

UNIVERSIDAD DE GRANADA
FACULTAD DE CIENCIAS
DEPARTAMENTO DE GEOMETRIA Y TOPOLOGIA

Memoria sobre:

ADELGAZAMIENTOS EN TOPOLOGIA DIGITAL

Realizada por el
Ldo.D. Juan T. Lopez Raya
para optar al grado de
Doctor en Ciencias Matemáticas,
bajo la dirección del
Dr.D. Ceferino Ruiz Garrido.

A mi hijo, Juan Luis.
A mi esposa, M^a Nati

Deseo manifestar públicamente mi agradecimiento a todas las personas que han ayudado y animado durante la preparación de esta memoria. Especialmente quiero agradecer la dirección científica y su entrañable apoyo humano a mi director el Profesor Dr. Ceferino Ruiz Garrido por las enseñanzas, ideas y buenos consejos que él me ha aportado. De igual forma, quiero resaltar la valiosa ayuda que he recibido del Profesor Dr. Alberto Márquez Pérez cuyo dinamismo, capacidad de organización y talla científica y humana han posibilitado la obtención de numerosos resultados que se exponen en esta memoria.

También quiero hacer patente mi agradecimiento al Departamento de Geometría y Topología de la Universidad de Granada y especialmente a su Director el Profesor Dr. Juan de Dios Pérez Jiménez, de la misma forma quiero manifestar mi gratitud a mis compañeros del Área de Geometría y Topología de la Universidad de Almería, de forma especial al Profesor Francisco García Arenas por las interesantes conversaciones mantenidas con él al inicio de este trabajo.

**ADELGAZAMIENTOS EN TOPOLOGIA
DIGITAL**

ADELGAZAMIENTOS EN TOPOLOGIA DIGITAL

TESIS DOCTORAL

**Realizada en el Departamento de Geometría y Topología,
por Dn. Juan T. López Raya,
bajo la dirección del Dr. Dn. Ceferino Ruiz Garrido,
y presentada en la Facultad de Ciencias de la Universidad de Granada.**

Esta Tesis fue defendida el día 18 de Febrero de 1994 ante el Tribunal compuesto por:

PRESIDENTE: Prf. Dr. Dn. Juan de Dios Pérez Jiménez, Prf. Titular y Director del Departamento de Geometría y Topología de la Universidad de Granada.

VOCALES: Prf. Dr. Dn. Tomás Recio Muñiz, Catedrático de Algebra de la Universidad de Cantabria.

Prf. Dr. Dn. Alberto Márquez Pérez, Prf. Titular de Geometría y Topología de la Universidad de Sevilla.

Dr. Dn. Cesar Carmona Moreno, Investigador del Centre Nationale de Etudes Spatiales, Toulouse, Francia.

SECRETARIO: Prf. Dr. Dn. Antonio Martínez López, Prf. Titular de Geometría y Topología de la Universidad de Granada.

y calificada con Apto cum laude por unanimidad.

INDICE

INTRODUCCION	1
CAPITULO I. PRELIMINARES.	7
1.1 Introducci3n.	8
1.2 Im3genes Digitales.	11
1.3 Topologfa Digital.	15
CAPITULO II. FUNDAMENTOS.	28
2.1 Introducci3n.	29
2.3 Conjuntos Parcialmente Ordenados.	
Espacios Celulares y Grafos.	34
2.4 CW-Complejos. Complejos Regulares.	51
2.4 Conclusiones.	58
CAPITULO III. ADELGAZAMIENTO DE I.D.B.	68
3.1 Introducci3n.	69
3.2 Un Criterio para el Adelgazamiento de Im3genes Digitales Binarias.	74
3.3 Un Algoritmo No Iterativo para la obtenci3n del Esqueleto.	80

CAPITULO IV. ADELGAZAMIENTO 3-DIMENSIONAL.	85
4.1 Introducción.	86
4.2 Preservación de la Topología en 3D2.	88
4.3 Adelgazamientosde I.D.B. tridimensionales.	90
CAPITULO V. IMAGENES NO BINARIAS.ADELGAZAMIENTOS.	95
5.1 Introducción.	96
5.2 Color. Transformaciones continuas de color.	100
5.3 Resultados Principales.	108
APENDICE.	123
BIBLIOGRAFIA.	166

INTRODUCCION

INTRODUCCION

Esta memoria está dedicada, fundamentalmente, a exponer resultados originales acerca de la operación de adelgazamiento en Imágenes Digitales, dentro del marco formal de la Topología Digital.

Para obtener estos resultados desde un marco teórico consistente (la topología) y enlazar con sus aplicaciones en el tratamiento de imágenes (topología digital), ha sido necesario, en primer lugar, fijar un modelo de este marco teórico que, sin ser igual a los que proponen los distintos autores que han tratado el tema, "armoniza" con los modelos teóricos actuales y contiene a los modelos clásicos; ligados éstos más a la implementación práctica de algoritmos. Por otra parte, para las imágenes no binarias, ha habido que construir un modelo, ya que estas imágenes no habían sido consideradas aún en topología digital. Este modelo es una generalización del propuesto para las imágenes binarias o monocromas.

Como norma general hemos pretendido abordar todos los aspectos relativos a la operación de adelgazamiento, desde su fundamentación teórica en una topología adecuada

INTRODUCCION

hasta su implementación práctica. En este sentido, se ha pretendido conseguir los resultados principales en todos los niveles posibles. Así, algunos de éstos aparecen como un resultado general, un teorema enunciable en el lenguaje clásico de grafos, un algoritmo y, finalmente, implementado en un lenguaje de programación en el apéndice de esta memoria.

Esta memoria está estructurada en los cinco capítulos que a continuación explicamos.

1- El Capítulo I está dedicado, en primer lugar, a situar la topología digital dentro del campo del tratamiento de imágenes y después a dar un resumen recopilatorio de los antecedentes de la topología digital y la operación de adelgazamiento. Aquí se exponen la notación y terminología clásicas y algunos resultados básicos en este campo.

2- En el capítulo segundo se establecen los fundamentos teóricos que se han elegido para basar la topología digital. El objetivo de este capítulo es, como se anuncia en el primero, el de establecer unos principios que nos permitan asociar a una imagen digital un espacio topológico de forma que este espacio tenga una relación directa con el espacio del que procede dicha imagen y dar unos métodos de cálculo en esta topología que sean apropiados para su uso en un ordenador.

Los resultados y conceptos topológicos que se manejan están tomados de la literatura clásica. Además, el modelo de espacio que se propone coincide en sus aspectos fundamentales o es equivalente a otros modelos aparecidos recientemente y que

INTRODUCCION

referenciamos en la bibliografía. Sin embargo, frente a otros modelos presenta la ventaja de considerar la modelización matemática del proceso de captación de una imagen dentro del propio modelo; la de establecer una equivalencia funtorial entre la topología de un objeto continuo del espacio euclídeo y la topología asociada a su imagen digital, lo que nos permite hablar con propiedad del "análogo continuo"; y la de obtener una subcategoría de conjuntos parcialmente ordenados, que probamos ser equivalente a cierta clase de espacios celulares, que resulta ser suficiente para nuestro estudio, respondiendo con ello a preguntas formuladas recientemente en la literatura y utilizando elementos de topología suficientemente generales y conocidos. Otra ventaja es la de englobar, mediante la asociación de un grafo, la "topología digital clásica". Debemos añadir que este modelo ha sido obtenido independientemente de otros modelos y que nuestra aportación ha consistido principalmente en dar un enfoque global de lo que hemos llamado topología digital, sin apartarse de la topología general, destacando los aspectos antes mencionados como parte de nuestro objetivo.

3- En el capítulo tercero se aborda el problema del adelgazamiento de imágenes digitales binarias de dimensión dos. En relación con este problema, que ha sido abundantemente tratado por distintos autores, se aporta en primer lugar un nuevo resultado que da condiciones necesarias para obtener aquellos puntos de la imagen que pueden ser borrados preservando la topología. Los cálculos necesarios para la implementación de este resultado han sido reducidos considerablemente con respecto a otros resultados conocidos hasta ahora. Teniendo este como referencia, se dan otros resultados que conducen a la obtención de un "método no iterativo para la determinación del esqueleto". Hemos de

INTRODUCCION

señalar que hasta ahora no hemos encontrado en la literatura ningún algoritmo que, en las condiciones de preservar la topología y de un comportamiento isotrópico, no necesite ser reiterado para conseguir el mismo fin. Este algoritmo trabaja, en el peor de los casos, en tiempo lineal, pudiendo ser implementado en paralelo (en un ordenador celular) y trabajar en un tiempo constante.

4- El Capitulo IV es una continuación y generalización del anterior para el caso tridimensional. Aquí revisamos los criterios para la preservar la topología en dimensión tres y después de elegir el criterio más consecuente con el marco teórico establecido, vemos que los métodos en topología digital desarrollados en el capítulo anterior son susceptibles de ser generalizados para este caso, obteniéndose un algoritmo para la obtención del esqueleto que trabaja en tiempo constante y que es el primero del que tenemos noticia en esta dimensión.

5-El Capítulo V está dedicado a estudiar las imágenes digitales no binarias y la operación de adelgazamiento en estas imágenes. Aquí se da el primer planteamiento, del que tenemos noticia en topología digital, sobre imágenes no binarias. Después de abordar el problema de dotar de una topología y de su grafo correspondiente a estas imágenes, se trata de clarificar el concepto de adelgazamiento en esta situación, viéndose que este concepto no se puede asimilar al concepto topológico de retracción como habían pretendido algunos autores.

La solución que se le da al problema consiste en asociar a una imagen con $n + 1$

INTRODUCCION

colores un espacio topológico con una aplicación característica que determina la partición en los $n+1$ colores y dotar al conjunto de todas las posibles particiones (o de todas las posibles aplicaciones características) de una topología, de forma que el paso de una partición a otra mediante un camino continuo sea lo que llamamos transformación continua del color. Esta operación generaliza para el caso binario la operación de adelgazamiento y constituye un primer paso fundamental para el adelgazamiento de imágenes no binarias.

Para finalizar damos en un apéndice la implementación de los algoritmos propuestos en los capítulos III, IV y V en un lenguaje de programación.

CAPITULO I. PRELIMINARES.

1.1 INTRODUCCION

Los problemas que trataremos desde el capítulo 2 en adelante corresponden a cuestiones planteadas en Topología Digital, cuyos antecedentes introduciremos en el apartado tercero de este capítulo, y que tiene su origen en el campo del estudio de la Visión Artificial.

Antes de continuar con la ubicación de nuestro trabajo, y dado que la visión artificial trata de simular la visión humana, quizás sea interesante esbozar brevemente el proceso de la visión humana. Aunque estudiar este proceso no es parte de nuestro objetivo,

ni siquiera lo son la mayoría de los aspectos relacionados con la visión artificial, si puede ser ilustrativo de como interactúa la topología con el fenómeno de la visión. Muchos estaremos de acuerdo en que la topología no es lo que percibimos visualmente, sino que se trata, en primera instancia, de una serie de conceptos básicos mediante los cuales interpretamos las escenas que vemos. Después nos sirve para hacer razonamientos acerca del espacio percibido y de todos los posibles espacios imaginados.

En el aprendizaje académico, posiblemente el primer contacto con esta disciplina sea la topología usual de \mathbb{R}^2 o \mathbb{R}^3 , plano y espacio con el cual se modeliza el mundo que nos rodea (localmente, ..hasta donde llega la vista). La topología usual resulta ser la abstracción y elaboración lógica de aquellos conceptos básicos a los que aludíamos más arriba, con esto quiero evidenciar la identificación mental constante entre las estructuras dadas a las escenas "visibles", a las escenas captadas por la vista y al modelo euclídeo de dichas escenas, y no solo entre las estructuras sino entre ellas mismas.

En lo que sigue trataremos de introducir de forma esquematizada y muy básica algunas nociones relativas a la visión y a los sistemas de tratamiento de imágenes, sin pretender profundizar en alguna de estas materias pero acentuando aquellos detalles que, posteriormente, trataremos de modelizar o que están en la base de los problemas que estudiamos.

PRELIMINARES

Si bien son bastante conocidas desde hace algún tiempo las propiedades físicas de la luz, no lo es tanto el fenómeno de la visión. Es decir, de la percepción de dicha luz de forma directa desde fuentes auto luminosas o bien reflejada por otros objetos. Distintas cuestiones acerca de descriptores de la impresión de luz (brillo, saturación, contraste,..) han sido respondidas al menos parcialmente, pero lo que tal vez esté más relacionado con nuestro objetivo sea la "mecánica " de la formación de la imagen en la retina, o más exactamente un modelo suficientemente simplificado de ello. Esta simplificación habría de ser obligada, además de por el desconocimiento a un nivel más profundo del tema, por la adecuación al propósito de este capítulo. En este sentido se pueden consultar los libros [32], [27], Y para una mayor profundidad [4].

1.2 IMAGENES DIGITALES

Las imágenes que se consideraron al inicio de estos estudios fueron, principalmente, imágenes captadas por dispositivos generadores de señales que, posteriormente, eran tratadas y almacenadas en forma de matriz. Las cámaras o dispositivos de captación han experimentado una notable evolución en las últimas décadas. Le dispositivo más utilizado actualmente y cuyo funcionamiento se adapta bien a la modelización matemática desde el punto de vista de la Geometría y de la Topología es el denominado CCD (coupled charge device).

El dispositivo de carga acoplada CCD puede ser considerado como una superficie fotosensible compuesta por un elevado número de fotodiodos en formación matricial, que

PRELIMINARES

determinarán los elementos mínimos de la imagen (pixel), situada sobre una base de silicio aislada mediante dióxido de silicio, en la que las cargas eléctricas originadas por los fotones de la luz de la imagen, son almacenadas primero y desplazadas después para, finalmente, obtener la salida de video. La imagen se capta entera en la matriz bidimensional y cada sensor

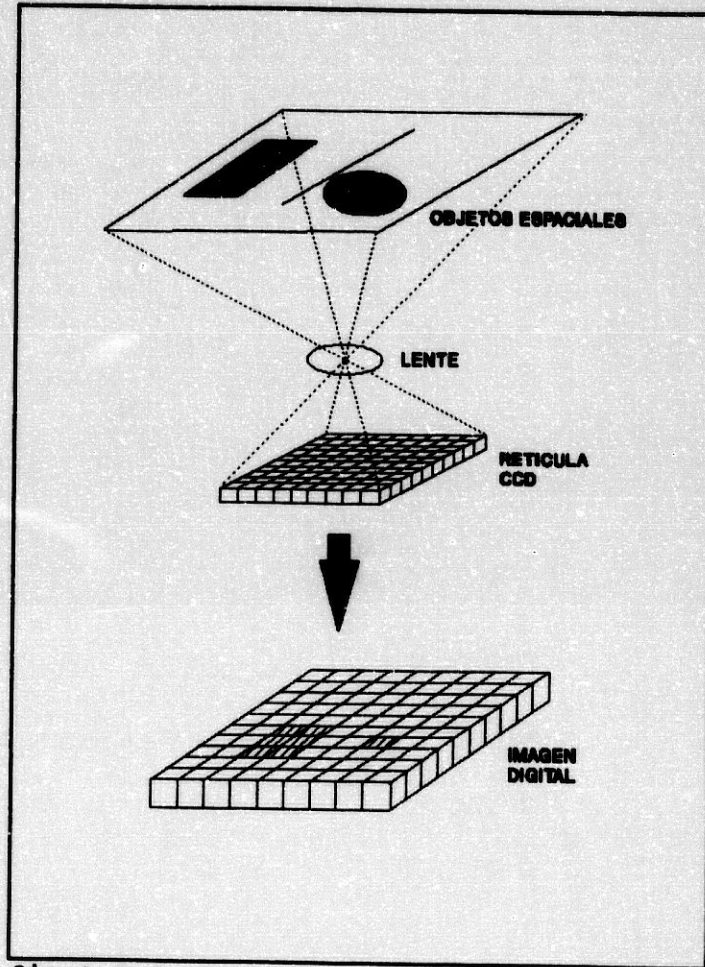


fig.1.2.1.

"ve" la luz incidente de un

área de la imagen fuente generando los electrones que se transmitirán en paralelo. Un esquema simplificado del funcionamiento de un sensor CCD se ilustra en la figura 1.2.1.

La luz captada y convertida en señal eléctrica por los sensores se expresa matemáticamente como una función de dos variables $f(x,y)$ que a cada punto del plano le hace corresponder un valor real (una n-tupla de valores reales en caso más general), y se

PRELIMINARES

anula fuera de una región acotada. Esta función que representa la luminiscencia de cada punto y que nosotros consideraremos como el "nivel de gris", necesita ser digitalizada para su tratamiento en un ordenador. La digitalización consiste en un proceso (muestreo y análisis estadístico) mediante el cual se divide la escala de grises en p intervalos y se le asigna a cada elemento de una matriz de $n \times m$ un valor entre 0 y p . El resultado de esta operación será nuestro primer concepto de imagen digital que llamaremos binaria, cuando $p = 1$.

Una imagen digital contiene distintas informaciones más o menos relevantes para el propósito de su estudio y para cada aplicación conviene acceder a la información deseada ignorando la restante, para ello es necesario transformar la imagen eliminando ruidos, segmentándola por niveles de gris y otras operaciones previas a su análisis. Al conjunto de todas esas operaciones se les llama tratamiento, manipulación o procesamiento de imágenes.

El análisis de imágenes está situado en las últimas fases de la extracción de información de una imagen digital, que puede ser datos numéricos u otras determinadas características, por medio de métodos automáticos. En la literatura, se distingue esta extracción de datos, de otro tipo de procesamientos de imágenes en los que se pretende obtener una imagen modificada desde una imagen primitiva, realizándola, restaurándola, suavizando contrastes... etc. . Dentro del campo del análisis de imágenes hay que situar

también las operaciones que conservan la información objeto del estudio.

Las características en las que centraremos este trabajo serán las propiedades topológicas de una imagen digital. Será entonces necesario establecer unos principios que nos permitan asociar a una imagen digital un espacio topológico de forma que este espacio tenga una relación directa con el espacio del que procede dicha imagen. Es decir se trata de estudiar este problema dentro de la topología como campo formalmente establecido, más que de desarrollar conceptos topológicos para el caso específico de una imagen digital, aunque sí será necesario desarrollar técnicas particulares y métodos de cálculo específicos para esta nueva situación.

1.3. TOPOLOGIA DIGITAL

Como ya hemos dicho más arriba la topología digital no debe de ser un cuerpo de doctrina distinto del que se conoce como topología (fundamentalmente topología general y algebraica). El nacimiento de esta disciplina surge como respuesta a la necesidad de introducir y manejar conceptos topológicos en unas situaciones muy concretas y con unas limitaciones dadas en cada momento por la naturaleza de los sistemas de adquisición de imágenes y por la arquitectura de las máquinas en las que se hacen los cálculos. Es destacable el hecho de que la mayoría de estos conceptos, introducidos en principio para dar respuestas prácticas, hayan podido ser justificados desde un punto de vista teórico. Es admirable la poderosa intuición de muchos autores, entre ellos destacamos a A. Rosenfeld, cuyos trabajos [30] [31] [18] han sido la base de la topología digital, en los que se han introducido conceptos que son imprescindibles, en primer lugar, para abstraer los problemas hasta el marco teórico de la topología y, después, para regresar desde este marco hasta la viabilidad de los cálculos y su utilidad práctica, y en los que se ha

PRELIMINARES

observado una gran precaución para detectar y esquivar paradojas a la vez que preocupación por la sencillez de las operaciones. A continuación damos los preliminares de lo que se ha llamado topología digital y que se encuentran , especialmente en " Digital Topology: Introduction and Survey" de Kong y Rosenfeld [18].

Los elementos integrantes de las imágenes que se considerarán en principio estarán dispuestos en retículas o matrices binarias ilimitadas, es decir sus elementos tendrán valor 1 o 0 según corresponda a un objeto representado en la imagen o a su complementario, y su dimensión será 2 o 3. Cada elemento de la matriz suele llamarse "pixel" o "voxel", según que estemos modelando el plano o el 3-espacio.

Asociando a cada elemento de la retícula sus coordenadas estamos en condiciones de introducir el importante concepto de la adyacencia entre pixeles o voxeles . Dos elementos en el plano se dicen 8-adyacentes si las coordenadas de un punto difieren de las coordenadas del otro a lo sumo en una unidad; dos puntos se dicen 4-adyacentes si son 8-adyacentes y difieren a lo sumo en una de sus coordenadas. De igual forma se definen entre puntos del 3-espacio la 26-adyacencia, la 18-adyacencia y la 6-adyacencia según que sus coordenadas distintas difieran a lo sumo en una unidad y estas sean a lo sumo en tres, dos o una, respectivamente. Estos conceptos están aclarados en las figuras desde la 1.3.1 hasta la 1.3.5.

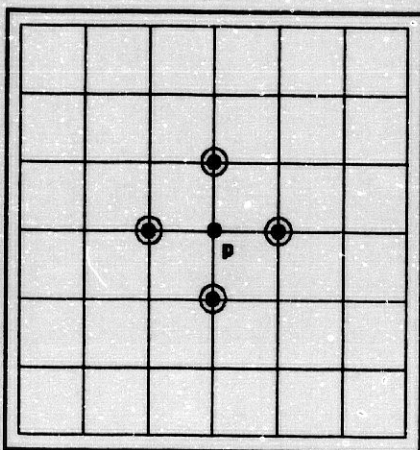


fig 1.3.1. 4-adyacencia

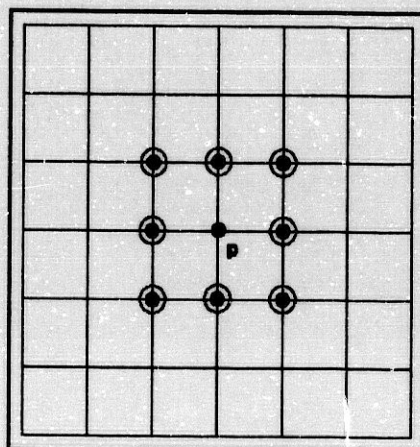


fig 1.3.2. 8-adyacencia

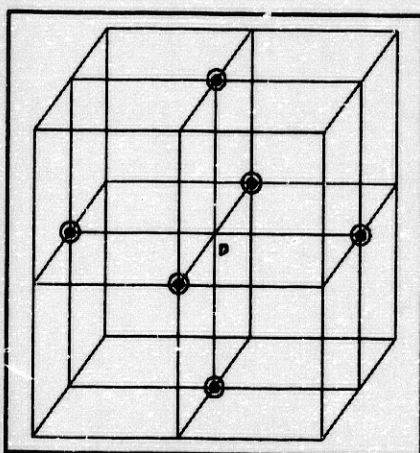


fig. 1.3.3 6-adyacencia

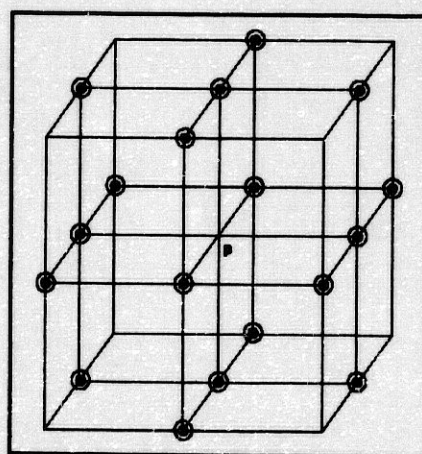


fig 1.3.4 18-adyacencia

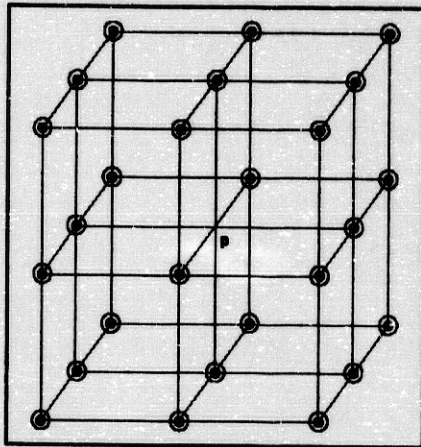


fig. 1.3.5. 26-adyacencia

Un "n-vecino" de un punto p en la retícula es un punto q que es n -adyacente a p , con $n = 4, 8, 6, 18, 26$. Dos conjuntos de puntos A y B se dirán n -adyacentes si existen puntos p y q de A y B respectivamente que son n -adyacentes. De esta forma podemos decir que un conjunto C es "n-conexo" (esta expresión no tendrá ambigüedad dentro de este contexto) si no puede dividirse en dos subconjuntos que no son n -adyacentes el uno del otro. La figura 1.3.6. muestra un conjunto de puntos que es 8-conexo y en la figura 1.3.7. este mismo conjunto vemos que tiene 4 componentes 4-conexas.

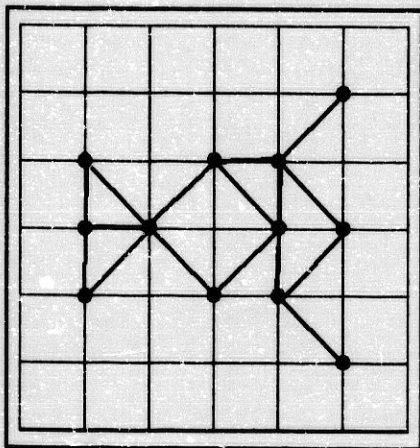


fig. 1.3.6.

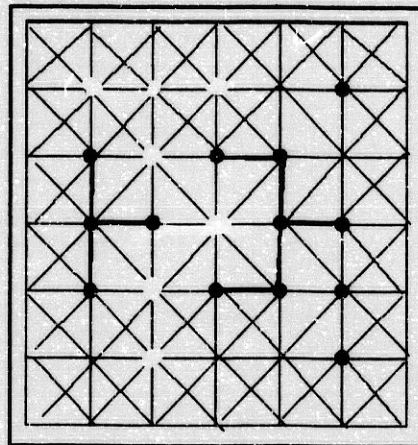


fig. 1.3.7.

Notaremos por $N(p)$ al conjunto de los 8-vecinos de p si p es un punto del plano, o al conjunto de los 26-vecinos de p si consideramos a p en el 3-espacio.

Un punto de la retícula asociado con un pixel o un voxel que tiene valor 1 lo llamaremos un punto "negro", en otro caso se dirá un punto "blanco". Motivado por la aparición de una paradoja que expondremos a continuación, se inicia el estudio e la topología digital haciendo uso de distintos tipos de adyacencia para los puntos negros y los puntos blancos, desarrollado por A. Rosenfeld y sugerido por Duda, Hart y Munson [30] [31] [7].

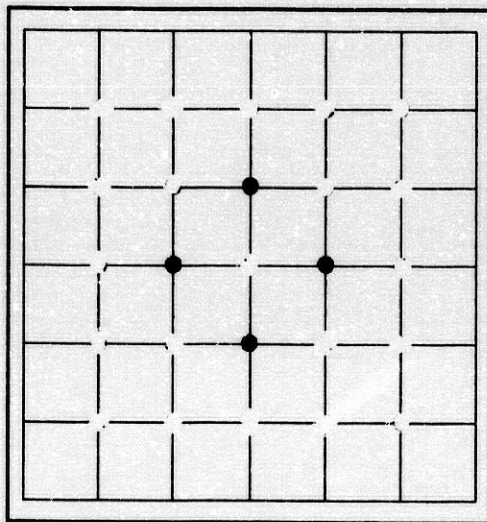


fig. 1.3.8.

En la figura 1.3.8. se ilustra una situación en la que, si consideramos el mismo tipo de conexión para todos los puntos, se obtienen situaciones paradójicas. Considerando 8-conexión tenemos una componente conexa de puntos blancos y otra componente de puntos negros que representan el análogo digital de una curva de Jordan y que sin embargo no separa el plano en dos componentes conexas. De otro, lado teniendo 4-conexión para todos los puntos tendríamos que los puntos negros forman un conjunto totalmente desconexo, cada punto es una componente conexa, que separa el plano en dos componentes conexas. Otras situaciones igualmente paradójicas podríamos encontrar en el 3-espacio si tomamos el mismo tipo de conexión para todos los puntos.

Introduciendo estos conceptos "topológicos" en el concepto de imagen se obtuvo la siguiente definición de imagen digital que convino a los propósitos de la nascente topología digital y que, como veremos más tarde, es generalizable dentro del contexto formal de la topología.

Definición 1.3.1. Una Imagen digital es una cuadrupla (V, m, n, B) , donde $V = Z^2$ o $V = Z^3$, $B \subset V$, y donde $(m, n) = (4, 8)$ o $(m, n) = (8, 4)$ si $V = Z^2$, y (m, n) es $(6, 26)$ o $(26, 6)$ o $(6, 18)$ o $(18, 6)$ si $V = Z^3$.

