia de los valores gaussianos para objetos diferentes, que permita auxiliar en la construcción de una medida de similitud.

Una ventaja de aprovechar esta teoría es que no sería necesario mantener el número de *voxels* constante para la medida de similitud.

8.1.2 Método de multi-resolución

Este método podría consistir en tomar a los objetos originales, representarlos por medio una gran cantidad de *spels* (es decir, una resolución suficiente para que los detalles de cada objeto se vean con una buena aproximación). Posteriormente, se procedería a aplicar, por ejemplo, un filtro gaussiano para que el número de *spels* disminuya y así obtener una resolución menor (menor número de *spels* respecto al original). Sin embargo, la forma del objeto iría degradándose de manera tan “suave” como se pueda, para que aún después del filtro los objetos intermedios sigan pareciéndose demasiado entre sí. Para realizar ésto, habría que investigar el número adecuado de veces en aplicar el filtro. Cuando se tengan N pares de objetos degradados en la resolución, se procederá a transformar a los objetos que tienen la menor resolución, una vez que nuevamente se ha normalizado en área. En este caso la cantidad de *spels* a mover será muy pequeña. Se guarda el trabajo realizado en esta transformación y se suma a los realizados entre los objetos intermedios del objeto transformado.

Véase un ejemplo: un objeto en forma de elefante se desea comparar con otro objeto en forma de automóvil (véase la figura 8.1). Se degrada la forma de la imagen como se mencionó arriba. Supóngase que se aplica tres veces un filtro gaussiano, hasta lograr una resolución de 16×16 *pixels*. A esta resolución, transformar el elefante y el auto no requiere de muchos movimientos de *pixels*. Se guarda el trabajo realizado en esta transformación, y se suma a los trabajos realizados en transformar el auto en la dirección en que indica la flecha.

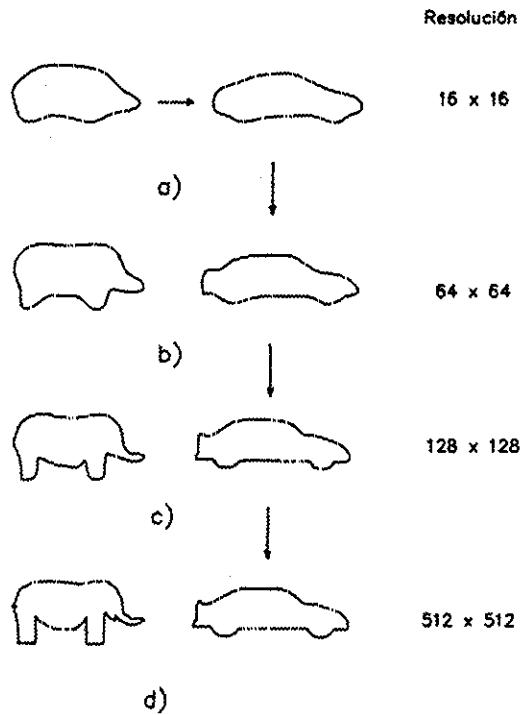


Figura 8.1. Un elefante y un automóvil a transformarse. Los objetos se degradan en resolución con filtros gaussianos, desde d) hasta a); a) se transforman primero los que tengan menor número de *pixels*. Posteriormente, se va transformando el segundo objeto a ser comparado (el auto), entre sus versiones a diferentes resoluciones (en la dirección de la flecha).

8.2 ¿ Cómo aplicar algoritmos genéticos al problema de optimización?

Como ya se sabe, obtener la distancia mínima de desplazamiento de los *voxels* en la transformación de los objetos consistió en encontrar un apareo óptimo, es decir, dada una gráfica bipartita $G < X, Y, E >$, saturar todos los vértices (que cada arista sea adyacente a un vértice) tal que la suma total de los pesos sea la menor de todas las posibles. Considérese el ejemplo dado por la siguiente gráfica, donde el conjunto X representa los *voxels* a mover y el conjunto Y las posiciones a alcanzar por los *voxels*.

Los pesos de esta gráfica pueden ponerse como elementos de una matriz cuadrada

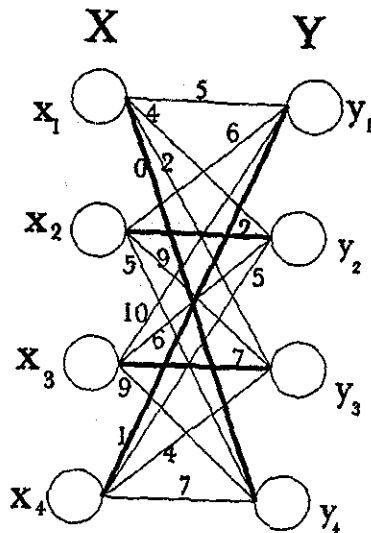


Figura 8.2. Apareo óptimo. Todos los vértices están saturados y la suma del subconjunto de aristas E es la menor.