Diremos que la imagen es n -dimensional, $n=2$ o $n=3$, si $V = Z^n$. A los puntos de B los llamaremos puntos negros y a los de $V - B$ puntos blancos. Dos puntos negros se dicen adyacentes si son m -adyacentes y dos puntos blancos o un punto blanco y uno negro se dicen adyacentes si son n -adyacentes. Llamaremos adyacencia al segmento que une dos puntos adyacentes. De lo anterior se deduce que podemos mirar una imagen digital como un grafo donde los nodos son los elementos de la imagen (píxeles o Voxeles) y las aristas son las adyacencias. Esta noción fue introducida por Rosenfeld [30] en los años setenta y ha estado vigente hasta el principio de los noventa.

Continuamos con el traslado de conceptos topológicos hacia el campo del tratamiento de imágenes que, como ya hemos dicho, han realizado de forma meritoria y con un notable esfuerzo algunos profesionales del proceso de imágenes y de las ciencias de la computación y que será necesario para comprender los objetivos, las operaciones y

PRELIMINARES

también las limitaciones del embrión de lo que será la topología digital . Para cualquier conjunto de puntos S un camino en S es una secuencia $\{p_i, 0 \leq i \leq n\}$ de puntos en S tal que p_i es adyacente a p_{i+1} para todo $0 \leq i \leq n-1$. Tal camino se dice entre p_0 y p_n y se dice cerrado si ambos coinciden. Claramente la conexión definida antes coincide con la que podría definirse mediante estos caminos.

En una imagen digital (V, m, n, B) un conjunto conexo X se dice que "envuelve" a un conjunto Y si Y está contenido en una componente conexa finita de $V - X$; la relación binaria que se desprende de esta definición es una relación de orden parcial. Cuando X es una componente de puntos negros que envuelve a un conjunto de puntos blancos Y adyacente a él, se dice que Y es un "agujero" o una "cavidad" según que estemos en dimensión dos o tres. En [3] Buneman introduce el "árbol de adyacencias" para una imagen digital como un grafo cuyos vértices representan a las distintas componentes conexas, blancas o negras, de la imagen, donde dos vértices están unidos por una arista si y solo si las correspondientes componentes son adyacentes una de la otra.

Un punto negro se dirá punto borde si es adyacente a algún punto blanco, en otro caso se llamará punto interior. El borde de una componente conexa negra es el conjunto de todos sus puntos bordes. Obsérvese que la definición de punto borde tiene aquí sentido solo para los puntos negros y no para los puntos blancos, esto sugiere que en algún sentido se está pensando que los puntos negros serían un "conjunto cerrado", si bien el marco

teórico en el que estamos no nos permitiría identificar el interior del conjunto de puntos blancos con los puntos blancos.

A fin de resolver problemas relacionados con la detección de objetos y otras operaciones previas al reconocimiento de formas, Rosenfeld [30] estudió los conceptos de punto borde y de frontera tratando de adecuar tanto como fuera posible estos conceptos a sus homólogos en topología y de forma que sirvieran a sus propósitos prácticos. En este sentido se vio la necesidad de que la frontera de una componente conexa sin agujeros fuera conexa y que además pudiera expresarse como un arco cerrado (un lazo) de tal forma que fuese obtenible mediante un algoritmo que se llamó "seguimiento de bordes" (border following)[30].

Otra aproximación a este concepto fue la dada mediante "aristas". Un par ordenado (p,q) donde p es un punto negro y q es un punto blanco que es 4-adyacente se llama una arista. Este nombre lo recibe al poder ser considerada como la arista común entre los pixeles representados por p y q . De esta forma el concepto de frontera estaría más próximo al usual, sin embargo la representación de la frontera estaría dada por un par de curvas (una bicurva) [31].

Uno de los puntos fundamentales del trabajo que sigue está centrado en torno a una operación que resulta necesaria en el análisis de imágenes y en la que interesa preservar las propiedades topológicas de la imagen. Esta operación consiste en el "borrado de puntos

negros" o dicho de otra forma en el cambio de color de puntos negros de los objetos a puntos blancos del fondo.

Definición 1.3.2 Sean $\phi = (V, m, n, B)$ y $\phi' = (V, m, n, B')$ imágenes digitales. Decimos que ϕ' ha sido obtenida de ϕ mediante el borrado de los puntos de D si $B' \subset B$ y $D = B - B'$.

Reservaremos el termino "adelgazamiento" para una operación tal que mediante el borrado sucesivo de puntos en B se obtiene un subconjunto minimal de B de forma que se mantienen las propiedades topológicas de la imagen a través de la operación.

Hasta ahora las propiedades topológicas de la imagen se han expresado en términos de conexión. En este sentido, para imágenes 2-dimensionales planas, Stefanelli y Rosenfeld [35] [18] establecieron el siguiente criterio:

Criterio 1.3.1. Sea $\phi = (Z^2, m, n, B)$ una imagen digital 2-dimensional. Entonces el borrado de los puntos de un subconjunto D de B preserva la topología si y solo si:

- 1.- Cada componente conexa negra de ϕ contiene exactamente una componente conexa negra de la imagen transformada, y
- 2.- Cada componente conexa blanca de la imagen transformada contiene exactamente una componente conexa blanca de ϕ .

$(m,n) = (8,4)$. Sea $C(\rho) = C_0 \cup C_1 \cup C_2$.

$C(\rho)$ se ilustra en las figuras 1.3.9 y 1.3.10. Las características de Euler serían respectivamente 0 y 2.

Proposición 1.3.1 $C(\rho)$ es un poliedro plano.

Definición 1.3.3 El poliedro $C(\rho)$ se llama poliedro asociado a la imagen ρ .

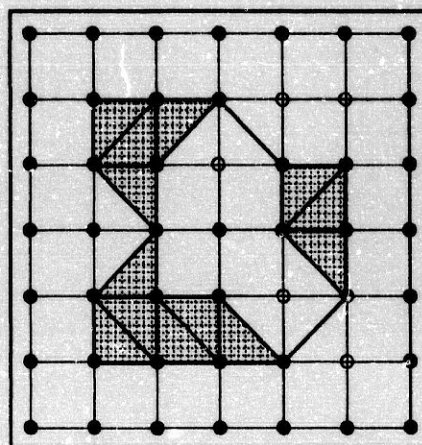


fig 1.3.9.

De esta forma surge la primera asociación de una imagen con un poliedro, más adelante veremos que desde el punto de vista del rigor no es completamente satisfactoria. No obstante estamos obligados a mencionarla ya que constituye el germen de todo el desarrollo posterior y los cálculos que con esta asociación se realizan coincidirán con otros realizados desde un marco teórico riguroso.

Distintos autores han dado fórmulas para el cálculo de esta característica en imágenes digitales, de ellas presentamos la siguiente debida a Kong

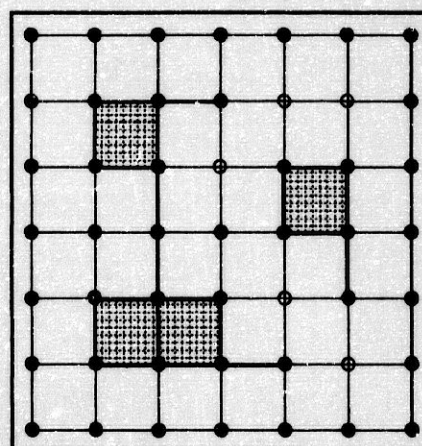


fig 1.3.10.

PRELIMINARES

(Tesis Doctoral [17]) que nos ha parecido la más adecuada a la línea de este trabajo y que es generalizable para un tipo de imágenes digitales en las que se consideran otro tipo de adyacencias.

Para una celda unitaria de dimensión dos o tres S , sean S^0 el conjunto de vértices de S , S^1 el conjunto de aristas de S y S^2 el conjunto de las caras de S si S es tridimensional. Entonces para un poliedro plano P o para un poliedro tridimensional se definen respectivamente:

$$\chi(P; S) = \chi(F \cap S) - \chi(F \cap S^1) / 2 - \chi(F \cap S^0) / 4$$

o

$$\chi(P; K) = \chi(F \cap K) - \chi(F \cap K^2) / 2 - \chi(F \cap K^1) / 4 - \chi(F \cap K^0) / 8$$

Consideraremos también imágenes digitales con otro tipo de adyacencia. Serán las imágenes digitales con la 6-adyacencia, en la que un punto p de coordenadas (x, y) se considerará adyacente a los puntos de coordenadas $(x-1, y-1)$, $(x, y-1)$, $(x-1, y)$, $(x+1, y)$, $(x, y+1)$ y $(x+1, y+1)$. No supone ningún problema considerar el mismo tipo de adyacencia para todos los puntos, blancos y negros, resultando ser una relación de adyacencia homogénea. Presenta un inconveniente de tipo estético al trabajar en una retícula ortogonal que desaparecería en un monitor con retícula hexagonal.

CAPITULO II FUNDAMENTOS.

En este capítulo se trata el problema de determinar cuales de los espacios parcialmente ordenados son apropiados para un planteamiento teórico de la topología digital. Daremos la equivalencia de esta subfamilia de CPO con una parte de los espacios celulares y la equivalencia de estos con los poliedros generalizados embebidos en el espacio euclídeo. Asociaremos a cada uno de estos CPO o espacios celulares un grafo que generalizará los tipos de adyacencias dados en el capítulo de preliminares.

Se ha pretendido que este capítulo sea lo más autocontenido posible, al menos en lo concerniente a resultados sobre espacios finitos y a los aspectos topológicos que inciden en otras áreas del tratamiento de imágenes fuera de la topología, por ello se dan demostraciones de algunos resultados que son bastante conocidos en topología.

2.1. INTRODUCCION

Como ya hemos anunciado en el capítulo anterior, nuestro objetivo ahora es el de fundamentar el tratamiento topológico de las imágenes digitales, de manera que el modelo formal para este estudio esté en consonancia con el modelo utilizado en el estudio topológico de los objetos de los cuales provienen las imágenes digitales. Los objetos susceptibles de observación son, en primera instancia, subespacios del espacio

euclídeo tridimensional, de los que recibimos la luz reflejada o emitida por "recintos" de su superficie. En la imagen digital tenemos un conjunto finito de elementos primarios (pixel) que representan aquellas áreas emisoras de luz, que también son áreas emisoras de luz, y que sin embargo, al ser indivisibles, tienen un tratamiento informático como puntos.

En el artículo [21] Kovalevsky consideraba la necesidad de estudiar una topología consistente para conjuntos finitos y se lamentaba de que los topólogos no hubieran hecho ningún esfuerzo por implementar sus conocimientos en el procesamiento de imágenes y en otros campos de la computación.

Frente al "grafo de adyacencias" dado por Rosenfeld en [31], Kovalevsky propone como modelo teórico la Topología de los Espacios Celulares, donde desaparecen los problemas relativos a la definición de frontera, la paradoja de la curva de Jordan y se puede hacer uso con propiedad de las nociones de conjunto abierto, entorno, continuidad, etc.

En una imagen digital (plana) los pixel corresponderían a los elementos de dimensión dos, mientras que para los elementos 0-dimensionales y 1-dimensionales no hay una realización material que los soporte tanto en la memoria como en la presentación gráfica de una imagen. Este autor asegura la viabilidad de codificar los

elementos de baja dimensión (que puede demandar hasta una cantidad de memoria cuatro veces superior a la necesitada para codificar los pixel de la imagen) así como la posibilidad de construir el hardware correspondiente para este propósito . En cualquier caso la ventaja matemática que supone para este modelo incluir los elementos de dimensiones 0 y 1 conlleva una desventaja informática respecto al modelo de los grafos de adyacencias.

Por otra parte, si los objetos de la imagen están formados por el subespacio de pixel iluminados, o en la terminología del capítulo anterior puntos negros, se hace necesario decidir la pertenencia, o el color, de los vértices y de las aristas. Estaríamos ante un modelo en el que el usuario tiene que decidir la estructura adecuada para cada imagen en función de un conocimiento a priori de ella, esto supone una gran desventaja para la implementación del procesamiento automático, como señalaban E. Dominguez, A. Francés y A. Márquez en [5]. Kovalevsky propone en su artículo [21] dos reglas para una decisión global de la pertenencia. A continuación exponemos la "Regla de Etiquetado por el Máximo" (Maximun Label Rule) de la que el autor asegura, sin especificar nada más, que es una generalización de la propuesta de Rosenfeld de usar las 4-adyacencias para el fondo y las 8-adyacencias para los objetos. En el apartado segundo de este capítulo veremos, después de introducir un grafo asociado a un espacio celular, que esta regla no generaliza completamente el modelo de Rosenfeld si bien este modelo puede ser generalizado dentro del modelo de los espacios celulares.

Regla de Etiquetado por el Máximo Si suponemos que las etiquetas de los pixel pertenecen a un conjunto ordenado (como puede ser la escala de grises). Entonces cada elemento x de dimensión 0 o 1 toma una etiqueta igual al máximo de las etiquetas de los pixel a los que bordea.

En un meritorio afán de incluir la topología digital en el contexto de la topología general en [21] se recuerda que todo espacio celular es un espacio topológico T_0 y se da para ver el recíproco en el caso finito se da una definición de función dimensión que no resulta completamente satisfactoria para un modelo de imágenes digitales, este hecho fue destacado por G.T.Herman en [10] por lo que este autor cuestiona la conveniencia del modelo de los espacios celulares proponiendo como modelo los espacios finitos parcialmente ordenados, aunque también añade que quizás la clase de estos espacios sea demasiado grande e incluya espacios no aptos para el propósito de modelizar las imágenes digitales.

Un modelo propuesto por los profesores E. Dominguez, A. Francés y A. Márquez para el estudio de los objetos discretos, provenientes de una digitalización, es el de los "Espacios Digitales" [5]. Este concepto está basado en una restricción de la noción de proximidad, y resulta bastante apropiado para establecer relaciones entre los elementos de pantalla, sin tener que recurrir a otros elementos añadidos que harían dificultosos los cálculos. Entre los grafos obtenidos desde estos espacios, se encuentran el de las 8-adyacencias y el de las 26-adyacencias como casos particulares. También en [5]

aparece por primera vez rigORIZADO el concepto de "digitalización".

En lo que sigue nos proponemos responder a las cuestiones propuestas o suscitadas implícitamente en los artículos [10] [21], eligiendo aquellos espacios parcialmente ordenados a los que se le puede dotar de una función dimensión de tal forma que el espacio celular obtenido no chocha con nuestra intuición, más precisamente damos una relación biunívoca entre estos espacios celulares y los objetos n -dimensionales (poliedros) observables en un R^n . Además probamos que el modelo dado por Rosenfeld y otros modelos quedan generalizados por este mediante la asociación de grafos al conjunto de elementos de máxima (o de mínima) dimensión del espacio celular, evitándose de esta forma complicar excesivamente los cálculos en el procesado de imágenes digitales; al tiempo que se dispone de un modelo teórico suficientemente general, en el que se pueden utilizar todas las herramientas y conceptos que son propios de la Topología General y de la Topología Algebraica.

Este modelo lleva implícita una estructura algebraica, la de retículo, que nos permite el uso de la combinatoria y de la aritmética como herramientas muy útiles, tanto desde el punto de vista teórico como práctico.

Hemos puesto también un especial interés en establecer una relación directa entre objetos continuos y discretos.

2.2 CONJUNTOS PARCIALMENTE ORDENADOS. ESPACIOS CELULARES Y GRAFOS.

[1] [33] [2]. Un conjunto parcialmente ordenado es un par (A,R) donde A es un conjunto y R una relación de orden (orden parcial) en A . En lo que sigue se abreviará CPO (POSET en terminología anglosajona). Para cualesquiera dos elementos x e y de A tales $x R y$, se dirá que x bordea, cubre, es menor o igual o es una cara de y .

Un subconjunto S de A con el orden parcial inducido $R|_S$ es un subconjunto parcialmente ordenado. Un subconjunto parcialmente ordenado (subCPO en lo sucesivo) I de un CPO A se dirá un ideal de A si siempre que dos elementos de A x e y estén relacionados en A , $x R y$, y además ocurra que y pertenece a I , se tiene que x pertenece a I . En particular el conjunto $A_{R_x} = \{y \in A; y R x\}$ se llama ideal principal generado por x . Si $x R y$, al subCPO $[x,y] = \{z \in A; x R z R y\}$ lo llamaremos intervalo cerrado determinado por x e y .

Un CPO (A,R) se dirá acotado si existe un elemento $\hat{0}$ que llamaremos ínfimo y un elemento $\hat{1}$ que llamaremos supremo de tal forma que para todo elemento x perteneciente a A se tiene que $\hat{0} R x R \hat{1}$. Dados dos elementos x e y de un CPO acotado se define el supremo y el ínfimo de $\{x,y\}$ como la mayor y la menor de las cotas, respectivamente, de $\{x,y\}$. En un CPO acotado en el que para cualesquiera dos elementos existe el supremo y el ínfimo, se obtienen dos operaciones internas que lo dotan de estructura de RETICULO.

Una cadena en (A,R) es un subconjunto de A totalmente ordenado. Se define la subdivisión baricéntrica de un CPO (A,R) , y se notará por $sd(A)$, como el conjunto de todas las cadenas finitas y no vacías de A , ordenadas por inclusión. Los morfismos que consideraremos entre dos CPO A y B serán las aplicaciones f que preservan el orden, estas aplicaciones inducen morfismos de retículo si A y B lo son. Es también inmediato definir la subdivisión baricéntrica de una tal aplicación f , $sd(f)$, entre $sd(A)$ y $sd(B)$.

Todo CPO (A,R) tiene asociado de forma natural un dual $(A;R^*)$, donde R^* se define mediante R como $\forall x,y \in A \quad x R^* y$ si y solo si $y R x$.

Definición 2.2.1 Un Espacio Celular es una terna de la forma (A,R,dim) donde (A,R) es un conjunto parcialmente ordenado y "dim" es una aplicación de A en el conjunto de los números naturales tal que para cualesquiera

elementos x e y de A con $x R Y$ se tiene que $\dim(x) \leq \dim(y)$.

A la imagen por la función \dim de cualquier elemento x , $\dim(x) = n$, la llamaremos dimensión de x y diremos que x es una n -celda. Un espacio celular A se dirá k -dimensional o k -espacio si $\dim(x) \leq k \quad \forall x \in A$. Al conjunto $\{ x \in A; \dim(x) \leq k \}$ lo llamaremos el k -esqueleto de A .

Las aplicaciones entre espacios celulares que preservan el orden las llamaremos aplicaciones celulares. Dos espacios celulares se dirán isomorfos si entre ellos existe una aplicación celular biyectiva f tal f^{-1} es también celular. Todo espacio celular de dimensión finita tiene asociado de forma natural un espacio celular dual.

Mediante el funtor de olvido tenemos que la categoría de espacios celulares se puede llevar a la de conjuntos parcialmente ordenados, por otra parte basta asignar una dimensión de forma trivial ($\dim(x) = n$, n fijo) para considerar cualquier CPO como un espacio celular. Sin embargo para que la estructura de espacio celular sea para nosotros más interesante que la de CPO convendrá que la función dimensión tenga un significado topológico y además que, si pretendemos modelar objetos del espacio continuo, los valores de esta función no choquen con nuestra intuición. Sobre este aspecto volveremos más tarde.

Veremos a continuación que estas estructuras (casi equivalentes) equivalen a la estructura de ciertos espacios topológicos.

Sea (A, R) un CPO y x un elemento de A . Consideramos el conjunto $B(x) = \{y \in A; x R y\}$. Se tiene:

1) $\forall x \in A, x R x$, y por tanto $x \in B(x)$, así pues $\bigcup_{x \in A} B(x) = A$

2) Si $\exists x, y \in A; B(x) \cap B(y) \neq \emptyset$ entonces $\forall z \in B(x) \cap B(y)$ se tiene que $x R z, y R z$ y por tanto $\forall h \in A$, tal que $z R h$ es $x R h, y R h$ luego $B(z) \subset B(x) \cap B(y)$.

De esta forma la familia $\{ B(x); x \in A \} \cup \{ \emptyset \}$ es base de una topología y podemos asociar al CPO (A, R) un espacio topológico (A, T) relacionado directamente con su estructura de orden.

Dada una aplicación f que preserva el orden entre los CPO (A, R) y (A', R') , si consideramos f como una aplicación entre los espacios topológicos asociados (A, T) y (A', T') , f resulta claramente continua. Por otra parte si x e y son elementos de A tales que $x R y$, y f es una aplicación entre A y A' tal que $f(x)$ no está relacionado con $f(y)$ entonces $f(y) \notin B(f(x))$, lo que equivale a que $y \notin f^{-1}(B(f(x)))$ y por tanto $B(x) \not\subset f^{-1}(B(f(x)))$ y f no sería continua. Luego toda aplicación continua entre los espacios topológicos asociados (A, T) y (A', T') es una aplicación que conserva el orden entre (A, R) y (A', R') .

Parece claro que cualquier concepto topológico que se de en (A,R) tenga un concepto homólogo en (A,T) y recíprocamente. En el capítulo I se daba el concepto de conexión en imágenes digitales mediante unos subconjuntos de puntos verificando unas relaciones de adyacencia entre ellos y que llamábamos arcos . En espacios celulares o en CPO se puede dar un concepto análogo de la siguiente forma:

Definición 2.2.2. Una sucesión de elementos $L = \{x_1, \dots, x_n\}$ de un CPO (A,R) se dirá un arco entre x_1 y x_n si y solamente si para cualesquiera dos elementos $x_i, x_j \in L$, con $j = i+1$ ó $j = i-1$ es $x_i R x_j$ o $x_j R x_i$.

Definición 2.2.3. Un subconjunto $B \subset A$ se dirá conexo si para cualesquiera dos elementos x, y de B , x e y se pueden unir por un arco en B .

Proposición 2.2.1. Sea (A,R) un CPO y (A,T) su espacio topológico asociado. Sea $B \subset A$ y A numerable. Se tienen:

- i) La conexión de B en el sentido de la definición 2.2.3 equivale a la conexión de B como subespacio topológico de (A,T) .
- ii) Los arcos introducidos en la definición 2.2.2 para (A,R) coinciden con los arcos como aplicaciones continuas $\alpha : [0, 1] \rightarrow A$ para (A,T) .
- iii) La conexión en (A,T) equivale a la conexión por arcos.

En efecto:

Sea B conexo en el sentido de la definición 2.2.3., veamos que B con la topología restringida de (A, T) es conexo.

Supongamos que B lo podemos expresar como $B = O_1 \cup O_2$ donde O_1 y O_2 son abiertos y tales que $O_1 \cap O_2 = \emptyset$. Sean x e y con $x \in O_1$, $y \in O_2$ y $L = \{x = x_1, \dots, x_n = y\}$ un arco uniendo entre x e y . Tomamos $L \cap O_1 = \{x_1, \dots, x_i\}$ y $L \cap O_2 = \{x_{i+1}, \dots, x_n\}$, x_i y x_{i+1} han de estar relacionados y podemos suponer que $x_i R x_{i+1}$ de donde se tiene que O_1 no contiene a ningún abierto básico que contenga a x_i y por tanto no puede ser abierto.

Supongamos ahora que (A, T) es conexo, sea $x \in A$ y sea S el subconjunto de A formado por los puntos que se pueden unir por un arco a x en (A, R) . Veamos que $S = A$.

$S \neq \emptyset$ ya que $x \in S$. S es un conjunto abierto ya que para todo elemento y perteneciente a S es $B(y) \subset S$. Por otra parte si y pertenece a la adherencia de S ($\text{cl}(S)$ en lo sucesivo) es $B(y) \cap S \neq \emptyset$ y $\exists z, z \in B(y) \cap S$ de forma que x se puede unir con y mediante un arco $\{x, \dots, z, \dots, y\}$. Luego $y \in S$ y por tanto S es cerrado, de donde resulta que $S = A$. De esta forma queda probado el aserto i)

Sea ahora $\alpha: [0, 1] \rightarrow A$ un arco en (A, T) y sea $\alpha([0, 1]) = S \subset A$, se tiene que S es conexo y sea L un arco (en el sentido de la definición 2.2.2) maximal en S . Si se tiene que $L \neq S$ existiría $x \in S$ tal que $x \notin L$. Pero S es conexo y esto, por

i), equivale a que x se pueda unir a cualquier punto de L , podemos obtener así un arco contenido en S y que contiene propiamente a L , por tanto L no sería maximal. Queda entonces probado que $L=S$.

Consideramos ahora un arco $L=\{x=x_1, \dots, x_n=y\}$ en (A,R) . Es fácil ver, por inducción sobre n , que L coincide con la imagen por una aplicación continua del intervalo $[0,1]$ en (A,T) .

Si $n = 2$ se tiene que $x R y$ (a) o bien $y R x$ (b), para caso tomamos una aplicación f desde $[0,1]$ definida :

a) $f(0) = x$, $f(1) = y$.

b) $f(0) = x$, $f(1) = y$.

En ambos casos la aplicación es claramente continua y desde aquí es inmediato probar que cualquier arco L de cualquier longitud es la imagen de un arco en (A,T) .

A partir de i) y de ii) se tiene iii). ■

Hasta ahora hemos visto cómo a todo CPO o a todo espacio celular se le puede asociar un espacio topológico determinado por su estructura de orden. Observando las características de estos espacios topológicos podemos establecer una correspondencia recíproca que nos conduciría a un isomorfismo de categorías. Estas características se resumen en la siguiente proposición.

Proposición 2.2.2 Para cualquier CPO (A,R) con una relación R (no trivial), el espacio topológico asociado (A,T) es T_0 , no es T_1 y la intersección arbitraria de abiertos es un abierto.

En efecto:

El hecho de ser T_0 se desprende de la antisimetría de la relación R . Para ver que no es T_1 basta elegir dos elementos x e y tales que $x R y$, aquí no es posible encontrar un abierto que contenga a x y no contenga a y . Por otra parte, para cualquier elemento x de A y cualquier abierto O de T conteniendo a x , se tiene que $x \in B(x) \subset O$.

Entonces, para cualquiera que sea la familia $\{O_i\}$ con $i \in I$ de abiertos, y para todo

$$x \in \bigcap_{i \in I} O_i \text{ es } B(x) \subset \bigcap_{i \in I} O_i \text{ y por tanto } \bigcap_{i \in I} O_i \in T. \blacksquare$$

Veamos que la proposición anterior caracteriza a la topología proveniente de un CPO.

Proposición 2.2.3. Sea (A,T) espacio verificando:

- i) (A,T) es T_0
- ii) La intersección de cualquier familia arbitraria de abiertos es un abierto

Entonces existe un orden parcial R en A de forma que los conjuntos

$B(x) = \{y \in A; x R y\}$, con $x \in A$, forman una base de T .

En efecto:

Definimos $x R y$ si y solo si $x \in \text{cl}(\{y\})$. Está claro que la relación R es reflexiva y, por ser el espacio T_0 , es también antisimétrica. Llamando $B(x) = \bigcap_{\alpha \in T, x \in O} O$ tenemos que para todo x, y, z verificando $x R y$, y $R z$, es $z \in B(y)$ y además $B(y) \subset B(x)$, por lo que se tiene que $x R z$.

Por la forma de definir $B(x)$, se puede ver fácilmente que $B(x) = \{y \in A; x R y\}$ y que la familia de los $B(x)$ con $x \in A$ es una base para T . ■

En particular, se tiene que la categoría de espacios topológicos finitos es isomorfa a la categoría de conjuntos finitos parcialmente ordenados y de las aplicaciones que preservan el orden entre ellos, ya que la clausura de cualquier subconjunto está dada por la unión de las clausuras de sus elementos.

Hemos visto que el concepto de CPO, y el de su espacio topológico equivalente, nos brinda un modelo en el que podremos disponer de conceptos y métodos algebraicos [2] [33], que ayudarán notablemente al desarrollo de una topología digital, permitiendo establecer una relación directa entre el modelo continuo, para el estudio de un objeto observable, y el modelo discreto para una "imagen" captada de ese objeto. La aritmética nos dará también la posibilidad de calcular efectivamente los invariantes de un espacio topológico. Este aspecto es muy interesante ya que uno de los propósitos de

de la topología digital consiste en realizar operaciones y cálculos topológicos mediante sistemas automáticos.

Hasta ahora no hemos hablado de la dimensión, salvo en la definición de espacio celular. Herman en [10] se cuestionaba la necesidad de dotar a un CPO de una función dimensión y expresaba su desacuerdo con la función "dim", presentada a continuación, y definida por Kovalevsky en [21] como un forma natural de dotar a un CPO de estructura de espacio celular.

Para cada elemento x de un CPO finito (A,R) se definió "dim" (x) como:

$$\dim(x) = \max_{y \in A} \{ \text{card}(B(y)) - \text{card}(B(y)) \}$$

Tomando un cubo en R^3 y considerando el conjunto de sus caras, aristas, vértices y la cara tridimensional, con la relación "ser borde de", vemos que la definición de dimensión anterior no es muy apropiada. Esto es así porque los elementos de una imagen captada (pixel, voxel,...) junto con los elementos que forman su borde, tienen dada una dimensión que es la dimensión de los elementos del objeto continuo al que representan.