de tamaño $n \times n$, donde las columnas representan las y 's y los renglones las x 's. Por ejemplo el elemento $(x_2, y_3) = (2, 3)$ es la arista cuyo peso es $w(x_2, y_3) = 9$. La forma de resolver este problema es muy semejante a la del problema del agente viajero (PAV). Para encontrar el apareo óptimo de esta gráfica bipartita, la matriz que representa los pesos de la gráfica podría escribirse como una cadena de ceros y unos. Enseguida, se pueden utilizar las técnicas de cruzamiento y mutación durante diferentes iteraciones hasta encontrar la mejor cadena. Sin embargo, se ha observado que manejar el problema con *bits* no resulta muy adecuado, pues como se puede ver en [75], utilizar código binario para los vértices (o ciudades en el caso del problema del agente viajero, PAV) y emplear cualquier mínima modificación en la cadena, por mutación, resulta en inconsistencias (como vértices doblemente saturados por ejemplo, o duplicaciones de ciudades en un viaje, si se quiere ver como PAV). Además, recientemente [76](Lidd,

1991) se ha reportado que el uso de código binario ha logrado encontrar la distancia mínima para 100 vértices solamente (con una PC de 350Mhz y 64Mb de memoria el que escribe ha logrado encontrar la distancia mínima usando 200 vértices en menos de 10 minutos). Por ejemplo, se podría intentar algo parecido a uno de los métodos clásicos para resolver el problema del agente viajero, en el que se emplea operadores de ciclo *crossover* (CX) [77](Goldberg, 1989), [78](Michalewicz, 1996).

La idea es la siguiente:

Considérese dos cadenas: la primera representa al conjunto X de vértices de la gráfica bipartita original y la otra al conjunto Y . Se llamarán cromosomas padre:

$$x_1x_2\dots x_n,$$

$$y_1y_2\dots y_n.$$

Se puede utilizar como ejemplo la gráfica de la Figura 8.2, con cuatro vértices en cada conjunto, de manera que la segunda cadena sea la primera en orden aleatorio, para formar dos cromosomas padres.

$$1234,$$

$$3412.$$

Se coloca el primer gen (el 1) del primer padre en un nuevo cromosoma hijo, y se compara con el primer gen del segundo padre. Se busca este gen (el 3) en el primer padre y se coloca en la misma posición en el cromosoma hijo. Enseguida se compara con el correspondiente gen del segundo padre (que es un 1) y se repite la operación. Debido a que se encontró un número ya utilizado del primer padre, se termina el algoritmo:

1 - 3 - -

Se lleva a cabo una recombinación de genes al colocar, en los genes faltantes del cromosoma hijo, los números correspondientes del segundo padre. Lo mismo se hace para el segundo cromosoma hijo:

1432,

3214.

Se registra la distancia total (en este caso es $2 + 5 + 10 + 5 = 22$) y se repite el procedimiento, registrando sólo aquellas sumas que cumplan con:

$$\sum w_{(i-1)}(x_j y_j) \leq w_{(i)}(x_j y_j),$$

donde i es la iteración hecha después de la iteración $i-1$. A diferencia de CX original, aquí sí se pueden repetir los valores (alelos) de los genes de la misma posición. De hecho, después de cierta cantidad de iteraciones, los cromosomas hijos pueden quedar como:

1234,

4231,

que corresponde a la suma $0+2+7+1 = 10$. La cual es la menor que se puede registrar en todas las iteraciones. Este es un método con pocas restricciones y habría que probar su eficiencia.

8.2.1 Restricciones sobre los AGs basadas en el método húngaro

Un método con más restricciones podría ser el siguiente. Considérese esta vez cadenas de ceros y unos. Sólo que se debe tener cuidado de no hacer cualquier tipo de recombinación o mutación arbitraria (hay que poner más restricciones que en el caso anterior).

Este procedimiento que se propone se basa en utilizar restricciones dadas por el método húngaro y uso de matrices. Según ese método [65](Jonker, 1986), primero se minimiza la matriz, esto es, se resta el menor de cada columna de la matriz y después el menor de cada renglón. Dado el ejemplo de la matriz anterior, la matriz minimizada queda como:

$$\begin{bmatrix} 5 & 4 & 1 & 0 \\ 4 & 0 & 6 & 3 \\ 4 & 0 & 0 & 3 \\ 0 & 4 & 2 & 6 \end{bmatrix}$$

La posición de los ceros de esta matriz corresponde a posibles aristas de la gráfica bipartita que forman parte de un apareo óptimo. Para el AG que se desea construir, considérense dos cadenas padre de longitud $n \times n$ (donde n es el número de renglones de la matriz), en el ejemplo expuesto la longitud es de $4 \times 4 = 16$. En el primer padre cada cadena de n elementos representa un renglón de la matriz (de izquierda a derecha), y cada cadena de n elementos del segundo padre representa una columna (de arriba abajo). Sólo habrá unos y ceros. Los unos son los ceros de la matriz, mientras que los ceros representa cualquier número diferente de cero de la matriz original:

0001010001101000,

0001011000101000.