Así pues será necesario encontrar aquellos CPO que sean susceptibles de representar a un objeto continuo observable, y a los que se le pueda dotar de una

dimensión en consonancia con la dimensión del objeto continuo. Este es un nuevo planteamiento de las preguntas formuladas por Herman en [10] y que nos acerca más a un modelo de topología digital.

Siguiendo con la idea de encontrar un modelo matemático para fundamentar el análisis de imágenes que actualmente se hace (imágenes bidimensionales y tridimensionales), y que además sea apto para otros posibles casos que sean generalizaciones de éstos, conviene recordar que en estas imágenes, captadas de recintos acotados de un espacio euclídeo, siempre se distinguen dos subconjuntos, el que llamamos de los objetos y su complementario o fondo. En cuanto a las dimensiones, se ha supuesto que los objetos, de dimensión n , estaban inmersos en un espacio de dimensión n ó $n+1$. Por todo esto se puede proponer estudiar, en principio, los pares (A,B) formados por un CPO compacto A y un subCPO compacto B . Las dimensiones de A y de B , que estudiaremos más adelante, deberán guardar la relación que hemos indicado.

Que un CPO sea compacto equivale a que sea finito. Desde el punto de vista teórico esta restricción, que resulta necesaria para B , se puede debilitar para A exigiendo solo que para cualquier elemento x de A sea $B(x)$ finito. Pero desde el punto de vista computacional obviamente ambas restricciones son indispensables.

Dado un CPO (A,R) finito tenemos una aplicación f desde $sd(A)$ en A definida como sigue [2] [33]:

Para toda cadena $c=[x,y]$ en $sd(A)$, es $f(c) = y$.

Esta aplicación es continua y la imagen recíproca de cualquier cerrado C es homeomorfo a $sd(C)$.

Definiendo la n -subdivisión baricéntrica de A como $sd^n(A) = sd(sd^{n-1}(A))$, para $n > 1$, obtenemos:

$$\dots sd^{n+1}(A) \rightarrow sd^n(A) \rightarrow \dots \rightarrow sd(A) \rightarrow A$$

Definición 2.2.4. Si (A,R) es un CPO finito, se define la realización geométrica de A y se notará por $|A|$ como $|A| = \varinjlim sd^n(A)$. (Ver [33], [6] [8]).

De la misma forma, dado que para toda aplicación continua f entre dos CPO A y B tenemos definida una aplicación $sd(f)$ entre $sd(A)$ y $sd(B)$, podemos considerar la realización geométrica de f , que notaremos por $|f|$.

Teorema 2.2.1. La realización geométrica de un CPO finito (A,R) es un poliedro P . La realización geométrica de un aplicación continua es también continua (y celular).[33]

Los vértices de P están en correspondencia biyectiva con los elementos de A , las aristas de P se corresponden con pares de elementos relacionados en (A,R) y en general las n -caras de P están vinculadas a las cadenas de longitud n en (A,R) , o equivalentemente con los elementos de $sd(A)$ que son subconjuntos de A de cardinal n .

A un CPO finito (A,R) y a su dual (A,R') le corresponde la misma realización geométrica $/A/$, ya que para ambos espacios la subdivisión baricéntrica es el mismo conjunto $sd(A)$.

Si representamos un CPO finito mediante un diagrama de puntos y aristas conectando puntos, de forma que dos elementos directamente relacionados se unen por una arista, las figuras fig 2.2.1 y fig 2.2.2 representan respectivamente a tres CPO y a sus correspondientes realizaciones geométricas.

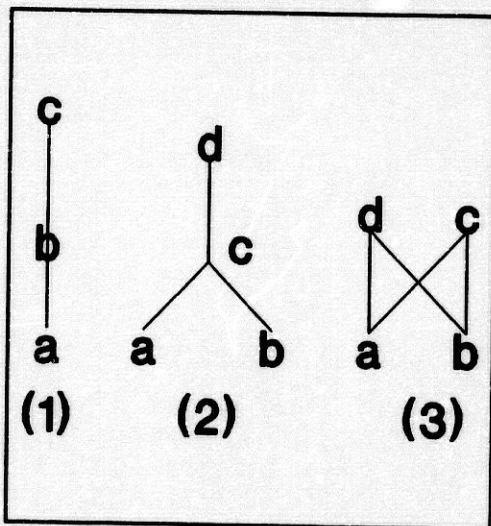


fig 2.2.1

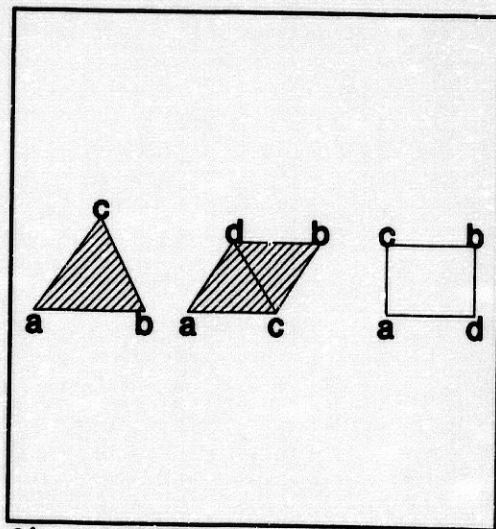


fig. 2.2.2

Observamos en las figuras anteriores que es posible que a dos CPO no homeomorfos le correspondan realizaciones geométricas homeomorfas como espacios topológicos, sin embargo estas realizaciones geométricas tienen distinta estructura como poliedros.

Observamos también que si tomamos la superficie de un rectángulo en el plano, determinado por los cuatro vértices $a, b, c,$ y d . Podemos considerar el conjunto $\{a, b, c, d, \{a, b\}, \{b, c\}, \{c, d\}, \{a, d\}, \{a, b, c, d\}\}$ junto con la relación de inclusión como un CPO, su realización geométrica nos da el rectángulo de partida pero con otra estructura como poliedro. (fig.2.2.3)

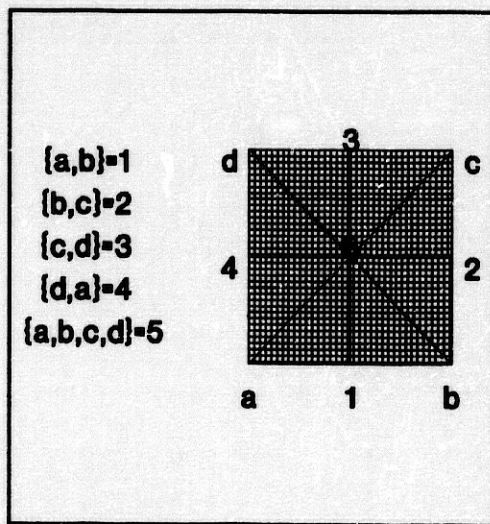


fig.2.2.3.

Si tenemos en cuenta que la realización geométrica de una cadena de longitud n es homeomorfa a la envolvente convexa de los puntos $p_1=(1,0,\dots,0)$, $p_2=(0,1,0,\dots,0),\dots,p_n=(0,0,\dots,1)$ de \mathbb{R}^n , podemos dar una definición de dimensión de un CPO que nos será muy útil.

Definición 2.2.5. Dado un CPO (A,R) en el que toda cadena tiene longitud finita, definimos la dimensión de (A,R) como el máximo de las longitudes de las cadenas contenidas en él, disminuido en una unidad. Si no hay ambigüedad sobre la relación R , notaremos esta dimensión como $\dim(A)$.

Se tiene de forma inmediata la siguiente proposición:

Proposición 2.2.4. La dimensión de un CPO finito coincide con la dimensión de su realización geométrica.

Desde la definición 2.2.5 es fácil dotar a un CPO de estructura de espacio celular de forma canónica. Dado (A,R) definimos la aplicación $d:A \rightarrow \mathbb{N}$ por $d(x) = \dim(/R_x/)$. De esta forma no será necesario hacer ninguna distinción entre un CPO (A,R) de dimensión finita y los espacios celulares de dimensión finita dotados de una aplicación dimensión d verificando 1 y 2 :

- 1) Para cualesquiera dos elementos distintos x, y de A , tal que $x R y$ se tiene que $d(x) < d(y)$.
- 2) Para toda aplicación dimensión d' en A verificando 1) y para todo elemento x de A se tiene $d(x) \leq d'(x)$.

En lo sucesivo nos referiremos únicamente a estos espacios celulares.

Dado un espacio celular de dimensión n , podemos asociarle un par de grafos de (G, G') de la siguiente forma:

- a) Los nodos de G y de G' serán los elementos x de A tales que $d(x) = n$.
- b) las aristas de G serán todos los arcos en A uniendo nodos x e y de la forma $\{x, a, y\}$. Las aristas de G' serán aquellas aristas de G para las que $d(a) = n-1$.

En particular, se tiene que si partimos de una imagen digital a la que asociamos un espacio celular, donde las 3-caras y las 2-caras son los voxel y los pixel respectivamente, el par de grafos (G'', G') donde G'' es G restringido al conjunto de los objetos, es una generalización del grafo de adyacencias definido por Rosenfeld. Para una imagen de dimensión dos obtenemos los tipos de adyacencia $(8,4)$ o $(4,8)$ según que tomemos el conjunto de objetos como un cerrado o un abierto respectivamente. De la misma forma, en dimensión tres, se obtienen los tipos de adyacencia $(26,6)$ y $(6,26)$.

Estos espacios son fáciles de reconstruir desde el par de grafos asociados y conociendo su dimensión. Sabemos cuales son los elementos de dimensión n y los de dimensión $n-1$, los elementos de dimensión $n-i$ serán aquellos representados por aristas de G que se pueden obtener mediante i aristas de G' . Así pues todos los aspectos topológicos de la imagen digital están bien representados en sus grafos asociados. Sin embargo esto no es posible en todos los espacios celulares, como por ejemplo en el espacio (A,R) donde $A=\{1,2,3,4,5\}$ y R está generado por el conjunto de pares $\{(1,3),(2,3),(3,4),(3,5)\}$. Volveremos sobre este aspecto después de estudiar cuales son los espacios celulares susceptibles de representar a una imagen digital en el apartado siguiente.

2.3.CW-COMPLEJOS. COMPLEJOS REGULARES.

[28] [34] [6] [8]

Veremos ahora algunas estructuras que son útiles para modelizar los objetos de los cuales queremos obtener una imagen digital. Trataremos en primer lugar algunas nociones de los CW-Complejos, como una estructura suficientemente general y bastante estudiada, para particularizar hasta un tipo de complejos que están muy próximos, o son casi iguales , a los complejos simpliciales o a los poliedros.

En lo sucesivo haremos uso de la siguiente notación:

$$S^{n-1} = \{x \in \mathbb{R}^n, |x| = 1\}; E^n = \{x \in \mathbb{R}^n, |x| < 1\}; D^n = \{x \in \mathbb{R}^n, |x| \leq 1\}.$$

Sea (X, T) un espacio topológico, una partición P de X se dice celular si, sobre el conjunto de elementos de esta partición , se tiene una función "d" con valores enteros no negativos, tal que para cada elemento "c" de P existe una aplicación continua

$$f_c: D^{d(c)} \rightarrow X$$

Verificando:

1) f es un homeomorfismo de el interior de $D^{d(c)}$ sobre c .

2) f aplica $S^{d(c)-1}$ sobre $\bigcup_{i=1}^{d(c)} d^{-1}(d(c)-i) \subset P$.

Llamaremos celdas a los elementos de la partición P y celdas cerradas a la clausura de éstas en (X, T) , a $d(c)$ lo llamaremos dimensión de c .

Un espacio de Hausdorff provisto de una partición celular P , diremos que es un CW-Complejo si y solo si se verifican [28]:

a) La clausura de cada celda consta de un número finito de celdas.

b) Un subconjunto A de X es cerrado si y solo si la intersección de A con cualquier celda cerrada de la partición P es cerrada en la topología T de X .

Llamaremos n -esqueleto de la partición, y lo notaremos por X^n , al conjunto $\{c \in P; d(c) \leq n\}$. Si existe $n \in \mathbb{N}$ tal que $X = X^n$, diremos que X es finito dimensional y que su dimensión es n .

Una aplicación f de un complejo celular (X, T, P) en otro complejo celular (Y, T', P') se dice celular si es continua y, para todo r , f aplica X^r en Y^r .

Nos interesará ver un tipo particular de CW-complejos en los que encontraremos ciertas propiedades que nos serán de mucha utilidad para establecer una equivalencia funtorial entre espacios celulares discretos y estos CW-complejos.

Definición 2.3.1 [8] Un CW-complejo se dirá regular si y sólo si para cada celda c con $d(c) = n$ existe una aplicación característica $f_c: D^n \rightarrow \text{cl}(c)$ que es un homeomorfismo, y además, si $c \subset \text{Cl}(c')$, es $f_c = f_{c'}/c$.

La primera condición de la definición es usada en topología algebraica para asegurarse de que los grupos de homología se pueden efectivamente calcular []. La particularidad de estos espacios consiste en que la aplicación característica f_c es también un homeomorfismo de la esfera $n-1$ -dimensional sobre su imagen en X^{n-1} , realmente sobre el borde de c . De aquí se deducen las siguientes propiedades:

- 1) Si $m, n \in \mathbb{Z}$, $m < n$ y c_1, c_2 son dos celdas tales que $d(c_1) = m$ y $d(c_2) = n$ y además $c_1 \cap \text{Fr}(c_2) \neq \emptyset$. Entonces se tiene que $c_1 \subset \text{Fr}(c_2)$.
- 2) $\text{Fr}(c)$ es la unión de clausuras de $(d(c)-1)$ -celdas.
- 3) Sean c_1, c_2 son dos celdas tales que $d(c_1) = n$ y $d(c_2) = n+2$, además $c_1 \subset \text{Cl}(c_2)$. Entonces existen exactamente dos $(n+1)$ -celdas c_i verificando:
 $c_i \subset \text{Cl}(c_1) \subset \text{Cl}(c_2)$. [8].

Admitiendo que $d(\emptyset) = -1$, de 2) y 3) se deduce que si $d(c) = 1$ entonces existen exactamente dos 0-celdas que constituyen su borde. También de las afirmaciones 1), 2) y 3) se desprende que si c_1, c_2 son dos celdas tales que $d(c_1) = n < d(c_2) = n+h$ y $c_1 \subset \text{Fr}(c_2)$. Entonces existen h celdas tales que:

$$c_i \subset \text{Fr}(c_{i1}) \subset \dots \subset \text{Fr}(c_{i,i-1}) \subset \text{Fr}(c_i), \text{ con } d(c_{ij}) = n+j.$$

Sea (X, T) espacio topológico con una estructura de CW-complejo regular P de dimensión finita n . Añadimos al esqueleto 0-dimensional X^0 un conjunto de puntos de X que se obtendrá de la siguiente forma:

Para cada c_{ij} de X^i con $i > 0$, tomamos un punto p_{ij} del interior de c_{ij} y ampliamos X^0 a $X^{0n} = X^0 \cup (\bigcup_i \bigcup_j \{p_{ij}\})$.

La condición de regularidad aseguraba que para cada celda c_{ij} existe un homeomorfismo f_{ij} entre $\text{Cl}(c_{ij})$ y D^i . Consideramos los puntos $x_{ij} = f_{ij}(p_{ij})$ y realizamos una división en D^i de forma que las nuevas caras de dimensión r estén determinadas por x_{ij} y r vértices v verificando que $v \in S^{i-1}$ y existen n, j tal que $v = x_{nj} \in X^{0n}$, como el interior de la envolvente convexa de dichos vértices en D^i , con la topología restringida a esta envolvente convexa.

Se obtiene una subdivisión P' de P tomando en X las imágenes por las aplicaciones f_{ij} de las divisiones en D_i . Es fácil ver que dos subdivisiones de un mismo CW-complejo regular son isomorfas. Por otra parte, si f es una aplicación celular entre dos CW-complejos (X, T, P) e (Y, T', P') , dada una subdivisión del primer espacio, es posible encontrar una subdivisión del segundo de forma que f sigue siendo celular entre ellas [28].

Se desprende de la forma de construir de la subdivisión de un complejo regular que si es $V = \{x_1, \dots, x_{m+1}\}$ el 0-esqueleto de una celda cerrada n -dimensional, entonces cualquier subconjunto de V formado por $m+1$ elementos es el 0-esqueleto de alguna m -celda cerrada.

Podemos construir ahora lo que llamaremos la realización geométrica de un CW-complejo regular compacto. Sea (X, T, P) un CW-complejo regular compacto y sea $k+1$ el número de vértices de una subdivisión suya (X, T, P') . Entonces construimos una aplicación f entre X y \mathbb{R}^k de la siguiente forma: Tomamos en \mathbb{R}^k $k+1$ puntos unitarios y afínmente independientes y establecemos una biyección entre estos puntos y los vértices de X de forma que para toda celda c sea $f(c)$ el simplex euclídeo generado por los vértices x_i imágenes de los vértices de c . Al poliedro $S = f(X)$ lo llamaremos realización de X y lo notaremos por $/X/$ [6] [8] [28] [34].

Si conocemos un CW-complejo regular y una subdivisión suya, o lo que se dará más generalmente, si distinguimos un subconjunto de vértices como previos a una subdivisión, podemos realizar una operación inversa a la subdivisión que llamaríamos "agrupación" [6] [8].

Resumiendo lo anterior podemos dar la siguiente proposición:

Proposición 2.3.1. Todo CW-complejo regular es isomorfo a un CW-complejo regular embebido en un espacio euclídeo.

Es conocido el hecho de que si la dimensión de uno de estos espacios X es d entonces X se pueden embeber en un espacio euclídeo de dimensión $2d+1$ [12].

Veremos ahora una asociación natural de un espacio celular (discreto) a un CW-complejo regular. Sea (X, T) un espacio topológico T_2 y P una partición que lo dota de estructura de CW-complejo regular de dimensión finita, consideramos el espacio topológico cociente $(X/P, T/P)$, este espacio se identifica con los elementos de P y los abiertos de T/P son imágenes por la proyección natural π de los abiertos P -saturados en (X, T) .

En X/P consideramos el orden parcial dado por $[x] < [y]$ sii $\pi^{-1}([x]) \subset \text{Fr}(\pi^{-1}([y]))$ o equivalentemente sii $[x] \in \text{Cl}[y]$ en la topología T/P y se tiene que la topología cociente coincide con la topología que proviene del orden " $<$ ". Tenemos también definida de forma natural una aplicación dimensión, coincidente con la aplicación dimensión del CW-complejo.

Se tiene que la proyección natural conmuta con la subdivisión de CW-complejos regulares y la subdivisión baricéntrica de CPO (en este caso de espacios celulares).

Desde ahora, en ambos casos, las llamaremos subdivisiones baricéntricas.

La proyección de una aplicación entre dos CW-complejos regulares X e Y es una aplicación celular entre las proyecciones de X y de Y . Todo lo anterior se puede resumir diciendo que la proyección natural es un funtor entre la categoría de CW-complejos regulares y la categoría de espacios celulares. También, en función del embebimiento antes señalado, podemos considerar los complejos regulares como subconjuntos de un espacio euclídeo.

2.4. CONCLUSIONES.

A lo largo de los apartados 2.2 y 2.3 hemos visto el funtor "realización geométrica" entre la categoría de CPO y la categoría de poliedros y el funtor proyección natural entre la categoría de complejos regulares y la de espacios celulares. Nuestro objetivo en este apartado será dar una equivalencia funtorial entre los complejos regulares compactos y una parte de los CPO para los cuales existe una forma natural de dotarlos de estructura de espacio celular.

La interpretación de la proyección natural es el proceso de captación de una imagen. Cada elemento de la retícula de la cámara capta la radiación emitida por un recinto del objeto. Podemos modelizar el objeto, inmerso en el espacio euclídeo, con la descomposición celular dada por estos recintos, es decir la estructura celular del objeto está dada por la estructura de la retícula de la cámara. También se tiene que los elementos de la imagen se corresponden con los elementos de esta retícula, así pues

cada celda de dimensión máxima, de la estructura celular dada al objeto, determina un elemento de la imagen digital.

El interés de encontrar una equivalencia funtorial estriba en la necesidad de restringir el modelo de los CPO a aquellos que, por un proceso como el señalado más arriba o una generalización suya, son susceptibles de representar un objeto de el espacio euclídeo sin ambigüedad (salvo homeomorfismos).

De las propiedades 1),2) y 3) de los CW-complejos regulares (X,T,P) se deducen las siguientes propiedades del espacio celular $(X/P,T/P, d)$:

a) Para todo elemento $x \in X/P$ tal que $d(x) = n$ existe un subconjunto $\{x_0, \dots, x_{n-1}\}$ en X/P verificando que $d(x_i) = i$ y que $x_0 \leq x_1 \leq \dots \leq x_{n-1} \leq x$.

b) Para cualesquiera dos elementos x, y de X/P con $d(x) = d(y) + 2$, existen exactamente dos elementos z_1 y z_2 tal que $x < z_i < y$ $i=1,2$ y $d(z_i) = d(x) + 1$. Además todo elemento 1-dimensional está bordeado por exactamente dos elementos 0-dimensionales.

c) Para cualesquiera dos elementos x, y de X/P con $d(x) = m < d(y) = n$, existe un subconjunto $\{x_1, \dots, x_{n-m+1}\}$ en X/P verificando que $d(x_i) = m+i$ con $i=1 \dots n-m+1$ y que $x = x_0 \leq x_1 \leq \dots \leq x_{n-m} \leq y$.

En un espacio celular arbitrario $(X, <, d)$ los asertos a) y b) no implican necesariamente c), como se ilustra en el diagrama de la figura 2.4.1.

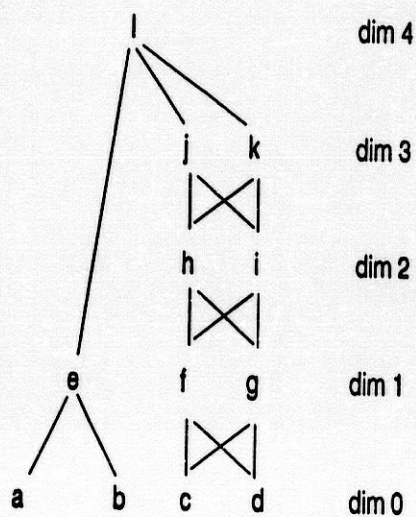


fig 2.4.1.

La propiedad c) junto con la afirmación de que para cada elemento x de X/P existe un elemento y con $d(y)=0$ y además $y < x$, implican la propiedad a). Designaremos esta nueva propiedad por c').

Entonces, las propiedades b) y c') son condiciones necesarias para que un espacio celular finito arbitrario represente el cociente natural de un CW-complejo regular compacto.

Veremos ahora las condiciones que debe verificar un CPO finito para que admita una aplicación dimensión verificando b) y c').

Sea $(X, <)$ un CPO finito y sea "d" la aplicación definida en el apartado 2.2. El valor de la aplicación "d" sobre cualquier elemento x es el máximo de las longitudes de las cadenas contenidas en $Cl\{x\}$. En estos espacios la condición c') equivale a que todas las cadenas maximales contenidas en la clausura de cualquier elemento tengan idéntica longitud. Por otra parte, es fácil ver que en cualquier espacio celular $(X, <, d')$ en el que se verifique b) y c'), la aplicación dimensión d' coincide con la definida anteriormente. En lo sucesivo, a esta aplicación dimensión la llamaremos dimensión natural.

Sea ahora $(X, <)$ un CPO finito, consideramos $sd(X)$ y la realización geométrica de $(X, <)$ $/X/$. El espacio $/X/$ es un poliedro y sus vértices están en correspondencia biyectiva con los elementos de $sd(X)$. Para que en $/X/$ se pueda hacer una "agrupación" ($Agr_{(X, <)}(/X/)$), siguiendo la información de $(X, <)$, tal que su proyección natural esté representada por $(X, <)$, es suficiente y necesario que $(X, <)$ con la dimensión natural verifique las condiciones b) y c'), a las que llamaremos condiciones de regularidad. Para estos CPO aludiremos a su realización geométrica como $Agr_{(X, <)}(/X/)$. Tenemos entonces la siguiente caracterización de los CPO, que poseen una estructura canónica de espacio celular, que son un modelo adecuado para la representación discreta de un objeto espacial continuo :

Teorema 2.4.1. La categoría de CW-complejos regulares compactos y aplicaciones celulares es equivalente a la categoría de CPO finitos verificando las condiciones b) y c').

Los siguientes diagramas son conmutativos:

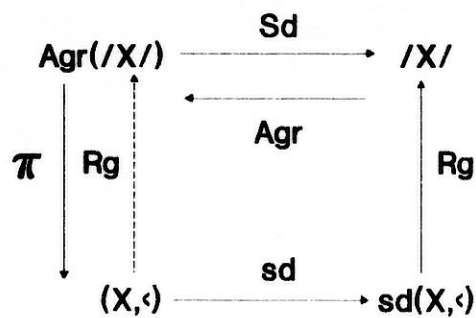
$$\begin{array}{ccc}
 (X, T, P) & \xrightarrow{f} & (Y, T', P') \\
 \downarrow \pi & & \downarrow \pi \\
 (X/P, T/P) & \xrightarrow{f/p} & (Y/P', T'/P')
 \end{array}$$

Funtor proyección natural

$$\begin{array}{ccc}
 /X/ & \xrightarrow{/g/} & /Y/ \\
 \uparrow Rg & & \uparrow Rg \\
 (X, \langle \rangle) & \xrightarrow{g} & (Y, \langle \rangle)
 \end{array}$$

Funtor Realización Geométrica

Estos funtores junto con los que hemos llamado subdivisión , subdivisión baricéntrica y agrupación, nos permiten definir otro funtor realización geométrica, que aparece en el diagrama en trazo discontinuo, entre la categoría de CW-complejos regulares y compactos y la categoría de CPO finitos con las condiciones de regularidad. Este funtor hace conmutativo el siguiente diagrama y junto con la proyección determina el isomorfismo de categorías.



Después de lo anterior queda respondida la cuestión planteada por Herman en [10] acerca de un modelo teórico adecuado para la Topología Digital. Además, la terna $(X, X/P, \pi)$, donde (X, P) es un CW-complejo regular y compacto, es una "digitalización completa" según la definición dada en [5]. También, el grafo asociado a X/P es un "Espacio Digital" como los definidos en [5].

De esta forma, trabajando con CPO regulares, tenemos la seguridad de estar considerando un objeto real, un poliedro, y de que los conceptos topológicos usados en el estudio del CPO tienen una correspondencia perfecta con la topología usual en un entorno de un poliedro. En cuanto a la tratabilidad de estos conceptos en un ordenador, tenemos por una parte la posibilidad de usar los grafos asociados o los espacios digitales, y por otra parte la potencia de cálculo de una estructura aritmética como es la de retículo que subyace en el CPO.

Los grafos son una de las herramientas más usadas en topología digital para el tratamiento de imágenes. Aquí hemos tratado de exponer que el uso de determinados grafos, los grafos asociados a un CPO regular, está perfectamente justificado (y es el más indicado) para tratar problemas relativos a la conexión y la continuidad en el espacio topológico asociado a una imagen digital. Caracterizar los grafos que son inducidos por una topología es un problema que ha sido abordado por E. Domínguez y A. Francés .

Un ejemplo de la potencia de cálculo en un retículo lo tenemos en el cómputo de la característica de Euler, en la definición de punto simple y en la comprobación efectiva de la simplicidad de puntos en una imagen digital.

Para abordar estas cuestiones necesitamos introducir la "función de Moebius"

en un CPO. Un estudio suficientemente detallado se puede encontrar en [2].

Definición 2.4.1. Sea (A, R) un CPO finito y $c_i(A)$ el número de cadenas en A de longitud i . Se define el número de Moebius de A como:

$$\mu(A) = -1 \cdot \sum_i (-1)^i c_i(A)$$

Proposición 2.4.1. Si notamos por R_x y ${}_xR$ los elementos de A que anteceden o siguen, respectivamente, a x , tenemos que :

$$\mu(A) = -1 - \sum_{x \in A} \mu(R_x)$$

y que:

$$\mu(A) = -1 - \sum_{x \in A} \mu({}_xR)$$

Tenemos que el número de Moebius de un CPO $(A, <)$ más una unidad, es la característica de Euler de $sd(A)$ que coincide con la característica de su realización geométrica y por tanto con la de $|A|$. Así pues se tiene que :

$$\mu(A) = \chi(A) - 1 = -1 + \sum_i (-1)^i H_i(A)$$

Donde $H_i(A)$ es i -ésimo grupo de homología simplicial del poliedro A .

El cálculo efectivo de los grupos de homología puede verse en [2] [33].

Cuando la relación de orden parcial R es fija y la consideramos en distintos subconjuntos C de un conjunto A, a los conjuntos R_x los notaremos por A_x y C_x . De esta forma se tiene que si $C \subset A$ y C abierto es:

$$\mu(A) = -1 - \sum_{x \in A} \mu(A_x); \mu(A-C) = -1 - \sum_{x \in A-C} \mu((A-C)_x)$$

$$\mu(A) - \mu(A-C) = -\sum_{x \in A} \mu(A_x) + \sum_{x \in A-C} \mu(A-C)_x = -\sum_{x \in A} \mu(A_x) + \sum_{x \in A-C} \mu(A_x)$$

* Esto es cierto porque al ser C abierto es A-C cerrado y por tanto los elementos que anteceden a x, es decir su clausura, están contenidos en A-C.