Se procede a buscar n unos *independientes*. Esto es, en cada subcadena de cuatro elementos se busca al menos un uno, que ninguno repita la misma posición en las subcadenas restantes. Por ejemplo, la subcadena 0001 (del primer padre) y 0100 tienen dos unos independientes, están en la posición 4 y 2 respectivamente. En la tercera subcadena, el primer uno repite la posición 2, por lo que se hace una mutación: $1 \rightarrow 0$. Pero el siguiente uno de esa misma subcadena tiene la posición 3, y el uno de la cuarta subcadena tiene la posición 1. Así que la cadena queda como un cromosoma hijo:

0001010000101000

con cuatro unos independientes. Los valores de los pesos para estos unos en la matriz original son 0,2,7,1, cuya suma da 10 y es la distancia mínima. Sin embargo, puede ser que la matriz minimizada represente los cromosomas padre de la siguiente forma:

0010001010000101,

0010000111000001.

En el primer hijo de la mutación del primer padre no se encontraron cuatro unos independientes, ya que si se aplica la mutación $1 \rightarrow 0$, quedaría una segunda subcadena de cuatro ceros. Una forma de resolver esto sería mutar algunos de los otros ceros

de la primera subcadena. Se debe saber con cierta certidumbre qué ceros mutar. Para ello, se cubren subcadenas que contengan unos y se mutan genes para generar más unos. Se cuenta el número de unos que se repitan en la misma subcadena, así como de la subcadena respectiva del segundo padre, dada por la posición en la subcadena del primer padre, cubriendo¹ la subcadena que tenga más unos. En el ejemplo que se está tratando, el uno está en la tercera posición y hay un sólo uno en esta subcadena. Se cuentan los unos de la tercer subcadena del segundo cromosoma. Hay dos unos y se marca la subcadena: 1*1*0*0*. Se salta la segunda subcadena del primer padre porque se repite la posición del uno. En la tercera subcadena hay un solo uno, que está en la primera posición, por lo tanto se cuenta el número de unos de la primer subcadena del segundo padre, hay también un sólo uno. Así que se marca la tercera subcadena del primer padre: 1*0*0*0*, la siguiente subcadena (la 4) del primer padre, tiene dos unos, el primero de ellos está en la segunda posición, así que se cuentan los unos de la segunda subcadena del segundo padre. Hay un solo uno. Así que se marca la cuarta subcadena del primer padre: 0*1*0*1*. Claramente, no pueden estar todas las subcadenas (de cualquier padre) marcadas al mismo tiempo, pues ya no podrían generarse más ceros en la matriz original. Si esto pasa, se puede invertir la regla anteriormente dada para marcar la subcadena del primer padre cuando haya igual número de unos, en vez de ello se marcaría la subcadena correspondiente del segundo padre. El procedimiento se repite para las demás subcadenas del primer padre. Si se observa este paso como una matriz en vez de cadenas, quedaría así:

$$\begin{bmatrix} 0 & 0 & 1* & 0 \\ 0 & 0 & 1* & 0 \\ 1* & 0* & 0* & 0* \\ 0* & 1* & 0* & 1* \end{bmatrix}$$

¹La forma de cubrir una cadena se hará marcando sus elementos con un *.

Como se observa, el elemento, o valor del gen de la posición del tercer alelo de la cadena marcada $1*1*0*0*$, también aparece en la tercer subcadena del primer padre en la misma posición. Lo mismo ocurre con el valor de la cuarta posición: aparece en la cuarta subcadena en la tercera posición del primer padre. Se dice entonces que estos alelos están doblemente marcados. Siguiendo con las condiciones del algoritmo húngaro, los valores de la matriz original en las posiciones dadas por estos alelos, se incrementan en una cantidad igual al valor más pequeño del resto de la matriz (original) que no tiene su contraparte marcada en las subcadenas. Y el resto de la matriz original, (la que no está marcada en su contraparte de subcadenas) se minimiza nuevamente restando el valor más pequeño. Con ello se garantiza tener más ceros. Localizando estos ceros en la matriz minimizada, se pueden mutar los cromosomas generando más unos. Se vuelven a buscar n unos independientes. Si los hay el algoritmo termina y se ha encontrado la distancia mínima. Si no, se repite todo el procedimiento. El objetivo principal de este AG es generar tantos unos como sea permisible por el algoritmo húngaro, éstos serán candidatos “fuertes” a ser parte del apareo óptimo en la gráfica bipartita. De esta manera el AG irá convergiendo a la distancia mínima.

8.2.2 Eficiencia de un AG para el cálculo de la distancia mínima en la transformación de dos objetos

Lo que puede hacer un AG es acercarse a la solución en un tiempo relativamente corto, cuya solución sea mejor que la de Bribiesca pero no mejor que la dada por el algoritmo húngaro, debido a que, revisando la literatura [78](Michalewicz, 1996), lo más que se ha logrado, para un caso simétrico del agente viajero, es acercarse a

la solución óptima hasta por un 60%, con 532 nodos (claramente la simetría puede reducir un poco el tiempo de búsqueda en la solución del problema). Los demás casos reportados han estado por debajo de esta cantidad de nodos o vértices. Sería interesante probar los dos algoritmos descritos. El primero tiene pocas restricciones, pero su implementación es más sencilla. Habría que ver si la aleatoriedad de la formación del segundo cromosoma padre permite encontrar rápidamente una solución cercana a la óptima en poco tiempo. O tal vez sea bueno para algún tipo de matrices que representen la transformación de ciertas figuras 3D. El segundo método puede ser viable, ya que sus restricciones se basan en la versión del algoritmo húngaro presentado por [65](Jonker, 1986). Si este AG llega a los n unos independientes en poco tiempo, con n lo suficientemente grande (más de 5000), sería muy apropiado para encontrar la distancia óptima.