Queda que :

$$\mu(A) - \mu(A-C) = -\sum_{x \in C} \mu(A_x)$$

Por otra parte tenemos que:

$$\mu(Cl(C)) = -1 - \sum_{x \in Cl(C)} \mu(Cl(C))_x = -1 - \sum_{x \in Fr(C)} \mu(Cl(C))_x - \sum_{x \in C} \mu(Cl(C))$$

$$= ** -1 - \sum_{x \in Fr(C)} \mu(Fr(C))_x - \sum_{x \in C} \mu(A_x)$$

** se tiene por ser Fr(C) y Cl(C) conjuntos cerrados.

Así pues se tiene que:

$$\mu (Cl(C)) - \mu (Fr(C)) = - \sum_{x \in C} \mu (A_x)$$

De todo lo anterior concluimos que:

Proposición 4.2.2 En las condiciones anteriores se tiene que:

$$\mu (A) - \mu (A-C) = \mu (Cl(C)) - \mu (Fr(C))$$

Si C es el menor abierto que contiene a un punto , tenemos que la ecuación anterior caracteriza a los puntos simples, además, si C es un entorno conexo de un conjunto de puntos, tenemos una caracterización de un conjunto de puntos "borrables" en un adelgazamiento.

**CAPITULO III ADELGAZAMIENTO DE
I.D.B.**

En este capítulo, se estudia el problema del adelgazamiento de IDB bidimensionales. Dándose un conjunto de resultados en topología digital conducentes a la obtención de un algoritmo de adelgazamiento que, verificando unos requisitos tomados como ideales, se puede ejecutar en tiempo constante.

3.1 INTRODUCCION.

En el capítulo anterior se introdujo una topología consistente y suficientemente general para el tratamiento de IDB. También se vio como, mediante el grafo asociado al espacio topológico, se obtenían como casos particulares los grafos de adyacencias introducidos por Rosenfeld. Será sobre estos modelos en los que centraremos nuestra atención para abordar por primera vez, en este trabajo, el concepto de adelgazamiento en IDB. Utilizaremos la notación y terminología que ha venido siendo usual en este tipo de operaciones en el marco de los modelos citados y que se encuentran expuestos casi en su totalidad en el capítulo primero de preliminares.

Los resultados que presentamos aquí aparecen ligados, además de a cuestiones teóricas relativas a operaciones que "preservan la topología" de un cierto espacio, a problemas prácticos de implementación eficiente de algoritmos que se usarán en el tratamiento de imágenes. Los principales resultados que presentamos también se enuncian y prueban dentro de un resultado más general en el marco de los espacios celulares en el capítulo IV de esta memoria.

La idea más general de un adelgazamiento en una imagen digital binaria consistiría en una operación sobre los objetos de dicha imagen (puntos negros en la terminología clásica), mediante la cual la información necesaria para representar a estos objetos se reduce con respecto a la original. Se le exige además al resultado de esta operación que presente las mismas propiedades de conexión que la imagen de partida.

Esta operación se utiliza fundamentalmente como un preproceso en el reconocimiento de formas y también es aplicable en tareas de codificación de imágenes, de forma que se optimice la cantidad de memoria necesaria para su almacenamiento. Para estos fines se requiere una condición no topológica consistente en que las partes "elongadas" de la imagen inicial se puedan representar mediante un arco en la imagen tratada.

Será necesario precisar los conceptos a los que hemos aludido, hasta ahora solo de forma intuitiva, en esta introducción. Tenemos que aclarar que estos conceptos han sido

tratados abundantemente en la literatura para imágenes 2-dimensionales y dentro del modelo del grafo de adyacencias. El caso de imágenes tridimensionales no es muy conocido y aún se seguían cuestionando la formalización de algunos de estos conceptos.

En los términos de las definiciones 1.3.1, 1.3.2. y del criterio 1.3.1, podemos afirmar que una operación sobre una IDB preserva la topología si el resultado de esta operación presenta el mismo "árbol de adyacencia". Donde el árbol de adyacencia de una IDB P (introducido por Buneman [3]) es un grafo cuyos vértices son las distintas componentes conexas de B o de $Z^2 - B$, de forma que dos vértices están unidos por una arista si y solo si las correspondientes componentes son adyacentes una de la otra.

Un punto se dirá simple si su borrado preserva la topología, en el sentido antes mencionado. En dimensión 2 existen caracterizaciones útiles de puntos simples [18].

Si notamos por Σ el conjunto de los puntos simples de B , se tiene claramente que $\Sigma \subset \delta \subset B$ ($\delta = Fr(B)$). Es conocido que el borrado simultáneo de los puntos simples no preserva la topología, se hace necesario un borrado secuencial de estos, Ronse [29].

De cara a la aplicación práctica, es necesario un algoritmo capaz de borrar puntos secuencialmente siguiendo el borde de una IDB, y sobre todo algoritmos paralelizables, es decir, algoritmos en los que las operaciones sobre cada punto solo dependan de la

información obtenida de puntos cercanos, a fin de poder ser implementados en ordenadores con múltiples procesadores. Otra condición deseable consiste en la isotropía del algoritmo. De esta forma estamos acercándonos a un ideal de algoritmo capaz de borrar puntos en todas las direcciones en paralelo. este ideal deja fuera de nuestra consideración a muchos algoritmos que sin embargo se siguen empleando porque se ha impuesto su sencillez de implementación y la velocidad de ejecución sobre las condiciones que hemos expuesto como ideales.

Un teorema obtenido por Fu y Tsao en [36], aplicable a los algoritmos que nos ocupan es el siguiente:

Teorema 3.1.1 Sea P una (m,n) Imagen digital binaria, donde (m,n) es $(4,8)$ o $(8,4)$, y sea R el conjunto de puntos simples de P tales que las siguientes condiciones se verifican para todo punto p en R :

- i) El conjunto de puntos negros no simples en N_p es conexo, y
- ii) es adyacente a p y a cualquier punto negro que es adyacente a p .

Entonces el borrado de puntos de R preserva la topología.

Estas condiciones necesitan conocer previamente que puntos son simples y almacenarlos, lo que requiere una gran cantidad de cálculos previos, de otra forma los cálculos se extenderían más allá de un entorno de cada punto borde. Estas objeciones

empañan su adecuación al ideal teórico y lo hacen poco apropiado para una implementación práctica [18].

3.2 UN CRITERIO PARA EL ADELGAZAMIENTO DE IMAGENES DIGITALES BINARIAS.

En este apartado se dan unos resultados para el adelgazamiento en IDB que están en la línea del ideal antes descrito, en los que los cálculos necesarios para el procedimiento de borrado se reducen notablemente con respecto a otros resultados conocidos (teorema 3.1.1.), ya que no es necesario obtener explícitamente el conjunto de puntos simples de una I.D.B. En una retícula ortogonal con adyacencias del tipo (8,4) o (4,8) este conjunto viene dado por el siguiente:

Teorema 3.2.1. Sea T el subconjunto de δ tal que:

- i) $\forall p \in T, N_p \cap (B - \delta) \neq \emptyset$ y conexo.
- ii) $\forall q \in (N_p - \{p\}) \cap \delta$ si q es adyacente a p , entonces
 $\exists r \in N_p \cap (B - \delta)$ tal que r es adyacente a q .

Entonces el borrado de los puntos de T preserva la topología.

Es fácil ver que todos los puntos de T son simples y las condiciones i) y ii) se entienden como un predicado sobre un δ -entorno de cada punto.

La escasez de demostraciones de teoremas sobre adelgazamiento en la literatura nos motiva a escribir la siguiente demostración explícita, dentro del modelo en el que hemos ubicado el resultado, sin perjuicio de una generalización, en un marco mucho más general, que se dará en el Capítulo IV.

Demostración.

Sea T_p una componente conexa de T en la imagen P . Probaremos que el borrado de T_p preserva la topología.

Si $T_p = \{p\}$, el resultado es inmediato ya que p es un punto simple. Supongamos ahora que T_p es distinto de $\{p\}$, entonces dado que $\forall p \in T, N_p \cap (B - \delta) \neq \emptyset$ tenemos que el número de componentes conexas de B no decrece.

Llamaremos B_p a la componente conexa de p en B , y $NT_p = \bigcup_{q \in T} N_q$. Tenemos que $T_p \subset B_p$.

Para ver que $B_p - T_p$ es conexo solo necesitamos comprobar que: $\forall x, x \in B_p \cap NT_p$ y $x \notin T_p$, x es adyacente a $B_p - T_p$.

Ya que la frontera con respecto a una componente conexa "blanca" es conexa, existe un camino en $\delta - T$ uniendo x con $\delta \cap (N_p - \{q\})$, para algún $q \in T_p$. Si no existe un tal camino en $\delta - t$, entonces estamos en la situación anterior. Por tanto x es conectable con algún punto del borde que es adyacente a $B_p - \delta$ y por tanto x es conectable con $B_p - T_p$.

Hemos visto que $B_p - T_p$ es conexo, y de esta forma cada componente conexa de B contiene exactamente una componente de $B - T$.

Probamos ahora que el borrado de T no puede unir dos componentes conexas blancas distintas. Sean W_1 y W_2 dos componentes conexas blancas con $W_1 \cap W_2 = \emptyset$. Si después de borrar T_p estas componentes quedan unidas, debe existir un camino C , $C = \{x_1, \dots, x_n; x_1 \sim w_1 \in W_1, x_n \sim w_2 \in W_2\}$ y $C \subset T_p$.

Por ser $T_p \subset \delta$, se tiene que $\forall i \in \{1, \dots, n\} \exists w_i \in Z^2 - B$, de forma que x_i es adyacente a w_i . Así, el camino C se reduce a $\{x_1, x_2\}$. Además x_1 y x_2 han de ser 4-adyacentes (el mismo o menor tipo de adyacencia que el de los puntos blancos). Por tanto $N_{x_1} \cap N_{x_2}$ son seis puntos, además $x_1, x_2 \in T$ implica que x_i es adyacente a $N_{x_j} \cap (B - \delta)$, con $i, j: 1, 2$.

De esta forma, no existen puntos negros del borde en $N_{x_1} \cap N_{x_2}$ localizados a un lado de la adyacencia dada por (x_1, x_2) , mientras que en el lado opuesto solo puede haber puntos del borde. Estos puntos del borde no pueden ser adyacentes a ningún punto interior (punto negro no borde). Esto está en contradicción con i) y con que x_1, x_2 pertenezcan a T , por tanto tienen que ser puntos blancos. Esto significa que algún punto x_i , $i: 1, 2$ es un punto borde con respecto a ambas componentes blancas W_1 y W_2 y por tanto no es un punto simple.

De todas estas consideraciones deducimos que dos componentes conexas blancas no pueden quedar unidas mediante el borrado de puntos de T . Así hemos probado que cada componente conexa blanca en $P' = (Z^2, m, n, B - T)$ contiene una sola componente blanca de $P = (Z^2, m, n, B)$. ■

En Z^2 se definían también las 6-adyacencias para cada punto, en este tipo de I.D.B. seguiremos notando N_p al conjunto de puntos que son 6-adyacentes a p .

Para cada punto p perteneciente a B los siguientes asertos son equivalentes:

- a) $(N_p - \{p\}) \cap B$ es conexo.
- b) $N_p \cap W$ es conexo, donde $W = Z^2 - B$.

Así pues, podemos comprobar si un punto es simple solo atendiendo a a).

En la topología definida por las 6-adyacencias en las imágenes digitales podemos caracterizar los puntos de T mediante un simple recuento de los puntos bordes y de todos los puntos negros de $N_p - \{p\}$. La siguiente proposición nos da una equivalencia entre las condiciones del teorema 3.2.1. y esta caracterización.

Proposición 3.2.1. En una (6,6) imagen digital, las hipótesis del teorema 3.2.1 pueden enunciarse de la siguiente forma:

Un punto $p \in \delta$ es $p \in T$ si y solo si se verifican las condiciones:

- a) $(N_p - \{p\}) \cap B$ tiene al menos tres puntos y
- b) $(N_p - \{p\}) \cap \delta$ tiene exactamente dos puntos.

En efecto:

Ya que p es un punto de la frontera, debe existir algún punto blanco w en N_p , entonces hay al menos dos puntos b_1, b_2 en $(N_p - \{p\}) \cap \delta$ y, ya que I_p es no vacío se tiene la condición a). Por ser I_p conexo deben haber solo dos puntos bordes en $N_p - \{p\}$ adyacentes a p . Pero por hipótesis, todo punto borde de N_p debe ser adyacente a I_p , así se tiene que hay solamente dos puntos bordes en $N_p - \{p\}$. El recíproco es fácil de ver de la misma forma. ■

Teorema 3.2.2. En la topología definida por las 6-adyacecias, sea T el subconjunto de δ tal que:

- a) $(N_p - \{p\}) \cap B$ tiene al menos tres puntos y
- b) $(N_p - \{p\}) \cap \delta$ tiene exactamente dos puntos.

.Entonces el borrado de los puntos de T preserva la topología.

En efecto:

Veamos primero que los puntos de T son puntos simples. Para cada punto p de B consideramos el conjunto $I_p = N_p \cap (B - \delta)$. Sea p un punto de T , entonces $I_p \neq \emptyset$, I_p es conexo y para cada punto q de N_p , q es adyacente a p , por tanto q es adyacente a algún punto en I_p . Así pues $(N_p - \{p\}) \cap B$ es conexo.

Por otra parte, si $p, q \in T$ y son adyacentes tenemos que I_p y I_q son no vacíos.

Además q es adyacente a I_p y p es adyacente a I_q lo que implica que $I_p \cap I_q$ es conexo.

Concretamente $I_p \cap I_q = \{r\}$.

Así, $\forall x \in (N_p - \{p\}) \cup (N_q - \{q\})$ es $x \in \delta$ o $x \in B - \delta$, en ambos casos x es conectable a r . De donde se tiene que $(N_p - \{p\}) \cup (N_q - \{q\})$ es conexo.

De todo lo anterior se obtiene que $NT_p - T_p$ es conexo y $B_p - T_p$ es conexo. ■

El Teorema 3.2.2. es la versión del Teorema 3.2.1. para las (6,6) imágenes digitales.

3.3 UN ALGORITMO NO ITERATIVO PARA LA OBTENCION DEL ESQUELETO EN I.D.B.

Hasta ahora hemos hablado del problema relativo al borrado de puntos en una I.D.B que necesariamente habían de pertenecer a la frontera del conjunto denominado objetos. Estas operaciones de borrado se pueden repetir hasta obtener una imagen en la cual ya no es posible continuar con el proceso. En lo sucesivo entenderemos por esqueleto de una I.D.B un subconjunto minimal del conjunto objetos, obtenido por adelgazamiento (adelgazamientos) de estos.

No hemos encontrado en la literatura algoritmos para la obtención del esqueleto, en el sentido que de este concepto aquí manejamos de preservar la topología, que no requieran reiterar adelgazamientos sobre los conjuntos de puntos borde que, tras cada operación aparecen como tales.

Definición 3.3.1. Dada una I.D.B P y un punto p de B , llamamos nivel de p y lo notaremos por $L(p)$ al mínimo de las longitudes de los caminos en B que unen

p con la frontera de B.

En [32] se dan algoritmos, tanto secuenciales como paralelos, para calcular el nivel de cada punto de B .

Es importante resaltar que resultados como el teorema 3.2.1. es necesario aplicarlos al menos tantas veces como el máximo nivel de una I.D.B para obtener su esqueleto. El propósito de este apartado es probar que, si el nivel de cada punto es conocido (mediante un preproceso), entonces es posible obtener el esqueleto de una I.D.B. de una forma mucho más corta, exactamente en dos pasos [24]. Para obtener nuestro resultado, usando el teorema 3.2.1., damos las siguientes definiciones.

Definición 3.3.2. Un punto p perteneciente a los objetos de una IDB se dirá primer punto base si no verifica alguna de las siguientes condiciones:

1. El conjunto $\{q \in N_p; L(q) > L(p)\}$ es no vacío y conexo.
2. Para cualquier punto q adyacente a p y tal que $L(q) = L(p)$, existe otro punto r en N_p adyacente a q con $L(r) > L(q)$.

El conjunto de los primeros puntos base de una imagen digital binaria P se notará $B_1(P)$.

De la definición anterior y como una consecuencia obvia del teorema 3.2.1.

tenemos:

Lema 3.3.1. En una IBD P , $B_1(P)$ es un subconjunto del esqueleto de P .

En general, se tiene que el conjunto de los primeros puntos base es un subconjunto propio del esqueleto. Para obtener los puntos restantes del esqueleto damos la siguiente definición.

Definición 3.3.3. En una IDB llamaremos segundos puntos base a los puntos p de B que verifiquen las condiciones:

1. $p \notin B_1(P)$.
- 2 Existe $q \in (B_1(P)) \cap N_p$ con $L(q) < L(p)$ tal que q no es adyacente a ningún punto del conjunto $\{r \in N_p; L(r) > L(p)\}$.

El conjunto de los segundos puntos base se notará $B_2(P)$. Los puntos de B en P que son primeros o segundos puntos base se dirán puntos base de P y se notarán $B(P)$.

Teorema 3.3.1. En una imagen digital binaria P , $B(P)$ es el esqueleto de P .

En efecto:

En primer lugar, es fácil comprobar que $B_2(P)$ es parte del esqueleto de p . Queda

solamente ver que cualquier punto del esqueleto es un punto base.

Sea p un punto del esqueleto de p con $L(p) = i$, después de $i+1$ iteraciones del teorema 3.2.1. obtendríamos que p es un punto de la frontera de la imagen resultante y que p es no borrable, así pues no verifica alguna de las condiciones del citado teorema. Si ocurre que p no verifica la primera condición o ningún punto en su entorno tiene nivel mayor que el, entonces p es primer punto base. De otra forma, es inmediato comprobar que p es segundo punto base, y por tanto, en cualquier caso, p es un punto base. ■

El teorema 3.3.1. es la base para un algoritmo que calcula el esqueleto de una IDB.

Este algoritmo es el siguiente:

ESQUELETO(P)

1. Calcular el nivel de cada punto en P
2. Encontrar el conjunto $B_1(P)$.
3. Encontrar el conjunto $B_2(P)$.

El primer paso puede ser considerado como un preproceso y es posible usar alguno de los algoritmos descritos en [32]. El segundo paso consiste en seguir la definición 3.3.2. siendo necesario comprobarla en todos los puntos de B , así en un algoritmo secuencial este paso correría en tiempo lineal. Para el tercer paso bastaría mirar en un entorno del conjunto $B_1(P)$ y en el peor caso también sería necesario un tiempo lineal.

Proposición 3.3.1. Es posible encontrar el esqueleto de una imagen digital binaria P con N puntos en tiempo $O(N)$ con $O(N)$ tiempo de preproceso y de almacenamiento.

Tenemos además que el algoritmo ESQUELETO es muy eficiente sobre un ordenador celular, ya que cada operación puede ser realizada en todos los puntos en paralelo. Es conocido que el primer paso requiere un tiempo proporcional al mayor de los niveles de los puntos en P , los pasos siguientes pueden ser realizados en un tiempo constante.

De lo anterior se deduce el siguiente teorema.

Teorema 3.3.2. Es posible encontrar en paralelo el esqueleto de una imagen digital binaria P con N puntos en tiempo $O(1)$ con tiempo de preproceso $O(d)$ y $O(N)$ procesadores y almacenamiento, donde d denota el mayor nivel en P .

**CAPITULO IV ADELGAZAMIENTOS
TRIDIMENSIONALES**

En este capítulo, se estudia el problema del adelgazamiento de IDB en tres dimensiones. Comprobándose que el método desarrollado en el capítulo anterior para IDB 2-dimensionales es susceptible de ser aplicado en el caso tridimensional. Obteniéndose igualmente un algoritmo que determina el esqueleto de una IDB tridimensional en tiempo constante.

4.1 INTRODUCCION

En el magnífico artículo de Kong y Rosenfeld [18] , que recogía la mayoría de los aspectos fundamentales y los resultados más sobresalientes en el campo de la topología digital, se concluía que, si bien el caso dos dimensional había sido abundantemente tratado, no ocurría lo mismo para el caso tridimensional, donde era difícil generalizar los resultados que en este campo se habían obtenido, quedando además algunos problemas formales por resolver.

Abordaremos en este apartado el marco formal en el que, más adelante, situaremos nuestro algoritmo de adelgazamiento.

En este capítulo una IDB será siempre un subconjunto finito de la retícula Z^3 con las (m,n) -adyacencias, donde (m,n) puede ser $(26,6)$ o $(6,26)$ (nuestros métodos serán también válidos para otro tipo de topología).

Las condiciones descritas para preservar la topología en dimensión 2 no son suficientes cuando tratamos con IDB 3-dimensionales, ya que objetos tan distintos como una esfera y un toro embebidos en R^3 tienen el mismo árbol de adyacencias; así pues es necesario definir más precisamente que significa preservar la topología.

También es importante definir en que tipo de IDB tiene sentido práctico la operación de adelgazamiento, ya que el resultado de un algoritmo de adelgazamiento debe preservar, aproximadamente, la forma de los objetos originales. Nosotros consideraremos aquí imágenes sin un gran volumen con respecto a su longitud o a su área. Diremos que una IDB no tiene un gran volumen con respecto a su longitud o a su área si el factor $V/(2d)^3$ es al menos 4, donde V representa el número de puntos (voxels) en B , es decir su volumen, y d es el máximo nivel. Se trata de un equivalente de las imágenes elongadas en dimensión 2. Estas ideas son originales del profesor A. Márquez y están recogidas en [25].

Como ya ha sido señalado, y dado que el árbol de adyacencias no es suficiente fuera del caso 2-dimensional (donde además ya no sería un árbol), es importante fijar que significa para un algoritmo 3-D "preservar la topología".

4.2 PRESERVACION DE LA TOPOLOGIA EN 3D

En la literatura se pueden encontrar abundantes definiciones de que significa para un algoritmo 3-D "preservar la topología". Pero, quizás la más natural (y la más fácil de comprobar, aunque la más restrictiva) es exigir que la IDB final sea obtenida desde la imagen de partida mediante un borrado secuencial de puntos simples (y así, los puntos simples deben ser definidos independientemente, como fue señalado en [18], esto se puede hacer definiendo los puntos simples como aquellos que satisfacen las tres condiciones en [18], Teorema 9.1] o condiciones similares). Otros criterios menos restrictivos (y en opinión del prof. Rosenfeld más naturales) para la conservación de la topología son los que aparecen como criterio 10.1 y criterio 10.2 en [18], que resultan ser no equivalentes y que presentan algunos aspectos no deseables como la falta de precisión de algunos conceptos utilizados en el primero y la necesidad de calcular grupos de homotopía en imágenes digitales para el segundo.

Otra posible definición de que es preservar la topología, reseñada en [18], es la que establece que " si una IDB P_1 puede ser obtenida de una IDB P_2 mediante el borrado de un conjunto de puntos negros, entonces la transformación de P_1 a P_2 conserva la topología si y solo si el análogo continuo $C(P_2)$ puede ser obtenido mediante retracto de deformación

si y solo si el análogo continuo $C(P_2)$ puede ser obtenido mediante retracts de deformación del análogo continuo $C(P_1)$ ". Para aceptar esta definición habría que considerar el análogo continuo como un espacio de Hausdorff, ya que si queremos usar esta definición dentro de la estructura de poliedro (conservándola) se presentan algunos problemas (que veremos detallados en el Capítulo V de esta memoria).

La definición primera no tiene mucho sentido desde el punto de vista topológico. Pero, por otra parte, en el Análogo Continuo de una IDB según [4], o más precisamente, en el marco del capítulo II de esta memoria, en los espacios (poliedros generalizados) que son Realización Geométrica de los espacios celulares que nosotros considerábamos y entre los que se tenía una equivalencia de categorías, el resultado de eliminar un punto simple es exactamente lo que se conoce en PL-Topología como un colapso elemental (ver [11]). Así pues, siguiendo a Kong en [17], podemos definir equivalentemente que una operación sobre una IDB preserva la topología si la operación produce un colapso en las realizaciones geométricas de las imágenes inicial y final.

Finalmente, diremos que un objeto 3-d es colapsable si colapsa a un punto. También lo llamaremos contráctil.

4.3 ADELGAZAMIENTO DE IDB TRIDIMENSIONALES

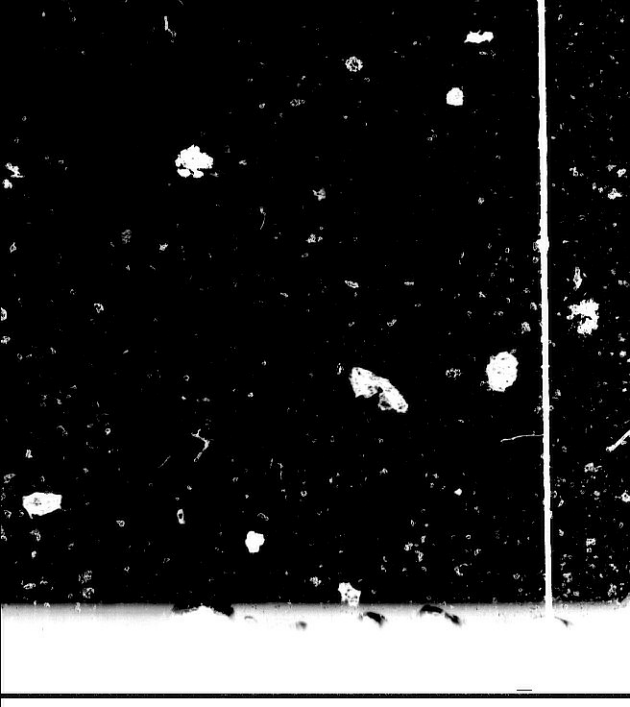
La aportación principal de este capítulo consiste en la generalización para dimensión 3 de los resultados expuestos en el capítulo tercero de esta memoria, consiguiéndoos un algoritmo de adelgazamiento que trabaja en tiempo constante, es de resaltar que a pesar de los muchos esfuerzos realizados (ver [18]) este es el primer algoritmo presentado que logra tal objetivo. En este sentido, la clave de nuestro algoritmo es el siguiente teorema.

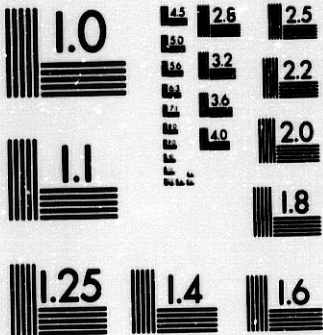
Teorema 4.3.1 Sea P una IDB y sea T un subconjunto del borde de P tal que:

1. Para cada punto p de T el conjunto $N_p \cap (B-\delta)$ es no vacío y contráctil.
2. Para cada punto $q \in \delta$ adyacente a p , existe otro punto r con $r \in (P-\delta) \cap N_p$ y r es adyacente a p .

Entonces el borrado de T preserva la topología.

En el marco teórico en el que situamos nuestro trabajo, y la naturaleza de los





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

4.3 ADELGAZAMIENTO DE IDB TRIDIMENSIONALES

La aportación principal de este capítulo consiste en la generalización para dimensión 3 de los resultados expuestos en el capítulo tercero de esta memoria, consiguiéndose un algoritmo de adelgazamiento que trabaja en tiempo constante, es de resaltar que a pesar de los muchos esfuerzos realizados (ver [18]) este es el primer algoritmo presentado que logra tal objetivo. En este sentido, la clave de nuestro algoritmo es el siguiente teorema.

Teorema 4.3.1 Sea P una IDB y sea T un subconjunto del borde de P tal que:

1. Para cada punto p de T el conjunto $N_p \cap (B-\delta)$ es no vacío y contráctil.
2. Para cada punto $q \in \delta$ adyacente a p , existe otro punto r con $r \in (P-\delta) \cap N_p$ y r es adyacente a p .

Entonces el borrado de T preserva la topología.

En el marco teórico en el que situamos nuestro trabajo, y la naturaleza de los

ADELGAZAMIENTOS EN 3D

conjuntos que manejamos en las hipótesis del teorema, se tiene que el hecho de que un conjunto sea colapsable y el hecho de ser contráctil son equivalentes.

En efecto:

La demostración se basa en el hecho de que cualquier punto de T es simple y de que el borrado de un punto de T no afecta a otros puntos de T . En otras palabras, cualquier orden en los puntos de T define una secuencia de colapsos elementales en P .

Las hipótesis del teorema, que están enunciadas en el grafo de adyacencias del tipo (26,6), se pueden trasladar fácilmente a la estructura de espacio celular como las condiciones de la definición 5.3.1 en la que $n = \{0,1\}$ y los objetos de la imagen son un cerrado.