Apéndice A

Algunos objetos 3D regulares

Se mostrarán algunos objetos cuyas alineaciones son fáciles de obtener, ya que sus ejes principales coinciden con sus ejes de simetría y, como se observará, las formas de estas figuras pueden identificarse claramente con tan sólo pocos *voxels*. Fue en el transcurso de estos experimentos que se observaron las limitaciones del algoritmo propuesto en la sección 4.4, debido a que ya no pudo ampliarse la resolución de los objetos que a continuación se muestran.

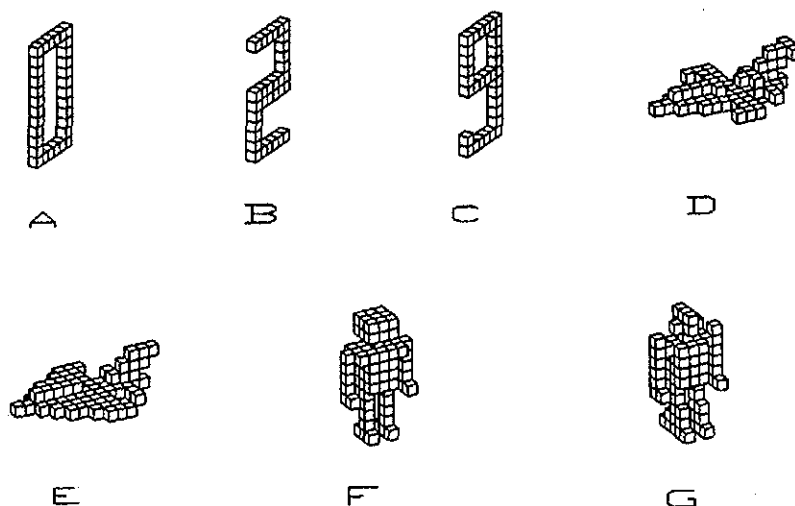


Figura A1. Siete objetos sintéticos empleados para transformarse entre algunos de ellos, usando el algoritmo planteado por Gould de la sección 4.1

La Figura A1 muestra a los objetos utilizados para realizar las siguientes trans-

Transformación	Voxels comunes	Voxels a mover
$A \rightarrow B$	24	8
$A \rightarrow C$	28	4
$B \rightarrow C$	24	8
$D \rightarrow E$	75	12
$F \rightarrow G$	75	12

Tabla A1. Número de *voxels* comunes a ambos objetos y *voxels* positivos a mover en cada transformación.

formaciones: $A \rightarrow B$, $A \rightarrow C$, $B \rightarrow C$, $D \rightarrow E$ y $F \rightarrow G$.

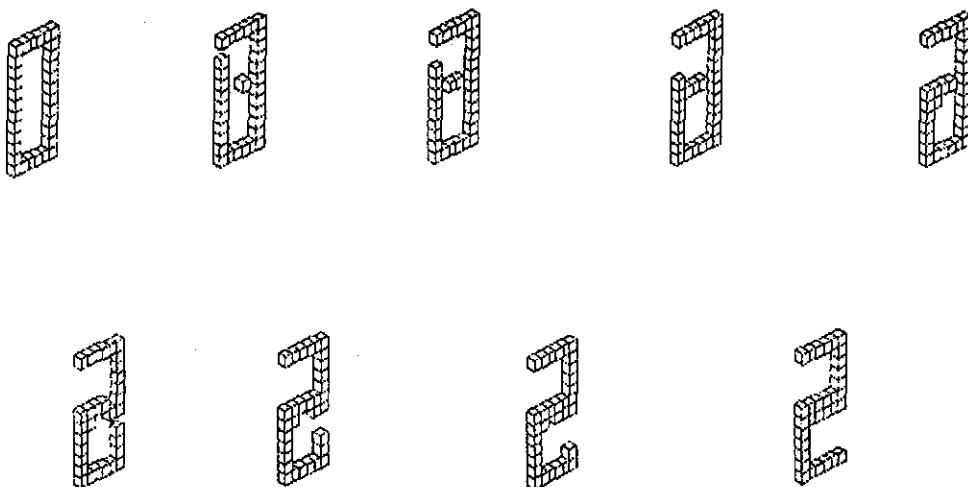


Figura A2. Transformación $A \rightarrow B$.

Las tablas A1 y A2 muestran los resultados de haber realizado las transformaciones entre los objetos de la Figura A1, en la forma que se muestra en las Figuras A2 y A3 para los casos $A \rightarrow B$, $D \rightarrow E$ y $F \rightarrow G$. En estas figuras aparecen las transformaciones en forma progresiva.