Con la definición 3.3.1 y en la línea del teorema anterior, generalizamos las definiciones 3.3.2 y 3.3.3 y el teorema 3.3.1 de la siguiente forma:

Definición 4.3.1 Un punto p perteneciente a los objetos de una IDB se dirá primer punto base si no verifica alguna de las siguientes condiciones:

1. El conjunto $\{q \in N_p; L(q) > L(p)\}$ es no vacío y contráctil.
2. Para cualquier punto q adyacente a p y tal que $L(q) = L(p)$, existe otro punto r en N_p adyacente a q con $L(r) > L(q)$.

Definición 4.3.2 En una IDB llamaremos segundos puntos base a los puntos p de B que verifiquen las condiciones:

1. $p \notin B_1(P)$.

2 Existe $q \in (B_1(P)) \cap N_p$ con $L(q) < L(p)$ tal que q no es adyacente a ningún punto del conjunto $\{r \in N_p; L(r) > L(p)\}$.

Aquí seguimos llamando $B_1(P)$ y $B_2(P)$ a los conjuntos de los primeros y segundo puntos base respectivamente y $B(P)$ al conjunto de todos los puntos base. De forma análoga a la del capítulo anterior se tiene el siguiente teorema:

Teorema 4.3.2 En una imagen digital binaria tridimensional P , $B(P)$ es el esqueleto de P .

En efecto:

En primer lugar, es fácil comprobar que $B_2(P)$ es parte del esqueleto de p . Queda solamente ver que cualquier punto del esqueleto es un punto base.

Sea p un punto del esqueleto de p con $L(p) = i$, después de $i+1$ iteraciones del teorema 4.3.1. obtendríamos que p es un punto de la frontera de la imagen resultante y que p es no borrrable, así pues no verifica alguna de las condiciones del citado teorema. Si ocurre que p no verifica la primera condición o ningún punto en su entorno tiene nivel mayor que el, entonces p es primer punto base. De otra forma, es inmediato comprobar que p es segundo punto base, y por tanto, en cualquier caso, p es un punto base. ■

En esta nueva situación tenemos igualmente el algoritmo ESQUELETO.

ESQUELETO(P)

1. Calcular el nivel de cada punto en P
2. Encontrar el conjunto $B_1(P)$.
3. Encontrar el conjunto $B_2(P)$.

Siguen siendo ciertos para este caso la proposición 3.3.1 y el teorema 3.3.2., que aquí enunciamos como proposición 4.3.1 y teorema 4.3.3, respectivamente.

Proposición 4.3.1. Es posible encontrar el esqueleto de una imagen digital binaria tridimensional P con N puntos en tiempo $O(N)$ con $O(N)$ tiempo de preproceso y de almacenamiento.

Seguimos teniendo que cada operación puede ser realizada en todos los puntos en paralelo y que por tanto el algoritmo ESQUELETO es eficiente sobre un ordenador celular. Y si en el primer paso se sigue necesitando un tiempo proporcional al mayor de los niveles de los puntos en P, los pasos siguientes nuevamente pueden ser realizados en un tiempo constante. De todo ello se deduce el siguiente teorema.

Teorema 4.3.3. Es posible encontrar en paralelo el esqueleto de una imagen digital binaria tridimensional P con N puntos en tiempo $O(1)$ con tiempo de preproceso $O(d)$ y $O(N)$ procesadores y almacenamiento, donde d denota el mayor nivel en P .

CAPITULO V
IMAGENES NO BINARIAS.
ADELGAZAMIENTOS.

5.1 INTRODUCCION.

Hasta ahora, cuando hemos hablado de topología digital, nos hemos referido a una terna de la forma (X,T,B) donde (X,T) era un espacio topológico numerable y B un subconjunto de X , que con la topología restringida constituía en subespacio de los objetos. La representación de esta terna es lo que hemos llamado una imagen digital, o más propiamente una imagen digital binaria, donde al conjunto B se le asignaba el color negro y a su complementario el color blanco.

Nos planteamos ahora el problema de considerar imágenes digitales cuyo conjunto de objetos posea más de un color; podemos pensar en imágenes con una escala de grises o más generalmente imágenes con un número finito de colores. En principio podemos seguir considerando una terna de la forma (X,T,B) donde ahora B es una familia finita de subconjuntos disjuntos B_1, B_2, \dots, B_n , y cada B_i sería representado por los elementos de color i .

En capítulos anteriores hemos analizado la relación existente entre la imagen digital binaria, como conjuntos de pixels de pantalla junto con un grafo de adyacencias;

IMAGENES NO BINARIAS

que es la base de los razonamientos, que decimos topológicos, sobre dicha imagen. Concretamente, en dos dimensiones, nos hemos referido al grafo de adyacencias de los tipos (4,8) y (8,4) como criterios globales correctos, en relación al problema planteado por la curva de Jordan, de uso extendido y otro de tipo (6,6) como criterio global correcto y homogéneo.

Para el caso que nos ocupa ahora, donde además los objetos de un determinado color pueden ser considerados separadamente del resto de los colores, se hace necesario elegir un buen tipo de adyacencias a fin de evitar nuevamente la paradoja de la curva de Jordan en cada uno de los colores con respecto a los demás. La figura 5.1 ilustra el problema de elegir un tipo de adyacencias para cada uno de los colores de forma global, los números 1, 2 y 3 indican el color de cada pixel.

2	1	2	3	2	3
1	2	1	2	1	2
2	1	2	3	2	3
1	3	1	2	3	1
3	1	3	3	2	3

fig.5.1

Una buena elección de tipo global para el grafo de adyacencias podría ser el determinado por la topología exagonal u otro de los sugeridos por Kowalewsky en * como "regla de etiquetado por el máximo o por el mínimo".

IMAGENES NO BINARIAS

Después de abordar el problema de dotar de una topología y de su grafo correspondiente a la imagen no binaria, debe tratarse de clarificar el concepto de adelgazamiento para esta situación. Así pues, tiene sentido que los objetos de un determinado color adelgacen con respecto a los demás colores, es decir respecto a los demás objetos y respecto al fondo (el fondo común a todos los objetos sería un cierto B_0). Sin embargo carece de sentido el adelgazamiento simultáneo de todos los objetos respecto a todos los colores, ya que el espacio ambiente, la pantalla, permanece fijo y el adelgazamiento de un objeto conlleva una transformación de su complementario en ese espacio.

En un lenguaje intuitivo podemos referirnos a esta situación diciendo que cuando un objeto adelgaza se produce un engrosamiento en el complemento de dicho objeto, y esto hace que este concepto no se pueda asimilar al concepto topológico de retracción como han pretendido algunos autores.

De lo dicho anteriormente se desprende la necesidad de hacer algunas puntualizaciones que clarifiquen el planteamiento de nuestro problema. Así pues en una terna de la forma (X, T, B) donde $B = \{B_i\}_{i=1, \dots, n}$ y $B_0 = X - B$, si llamamos n al conjunto de índices de los colores $n = \{0, 1, 2, \dots, n\}$, hay que asegurarse de que cualesquiera que sean los índices $i, j \in n$ con $i \neq j$, entonces si los objetos de color i pueden adelgazar respecto al color j , no ocurre que los objetos de color j puedan adelgazar respecto del color i . En ese caso para cualesquiera subconjuntos A_1 y A_2 con $A_1 \subset B_i, A_2 \subset B_j$ y para

IMAGENES NO BINARIAS

todo elemento $x \in \text{Fr}(A_1) \cap \text{Fr}(A_2)$ es $x \notin A_2$ (es decir si $(x \in \text{Fr}(A_1) \cap \text{Fr}(A_2)) \subset (A_1 \cup A_2)$ entonces $x \in A_1$). Decir esto es tanto como asegurar que debe existir una relación entre los índices de n , al menos una prioridad de unos respecto de otros, que tenga una repercusión en la topología de (X, T) .

5.2 COLOR. TRANSFORMACIONES CONTINUAS DE COLOR

En este apartado tomaremos las consideraciones que hemos hecho en el anterior y trataremos de situarnos en una posición suficientemente general, que nos permita rigORIZAR los conceptos y las puntualizaciones que allí se vislumbraban.

Antes de continuar veamos un ejemplo con el que ilustraremos la necesidad de revisar el concepto de adelgazamiento y nos permitirá exponer intuitivamente los propósitos de este apartado.

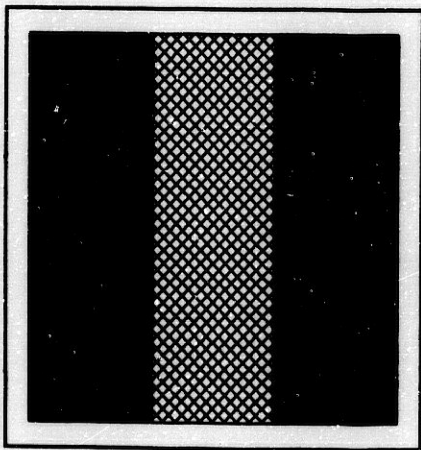


fig.5.2

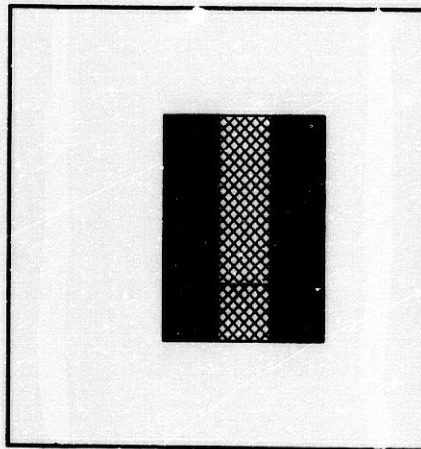


fig.5.3

IMAGENES NO BINARIAS

La figura 5.2 muestra una bandera de dos colores sobre un fondo y la figura 5.3 muestra lo que podría ser un adelgazamiento de la figura 5.2. Notaremos los colores que aparecen por 0 para el fondo, 1 para la parte lisa y 2 para la parte punteada. Observamos que los colores 1 y 2 han adelgazado sobre el fondo 0, y que además el color 2 ha adelgazado sobre el color 1; también vemos que el color 1 en la figura 5.3 está situado en parte sobre lo que era color 2 en la figura 5.2, así pues no se puede tratar de un retracto de deformación la operación sobre el color 1. Siguiendo con nuestra explicación intuitiva los objetivos, podemos concluir que la operación ha consistido en que "algunos pixels de color i , vecinos de otros de color j , con $j < i$, se han sustituido por estos de forma que la topología de los subespacios B_1 , B_2 y $B_1 \cup B_2$ se ha mantenido". Naturalmente la operación habría tenido otro resultado si hubiésemos cambiado el orden de los índices o si la situación relativa de los subconjuntos de color hubiese sido distinta, como puede verse en la figura 5.4.

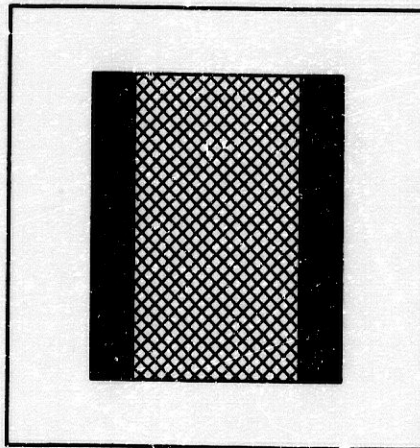


fig. 5.4

El ejemplo anterior nos sugiere que debemos tratar nuestro problema como un

IMAGENES NO BINARIAS

problema de particiones y operaciones sobre el conjunto de las particiones.

Sea (X, T) un espacio topológico y sea $n = \{0, 1, \dots, n\} \subset \mathbb{N}$, en n definimos una topología T' por $T' = \{[0, i]; i \in n\}$.

Definición 5.2.1 Decimos que una partición P de X de $n+1$ elementos $P = \{A_0, A_1, \dots, A_n\}$ es compatible con la topología T' si la aplicación f , definida :

$$f: (X, T) \rightarrow (n, T')$$

$$f(x) = \sum_{i=1}^n i \cdot \chi_i(x)$$

donde χ_i es la aplicación característica de A_i , es continua.

Si como hasta ahora en X se ha considerado una topología T que lo dotaba de espacio parcialmente ordenado, al que fácilmente se le podía dotar de una función dimensión, que lo convertía en un espacio celular y en n hemos tomado la topología $T' = \{[0, i]; i \in n\}$, la compatibilidad de una partición P representada por una aplicación característica f se traduce en que un color i ($f^{-1}(i)$) es cerrado sobre cualquier color j siempre que j sea menor que i y, recíprocamente, es abierto sobre cualquier color j siempre que j sea mayor que i . Esto da una regla global para la asignación de un índice a los elementos de baja dimensión (vértices y aristas), en función del índice del elemento de dimensión máxima al que bordean (caras).

IMAGENES NO BINARIAS

Las siguientes figuras 5.5 y 5.6 ilustran esa regla.

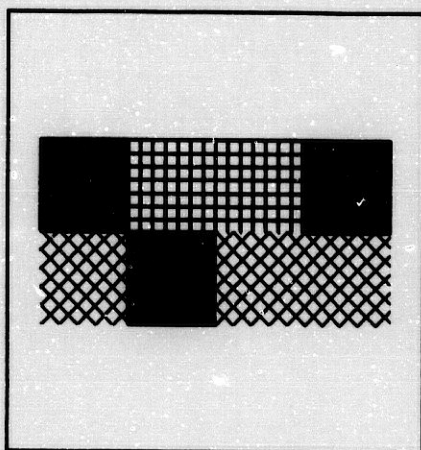


fig.5.5

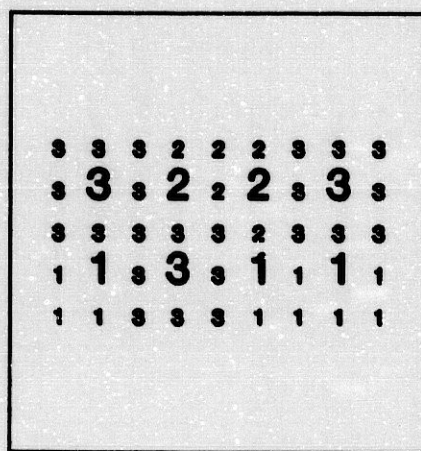


fig.5.6

Para poder hablar de transformaciones continuas de una partición P compatible con una topología T' , necesitamos dotar de una topología al conjunto de tales particiones.

Sea $\text{Hom}(X, \mathfrak{n}) = \{f: (X, T) \rightarrow (\mathfrak{n}, T'); f \text{ continua}\}$. Necesitamos dotar a este conjunto de una topología que de significado a lo que podría ser un cambio continuo de color. Una transformación continua entre una aplicación f_0 y otra aplicación f_1 ,

IMAGENES NO BINARIAS

ambas pertenecientes a $\text{Hom}(X;n)$, puede ser mirado como una homotopía entre ellas o bien como la existencia de un camino continuo que las une en $\text{Hom}(X;n)$ con alguna topología $T_{x,n}$. La topología compacto-abierta nos aseguraría que cualquier homotopía entre dos de tales aplicaciones de X a n determina un camino continuo en $\text{Hom}(X;n)$, sin embargo esto carece de sentido cuando el espacio (n,T') es contráctil (como es el caso), ya que cualesquiera dos aplicaciones continuas de (X,T) a este espacio serán homotópicas.

También, en la nota 5.2, podemos encontrar una topología $T_{x,n}$ en la que se da una situación recíproca. Se tiene que el camino α , del intervalo unidad I en $\text{Hom}(X;n)$, es homotopía entre las aplicaciones $\alpha(0)$ y $\alpha(1)$ y además dos aplicaciones homotópicas pueden no estar unidas por uno de tales caminos.

NOTA 5.2. Sea $\text{Hom}(X,n) = \{f: (X,T) \rightarrow (n,T'); f \text{ continua}\}$. Dotamos a $\text{Hom}(X,n)$ de una topología $T_{x,n}$ mediante una familia de subconjuntos V^f definidos:

$$V^f = \{g \in \text{Hom}(X,n); \forall x \in X, \forall O \in T, \exists O' \in T, x \in O' \subset O; g(O') \subset f(O)\}$$

Está claro que para cada f perteneciente al conjunto de particiones, se tiene que $f \in V^f$ y por tanto la unión de la familia $\{V^f; f \in \text{Hom}(X,n)\}$ es $\text{Hom}(X,n)$.

Por otra parte, dadas $f, g, h \in \text{Hom}(X,n)$ de forma que $h \in V^f \cap V^g$, se tiene que para cada $x \in O$ con $O \in T$, existen O', O'' con $x \in O' \subset O$ de forma que $h(O') \subset f(O)$, y también $x \in O'' \subset O$ con $h(O'') \subset g(O)$, luego para cada $x \in O' \cap O''$ y cada

IMAGENES NO BINARIAS

$h' \in V^h$ se verifica que existe $O''' \subset O' \cap O''$ y $h(O''') \subset h(O') \subset f(O)$, análogamente $h'(O''') \subset g(O)$. Así pues $h' \in V^f \cap V^g$.

Queda de esta forma comprobado que la familia $\{V^f; f \in \text{Hom}(X, n)\}$ es base de una cierta topología $T_{X,n}$.

Proposición 5.2.1. La aplicación $F: (X \times \text{Hom}(X, n), T'') \rightarrow (n, T')$ definida por

$$F(x, f) = f(x)$$

es continua, donde T'' es la topología producto de T y $T_{X,n}$.

Veamos que la topología T'' hace continua a la aplicación F . Sean $O' \in T'$ y $F^{-1}(O') = \{(x, f); f(x) \in O'\}$. Para cada par $(x, f) \in F^{-1}(O')$ es $x \in f^{-1}(O') = O \in T$, entonces existe $O'' \in T$ con $x \in O'' \subset O$, de tal forma que para cada $g \in V^f$ se tiene que $g(O'') \subset f(O) \subset O'$, luego $F(O'', V^f) \subset O'$. Así se tiene que $F^{-1}(O') \in T''$ y la aplicación F es continua.

Sea ahora α un camino continuo en $\text{Hom}(X, n)$

$$\alpha: I \rightarrow \text{Hom}(X, n)$$

definimos la aplicación L

$$L: X \times I \rightarrow X \times \text{Hom}(X, n)$$

IMAGENES NO BINARIAS

como $L(x,t) = (x, \alpha(t))$, entonces la aplicación H

$$H = F \circ L: X \times I \rightarrow n$$

es continua y determina una homotopía entre las aplicaciones características $f_0 = \alpha(0)$ y $f_1 = \alpha(1)$.

Sin embargo no es cierto que toda homotopía H entre dos aplicaciones f_0 y f_1 determine un camino continuo entre ellas en $\text{Hom}(X, n)$. Esto quedará de manifiesto después de ver la siguiente proposición.

Proposición 5.2.2 Sea α un camino en $(\text{Hom}(X, n), T_{X, n})$ con $\alpha(I) = \{f_t, t \in I\}$.

Entonces: $\forall t \in I, \exists \epsilon \in \mathbb{R}^+$ tal que $\forall i \in N, \forall x \in \text{int}(f_t^{-1}(i))$ se tiene que si

$$|t - t'| < \epsilon \text{ entonces } f_{t'}(x) = i.$$

En efecto:

Si $x \in \text{int}(f_t^{-1}(i))$ entonces existe $O \subset T$ con $x \in O$ y $f_t(O) \subset \{i\}$. Así para cada $g \in \mathcal{V}^i$ es $g(O) \subset f_t(O)$, por la continuidad de α se obtiene el resto.

Esto significa que en cada instante t , en un entorno suyo, el interior de cada color permanece de ese color y que las variaciones de color en un punto x ocurren

IMAGENES NO BINARIAS

cuando x está en la frontera de un color para alguna partición f_i . Es decir que si existe $x \in X$, $t \in I$ con $f_i(x) = i$ tal que $\forall \varepsilon \in \mathbb{R}^+$ se tiene que $\exists t'$ con $|t-t'| < \varepsilon$ y $f_i(x) \in n - \{i\}$, necesariamente ha de ocurrir que $x \in \text{Fr}(f_i^{-1}(i))$.

Tomando la topología T' , definida más arriba, en n el espacio topológico (n, T') es contráctil y por tanto cualesquiera dos aplicaciones f, g de (X, T) a este espacio son homotópicas, no obstante, en virtud de la proposición anterior, podemos encontrar dos de tales aplicaciones características entre las que no puede haber un camino continuo en $\text{Hom}(X, n)$, donde se ha tomado la topología $T_{x,n}$, basada en la familia de conjuntos V^i .

Las figuras 5.7 y 5.8 muestran dos imágenes digitales, a las que corresponden dos particiones de un espacio topológico, cuyas aplicaciones características están en la situación descrita más arriba.

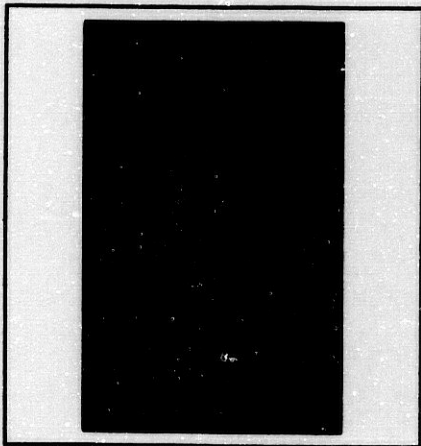


fig.5.7.

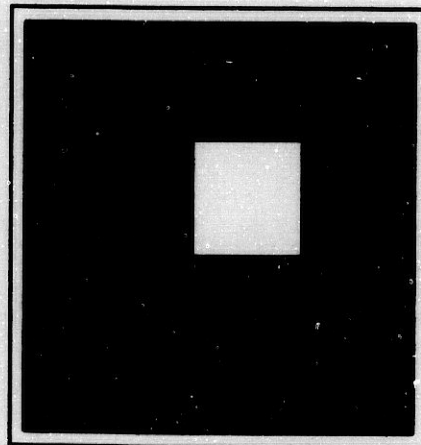


fig.5.8

5.3 RESULTADOS PRINCIPALES.

En los apartados anteriores de este capítulo hemos visto la conveniencia de considerar los colores de los objetos de una imagen digital como una familia de particiones en su espacio topológico asociado, y también la necesidad de dotar al conjunto de tales particiones de una topología adecuada, que nos permita hablar de transformaciones continuas del color. Con este concepto se ha querido aludir a la "deformación de objetos de un color en objetos del mismo color" de forma que se conservan las propiedades topológicas de estos objetos (subespacios) y la "posición relativa" de éstos con el resto de los objetos de cualquier color.

En todo lo que sigue consideraremos un espacio topológico (X, T) finito con estructura de espacio celular y una partición $P = \{A_0, A_1, \dots, A_n\}$ de X .

En $n = \{0, 1, \dots, n\} \subset \mathbb{N}$ consideraremos la topología $T' = \{[0, i]; i \in n\}$ y

IMAGENES NO BINARIAS

supondremos que la aplicación característica de P es compatible con la topología T' (Definición 5.2.1). Esto no supone ninguna restricción ya que la partición P proviene de una partición de una imagen digital, representando los $n+1$ colores de la misma. Es decir P está dado sobre los elementos de máxima dimensión de X , y tan solo es necesario decidir la pertenencia de los elementos de baja dimensión a cada uno de los conjuntos A_i . Elegir convenientemente esta pertenencia es asegurar la compatibilidad de P con T' .

La topología para $\text{Hom}(X,n)$ definida en la nota 5.2 tan solo ha servido para explicar que la existencia de un camino continuo entre dos aplicaciones características es una condición más fuerte que la existencia de una homotopía entre ellas, sin embargo la proposición, enunciada en esa nota, aseguraba que los cambios de color en un subconjunto de una partición solo pueden ocurrir en la frontera del mismo; lo cual no tiene ninguna correspondencia con el objeto del que proviene en una imagen digital, ya que en ésta solo se reflejarían los cambios producidos en los elementos de dimensión dos (pixel).

Definiremos para cada aplicación f una nueva familia V^f , a partir de la cual construiremos una topología que será más útil para nuestros propósitos, aunque no será tan simple y bonita como la anterior.

Definición 5.3.1 Para cada aplicación $f \in \text{Hom}(X,n)$ definimos $V^f = \{g \in \text{Hom}(X,n), \text{ t.q. } \forall x \in X \text{ con } g(x) \neq f(x) \text{ se tiene } 1,2,3\}$

IMAGENES NO BINARIAS

1) $g(O'_x) \subset f(O'_x)$

2) se tiene a) o b)

a) $g(O_x) \subset f(O_x)$

b) $\exists y \in fr(x); g(y) = g(x), g(y) \neq f(y)$

3) $fr(O_x) \cap Int(f^{-1}(f(x))) \neq \emptyset$ y contráctil.

$fr(O_x) \cap f^{-1}(f(x))$ es contráctil.

Donde O_x representa el menor abierto que contiene a x , y O'_x es el menor abierto que contiene a la clausura de $\{x\}$ ($Cl(\{x\})$). Las figuras 5.9 y 5.10 muestran a modo de ejemplo O_x y O'_x , respectivamente, para un vértice, una arista y una cara 2-dimensional.

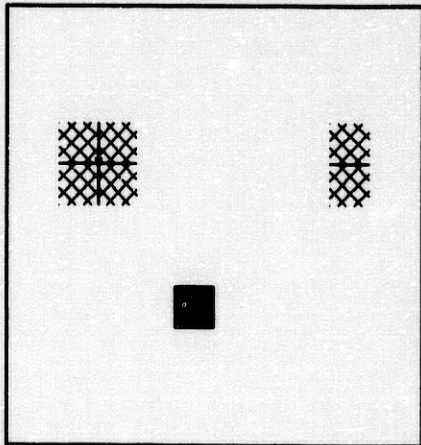


fig.5.9 O_x

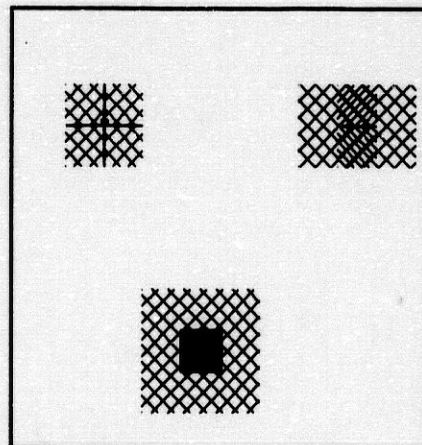


fig.5.10 O'_x

IMAGENES NO BINARIAS

Lema 5.3.1. $\forall g \in V'$ y $\forall x \in X$ tal que $f(x) \neq g(x)$ se tiene alguna de las siguientes afirmaciones:

a) $x \in (\text{fr}(f^{-1}(f(x))) \cap \text{fr}(f^{-1}(g(x))))$

b) $\exists y \in \text{fr}(f^{-1}(f(x)))$, y no se puede separar de x , $f(y) = f(x)$, $g(y) = g(x)$.

Observación : Que y no se pueda separar de x significa que $\forall O \in T, y \in O$, entonces $x \in O$.

En efecto:

Sea $x \in X$ tal que $f(x) \neq g(x)$ y $g \in V'$. Si $\{x\} = \text{Cl}(\{x\})$ entonces se tiene que $O'_x = O_x$ y por la condición 1 de la definición 5.3.1 es $g(O_x) \in f(O_x)$, en particular $g(x) \in f(O_x)$. Luego $O_x \cap f^{-1}(f(x)) \neq \emptyset$, y como se tiene que

$\forall O \in T$ tal que $x \in O$ es $O_x \subset O$ ocurre que

$$x \in \text{Fr}(f^{-1}(f(x))) \cap \text{Fr}(f^{-1}(g(x))).$$

Si no ocurre que $g(O_x) \subset f(O_x)$ entonces, por lo anterior, $\{x\}$ no es cerrado y por la segunda parte (b) de la condición 2 de la definición 5.3.1 $\exists y_1 \in \text{Fr}(\{x\})$ con $g(y_1) = g(x)$, $g(y_1) \neq f(y_1)$. Si $\{y_1\}$ no es cerrado, reiterando el mismo proceso, encontramos y_1, y_2, \dots, y_k de forma que $g(y_i) = g(x)$, $g(y_i) \neq f(y_i)$ con $y_i \in \text{Fr}(\{y_{i-1}\})$ y en consecuencia $y_i \in \text{Fr}(\{x\})$ con $\{y_i\}$ cerrado ($k \leq \dim(X, T)$) o bien $g(O_{y_k}) \subset f(O_{y_k})$. Así pues existe un punto y_k en la frontera de $f^{-1}(f(x))$ que no se puede separar de x . ■

IMAGENES NO BINARIAS

Lema 5.3.2 Para toda aplicación f , para cada $g \in V_f$ y para cada $x \in X$ tal que $f(x) \neq g(x)$ se tienen:

a) $g(x) < f(x)$.

b) Si $x \in \text{Fr}(f^{-1}(f(x)))$ entonces $\exists y \in \text{Int}f^{-1}(f(x))$, $x \in \text{fr}(\{x\})$,

con $g(y) = g(x)$.