Se hace una comparación entre el método empleado en este trabajo y el de ele-

Transformación	W_{EMC}	W_H
$A \rightarrow B$	26.73	26.26
$A \rightarrow C$	14.14	14.14
$B \rightarrow C$	30.75	29.95
$D \rightarrow E$	65.36	60.51
$F \rightarrow G$	103.91	99.98

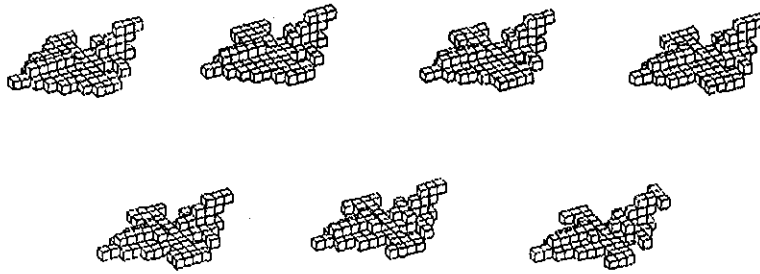
Tabla A2. Comparación del trabajo realizado según el método aplicado para transformar a los objetos.

mentos más cercanos (EMC).

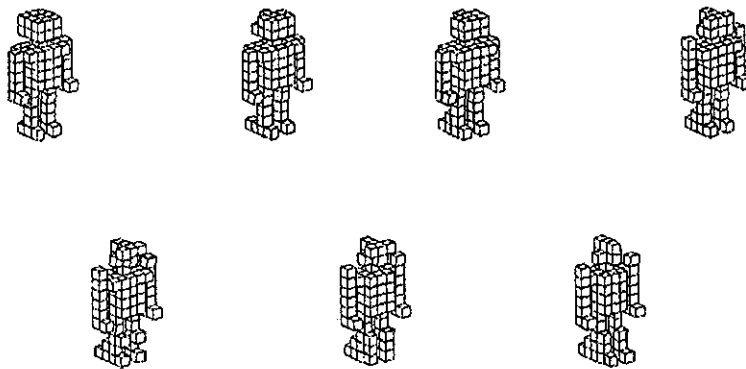
Se puede ver que el trabajo realizado al hacer las transformaciones empleando el algoritmo EMC , es mayor o igual que el trabajo realizado al emplear el algoritmo húngaro en la versión de Gould [60](1988).

Como se puede observar, el número de *voxels* a mover no excede los 12. ¿Qué sucede si se quiere transformar el *robot* en el primer *avión*? Ello significaría mover 58 *voxels*. ¿Qué pasa si se aumenta la definición o el tamaño de los objetos? ¿por qué no se transformaron algunos de estos casos con este algoritmo?

Se hizo un estudio en el capítulo 4 de la manera en que trabaja el algoritmo húngaro implementado en la versión de Gould, ello permite ver por qué no se puede aplicar el método a transformaciones en las que el número de *voxels* a mover exceda a 12, ya que en el transcurso de la búsqueda de los ceros independientes se llega a una etapa de la matriz como la presentada en la Figura 4.9 del Capítulo 4.



a)



b)

Figura A3. Transformaciones progresivas: a) $D \rightarrow E$ y b) $F \rightarrow G$.

Apéndice B

Objetos simétricos

Buscando la forma de mover muchos *voxels* al mismo tiempo, se encontró que transformar objetos con cierta simetría (ya sea respecto a un punto, un eje o un plano), la forma de calcular el trabajo de la transformación puede disminuir el tiempo de ejecución del programa. Se puede particionar a los sólidos y calcular el trabajo de transformación.

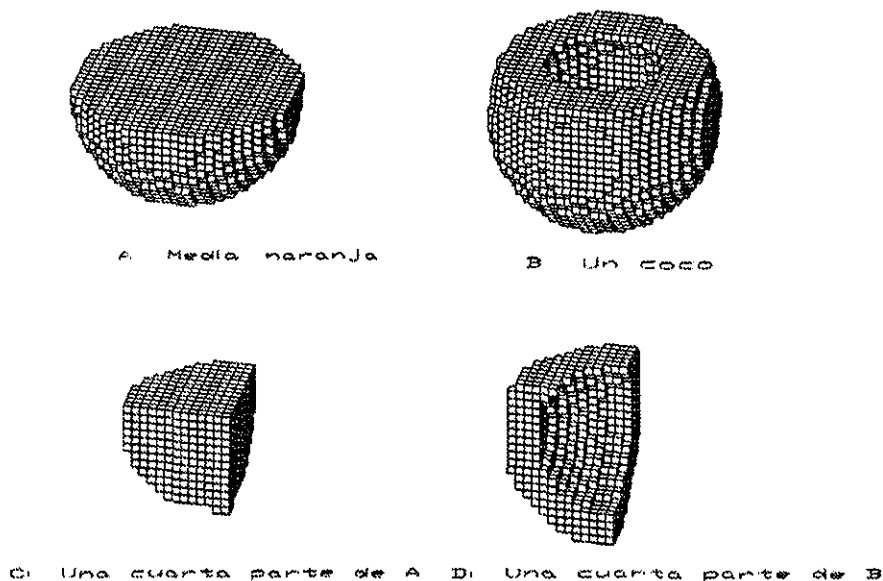


Figura B1. Para transformar A en B no es necesario mover los 1,916 *voxels* comunes, sino sólo una cuarta parte de ellos (479 *voxels*), pues las otras tres cuartas partes son idénticas entres sí.