En efecto:

a) Por el lema 5.3.1 y la condición 2 de la definición 5.3.1

$x \in \text{Fr}(f^{-1}(f(x))) \cap \text{Fr}(f^{-1}(g(x)))$ o bien existe $y \in \text{Fr}(\{x\})$,

$y \in \text{Fr}(f^{-1}(f(x))) \cap \text{Fr}(f^{-1}(g(x)))$, $g(y) = g(x)$, $g(y) \neq f(y)$; por tanto podemos suponer para simplificar que $x \in \text{Fr}(f^{-1}(f(x))) \cap \text{Fr}(f^{-1}(g(x)))$.

En este caso, dado que $f^{-1}([g(x), n])$ es cerrado, $\text{Fr}(f^{-1}(g(x))) \subset f^{-1}([g(x), n])$

y en particular $\text{Fr}(f^{-1}(f(x))) \cap \text{Fr}(f^{-1}(g(x))) \subset f^{-1}([g(x), n])$ con lo que

$x \in f^{-1}([g(x), n])$ y por tanto $f(x) \in [g(x), n]$.

b) Es consecuencia inmediata de a) y de la continuidad de g . ■

Lema 5.3.3 Sea $g \in V^f$, $x \in X$, $g(x) \neq f(x)$, siendo el elemento x perteneciente a $\text{Fr}(f^{-1}(f(x))) \cap \text{Fr}(f^{-1}(g(x)))$.

Entonces $g(O_x) \subset f(O_x)$.

IMAGENES NO BINARIAS

En efecto:

Si $\{x\}$ es cerrado, entonces por la condición 1 y por ser en este caso $O'_x = O_x$ se tiene la afirmación.

Si $\{x\}$ no es cerrado entonces, dado que $\{x\}$ tampoco puede ser abierto, según nuestras hipótesis, por la condición 2, apartado a), de la definición 5.3.1 se tiene que $g(O_x) \subset f(O_x)$, o bien reiterando si es necesario el apartado b) llegaríamos a que $\exists y \in \text{Fr}(\{x\})$, $\{y\}$ cerrado con $g(O_y) \subset f(O_y)$, además por ser $y \in \text{Fr}(\{x\})$ se tiene que $O_x \subset O_y$. ■

Repasamos a continuación el significado de las condiciones de la definición 5.3.1. y de los lemas 5.3.1, 5.3.2 y 5.3.3 : la condición 1 prevé que un elemento pueda cambiar de un color a otro siempre que este último se halle presente en un entorno "suficientemente pequeño" de él, pero no necesariamente en el más pequeño, es decir no en todo entorno de él (fig.5.11.).

La condición 2 asegura que si el nuevo color de un elemento no está presente en todos los entornos de este elemento x , (fig.5.11), entonces existe otro elemento y que si está en esa situación, es inseparable de x y además cambia de color junto a x (fig.5.12). Ambas condiciones aseguran que un elemento cambia de color si está en la frontera de ese color o en un entorno mínimo de la frontera. Es decir, no se generan agujeros en ninguno de los objetos. Estas afirmaciones están demostradas en el primer

lema.

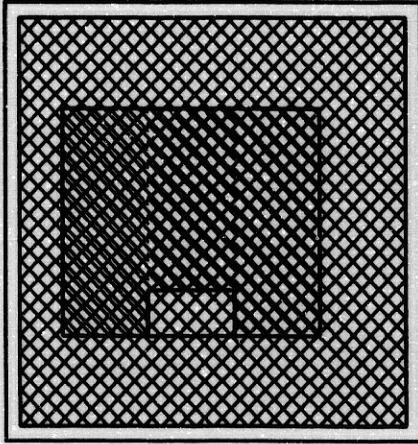


fig.5.11.

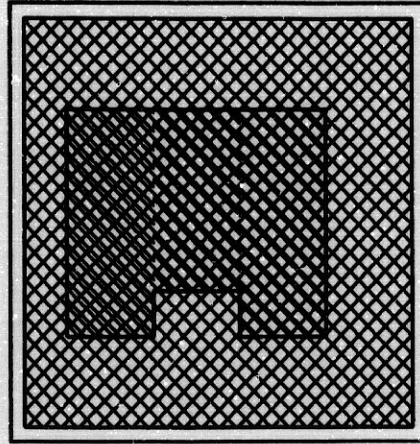


fig.5.12.

El segundo lema matiza al primero y da una especie de recíproco, asegurando que el cambio de color no puede ocurrir solo en la frontera sino que también tiene que darse en un entorno mínimo de ésta. Por otra parte, si convenimos en interpretar el orden de los colores como " $<$ " = "menos oscuro que", resulta que un elemento solo puede cambiar de un color a otro menos oscuro.

La condición 3 servirá para mantener la conexión de cada objeto. Esto significa que no solo los objetos no se pueden romper (0-conexión) sino que, en dimensiones

IMAGENES NO BINARIAS

superiores a dos, no se pueden formar "túneles" y en general que se mantiene la n -conexión.

Definición 5.3.2. Para cada $f \in \text{Hom}(X, n)$ tomamos $V(V^f) = \bigcup_{g \in V^f} V^g$.

Notaremos $V^0(f) = V^f$ y definimos $V^n(f)$ como $V(V^{n-1}(f))$. Con la notación anterior llamaremos $O(f)$ a:

$$O(f) = \bigcup_{n \in \mathbb{N}} V^n(f)$$

Proposición 5.3.1. Para cualesquiera dos aplicaciones $f, g \in \text{Hom}(X, n)$ tales que $g \in O(f)$ se tiene que $O(g) \subset O(f)$.

En efecto:

Si $g \in O(f) = \bigcup_{n \in \mathbb{N}} V^n(f)$, entonces $\exists n \in \mathbb{N}, g \in V^n(f)$ y por tanto $V(g) \subset V^{n+1}(f)$.

Luego $\forall n \in \mathbb{N}$ es $V^n(g) \subset O(f)$ y en consecuencia $O(g) \subset O(f)$. ■

Proposición 5.3.2. Para cualesquiera dos aplicaciones $f, g \in \text{Hom}(X, n)$ tales que $g \in O(f)$ y $f \in O(g)$, se tiene que $f = g$.

Proposición 5.3.3. Para toda aplicación $g \in O(f)$ existe una familia de aplicaciones $\{f_i\}_{i=0,\dots,n}$ de forma que se verifican:

a) $f_0 = f, f_n = g.$

b) $f_i \in V^i(f).$

c) $f_i \in V^{i+1}(f_{i+1}).$

En efecto:

La demostración es inmediata a partir de la definición de $O(f)$ y de la proposición 5.3.1 ya que al ser $g \in O(f)$ $\exists n \in \mathbb{N}, g \in V^n(f)$ luego $\exists f_{n-1} \in V^{n-1}(f)$ y $g \in V(f_{n-1})$, reiterando el proceso para cada una de las f_i llegaríamos a $f_0 = f.$ ■

Proposición 5.3.4. La familia $\{O(f); f \in \text{Hom}(X,n)\}$ es base para una topología $T_{X,n}$ en $\text{Hom}(X,n).$

En efecto:

Se tiene trivialmente que para cada $f \in \text{Hom}(X,n)$ es $f \in O(f)$ y por la proposición 5.3.1. se tiene que para cada h tal que $h \in O(f) \cap O(g)$, es $O(h) \subset O(g)$ y también $O(h) \subset O(f)$ de donde $O(h) \subset O(f) \cap O(g).$ ■

Proposición 5.3.5 La aplicación F definida:

$$F: (X \times \text{Hom}(X, n), T') \rightarrow (n, T')$$

$$F((x, f)) = f(x)$$

es continua. Donde T'' es una topología que tiene por base el producto $T \times T_{x, n}$.

En efecto:

Para cada abierto O' de T' es $F^{-1}(O') = \{(f, x); f(x) \in O'\}$ y $F^{-1}(O') = O \in T$, entonces para cada $x \in O$ y para cada $g \in O(f)$ es $g(x) \leq f(x)$, por la proposición 5.3.3., y por tanto $g(O) \subset O'$, de donde resulta que $F((O, O(f))) \subset O'$. ■

Proposición 5.3.6. Todo camino continuo en $(\text{Hom}(X, n), T_{x, n})$ es constante.

En efecto.

Sea α una aplicación continua desde el intervalo unidad I con la topología usual T_I restringida, hasta el espacio $(\text{Hom}(X, n), T_{x, n})$ y sean f_0 y f_1 respectivamente $\alpha(0)$ y $\alpha(1)$. Sea $t \in I$, consideramos $O(f_t)$, con $f_t = \alpha(t)$, entonces para cada $t' \in \alpha^{-1}(O(f_t)) = A \in T_I$ ha de ser $\alpha(t') \in O(f_t)$. Tomando un nuevo t'' en la frontera de A y reiterando el razonamiento anterior sobre el compacto I llegamos a que $\alpha(I) \subset O(f)$ y como t es cualquiera resulta que f_0 pertenece a $O(f_1)$ y recíprocamente f_1 pertenece a $O(f_0)$ de donde $O(f_0)$ es igual a $O(f_1)$ y por tanto $f_0 = f_1$. ■

Definición 5.3.3. Para cada aplicación $f \in \text{Hom}(X, \mathbb{R})$ llamamos adelgazamiento continuo de f a cualquier aplicación $g \in O(f)$.

Definición 5.3.4. Sea $\alpha: I \rightarrow \text{Hom}(X, \mathbb{R})$ decimos que α es semicontinua a la derecha si para cada $t \in I$ existe ε real y positivo tal que $t' - t < \varepsilon$ entonces $f_{t'} \in O(f_t)$. Es decir, es continua para la topología dada por los conjuntos de la forma $[t, t + \varepsilon)$ en I .

Proposición 5.3.7. Las afirmaciones siguientes son equivalentes:

- a) g es un adelgazamiento de f .
- b) Existe un camino semicontinuo a la derecha que une f con g .

Sea ahora (X, T) un espacio topológico y A, B dos subconjuntos de X , $B \subset A$, de forma que existe una retracción r de A a B y B es además un retracto fuerte de deformación de A . Podemos considerar la partición $P_0 = \{X - A, A\}$ y la partición $P_1 = \{X - B, B\}$.

Sea $H: A \times I \rightarrow A$ la homotopía entre la aplicación identidad en A y la retracción r , para cada $t \in I$ podemos considerar una partición de la forma

IMAGENES NO BINARIAS

$P_t = \{X - r_t(A), r_t(A)\}$, donde $r_t(A) = H(A, t)$. Si el espacio (X, T) es T_2 , entonces las particiones descritas son compatibles con la topología $T' = \{\{0\}, \{0, 1\}\}$ en $I = \{0, 1\}$. Tenemos entonces una familia de aplicaciones $\{f_t\}_{t \in I}$, correspondiente a la familia de particiones, que en la topología adecuada constituyen un camino continuo entre las aplicaciones f_0 y f_1 .

Podemos también encontrar dos particiones $P_0 = \{X - B, B\}$ y $P_1 = \{X - A, A\}$, con $A \subset B$, de un espacio topológico X , concretamente en un espacio celular, de forma que A no puede ser retracto de B y sin embargo P_1 es un adelgazamiento, o una transformación continua, de P_0 . Véanse las figuras fig.13 y fig.14 .

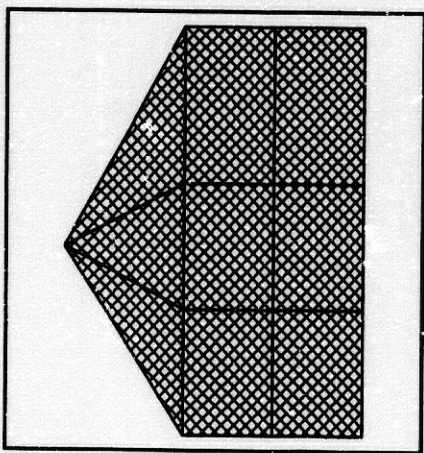


fig.13

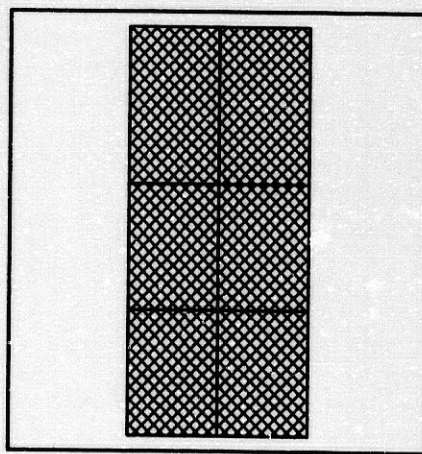


fig. 14.

IMAGENES NO BINARIAS

La razón por la cual el subespacio A que representa la figura 14 no puede ser un retracto del espacio B representado en la figura 13, es que no se puede encontrar una aplicación continua de B en A . Ya que el vértice común a los tres triángulos de $B-A$ no se puede aplicar de forma continua sobre ningún otro elemento de A .

Si comprobamos que en una deformación continua de una partición se conservan las propiedades topológicas del espacio en cuestión, habremos rigORIZADO completamente el concepto de adelgazamiento que ha sido usado hasta este capítulo y además la habremos dado un significado que resulta ser más general que el de retracto fuerte de deformación.

Observación En lo que sigue se entenderá ser " equivalente topológicamente a " como " poseer el mismo tipo de h -conexión ", con $h < \dim(X)$.

Teorema 5.3.1. Sea (X, T) espacio topológico con estructura de espacio celular. Sea $P_0 = \{A_0, A_1, \dots, A_n\}$ una partición de X compatible con la topología T' de $n = \{0, 1, \dots, n\} \subset \mathbb{N}$, $T' = \{[0, i]; i \in n\}$, con aplicación característica f y P_1 un adelgazamiento continuo de P_0 con aplicación característica g .

Entonces:

$\forall i \in \mathbb{N}$ es:

- a) $f^{-1}(i)$ equivalente topológicamente a $g^{-1}(i)$
- b) $f^{-1}([0, i])$ equivalente topológicamente a $g^{-1}([0, i])$

IMAGENES NO BINARIAS

Necesitaremos el siguiente lema.

Lema 5.3.4. Sea f la aplicación característica del teorema 5.3.1. y sea h tal que $h \in V'$. Entonces:

$\forall i \in \mathbb{N}$ es:

- a) $f^{-1}(i)$ equivalente topológicamente a $h^{-1}(i)$
- b) $f^{-1}([0,i])$ equivalente topológicamente a $h^{-1}([0,i])$

En efecto:

Podemos suponer que h y f difieren a lo sumo sobre un elemento de dimensión máxima. Ya que si es $D = \{x_1, \dots, x_m\}$ el conjunto de elementos de dimensión máxima sobre los que difieren h y f , consideraríamos las aplicaciones $h_1, \dots, h_m = h$ tales que h_i difiere de h_{i-1} en $\{x_i\}$ y naturalmente todas ellas pertenecen a V' .

Sea x el único elemento de máxima dimensión tal que $f(x) \neq h(x)$, dado que x no puede pertenecer a $\text{Fr}(f^{-1}(fx))$ se tiene por el lema 5.3.1. apartado b) que $\exists y \in \text{fr}(f^{-1}(f(x)))$, y no se puede separar de x , $f(y) = f(x)$, $h(y) = h(x)$. Además, por la continuidad de h , ha de ser $h(\text{Fr}(\{x\} \cap \text{Fr}(f^{-1}(fx)))) = h(x)$ y por el lema 5.3.2. apartado b) es $(\text{Fr}(\{x\} \cap \text{Fr}(f^{-1}(fx))))$ el conjunto de todos los elementos sobre los que f y h son distintas. Así pues es $h[\text{fr}\{x\} \cap \text{Int}(f^{-1}(f(x)))] = f(x)$ y por el apartado 3) de la definición 5.3.1. es $[\text{fr}\{x\} \cap \text{Int}(f^{-1}(f(x)))] \neq \emptyset$ y contráctil, de donde se sigue nuestra afirmación. ■

IMAGENES NO BINARIAS

La demostración del teorema 5.3.1. se sigue del lema anterior y de la proposición 5.3.3.

Corolario 5.3.1. En las condiciones del Teorema 5.3.1. se tiene:

$\forall i, j \in \mathbb{N}$

a) $f^1([i, n])$ equivale topológicamente a $g^1([i, n])$

b) $f^1([i, j])$ equivale topológicamente a $g^1([i, j])$

Corolario 5.3.2. En el caso de ser (X, T) un espacio de dimensión 2 y $\mathbb{n} = \{0, 1\}$, la definición 5.3.3. y el teorema 5.3.1. aportan una generalización del concepto de adelgazamiento expuesto por Rosenfeld en [18]

APENDICE

APENDICE

Los programas, cuyo listado damos a continuación, han sido realizados en colaboración con los programadores: Dn. Lorenzo Ramos Requena y Dn. Manuel Osorio Herrera y fundamentalmente con el Ldo. en Informática Dn. Luis Lopez Salvador, a quienes agradezco profundamente su dedicación a este trabajo.

PROGRAMA QUE IMPLEMENTA UN ALGORITMO DE ADELGAZAMIENTO SECUENCIAL.

```
CONST XM=319;
      YM=174;
TYPE
  NUMERO=0..3;
  PUNTERO=^I;
  I=RECORD
    BASE:ARRAY[1..XM,1..YM]OF NUMERO;
  END;
  EGA_screen = Array [0..16383] of BYTE;
  CAD3=STRING[3];
VAR PUNTOS,B:PUNTERO;
    panEGA:EGA_Screen absolute $A000:0;
    V:EGA_SCREEN;
    T1,T2,X,Y,MUDA,H,XO,YO,XP,YF,N1,N2:INTEGER;
    CH:CHAR;
    HOR,MIN,SEG,CENT:WORD;
    MIN3,SEG3,CENT3,ABORA:STRING;
    NOMBRE:STRING[15];
    TOPOLOGIA:STRING[3];
```

```
Procedure Iniciar;
  Var Driver , Mode : Integer;
  Begin
    Driver:=Detect;
    Mode:=0;
    InitGraph(Driver,Mode,'..\');
  end;
```

```
PROCEDURE PINTAB;
  BEGIN
    WITH B^DO
      FOR X:=XO-1 TO XF+1 DO
        FOR Y:=YO-1 TO YF+1 DO
```

APENDICE

```

        IF BASE[X,Y]=2 THEN PUTPIXEL(X+XM,Y,YELLOW);
    END;
PROCEDURE PINTAPUNTOS;
BEGIN
    WITH PUNTOS^ DO
    FOR X:=X0-1 TO XF+1 DO
        FOR Y:=Y0-1 TO YF+1 DO
            IF BASE[X,Y]=0 THEN PUTPIXEL(X+XM,Y,BLACK);
        END;
    END;
PROCEDURE PINTAR;
BEGIN
    WITH PUNTOS^ DO
    FOR X:=X0 TO XF DO
        FOR Y:=Y0 TO YF DO
            IF BASE[X,Y]=1 THEN PUTPIXEL(X+XM,Y,RED);
        END;
    END;

PROCEDURE PINTAINVER;
BEGIN
    WITH PUNTOS^ DO
    FOR X:=X0 TO XF+1 DO
        FOR Y:=Y0 TO YF+1 DO
            IF BASE[X,Y]=1 THEN PUTPIXEL(X+XM,Y,WHITE)
            ELSE PUTPIXEL(X+XM,Y,BLACK);
        END;
    END;

Procedure print(x,y:integer;s:string;co:integer);
begin;
    SetColor(co);
    OutTextxy(x,y,s);
end;
PROCEDURE CAJA(TOPOLOGIA:CAD3);
VAR ESP:STRING[4];
    CA,CA2:STRING;
BEGIN
    ESP:='  ';
    SETFILLSTYLE(SOLIDFILL,BLUE);
    BAR(3,YM+15,GETMAXX,GETMAXY);
    SETCOLOR(WHITE);
    RECTANGLE(1,1,639,349);
    LINE(1,YM,639,YM);
    LINE(1,YM+15,639,YM+15);
    RECTANGLE(1,1,XM,YM+15);
    LINE(1,233,639,233);
    LINE(1,291,639,291);
    LINE(130,YM,130,YM+15);
    PRINT(250,236,'MENU DE OPCIONES',WHITE);
    LINE(250,244,378,244);
    PRINT(3,YM+3,' Imagen origen. Topologia usada:      Imagen trasformada',cyan);
    PRINT(XM-40,YM+3,TOPOLOGIA,RED);
    CA:=ESP+'ambia Top. '+ESP+'delgazar.'+ESP+'squeleto.'+ESP+'nvertir.'+ESP+'ordes.';
    CA2:=ESP+'in.'+ESP+'eer fichero.'+ESP+'ision Ampliada.';
    PRINT(3,255,CA,12);

```

APENDICE

```

PRINT(3,270,CA2,LIGHTRED);
PRINT(11,255,'[C]          [A]          [E]          [I]          [B]',LIGHTGREEN);
PRINT(11,270,'[F]    [L]          [V]',LIGHTGREEN);
END;
Procedure VISIONAR;
var x,y:integer;
Procedure Punto(x,y,co:Integer);
var x1,y1:Integer;
label esc;
Begin
    x1:=2*x;
    y1:=2*y;
    if (x1>640) or (y1>360) then goto esc;
    putPixel(x1,y1,co);
esc:
end;
begin
SETACTIVEPAGE(1);
WITH B^ DO
BEGIN
for y:=1 to YM do
for x:=1 to XM do
begin
if BASE[x,y]=1 then begin;punto(x,y,RED);end;
if BASE[x,y]=0 then begin;punto(x,y,lightblue);end;
if BASE[x,y]=2 then begin;punto(x,y,yellow);end;
end;
end;
borrar_ventana(3,0);
print(200,300,'Pulsa la "S" para salir.',lightred);
print(200,310,'Informacion de colores',white);
print(200,320,'--- Puntos bordes',yellow);
print(200,330,'--- Puntos interiores',RED);
SETVISUALPAGE(1);
REPEAT UNTIL (READKEY='S') OR(READKEY='s');
END;
{*****PROCEDIMIENTOS DE CALCULO*****}
Function Borde(x,y:Integer):integer;
var ok:Integer;
LABEL ESC;
begin;
ok:=0;
WITH B^ DO
IF TOPOLOGIA='8x4' THEN
if BASE[x,y]=1 then
begin
ok:=0;
if base[x-1,y]=0 then begin;ok:=1;goto esc;end;
if base[x+1,y]=0 then begin;ok:=1;goto esc;end;
if base[x,y-1]=0 then begin;ok:=1;goto esc;end;
if base[x,y+1]=0 then begin;ok:=1;goto esc;end;
end;

```

APENDICE

```
IF TOPOLOGIA='4x8' THEN
  BEGIN
    ok:=0;
    WITH B^ DO
      IF BASE[X,Y]=1 THEN
        BEGIN
          If BASE[x-1,y]=0 then begin;ok:=1;goto esc;end;
          If BASE[x+1,y]=0 then begin;ok:=1;goto esc;end;
          if BASE[x,y-1]=0 then begin;ok:=1;goto esc;end;
          if BASE[x,y+1]=0 then begin;ok:=1;goto esc;end;
          if BASE[x-1,y-1]=0 then begin;ok:=1;goto esc;end;
          if BASE[x-1,y+1]=0 then begin;ok:=1;goto esc;end;
          if BASE[x+1,y-1]=0 then begin;ok:=1;goto esc;end;
          if BASE[x+1,y+1]=0 then begin;ok:=1;goto esc;end;
        END;
      END;
    ESC:
    borde:=ok;
    end;
  Procedure Bordes;
  var ok:integer;
  begin;
    WITH B^ DO
      for y:=YO to yF do
        for x:=XO to xF do
          begin
            ok:=0;
            ok:=borde(x,y);
            if ok=1 then BASE[x,y]:=2;
          end;
        end;
      Function CondT(x,y:integer):NUMERO; {Calcula la condicion T para ORTO.}
      var ok:Integer;
          XP:Integer;
          n_interior,XX,YY,NUMER:Integer;
          LOCAL:ARRAY[1..3,1..3] OF NUMERO;
      Label esc,LOOP2,ESC1,ESC2;
      begin;
        ok:=0;

        n_interior:=0;
        WITH B^ DO
          if BASE[x,y]=2 then
            begin
              IF TOPOLOGIA='8x4' THEN
                if BASE[x,y]=1 then
                  begin
                    ok:=0;
                    if base[x-1,y]=1 then begin;ok:=1;goto esc;end;
                    if base[x+1,y]=1 then begin;ok:=1;goto esc;end;
                    if base[x,y-1]=1 then begin;ok:=1;goto esc;end;
                    if base[x,y+1]=1 then begin;ok:=1;goto esc;end;
                  end;
            end;
          end;
        end;
```


APENDICE

```

IF TOPOLOGIA='4x8' THEN
  BEGIN
    ok:=0;
    WITH B^ DO
      IF BASE[X,Y]=2 THEN
        BEGIN
          If BASE[x-1,y]=1 then begin;ok:=1;goto esc;end;
          if BASE[x+1,y]=1 then begin;ok:=1;goto esc;end;
          if BASE[x,y-1]=1 then begin;ok:=1;goto esc;end;
          if BASE[x,y+1]=1 then begin;ok:=1;goto esc;end;
          if BASE[x-1,y-1]=1 then begin;ok:=1;goto esc;end;
          if BASE[x-1,y+1]=1 then begin;ok:=1;goto esc;end;
          if BASE[x+1,y-1]=1 then begin;ok:=1;goto esc;end;
          if BASE[x+1,y+1]=1 then begin;ok:=1;goto esc;end;
        END;
      END;
    ESC:
    FILLCHAR(LOCAL,SIZEOF(LOCAL),0);
    WITH B^ DO
      local[2,2]:=BASE[x,y];
      Local[1,1]:=BASE[x-1,y-1];
      local[1,2]:=BASE[x-1,y];
      Local[1,3]:=BASE[x-1,y+1];
      local[2,1]:=BASE[x,y-1];
      local[2,3]:=BASE[x,y+1];
      Local[3,1]:=BASE[x+1,y-1];
      local[3,2]:=BASE[x+1,y];
      local[3,3]:=BASE[x+1,y+1];
      {===== tipo 4x8 ===== }
      if TOPOLOGIA='4x8' then begin

        NUMER:=0;
        for xx:=1 to 3 do
          begin
            for yy:=1 to 3 do
              begin;
                if (local[xx,yy]=1) then begin;numer:=numer+1;end;
              end;
            end;
          if numer<1 then begin;ok:=0;goto loop2;end;
          for xx:=1 to 3 do
            begin
              for yy:=1 to 3 do
                begin;
                  numer:=0;
                  if (xx=2) and (yy=2) then begin;goto escl;end;
                  if local[xx,yy]=2 then
                    begin
                      if ((xx=2) and (yy=1)) or ((xx=1) and (yy=2)) or ((xx=2) and (yy=3)) or ((xx=3) and (yy=2)) then
                        begin;
                          if (xx-1>0) and (local[xx-1,yy]=1) then begin;numer:=1;end;
                          if (xx+1<4) and (local[xx+1,yy]=1) then begin;numer:=1;end;
                          if (yy-1>0) and (local[xx,yy-1]=1) then begin;numer:=1;end;

```

APENDICE

```

if (yy+1<4) and (local[xx,yy+1]=1) then begin;numer:=1;end;
if ((local[xx,yy]=2) and (numer=0)) then begin;ok:=0;goto loop2;end;
                                end;
                                end;
                                escl:
                                end;
                                end;
(===== CALCULA SI LOS INTERIORES SON CONEXOS =====)
xp:=0;
ok:=0;
if (local[2,1]=1) and (local[3,1]<>1) then begin;xp:=xp+1;end;
if (local[3,1]=1) and (local[3,2]<>1) then begin;xp:=xp+1;end;
if (local[3,2]=1) and (local[3,3]<>1) then begin;xp:=xp+1;end;
if (local[3,3]=1) and (local[2,3]<>1) then begin;xp:=xp+1;end;
if (local[2,3]=1) and (local[1,3]<>1) then begin;xp:=xp+1;end;
if (local[1,3]=1) and (local[1,2]<>1) then begin;xp:=xp+1;end;
if (local[1,2]=1) and (local[1,1]<>1) then begin;xp:=xp+1;end;
if (local[1,1]=1) and (local[2,1]<>1) then begin;xp:=xp+1;end;

if xp<>1 then begin;ok:=0;goto loop2;end;
ok:=1;
goto loop2;
end;

(===== tipo 8x4 ===== )
if TOPOLOGIA='8x4' then begin
numer:=0;
for xx:=1 to 3 do
begin
for yy:=1 to 3 do
begin;
if (local[xx,yy]=1) then begin;numer:=numer+1;end;
end;
end;
if numer<1 then begin;goto loop2;end;
numer:=0;
for xx:=1 to 3 do
begin
for yy:=1 to 3 do
begin;
numer:=0;
if (xx=2) and (yy=2) then begin;goto esc2;end;
if local[xx,yy]=2 then begin;
if (xx-1>0) and (local[xx-1,yy]=1) then begin;numer:=1;end;
if (xx+1<4) and (local[xx+1,yy]=1) then begin;numer:=1;end;
if (yy-1>0) and (local[xx,yy-1]=1) then begin;numer:=1;end;
if (yy+1<4) and (local[xx,yy+1]=1) then begin;numer:=1;end;
if (xx-1>0) and (yy+1<4) and (local[xx-1,yy+1]=1) then begin;numer:=1;end;
if (xx+1<4) and (yy+1<4) and (local[xx+1,yy+1]=1) then begin;numer:=1;end;
if (xx-1>0) and (yy-1>0) and (local[xx-1,yy-1]=1) then begin;numer:=1;end;
if (xx+1<4) and (yy-1>0) and (local[xx+1,yy-1]=1) then begin;numer:=1;end;
end;
if (local[xx,yy]=2 ) and (numer=0) then begin;ok:=0;goto loop2;end;