Proposición: dados dos objetos *voxelizados*, simétricos respecto a un punto, eje o plano de simetría, el trabajo realizado para transformar a los dos objetos es igual

Objetos a transformar:	1/4 de media naranja-1/4 de coco	Media naranja-coco
Número de <i>voxels</i> :	1,311	5,244
<i>Voxels</i> comunes:	832	2,456
<i>Voxels</i> a mover :	479	1,916
Trabajo realizado W_B :	35,940	143,760
Trabajo realizado W_H :	30,114	120,456

Tabla B1. Resumen de la transformación hecha para los objetos originales A y B. Nótese que para transformar figuras geométricas es suficiente con particionar los objetos.

a la suma de los trabajos realizados en transformar una de las particiones discretas, multiplicado por el número de particiones.

Como un ejemplo representativo, supóngase que se tiene una media naranja y un coco como los mostrados en la Figura B1.

Estos objetos tienen simetría axial. Si se quiere transformarlos, se podrían mover los 1,916 *voxels* no comunes que tienen entre sí. Pero debido a su simetría es suficiente con mover sólo 479 *voxels* y luego multiplicar este resultado por cuatro. La tabla B1 resume este hecho, válido para todos los objetos simétricos.

Apéndice C

Artículos derivados de esta tesis

1. Sánchez, H. Optimización de una medida de semejanza para objetos tridimensionales a partir de invariantes y transformaciones. *Computación y Sistemas*. 3(4), Abr-jun 2000.

En este artículo se propone la transformación de objetos simétricos y de baja resolución, utilizando el algoritmo húngaro en una versión modificada a base de minimización de matrices. Se demuestra experimentalmente que este algoritmo no es apropiado para transformar a los objetos de mayor resolución.

2. Sánchez, H. and Bribiesca, E. A method of optimum transformation of 3D objects used as a measure of shape dissimilarity. Sometido y aceptado con cambios menores a la revista *Image and Vision Computing*.

En este artículo se presentó el método propuesto en el presente trabajo aplicado a imágenes 3D, cuyos ejemplos fueron los volcanes cercanos a la cuenca de México.

Referencias

- 1 Bribiesca, E. Measuring 3-D Shape Similarity Using Progressive Transformations, *Pattern Recognition*, 29 (1996) 1117-1129.
- 2 Ballard, D. H. and Brown, C. *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ (1982).
- 3 Freeman, H. On the Encoding of Arbitrary Geometric Configurations, *IRE Trans. Electron. Comput.* EC-10 (1961) 260-268.
- 4 Bribiesca, E. A New Change Code. *Pattern Recognition* 32 (1999) 235-251.
- 5 Besl, P. J. and Jain, R. C. Three-dimensional Object Recognition, *ACM Computing Surveys*, 17 (1985) 75-139.
- 6 Bribiesca, E. A Chain Code for Representing 3D Curves. *Pattern Recognition*, 33 (5), (May 2000) 755-765.
- 7 Foley, James D.; Dam, Andries van Dam; Feiner, Steven K; Hughes, John F. *Computer Graphics: Principles and Practice*. Second edition. The system Programming Series. Addison-Wesley Publishing Company. (1987) 1175pp.
- 8 Russ, John C. *The Image Processing Handbook*.
- 9 Dickinson, S. J.; Pentland, A. P.; Rosenfeld, A. From volumes to views: an approach to 3-D object recognition, *CVGIP: Image Understanding*, 55 (1992) 130-154.
- 10 Lohmann, G. *Volumetric Image Analysis*, Wiley & Sons and B. G. Teubner Publishers, New York, NY, (1998).
- 11 Henk J.A.M. Heijmans, Alexander V. Tuzikov. Similarity and Symmetry Measures for Convex Shapes Using Minkowski Addition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (9) (Sept. 1998).

- 12 Otterloo, Van; Peter, J. A Contour-Oriented Approach to Shape Analysis. Prentice Hall, 368pp. (1991).
- 13 Arkin, Esther M.; Chew L. Paul; Huttenlocher, Daniel P.; Kedem Klara; and Mitchel Joseph S. B. An Efficient Computable Metric for Comparing Polygonal Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (3) 1991.
- 14 Jain K., Fundamentals of Digital Image Processing. Prentice Hall Information and System Sciences Series. Thomas Kailath, series Editor, (1989) 569pp.
- 15 Danielsson, Per-Erik. Note a New Shape Factor. Computer Graphics and Image Processing. 7, (1978) 292-299.
- 16 Boyse, J. W., Data Structure for Solid Modeller, NFS Workshop on the Representation on Three-Dimensional Objects, University of Pennsylvania (1979).
- 17 Brooks R., Model Based 3-D Interpretations of 2-D Images, IEEE Trans. Pattern Anal. Mach. Intell., 5 (2), (1983) 140-150.
- 18 Jain, A. K.; Hoffman, R. Evidence-based Recognition of 3-D Objects, IEEE Transactions on Pattern Analysis and Machine Intelligence, 10 (1988) 783-802.
- 19 Cohen-Or, D.; Levin D. and Solomovici, A. Three-dimensional Distance Field Metamorphosis, ACM Transactions on Graphics, 17 (1998) 116-141.
- 20 Adan, A.; Cerrada, C. and Feliu, V. Global Shape Invariants: a Solution for 3D Free-Form Objects Discrimination/Identification Problem, Pattern Recognition, 34 (2001) 1331-1348.
- 21 Holden, M.; Hill, D. L. G.; Denton, E. R. E.; Jarosz, J. M.; Cox, T. C. S.; Rohlfing, T.; Goodey, J. and Hawkes, D. J. Voxels Similarity Measures for 3-D Serial MR Brain Image Registration, IEEE Transactions on Medical Imaging, 19 (2000) 94-102.
- 22 Mokhtarian, F.; Khalili, N. and Yuen, P. Multi-Scale Free-Form 3D Object Recognition Using 3D Models, Image and Vision Computing, 19 (2001) 271-281.

- 23 Zhang, S.; Sullivan, G. and Baker, K. The Automatic Construction of a View-Independent Relational Model for 3D Object Recognition, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 15 (6) (1993) 778-786.
- 24 Bribiesca, E. and Wilson, R. G. A Measure of 2D Shape-of-Object Dissimilarity. *Appl. Math. Lett.*, 10 (6) (1997) 107-115.
- 25 Bribiesca, E. and Guzman, A. How to Describe Pure Form and How to Measure Differences in Shapes Using Shape Numbers. *Pattern Recognition*, 12 (1979) 101-112.
- 26 Besl, Paul J. and McKay, Neil D. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14 (2) (Feb, 1992) 239-256.
- 27 Haker, Steven; Angenent, Sigurd; Tannenbaum, Allen; Kikinis, Ron; Sapiro, Guillermo and Halle Michael. Conformal Surface Parameterisation for Texture Mapping. *IEEE Transactions on Visualization and Computer Graphics*, 6 (2) (April-June, 2000) 181-189.
- 28 Malandain, Grégoire and Rochisani, Jean-Marie. Matching of 3D Medical Images with a Potential Based Method, *Unité de Recherche INRA-Sophia Antipolis*. No. 1890 (Mar 1993) 1-37.
- 29 Louis, A. K., & Natterer, F. Mathematical Problems of Computer Tomography. *Proc. IEEE*, 71 379-389.
- 30 Levin, David. Multidimensional Reconstruction by Set-valued Approximations. *IMA Journal of Numerical Analysis*, (1986) 6, 173-184.
- 31 Payne, Bradley A. and Toga, Arthur W. Surface Mapping Function 3D Models. *Medical Imaging. IEEE CG&A*, 10 (5) (Sept 1990) 33-41.
- 32 Payne, Bradley A. and Toga, Arthur W. Distance Field Manipulation of Surface Models. *IEEE Computer Graphics and Applications*, 12 (1) (Jan 1992) 65-71.
- 33 Barequet, G. and Sharir, M. Piecewise Linear Interpolation Between Polygonal Slices. In *Proceedings of the Tenth Annual ACM Symposium on Computational Geometry (Stony Brook, NY)*, (1994) 93-102.

- Full version: *Comput. Vision Graph. Image Process. Image Understand*, (Jan,1996) 63, 1.
- 34 Mandelbrot, B. *The Fractal Geometry of Nature*, W. H. Freeman and Co., San Francisco (1982)[English translation of the French version published in 1975].
 - 35 Arakawa, Kenichi and Erick Krotkov. *Fractal Modeling of Natural Terrain: Analysis and Surface Reconstruction with Range Data*. *Graphical Models and Image Processing*, 58 (5), (Sept 1996) 413-436.
 - 36 Turiel, A.; del Pozo, A. *Reconstructing Images from their Most Singular Fractal Manifold*. *Image Processing, IEEE Transactions on*, 11 (4) (Apr 2002) 345 -350.
 - 37 Belloulata, K.; Konrad, J. *Fractal Image Compression with Region-Based functionality*. *Image Processing, IEEE Transactions on*, 11 (4) (Apr 2002) 351-362.
 - 38 Neil, G. and Curtis, K. M. *Shape Recognition Using Fractal Geometry*, *Pattern Recognition*, 30 (12) (1997) 1957-1969.
 - 39 Wood, Jeffrey. *Invariant Pattern Recognition: A Review*. *Pattern Recognition*, 29 (1) (Jan 1996) 1-17.
 - 40 Hu, M. K. *Visual Pattern Recognition by Moment Invariants*, *IEEE Transactions in Information Theory*, 8 (1962) 179-187.
 - 41 Karush, W. *Webster's New World Dictionary of Mathematics*, Simon & Schuster, Inc., New York, (1989).
 - 42 Sadjadi, F.A. and Hall, E. L. *Three Dimensional Moment Invariants*. *IEEE Trans. Pattern Anal. Machine Intell.*, vol PAMI-2, (Mar 1980) 127-136.
 - 43 Lo, C. and Don, H. *3-D Moment Forms: Their Construction and Application to Object Identification and Positioning*. *IEEE Trans. on Pattern Anal. and Machine Intelligence*, 11 No. 10. (1989) 1053-1064.
 - 44 Burel G. and Hnocq H. *Three-dimensional Invariants and Their Application to Object Recognition*. *Signal Processing*, 45, (1995) 1-22.