```

APENDICE

```
        esc2:
        end;
    end;
    {===== CALCULA SI LOS INTERIORES SON CONEXOS =====}
        xp:=0;
        ok:=0;
        if (local[2,1]=1) and (local[3,1]<>1) and (local[3,2]<>1) then xp:=xp+1;
        if (local[3,1]=1) and (local[3,2]<>1) then begin;xp:=xp+1;end;
        if (local[3,2]=1) and (local[3,3]<>1) and (local[2,3]<>1) then xp:=xp+1;
        if (local[3,3]=1) and (local[2,3]<>1) then begin;xp:=xp+1;end;
        if (local[2,3]=1) and (local[1,3]<>1) and (local[1,2]<>1) then xp:=xp+1;
        if (local[1,3]=1) and (local[1,2]<>1) then begin;xp:=xp+1;end;
        if (local[1,2]=1) and (local[1,1]<>1) and (local[2,1]<>1) then xp:=xp+1;
        if (local[1,1]=1) and (local[2,1]<>1) then begin;xp:=xp+1;end;
        if xp<>1 then begin;ok:=0;goto loop2;end;
            ok:=1;
            goto loop2;
        end;
    LOOP2:
    CONDT:=Ok;
end;

END;
Procedure Tes;
var ok:integer;
begin;
H:=0;
WITH puntos^ DO
for y:=YO to yF do
for x:=XO to xF do
begin
ok:=0;
ok:=Condt(x,y);
if ok=1 then BEGIN BASE[x,y]:=0;INC(H);END;
end;
end;
PROCEDURE ADELGAZAR;
BEGIN
MOVE(PUNTOS^,B^,SIZEOF(B^));
bordes;
TES;
END;
PROCEDURE ESQUELETO;
BEGIN
PRINT(200,300,'PROCESANDO INFORMACION',red);
REPEAT
ADELGAZAR;
UNTIL H=0;
PINTAPUNTOS;
borrar_VENTANA(1,2);
END;
Procedure invertir;
begin
```

APENDICE

```
PRINT(200,300,'PROCESANDO INFORMACION',red);
X0:=X0-1;
Y0:=Y0-1;
XF:=XF+1;
YF:=YF+1;
WITH PUNTOS^ DO
for x:=X0 to XF+1 do
  for y:=Y0 to YF+1 do
    begin
      if BASE[x,y]=0 then BASE[X,Y]:=1 else
      if BASE[x,y]=1 then BASE[x,y]:=0;
    end;
  PINTAINVER;
  BORRAR_VENTANA(1,2);
end;
PROCEDURE CAMBIATOP;
begin;
IF TOPOLOGIA='8x4' then Topologia:='4x8' else topologia:='8x4';
end;
{*****RESTO DE PROCEDIMIENTOS*****}
Procedure blood(nombre:string);
var a,al:byte;
    t:Integer;
    b:char;
    f:TEXT;
    top:INTEGER;
    tipe:string;
begin
top:= 16383;
Assign(f,nombre);
reset(i);
FOR T:=0 TO 6 DO read(f,b);
for t:=0 to top do
begin
read(f,b);
a:=Ord(b);
V[t]:=a;
end;
close(f);
end;
PROCEDURE CARGAR_PUNTOS;
BEGIN
X0:=XM;
Y0:=YM;
XF:=0;
YF:=0;
WITH PUNTOS^ DO
FOR X:=1 TO XM-1 DO
FOR Y:=1 TO YM-1 DO
BEGIN
IF GETPIXEL(X,Y)=0 THEN
BASE[X,Y]:=0
ELSE
```

APENDICE

```

    BEGIN
        BASE[X,Y]:=1;
        IF X<X0 THEN X0:=X;
        IF X>XF THEN XF:=X;
        IF Y<Y0 THEN Y0:=Y;
        IF Y>YF THEN YF:=Y;
    END;
END;

PROCEDURE INICIAR_TABLAS;
BEGIN
    fillchar(PUNTOS^,SIZEOF(PUNTOS^),0);
    fillchar(B^,SIZEOF(B^),0);
END;

*****
*****

{*****PROCEDIMIENTOS DE CONTROL DE TIEMPOS***}
PROCEDURE START(VAR HOR,MIN,SEG,CENT:WORD);
    BEGIN
        GETTIME(HOR,MIN,SEG,CENT);
        HOR:=0;
        T1:=MIN*6000+SEG*100+CENT;
    END;
PROCEDURE STOP;
    VAR HOR2,MIN2,SEG2,CENT2:WORD;
    BEGIN
        GETTIME(HOR2,MIN2,SEG2,CENT2);
        T2:=MIN2*6000+SEG2*100+CENT2;
        T2:=T2-T1;
        CENT2:=T2 MOD 100;
        SEG2:=(T2 DIV 100) MOD 60;
        MIN2:=(T2 DIV 100) DIV 60;
        STR(MIN2,MIN3);
        STR(SEG2,SEG3);
        STR(CENT2,CENT3);
        AHORA:='HA TARDADO '+MIN3+' Minutos '+SEG3+' Segundos '+CENT3+' Centesimas';
        PRINT(115,300,AHORA,WHITE);
    END;

BEGIN {-----PROGRAMA PRINCIPAL-----}
NEW(PUNTOS);
NEW(B);
INICIAR_TABLAS;
clrscr;
intro;
CH:=' ';
H:=1;
TOPOLOGIA:='4x8';
REPEAT

```

APENDICE

```
CH:=READKEY;
CASE CH OF
  '1':NOMBRE:='DEMO';
  '2':NOMBRE:='DEMO1';
  '3':NOMBRE:='DEMO2';
  '4':NOMBRE:='DEMO3';
  '5':NOMBRE:='DEMO4';
  '6':NOMBRE:='DEMO5';
  '7':NOMBRE:='DEMO6';
  ELSE BEGIN
    SOUND(1000);
    DELAY(100);
    NOSOUND;
    CH:=' ';
    END;
  END;
UNTIL CH<>' ';
CLRSCR;
TEXTMODE(C80);
INICIAR;
PRINT(200,300,'PROCESANDO INFORMACION',RED);
BLOAD(NOMBRE);
MOVE(V,PANEGA,SIZEOF(PANEGA));
CARGAR_PUNTOS;
PINTAR;
BORRAR_VENTANA(1,2);
CAJA(TOPOLOGIA);
REPEAT
CH:=READKEY;
CASE CH OF
  'A','a':BEGIN
    BORRAR_VENTANA(1,2);
    IF H=0 THEN PRINT(200,300,'No se puede adelgazar. ',LIGHTRED)
    ELSE
    BEGIN
      PRINT(200,300,'PROCESANDO INFORMACION',RED);
      START(HOR,MIN,SEG,CENT);
      ADELGAZAR;
      PINTAPUNTOS;
      BORRAR_VENTANA(1,2);
      STOP;
    END;
  END;
  'I','i':BEGIN H:=1;BORRAR_VENTANA(1,2);START(HOR,MIN,SEG,CENT);INVERTIR;STOP;END;
  'L','l':BEGIN H:=1;BORRAR_VENTANA(1,2);LEER;END;
  'B','b':BEGIN
    BORRAR_VENTANA(1,2);
    PRINT(200,300,'PROCESANDO INFORMACION',RED);
    START(HOR,MIN,SEG,CENT);
    move(puntos^,b^,SIZEOF(B^));
    BORDES;
    PINTAB;
    BORRAR_VENTANA(1,2);
```

APENDICE

```
        STOP;
        H:=1;
        END;
'E','e':BEGIN
        BORRAR_VENTANA(1,2);
        IF H=0 THEN PRINT(200,300,'No se puede adelgazar. ',CYAN)
        ELSE
        BEGIN
                START(HOR,MIN,SEG,CENT);
                ESQUELETO;
                STOP;
        END;
        END;
'F','f':CH:='F';
'C','c':BEGIN ;print(xm-40,ym+3,topologia,black); cambiatop;print (xm-40,ym+3,topologia,red);END;
'V','v':BEGIN BORRAR_VENTANA(1,2);
        PRINT(200,300,'PROCESANDO INFORMACION',RED);
        MOVE(PUNTOS^,B^,SIZEOF(PUNTOS^));
        BORDES;
        VISIONAR;
        SETVISUALPAGE(0);
        SETACTIVEPAGE(0);
        BORRAR_VENTANA(1,2);
        END;
ELSE BEGIN
        SOUND(1000);
        DELAY(100);
        NOSOUND;
        CH:=' ';
        END;
END;
UNTIL CH='F';
CLOSEGRAPH;
END.
```

APENDICE

PROGRAMA PARA LA OBTENCION DEL ESQUELETO DE FORMA NO ITERATIVA.

```
USES CRT,GRAPH,DOS,LUIS3;
CONST XM=319;
      YN=174;
      fichero = 'matrices\';
TYPE
  NUMERO=0..3;
  PUNTERO=^I;
  I=RECORD
    BASE:ARRAY[1..2,1..2]OF NUMERO;
  END;
      vGA_screen = Array [0..16383] of BYTE;
  CAD3=STRING[3];
VAR PUNTOS,B:PUNTERO;
      panvGA:vGA_Screen absolute $A000:0;
      V:vGA_SCREEN;
  X,Y,MUDA,H,XO,YO,XF,YF,N1,N2:INTEGER;
      T1,T2 : longint;
  CH:CHAR;
      HOR,MIN,SEG,CENT:longint;
  MIN3,SEG3,CENT3,AHORA:STRING;
  nombre : string[15];
  TOPOLOGIA:STRING[3];
Procedure Iniciar;
  Var Driver , Mode , resultado: Integer;
  Begin
  DetectGraph(driver mode);
  Mode:=VGAHi;
  InitGraph(Driver,Mode,'c:\tp\bgi');
  resultado:=graphresult;
  end;
PROCEDURE PINTAB;
  BEGIN
  WITH B^DO
  FOR X:=XO-1 TO XF+1 DO
    FOR Y:=YO-1 TO YF+1 DO
      IF BASE[X,Y]=2 THEN PUTPIXEL(X+XM,Y,YELLOW);
  END;
PROCEDURE PINTAPUNTOS;
  BEGIN
  WITH PUNTOS^ D20
  FOR X:=XO-1 TO XF+1 DO
    FOR Y:=YO-1 TO YF+1 DO
      IF BASE[X,Y]=0 THEN PUTPIXEL(X+XM,Y,BLACK);
  END;
PROCEDURE PINTAR;
  BEGIN
      FOR y:=1 TO MaxFila DO
        FOR x:=1 TO MaxColumna DO
          IF (MATRIZ[y]^x) AND $4000 <> 0) then
```


APENDICE

```
IF MATRIZ[y]^[x] and bas2<> 0 THEN PUTPIXEL(X+XM,Y,RED) ELSE
  PUTPIXEL(X+XM,Y,white) else
  if matriz[y]^[x] and $8000<>0 then
    putPixel(x+xm,y,blue);
```

```
END;
```

```
PROCEDURE PINTAR2;
BEGIN
  FOR y:=1 TO yM DO
    FOR x:=1 TO xM DO
      IF matriz[y]^[x] and $8000<> 0 THEN
        if matriz[y]^[x] mod 2 = 1 then PUTPIXEL(X+XM,Y,yellow)
        ELSE PUTPIXEL(X+XM,Y,blue);
```

```
END;
```

```
PROCEDURE PINTAR3;
BEGIN
  FOR y:=1 TO yM DO
    FOR x:=1 TO xM DO
      IF (MATRIZ[y]^[x] AND $4000 <> 0) then
        IF (matriz[y]^[x] mod 2 = 1) then
          PUTPIXEL(X+XM,Y,lightMagenta) else
            if matriz[y]^[x] mod 2 <> 1 then PUTPIXEL(X+XM,Y,lightMagenta)
            else
              if matriz[y]^[x] and $8000<>0 then
                putPixel(x+xm,y,blue);
```

```
END;
```

```
Procedure print(x,y:integer;s:string;co:integer);
begin;
  SetColor(co);
  OutTextxy(x,y,s);
end;
```

```
Procedure VISIONAR;
var x,y:integer;
Procedure Punto(x,y,co:Integer);
var x1,y1:Integer;
label esc;
Begin
  x1:=(2*x);
  y1:=2*y;
  if (x1>640) or (y1>360) then goto esc;
  putPixel(x1,y1,co);
esc:
end;
```

```
begin
SETACTIVEPAGE(1);
for y:=1 to YM do
  for x:=1 to XM do
    begin
      if matriz[y]^[x] and $4000 <>0 then
```

APENDICE

```
        if matr:[y]^x mod 2 =1 then punto(x,y,lightblue) else
            punto(x,y,yellow);
    end;
END;

(*****PROCEDIMIENTOS DE CALCULO*****)
Function Borde(x,y:Integer):integer;
var ok:Integer;
LABEL ESC;
begin;
ok:=0;
WITH B^ DO
if BASE[x,y]=1 then
begin
if BASE[x-1,y-1]=0 then begin;ok:=1;GOTO ESC;end;
if BASE[x,y-1]=0 then begin;ok:=1;goto esc;end;
if BASE[x-1,y]=0 then begin;ok:=1;goto esc;end;
if BASE[x+1,y]=0 then begin;ok:=1;goto esc;end;
if BASE[x+1,y+1]=0 then begin;ok:=1;goto esc;end;
if BASE[x,y+1]=0 then begin;ok:=1;goto esc;end;
end;
ESC:
borde:=ok;
end;
Procedure Bordes;
var ok:integer;
begin;
WITH B^ DO
for y:=Y0 to yF do
for x:=X0 to xF do
begin
ok:=0;
ok:=borde(x,y);
if ok=1 then BASE[x,y]:=2;
end;
end;
end;
Function CondT(x,y:integer):NUMERO; {Calcula la condicion T (Borrable) para retícula hexagonal.}
var ok:Integer;
n_borde:Integer;
n_interior:Integer;
Label esc;
begin;
ok:=0;
n_borde:=0;
n_interior:=0;
WITH B^ DO
if BASE[x,y]=2 then
begin
if BASE[x-1,y-1]=2 then begin;n_borde:=n_borde+1 ;end;
if BASE[x-1,y]=2 then begin;n_borde:=n_borde+1 ;end;
if BASE[x,y+1]=2 then begin;n_borde:=n_borde+1 ;end;
if BASE[x,y-1]=2 then begin;n_borde:=n_borde+1 ;end;
if BASE[x+1,y]=2 Then begin;n_borde:=n_borde+1 ;end;
```

APENDICE

```
        if BASE[x+1,y+1]=2 then begin;n_borde:=n_borde+1 ;end;
        end;

if n_borde<>2 then BEGIN OK:=0;GOTO ESC;END;
    begin
        WITH B^ DO
            if BASE[x,y-1]=1 then begin;n_interior:=n_interior+1 ;end;
        WITH B^ DO
            if BASE[x-1,y]=1 then begin;n_interior:=n_interior+1 ;end;
        WITH B^ DO
            if BASE[x,y+1]=1 then begin;n_interior:=n_interior+1 ;end;
        WITH B^ DO
            if BASE[x-1,y-1]=1 then begin;n_interior:=n_interior+1 ;end;
        WITH B^ DO
            if BASE[x+1,y]=1 then begin;n_interior:=n_interior+1 ;end;
        WITH B^ DO
            if BASE[x+1,y+1]=1 then begin;n_interior:=n_interior+1 ;end;
        end;
if n_interior<1 then begin;ok:=0;goto esc;end;
OK:=1;
ESC:
CondT:=ok;
end;
Procedure Tes;
var ok:integer;
begin;
H:=0;
WITH puntos^ DO
for y:=YO to yF do
for x:=XO to xF do
begin
ok:=0;
ok:=CondT(x,y);
if ok=1 then BEGIN BASE[x,y]:=0;INC(H);END;
end;
end;
Procedure k1(a,b,c:NUMERO);
var n:integer;
begin;
n:=0;
if a=1 then n:=n+1;
if b=1 then n:=n+1;
if c=1 then n:=n+1;
if n=0 then begin;n1:=n1;end;
if n=1 then begin;n1:=n1+1;end;
if n=2 then begin;n2:=n2+1;end;
if n=3 then begin;n1:=n1;end;
end;
Procedure Simplicex(x,y:integer);
var a,b,c,d:NUMERO;
begin
WITH PUNTOS^ DO
begin
```

APENDICE

```

        a:=BASE[x,y];
        b:=BASE[x-1,y];
        c:=BASE[x-1,y-1];
        d:=BASE[x,y-1];
        kl(a,b,c);
        kl(a,c,d);
    end;
end;

Procedure Euler;
var CONT:Integer;
    carac_Euler:real;
    AUX:string[35];
    AX:STRING;
begin

FOR CONT:=1 TO 35 DO AUX[CONT]:=' ';
n1:=0;
n2:=0;
for y:=2 to YM do
for x:=2 to XM do
simplices(x,y);
Carac_Euler:=(n1-n2)/6;
str(Carac_Euler:3:0,ax);
AUX:='';
AUX:='LA CARAC.EULER : '+AX;

PRINT(10,200,AUX,RED);

end;
PROCEDURE ADELGAZAR;
BEGIN
MOVE(PUNTOS^,B^,SIZEOF(B^));
bordes;
TES;
END;
PROCEDURE ESQUELETO;
BEGIN
PRINT(200,255,'PROCESANDO INFORMACION',red);
REPEAT
ADELGAZAR;
UNTIL H=0;
PINTAPUNTOS;
borrar_VENTANA(1,2);
END;
Procedure invertir;
begin
PRINT(200,255,'PROCESANDO INFORMACION',red);
X0:=X0-1;
Y0:=Y0-1;
XF:=XF+1;
YF:=YF+1;
```

APENDICE

```
WITH PUNTOS^ DO
for x:=X0 to XF+1 do
  for y:=Y0 to YF+1 do
    begin
      if BASE[x,y]=0 then BASE[X,Y]:=1 else
      if BASE[x,y]=1 then BASE[x,y]:=0;
    end;

  end;
(*****RESTO DE PROCEDIMIENTOS*****)
Procedure blood(nombre:string);
var a,a1:byte;
    t:Integer;
    b:char;
    f:TEXT;
    top:INTEGER;
    tipe:string;
begin
top:= 16383;
Assign(f,nombre);
reset(f);
FOR T:=0 TO 6 DO read(f,b);
for t:=0 to top do
  begin
    read(f,b);
    a:=Ord(b);
    V[t]:=a;
  end;
close(f);
end;
PROCEDURE CARGAR_PUNTOS;
BEGIN
X0:=XM;
Y0:=YM;
XF:=0;
YF:=0;
WITH PUNTOS^ DO
  FOR X:=1 TO XM-1 DO
    FOR Y:=1 TO YM-1 DO
      BEGIN
        IF GETPIXEL(X,Y)=0 THEN
          BASE[X,Y]:=0
        ELSE
          BEGIN
            BASE[X,Y]:=1;
            IF X<X0 THEN X0:=X;
            IF X>XF THEN XF:=X;
            IF Y<Y0 THEN Y0:=Y;
            IF Y>YF THEN YF:=Y;
          END;
        END;
      END;
END;
```

APENDICE

```
PROCEDURE INICIAR_TABLAS;
  BEGIN
    fillchar(PUNTOS^,SIZEOF(PUNTOS^),0);
    fillchar(B^,SIZEOF(B^),0);

    END;

PROCEDURE LEER;
  VAR CARACTER:CHAR;
      I:INTEGER;
  BEGIN
    BORRAR_VENTANA(2,2);
    PRINT(5,200,'INTRODUZCA EL NUMERO DEL FICHERO A TRATAR',RED);
    PRINT(5,298,'FICHEROS DISPONIBLES:',LIGHTRED);
    PRINT(5,318,'1-DEMO 2-DEMO1 3-DEMO2 4-DEMO3 5-DEMO4 6-DEMO5 7-DEMO6',LIGHTGRAY);
    CH:=' ';
    REPEAT
      CH:=READKEY;
    CASE CH OF
      '1':NOMBRE:='DEMO';
      '2':NOMBRE:='DEMO1';
      '3':NOMBRE:='DEMO2';
      '4':NOMBRE:='DEMO3';
      '5':NOMBRE:='DEMO4';
      '6':NOMBRE:='DEMO5';
      '7':NOMBRE:='DEMO6';
      ELSE BEGIN
        SOUND(1000);
        DELAY(100);
        NOSOUND;
        CH:=' ';
        END;
    END;
  UNTIL CH<>' ';
  INICIAR_TABLAS;
  CLEARVIEWPORT;
  PRINT(200,255,'PROCESANDO INFORMACION',RED);
  BLOAD(NOMBRE);
  MOVE(V,PANvGA,SIZEOF(PANvGA));
  LeePantalla(Matriz);

  PINTAR;
  END;

{*****PROCEDIMIENTOS DE CONTROL DE TIEMPOS***}
PROCEDURE START(VAR HOR,MIN,SEG,CENT:longint);

var   h,m,s,c : word;
      BEGIN
        GETTIME(H,M,S,C);
        HOR:=0;min:=m;seg:=s;cent:=c;
        T1:=MIN*6000+SEG*100+CENT;
```

APENDICE

```
END;
PROCEDURE STOP;
    VAR HOR2,MIN2,SEG2,CENT2:longint;
        h,m,s,c : word;
    BEGIN
        GETTIME(H,M,S,C);
        min2:=m;seg2:=s;cent2:=c;
        T2:=MIN2*6000+SEG2*100+CENT2;
        T2:=T2-T1;
        CENT2:=T2 MOD 100;
        SEG2:=(T2 DIV 100) MOD 60;
        MIN2:=(T2 DIV 100) DIV 60;
        STR(MIN2,MIN3);
        STR(SEG2,SEG3);
        STR(CENT2,CENT3);
        AHORA:='HA YARDADO '+MIN3+' Minutos '+SEG3+' Segundos '+CENT3+' Centesimas';
        PRINT(115,300,AHORA,WHITE);
    END;

BEGIN {-----PROGRAMA PRINCIPAL-----}
NEW(PUNTOS);
NEW(B);
INICIAR_TABLAS;
clrscr;
intro;
CH:=' ';
H:=1;
TOPOLOGIA:='6x6';
REPEAT
CH:=READKEY;
CASE CH OF
    '1':NOMBRE:='DEMO';
    '2':NOMBRE:='DEMO1';
    '3':NOMBRE:='DEMO2';
    '4':NOMBRE:='DEMO3';
    '5':NOMBRE:='DEMO4';
    '6':NOMBRE:='DEMO5';
    '7':NOMBRE:='DEMO6';
    ELSE BEGIN
        SOUND(1000);
        DELAY(100);
        NOSOUND;
        CH:=' ';
        END;
    END;
UNTIL CH<>' ';
CLRSCR;
TEXTMODE(C80);
INICIAR;
PRINT(200,300,'PROCESANDO INFORMACION',RED);
BLOAD(NOMBRE);
MOVE(V,PANvGA,SIZEOF(PANvGA));
LEEPANTALLA(MATRIZ);
```

APENDICE

```

PINTAR;
BORRAR_VENTANA(1,2);
REPEAT
CH:=READKEY;
CASE upcase(CH) OF
    'G' : BEGIN
        BORRAR_VENTANA(1,2);
        PRINT(200,255,'PROCESANDO INFORMACION',RED);
        START(HOR,MIN,SEG,CENT);
        profundidadTotalOrto(matriz);
        basicOrto(matriz);
        guardarOrto(matriz,fichero+nombre);
        BORRAR_VENTANA(1,2);
        STOP;
    END;
    'C' : BEGIN
        leerMatriz(matriz,fichero+nombre);
        pintar;
    end;
    'L','l':BEGIN H:=1;BORRAR_VENTANA(1,2);LEER;END;
    'B','b':BEGIN
        BORRAR_VENTANA(1,2);
        PRINT(200,255,'PROCESANDO INFORMACION',RED);
        START(HOR,MIN,SEG,CENT);
        ProfundidadTotalOrto(matriz);
        basicOrto(matriz);
        stop;
        pintar3;
        BORRAR_VENTANA(1,2);
        { H:=1;}
    END;
    'E','e':BEGIN
        BORRAR_VENTANA(1,2);
        START(HOR,MIN,SEG,CENT);
        PRINT(200,255,'PROCESANDO INFORMACION',RED);
        ProfundidadTotal(Matriz);
        Basic1(matriz);
        Basic2(Matriz);
        BORRAR_VENTANA(1,2);
        STOP;
        Pintar;
    END;
    'F','f':CH:='F';
    'P','p':BEGIN START(HOR,MIN,SEG,CENT);ProfundidadTotalOrto(Matriz);
        STOP;
        pintar2;
    END;
    'V','v':BEGIN BORRAR_VENTANA(1,2);
        PRINT(200,255,'PROCESANDO INFORMACION',RED);
    END;
VISIONAR;
SETVISUALPAGE(0);

```


APENDICE

```
                SETACTIVEPAGE(0);
                BORRAR_VENTANA(1,2);
                END;
ELSE BEGIN
    SOUND(1000);
    DELAY(100);
    NOSOUND;
    CH:=' ';
    END;
END;
UNTIL CH='F';
CLOSEGRAPH;
END.
```

UNIDAD AUXILIAR AL PROGRAMA

```
UNIT   LUIS3;

INTERFACE
USES
    CRT,GRAPH;

CONST
    MaxFila = 175;
    MaxColumna = 320;{140;}
    profundidad1 = $1FF;
    bas1 = $4000;
    bas2 = $2000;
    color = $8000;
    visto = $1000;
    marcaBloque = $80;
    profundidad = $04;
    derecha = $40;
    izquierda = $20;
    abajo = $10;
    arriba = $08;
    falso = false;
    cierto = true;

TYPE
    fila = array[0..MaxColumna+1] of word;
    Pfila = ^fila;
    tipoMatriz = array[0..MaxFila+1] of Pfila;

    Pproceso = ^proceso;
    proceso = object
        m : ^tipoMatriz;
        bloque : ^boolean;
        x,y : ^word;
        n : ^byte;
        incX,incY : shortInt;
        constructor inicia(var matriz : tipoMatriz;var nn : byte;
```

APENDICE

```

                                var Ibloque : boolean;var Ix,Iy : word;
                                IincX,IincY : shortInt);

    destructor acaba;
    function sigue : boolean;virtual;
end;

PprocesoBloque = ^procesoBloque;
procesoBloque = object(proceso)
    p,nx,ny : longInt;
    constructor inicia(var matriz : tipoMatriz;var nn : byte;
                                var Ibloque : boolean;var Ix,Iy : word;
                                IincX,IincY : shortInt);

    destructor acaba;
    function sigue : boolean;virtual;
end;

procedure iniciaMatriz(var M : tipoMatriz);
procedure profundidadTotal(var m : tipoMatriz);
procedure leePantalla(var m : tipoMatriz);
procedure Basico1(var m : tipoMatriz);
procedure Basico2(var m : tipoMatriz);
procedure Basico2New(var m : tipoMatriz);
procedure guardar(var M : tipoMatriz;nombre : string);
procedure leerMatriz(var M : tipoMatriz;nombre : string);
procedure profundidadTotalOrto(var m : tipoMatriz);
procedure Basico1Orto(var m : tipoMatriz);
procedure guardarOrto(var Matriz : tipoMatriz;nombre : string);

VAR
    matriz : tipoMatriz;

IMPLEMENTATION
constructor proceso.inicia(var matriz : tipoMatriz;var nn : byte;
                                var Ibloque : boolean;var Ix,Iy : word;
                                IincX,IincY : shortInt);

    begin
        m:=@matriz;
        n:=@nn;
        bloque:=@Ibloque;
        x:=@Ix;
        y:=@Iy;
        incX:=IincX;
        incY:=IincY;
    end;

destructor proceso.acaba;
begin
end;
function proceso.sigue : boolean;
begin
    if m^[y^+incY]^[x^+incX] and (bas1 or bas2 or visto) = bas1 then
        begin
            sigue:=falso;
        end;
end;