- 45 Khotanzad, A. and Hong, Y.H., Invariant Image Recognition by Zernike Moments. *IEEE Trans. on Pattern Anal. and Machine Intelligence*. Vol 12, (1990) 489-497.
- 46 Li, Y., Reforming the Theory of Invariant Moments for Pattern Recognition. *Pattern Recognition*, 25 (7) (1992).
- 47 Faber, Tracy L, Stokety, Ernest M. Orientation of 3D Structures in Medical Images. *IEEE Transactions on Pattern and Machine Intelligence*, 10 (5), (Sep 1988) 626-633.
- 48 Gálvez, J. M. and Canton, M. Normalisation and Shape Recognition of Three-Dimensional Objects by 3D Moments. *Pattern Recognition*, 26 (5) (1993) 667-681.
- 49 Bullock, H.; Dooley, L. and Wermser, D. Application of principal axes for Registration of NMR Image Sequences, *Pattern Recognition Letters*, 21 (2000) 329-336.
- 50 Borisenko, A. I. and Tarapov, I. E. *Vector and Tensor Analysis*, Dover Publications, Inc., New York (1979).
- 51 Wolfram, S. *Mathematica: A System for Doing Mathematics by Computer*, Addison-Wesley, Redwood City, California (1991).
- 52 Kanatani K., *Group of Theoretical Methods in Image Understanding*, Springer - Verlag. (1990).
- 53 Dirección electrónica:
<http://www.maths.tcd.ie/pub/histmath/people/hamilton/quaternions.html>
- 54 *The Transforms and Application Handbook*. Alexander D. Poularikas. IEEE Press. (2000).
- 55 Helgason, S., *The Radon Transform*. Birkhauser, Boston, (1980).
- 56 <http://saigon.ece.wisc.edu/waveweb/Coder/hvs.html>
- 57 Martens, JB. Local Orientation Analysis in Images by Means of the Hermite Transform. *IEEE Transactions on Image Processing*, 6 (8) (Aug. 1997) 1103-1116.

- 58 Bondy, J. A. and Murty, U. S. R. Graph Theory with Applications, The MacMillan Press Ltd, London and Basingstoke, (1976).
- 59 Chartrand, Gary. Introductory Graph Theory. Imprim. New York: Dover, 294pp (1985).
- 60 Gould, R., Graph Theory, Emory University, The Benjamin/Cummings Publishing Company, Inc. (1988)
- 61 Gass, Saul I. Linear Programming: Methods and Applications. New York: McGraw-Hill, 223pp (1958).
- 62 Hadley, George F. Linear Programming. Addison-Wesley, 520pp (1962).
- 63 Kuhn, H. W. The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly, 2 (1) (Mar 1955).
- 64 Evans, James R, and Minieka, Edward. Optimisation Algorithms for Networks and Graphs. Marcel Dekker, Inc, 470pp. 1992.
- 65 Jonker and Volgenant. Improving the Hungarian Assignment Algorithm. Operation Research Letters, 5 (4) Oct 1986.
- 66 Sánchez, Hermilo. Optimización de una medida de Semejanza para Objetos Tridimensionales a Partir de Invariantes y Transformaciones. Computación y Sistemas, 3 (4) (Abr-jun 2000).
- 67 Meisel, William S. Computer-Oriented Approaches to Pattern Recognition. Academic Press. (1974).
- 68 Bribiesca, E. Measuring 2-D Shape Compactness Using the Contact Perimeter. Comput. Math. Appl., 33 (11) (Jun 1997) 1-9.
- 69 Egerváry, J., Matrixok Kombinatorikus Tulajdonságairól. Matematika és Fizikai Lápok, vol 38 (1931), 16-28.
- 70 Bribiesca, E. A Measure of Compactness for 3D Shapes. Comput. Math. Appl., 40 (10-11) (Nov.-dec. 2000) 1275-1284.
- 71 Babaud, J.; Witkin, A. P.; Baudin, M. and Duda, R.O. Uniqueness of the Gaussian Kernel for Scale-Space Filtering, IEEE Trans. Pattern Analysis and Machine Intelligence, vol 8, no. 1, (1986) 26-33.

- 72 Koenderink, J. J. The Structure of Images, Biol. Cybern., vol 50, (1984) 363-370.
- 73 Christopher, E. Reid., Thomas B. Passin. Signal Processing in C. Wiley, 324pp (1992).
- 74 Bridgman, P. W. Dimensional Analysis. New haven: Yale University, 113pp (1931).
- 75 Whitley, D., Starkweather, T., and Fuquay, D'A., Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator, pp. 133-140.
- 76 Lidd, M. L., Traveling Salesman Problem Domain Application of a Fundamentally New Approach to Utilizing Genetic Algorithms, Technical Report, MITRE Corporation, (1991).
- 77 Goldberg, E., David., Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley Publishing Company, 412pp (1989).
- 78 Michalewicz, Z. Genetic Algorithms + Data Structures = Evolution Programs. Springer, 387pp (1996).