```

APENDICE

```

        y^:=y^+incY;x^:=x^+incX;
        m^[y^][x^]:=m^[y^][x^] or visto;
        case incY of
            -1 : n^:=n^ or arriba;
            1 : n^:=n^ or abajo;
        end;
        case incX of
            -1 : n^:=n^ or izquierda;
            1 : n^:=n^ or derecha;
        end;
    end else sigue:=cierto;
end;

constructor procesoBloque.inicia(var matriz : tipoMatriz;var nn : byte;
                                var Ibloque : boolean;var Ix,Iy : word;
                                IincX,IincY : shortInt);
begin
    proceso.inicia(matriz,nn,Ibloque,Ix,Iy,IincX,IincY);
end;

destructor procesoBloque.acaba;
begin
end;

function procesoBloque.sigue : boolean;
begin
    p:=m^[y^][x^] and profundidad1;
    nx:=(p-1)*incX+x^+incX;
    ny:=(p-1)*incY+y^+incY;
    bloque^:=falso;
    sigue:=falso;
    if (nx<=maxColumna) and (nx>0) and (ny<=maxFila) and (ny>0)
        and (p>1) then
        begin
            if m^[ny][nx] and (bas1 or bas2 or visto)=bas1 then
                begin
                    bloque^:=cierto;
                    n^:=n^ or marcaBloque;
                    while nx<>x^ do
                        begin
                            m^[ny][nx]:=m^[ny][nx] or visto;
                            nx:=nx-incX;
                        end;
                    while ny<>y^ do
                        begin
                            m^[ny][nx]:=m^[ny][nx] or visto;
                            ny:=ny-incY;
                        end;
                    x^:=(p-1)*incX+x^+incX;
                    y^:=(p-1)*incY+y^+incY;
                end else sigue:=cierto;
            end else sigue:=cierto;
        end;
end;

```

APENDICE

```

procedure iniciaMatriz(var M : tipoMatriz);
VAR   n : word;
      p : Pfila;
begin
  for n:=0 to maxFila+1 do
    begin
      new(p);
      m[n]:=p;
    end;
  for n:=0 to MaxColumna+1 do
    begin
      m[0]^n:=0;
      m[maxFila+1]^n:=0;
    end;
  for n:=0 to maxFila+1 do
    begin
      m[n]^0:=0;
      m[n]^maxColumna+1:=0;
    end;
  end;
procedure profundidadTotal(var m : tipoMatriz);
VAR   x,y,n : word;
function profundidadDescendente : word;
VAR   n : word;
begin
  n:=m[x]^y-1 and profundidad1;
  if n>m[x-1]^y-1 and profundidad1 then
    n:=m[x-1]^y-1 and profundidad1;
  if n>m[x-1]^y and profundidad1 then
    n:=m[x-1]^y and profundidad1;
  profundidadDescendente:=n+1;
end;
function profundidadAscendente : word;
VAR   n : word;
begin
  n:=m[x+1]^y+1 and profundidad1;
  if n>m[x]^y+1 and profundidad1 then
    n:=m[x]^y+1 and profundidad1;
  if n>m[x+1]^y and profundidad1 then
    n:=m[x+1]^y and profundidad1;
  inc(n);
  if n>m[x]^y and profundidad1 then
    n:=m[x]^y and profundidad1;
  profundidadAscendente:=n;
end;
begin
  for x:=1 to maxFila do
    for y:=1 to maxColumna do
      if (m[x]^y and $8000)=0 then
        m[x]^y:=m[x]^y and 61440 else
          m[x]^y:=m[x]^y or profundidadDescendente;
  for x:=maxFila downto 1 do
    for y:=MaxColumna downto 1 do

```

APENDICE

```

        if (m[x]^y and $8000)=0 then
            m[x]^y:=m[x]^y and 61440 else
                begin
                    n:=profundidadAscendente;
                    m[x]^y:=m[x]^y and 61440;
                    m[x]^y:=m[x]^y or n;
                end;
        end;
procedure profundidadTotalOrto(var m : tipoMatriz);
VAR    x,y,n : word;
function profundidadDescendente : word;
VAR    n : word;
    begin
        n:=m[x]^(y-1) and profundidad1;
        if n>m[x-1]^(y-1) and profundidad1 then
            n:=m[x-1]^(y-1) and profundidad1;
        if n>m[x-1]^(y+1) and profundidad1 then
            n:=m[x-1]^(y+1) and profundidad1;
        if n>m[x-1]^y and profundidad1 then
            n:=m[x-1]^y and profundidad1;
        profundidadDescendente:=n+1;
    end;
function profundidadAscendente : word;
VAR    n : word;
    begin
        n:=m[x+1]^(y+1) and profundidad1;
        if n>m[x]^(y+1) and profundidad1 then
            n:=m[x]^(y+1) and profundidad1;
        if n>m[x]^(y-1) and profundidad1 then
            n:=m[x]^(y-1) and profundidad1;
        if n>m[x+1]^(y-1) and profundidad1 then
            n:=m[x+1]^(y-1) and profundidad1;
        if n>m[x+1]^y and profundidad1 then
            n:=m[x+1]^y and profundidad1;
        inc(n);
        if n>m[x]^y and profundidad1 then
            n:=m[x]^y and profundidad1;
        profundidadAscendente:=n;
    end;
begin
    for x:=1 to maxFila do
        for y:=1 to maxColumna do
            if (m[x]^y and $8000)=0 then
                m[x]^y:=m[x]^y and 61440 else
                    m[x]^y:=m[x]^y or profundidadDescendente;
        for x:=maxFila downto 1 do
            for y:=MaxColumna downto 1 do
                if (m[x]^y and $8000)=0 then
                    m[x]^y:=m[x]^y and 61440 else
                        begin
                            n:=profundidadAscendente;
                            m[x]^y:=m[x]^y and 61440;
                            m[x]^y:=m[x]^y or n;
                        end;
            end;
        end;
    end;
end;

```

APENDICE

```

end;
end;
procedure leePantalla(var m : tipoMatriz);
VAR   x,y : word;
begin
    for y:=1 to maxFila do
        for x:=1 to maxColumna do
            if getPixel(x,y)=0 then m[y]^x:=0 else
                m[y]^x:=32768;
        end;
    end;
procedure Basicol(var m : tipoMatriz);
VAR   x,y,n,p : word;
begin
    for y:=1 to MaxFila do
        for x:=1 to MaxColumna do if (m[y]^x and $8000) <>0 then
            begin
                n:=0;
                m[y]^x:=m[y]^x or bas1;
                p:=m[y]^x and profundidad1;
                if p<=m[y]^(x-1) and profundidad1 then inc(n);
                if p<=m[y]^(x+1) and profundidad1 then inc(n);
                if p<=m[y+1]^(x+1) and profundidad1 then inc(n);
                if (n<3) and (p<=m[y+1]^x) and profundidad1 then inc(n);
                if (n<3) and (p<=m[y-1]^x) and profundidad1 then inc(n);
                if (n<3) and (p<=m[y-1]^(x-1)) and profundidad1 then inc(n);
                if n>=3 then
                    begin
                        n:=0;
                        p:=m[y]^x and profundidad1;
                        if p=m[y]^(x-1) and profundidad1 then inc(n);
                        if p=m[y]^(x+1) and profundidad1 then inc(n);
                        if (p=m[y+1]^(x+1) and profundidad1) then inc(n);
                        if (n<3) and (p=m[y+1]^x) and profundidad1 then
                            if (n<3) and (p=m[y-1]^x) and profundidad1 then
                                if (n<3) and (p=m[y-1]^(x-1)) and profundidad1
                                    if n=2 then m[y]^x:=m[y]^x and not(bas1);
                    end;
                end;
                {Fin de begin de for.}
                {Fin de begin de procedimiento.}
            end;
        end;
    end;
procedure BasicolOrto(var m : tipoMatriz);
VAR   x,y,n,p : word;
begin
    for y:=1 to MaxFila do
        for x:=1 to MaxColumna do if (m[y]^x and $8000) <>0 then
            begin
                n:=0;
                m[y]^x:=m[y]^x and (not bas1);
                p:=m[y]^x and profundidad1;
                if p<=m[y-1]^(x+1) and profundidad1 then inc(n);
            end;
        end;
    end;
end;

```

APENDICE

```

if (n<1) and (p<m[y+1]^x-1 and profundidad1) then inc(n);
if (n<1) and (p<m[y]^x-1 and profundidad1) then inc(n);
if (n<1) and (p<m[y]^x+1 and profundidad1) then inc(n);
if (n<1) and (p<m[y+1]^x+1 and profundidad1) then inc(n);
if (n<1) and (p<m[y+1]^x and profundidad1) then inc(n);
if (n<1) and (p<m[y-1]^x and profundidad1) then inc(n);
if (n<1) and (p<m[y-1]^x-1 and profundidad1) then inc(n);
if n=0 then m[y]^x:=m[y]^x or bas1;
end;
(Pin de begin de for.)
end;
(Pin de begin de procedimiento.)

procedure Basico2(var m : tipoMatriz);
VAR x,y,n,p : word;
    sigue : boolean;
function vecinoBasico : boolean;
begin
    if( m[y-1]^x-1 and $4000<>0) and
(m[y]^x and Profundidad1> m[y-1]^x-1 and Profundidad1)
then vecinoBasico:=true else
    if( m[y-1]^x and $4000 <>0) and
(m[y]^x and Profundidad1> m[y-1]^x and Profundidad1)
then vecinoBasico:=true else
    if( m[y]^x-1 and $4000 <>0) and
(m[y]^x and Profundidad1> m[y]^x-1 and Profundidad1)
then vecinoBasico:=true else
    if( m[y]^x+1 and $4000 <>0) and
(m[y]^x and Profundidad1> m[y]^x+1 and Profundidad1)
then vecinoBasico:=true else
    if( m[y+1]^x and $4000 <>0) and
(m[y]^x and Profundidad1> m[y+1]^x and profundidad1)
then vecinoBasico:=true else
    if( m[y+1]^x+1 and $4000 <>0) and
(m[y]^x and Profundidad1> m[y+1]^x+1 and Profundidad1)
then vecinoBasico:=true else
        vecinoBasico:=false;
end;

begin
for y:=1 to MaxFila do
for x:=1 to MaxColumna do
if ((m[y]^x and $8000) <>0) and (m[y]^x and $4000=0)
and vecinoBasico then
begin
m[y]^x:=m[y]^x or bas1 or bas2;
dec(x,2);dec(y);
end;
(
begin
sigue:=true;
p:=m[y]^x and profundidad1;
if( p>m[y-1]^x-1 and profundidad1) and (m[y-1]^x-1 and $4000<>0)
then
if (m[y-1]^x-1 and profundidad1>=m[y-1]^x and profundidad1) and
(m[y-1]^x-1 and profundidad1>= m[y]^x-1 and profundidad1)

```

APENDICE

```

and (m[y-1]^x and Profundidad1 > p
then
    begin
        m[y]^x := m[y]^x or bas2 or bas1;
        dec(x); dec(y);
        sigue := falso;
    end;
if sigue and (p > m[y-1]^x and Profundidad1) and (m[y-1]^x and $4000 < 0)
then
    if (m[y-1]^x and Profundidad1 >= m[y-1]^{x-1} and Profundidad1) and
        (m[y-1]^x and Profundidad1 >= m[y]^{x+1} and Profundidad1) then
        begin
            m[y]^x := m[y]^x or bas1 or bas2;
            dec(x); dec(y);
            sigue := falso;
        end;
if sigue and (p > m[y+1]^x and Profundidad1) and (m[y+1]^x and $4000 < 0)
then
    if (m[y+1]^x and Profundidad1 >= m[y]^{x-1} and Profundidad1) and
        (m[y+1]^x and Profundidad1 >= m[y+1]^{x+1} and Profundidad1) then
        begin
            m[y]^x := m[y]^x or bas1 or bas2;
            dec(x); dec(y);
            sigue := falso;
        end;
if sigue and (p > m[y+1]^{x+1} and Profundidad1) and (m[y+1]^{x+1} and $4000 < 0)
then
    if (m[y+1]^{x+1} and Profundidad1 >= m[y]^{x+1} and Profundidad1) and
        (m[y+1]^{x+1} and Profundidad1 >= m[y+1]^x and Profundidad1) then
        begin
            m[y]^x := m[y]^x or bas1 or bas2;
            dec(x); dec(y);
            sigue := falso;
        end;
if sigue and (p > m[y]^{x+1} and Profundidad1) and (m[y]^{x+1} and $4000 < 0)
then
    if (m[y]^{x+1} and Profundidad1 >= m[y-1]^x and Profundidad1) and
        (m[y]^{x+1} and Profundidad1 >= m[y+1]^{x+1} and Profundidad1) then
        begin
            m[y]^x := m[y]^x or bas1 or bas2;
            dec(x); dec(y);
            sigue := falso;
        end;
if sigue and (p > m[y]^{x-1} and Profundidad1) and (m[y]^{x-1} and $4000 < 0)
then
    if (m[y]^{x-1} and Profundidad1 >= m[y-1]^{x-1} and Profundidad1) and
        (m[y]^{x-1} and Profundidad1 >= m[y+1]^x and Profundidad1) then
        begin
            m[y]^x := m[y]^x or bas1 or bas2;
            dec(x); dec(y);
        end;
end;
{Fin de begin de for.}
{Fin de begin de procedimiento.}
end;

```


APENDICE

```
procedure Basico2New(var m : tipoMatriz);

VAR   x,y,n,p : word;
      signe : boolean;

function vecinoBasicoNew : boolean;

begin
  n:=0;
  if( m[y-1]^x-1 and $4000<>0)
  then inc(n);
      if( m[y-1]^x and $4000 <>0)
      then inc(n);
          if( m[y]^x-1 and $4000 <>0)
          then inc(n);
              if( m[y]^x+1 and $4000 <>0)
              then inc(n);
                  if( m[y+1]^x and $4000 <>0)
                  then inc(n);
                      if( m[y+1]^x+1 and $4000 <>0)
                      then inc(n);
                          if n=1 then vecinoBasicoNew:= true else
                          vecinoBasicoNew:=false;
end;

begin
for y:=1 to MaxFila do
  for x:=1 to MaxColumna do
    if ((m[y]^x and $8000) <>0) and (m[y]^x and $4000=0)
    and vecinoBasicoNew then
      begin
        m[y]^x:=m[y]^x or bas1 or bas2;
        dec(x,2);dec(y);
      end;
end;

End;

procedure guardar(var M : tipoMatriz;nombre : string);

CONST
  coordenadas = $80;
  derecha = $40;
  izquierda = $20;
  abajo = $10;
  arriba = $08;
  visto = $1000;
  profundidad = $04;

TYPE
  TipoCoor = record
```

APENDICE

```

                                case integer of
                                    1 : (x,x1,y,y1 : byte);
                                    2 : (xx,yy : word);
                                    3 : (tt : longint);
                                end;

VAR    f : file of byte;
        nl,n : byte;
        x,y : word;
        derecho : word;
        coor1,coor : TipoCoor;
        sigue : boolean;
        prof : word;

procedure PonPunto(x,y : word);

begin
putPixel(x+319,y,BLUE);
delay(20);
end;

begin
assign(f,nombre);
rewrite(f);
derecho:=0;
prof:=1;
for y:=1 to maxFila do
    for x:=1 to MaxColumna do
        if (m[y]^x and (bas1 or bas2 or visto)=bas1) then
            begin
                n:=coordenadas;
                sigue:=cierto;
                coor.xx:=x;coor.yy:=y;
                while sigue do
                    begin
                        m[y]^x:=m[y]^x or visto;
                        if prof+1=m[y]^x and profundidad1 then
                            n:=(n or $04) else
                                if prof-1=m[y]^x and profundidad1 then
                                    n:=n or $02 else
                                        if prof=m[y]^x and profundidad1 then
                                            n:=n or $06 else prof:=m[y]^x and
profundidad1;

                        if (m[y]^(x-1) and (bas1 or bas2 or visto)=bas1) then
                            begin
                                dec(x);
                                n:=n or izquierda;
                            end else
                                if (m[y]^(x+1) and (bas1 or bas2 or visto)=bas1) then
                                    begin
                                        inc(x);
                                        n:=n or derecha;
                                    end else

```

APENDICE

```

if (m[y+1]^x and (bas1 or bas2 or visto)=bas1) then
  begin
    inc(y);
    n:=n or (abajo);
  end else
if (m[y-1]^x and (bas1 or bas2 or visto)=bas1) then
  begin
    dec(y);
    n:=n or (arriba);
  end else
if (m[y+1]^x+1 and (bas1 or bas2 or visto)=bas1) then
  begin
    inc(y);inc(x);
    n:=n or (abajo or derecha);
  end else
if (m[y+1]^x-1 and (bas1 or bas2 or visto)=bas1) then
  begin
    dec(x);inc(y);
    n:=n or (izquierda or abajo);
  end else
if (m[y-1]^x-1 and (bas1 or bas2 or visto)=bas1) then
  begin
    dec(x);dec(y);
    n:=n or (izquierda or arriba);
  end else
if (m[y-1]^x+1 and (bas1 or bas2 or visto)=bas1) then
  begin
    inc(x);dec(y);
    n:=n or (derecha or arriba);
  end else sigue:=falso;
write(f,n);ponPunto(x,y);
if n and coordenadas= coordenadas then
  begin
    coor1.tt:=coor.xx+coor.yy*(MaxColumna);
    write(f,coor1.x,coor1.x1,coor1.y);
  end;
n1:=n and $06;
if n1=4 then inc(prof);
if n1=2 then dec(prof);
if n1=0 then
  begin
    coor1.vv:=prof;
    write(f,coor1.x,coor1.x1);
  end;
n:=0;
end;
x:=coor.xx;
y:=coor.yy;
end;
close(f);
end;
{Fin de while.}
{Fin de if.}
{Fin de procedimiento guardar.}

```

APENDICE

```
procedure leerMatriz(var M : tipoMatriz;nombre : string);
CONST
    coordenadas = $80;
TYPE
    coor1 = record
        case integer of
            1 : (x,x1,y,y1 : byte);
            2 : (xx,yy : word);
            3 : (tt : longInt);
        end;
VAR
    f : file of byte;
    n : byte;
    x,y : word;
    prof,derecho : word;
    coor : coor1;
    derechal,izquierdal,abajol,arribal : boolean;
begin
    for y:=0 to maxFila+1 do
        for x:=0 to maxcolumna+1 do m[y]^x:=0;
    assign(f,nombre);
    reset(f);
    prof:=1;
    while not eof(f) do
        begin
            read(f,n);
            if n and coordenadas = coordenadas then
                begin
                    read(f,coor.x,coor.x1,coor.y);
                    x:=(coor.tt mod maxColumna);
                    y:=(coor.tt div MaxColumna);
                    end;
                if n and $06 = 0 then
                    begin
                        read(f,coor.x,coor.x1);
                        prof:=coor.xx;
                    end;
                if n and $06 = 4 then inc(prof);
                if n and $06 = 2 then dec(prof);
                m[y]^x:=m[y]^x or (color or bas1 or prof);
                if n and derecha = derecha then inc(x);
                if n and izquierda <>0 then dec(x);
                if n and abajo<>0 then inc(y);
                if n and arriba<>0 then dec(y);
            end;
        m[y]^x:=m[y]^x or (color or bas1);
    close(f);
end;
procedure guardarOrto(var Matriz : tipoMatriz;nombre : string);
TYPE
    direcciones = array[0..11]of Pproceso;
    coor1 = record
        case integer of
            1 : (x,x1,y,y1 : byte);
```

APENDICE

```
                2 : (xx,yy : word);
                3 : (tt : longInt);
            end;
VAR
    f : file of byte;
    d : direcciones;
    bloque : boolean;
    x,y,xx,yy : word;
    n,m : byte;
    seguir,prof : boolean;
    coor : coor1;
    t : longInt;
procedure ponPunto(x,y : word);
begin
    putPixel(x+319,y,blue);
    delay(20);
end;
begin
assign(f,nombre);
rewrite(f);
d[0]:=new(Pproceso,incia(matriz,n,bloque,xx,yy,-1,0));
d[1]:=new(Pproceso,incia(matriz,n,bloque,xx,yy,1,0));
d[2]:=new(Pproceso,incia(matriz,n,bloque,xx,yy,0,1));
d[3]:=new(Pproceso,incia(matriz,n,bloque,xx,yy,0,-1));
d[4]:=new(Pproceso,incia(matriz,n,bloque,xx,yy,1,1));
d[5]:=new(Pproceso,incia(matriz,n,bloque,xx,yy,-1,1));
d[6]:=new(Pproceso,incia(matriz,n,bloque,xx,yy,-1,-1));
d[7]:=new(Pproceso,incia(matriz,n,bloque,xx,yy,1,-1));
d[8]:=new(PprocesoBloque,incia(matriz,n,bloque,xx,yy,-1,0));
d[9]:=new(PprocesoBloque,incia(matriz,n,bloque,xx,yy,1,0));
d[10]:=new(PprocesoBloque,incia(matriz,n,bloque,xx,yy,0,1));
d[11]:=new(PprocesoBloque,incia(matriz,n,bloque,xx,yy,0,-1));
for y:=1 to maxFila do
    for x:=1 to maxColumna do
        if matriz[y]^x and (bas1 or bas2 or visto)=bas1 then
            begin
                matriz[y]^x:=matriz[y]^x or visto;
                xx:=x;yy:=y;
                prof:=cierto;
                seguir:=falso;
                repeat
                    m:=8;
                    n:=0;
                    {$B-}
                    while (m<12) and (d[m]^sigue) do inc(m);
                    m:=0;
                    if not bloque then while (m<8) and (d[m]^sigue) do inc(m);
                    if m=8 then seguir:=cierto;
                    PonPunto(xx,yy);
                    if prof then
                        begin
                            n:=n or profundidad;
                            write(f,n);
```

APENDICE

```

                                coor.yy:=matriz[y]^[x] and profundidad1;
                                write(f,coor.x,coor.x1);
                                t:=yy;
                                coor.tt:=t*MaxColumna+xx;
                                write(f,coor.x1,coor.y,coor.y1);
                                prof:=falso;
                                end else write(f,n);
                                until seguir;
                                end;                                {Fin de if bas1}
close(f);
end;                                {Fin de procedimiento}
begin
iniciaMatriz(matriz);
END.
```

APENDICE

PROGRAMA PARA EL ADELGAZAMIENTO EN COLOR

```
program gr;
(Programa para tener una propia rutina de gráficos.)
USES
    mgrafico,CRT;

TYPE
    tipoMatricilla = array[0..7]of boolean;
    tipoDato = record
        color : byte;
        borde : boolean;
        bordeC : byte;
        borrrable : boolean;
    end;
    Pcolor = ^Pcolor1;
    Pcolor1 = array[0..159,0..199]of byte;
    Pborrrable = ^Pborrrable1;
    Pborrrable1 = array[0..159,0..199]of boolean;
    Pgraficos1 = ^graficos1;
    graficos1 = object (graficos)
        constructor inicia;
        destructor acaba;
        function cojeColor(x,y : word) : word;virtual;
    end;
    pTopologia = ^topologia;
    topologia = object
        matricilla : TipoMatricilla;
        constructor inicia;
        destructor acaba;
        function condicion(x,y : word) : boolean;virtual;
        procedure borde(x,y : word);virtual;
    end;
    Pseis_seis = ^seis_seis;
    seis_seis = object(topologia)
        function condicion(x,y : word) : boolean;virtual;
        procedure borde(x,y : word);virtual;
    end;
    Pocho_cuatro = ^ocho_cuatro;
    ocho_cuatro = object(topologia)
        function condicion(x,y : word) : boolean;virtual;
        function puntoInterior(x,y,c : word) : boolean;
        procedure borde(x,y : word);virtual;
        function Cy(y,c : word) : word;
        function Cx(x,c : word) : word;
        function conexionInterior(x,y : word) : boolean;
        function conexionBordeInterior(x,y : word) : boolean;
    end;

VAR
    f : file of tipoDato;
    Mcolor,MbordeC : Pcolor;
```

APENDICE

```

    Mborde,Mborrable : Pborrable;
    nombre : string;
    tipoTopologia : Ptopologia;
    topo : byte;

constructor graficos1.inicia;
begin
    graficos.inicia($13);
end;
destructor graficos1.acaba;
begin
    graficos.acaba;
end;
function graficos1.cojeColor(x,y : word) : word;
begin
    cojeColor:=mem[$A000:y*320+x];
end;
function lineal(x,y : word) : longInt;
VAR    n,m : longInt;
begin
    n:=x;
    m:=y;
    lineal:=y*160+x;
end;
function punto(x,y : word) : word;
VAR    a : tipoDato;
begin
    (
        seek(f,lineal(x,y));
        read(f,a);
        punto:=a.color;
        punto:=Mcolor^[x,y];
    )
end;

procedure ponBorde(x,y,b : word);

VAR    a : tipoDato;
begin
    Mborde^[x,y]:=cierto;
    MbordeC^[x,y]:=b;
end;

procedure bordes;
VAR    x,y : word;
    procedure ini(x,y : word);
        begin
            for x:=1 to 158 do
                for y:=1 to 198 do
                    begin
                        Mborde^[x,y]:=falso;
                        Mborrable^[x,y]:=falso;
                    end;
                end;
            end;
        end;
end;
```


APENDICE

```
begin
  ini(x,y);
  for x:=1 to 158 do
    for y:=1 to 198 do
      tipoTopologia^.borde(x,y);
    end;
  end;
procedure carga;
VAR   x,y : word;
      a : tipoDato;
begin
  for y:=0 to 199 do
    for x:=0 to 159 do
      begin
        Mborde^[x,y]:=falso;
        MbordeC^[x,y]:=255;
        Mborrable^[x,y]:=falso;
        Mcolor^[x,y]:=p^.cojeColor(x,y);
      end;
    end;
  end;
procedure dibujaBordes(d : word);
VAR   x,y : word;
      a : tipoDato;
begin
  for x:=0 to 159 do
    for y:=0 to 199 do
      begin
        if Mborde^[x,y] then
          begin
            p^.ponColor(MbordeC^[x,y]);
            p^.ponPixel(x+d,y);
          end;
        end;
      end;
    end;
  end;
procedure dibujaColor(d : word);
VAR   x,y : word;
begin
  for x:=0 to 159 do
    for y:=0 to 199 do
      begin
        p^.ponColor(Mcolor^[x,y]);
        p^.ponPixel(x+d,y);
      end;
    end;
  end;
function colorEntorno(m : Pcolor;x,y,n : word) : byte;
begin
  case n of
    1 : colorEntorno:=m^[x-1,y-1];
    2 : colorEntorno:=m^[x,y-1];
    3 : colorEntorno:=m^[x-1,y];
    4 : colorEntorno:=m^[x+1,y];
    5 : colorEntorno:=m^[x,y+1];
    6 : colorEntorno:=m^[x+1,y+1];
  end;
end;
```

APENDICE

```
end;
function bordeEntorno(m : Pborrable;x,y,n : word) : boolean;
begin
    case n of
        1 : bordeEntorno:=m^[x-1,y-1];
        2 : bordeEntorno:=m^[x,y-1];
        3 : bordeEntorno:=m^[x-1,y];
        4 : bordeEntorno:=m^[x+1,y];
        5 : bordeEntorno:=m^[x,y+1];
        6 : bordeEntorno:=m^[x+1,y+1];
    end;
end;
constructor topologia.inicia;
begin
end;
destructor topologia.acaba;
begin
end;
function topologia.condicion(x,y : word) : boolean;
begin
    condicion:=falso;
end;
function seis_seis.condicion(x,y : word) : boolean;
VAR b,c : byte;
function entorno(x,y : word) : boolean;
VAR c : boolean;
    n : byte;
begin
    c:=falso;
    n:=1;
    repeat
        if (Mcolor^[x,y]<>colorEntorno(Mcolor,x,y,n)) and
            (Mborde^[x,y]<>colorEntorno(Mcolor,x,y,n)) then c:=cierto;
        inc(n);
    until c or (n>6);
    entorno:=c;
end;
begin
    b:=0;
    c:=0;
    if not entorno(x,y) then
        begin
            if Mcolor^[x,y]=Mcolor^[x-1,y-1] then
                begin
                    inc(c);
                    if Mborde^[x-1,y-1] then inc(b);
                end;
            if Mcolor^[x,y]=Mcolor^[x,y-1] then
                begin
                    inc(c);
                    if Mborde^[x,y-1] then inc(b);
                end;
            if Mcolor^[x,y]=Mcolor^[x-1,y] then
```

APENDICE

```

begin
    inc(c);
    if Mborde^[x-1,y] then inc(b);
end;
if Mcolor^[x,y]=Mcolor^[x+1,y+1] then
begin
    inc(c);
    if Mborde^[x+1,y+1] then inc(b);
end;
if Mcolor^[x,y]=Mcolor^[x+1,y] then
begin
    inc(c);
    if Mborde^[x+1,y] then inc(b);
end;
if Mcolor^[x,y]=Mcolor^[x,y+1] then
begin
    inc(c);
    if Mborde^[x,y+1] then inc(b);
end;
end;
if (c>=3) and (b=2) then condicion:=cierto else condicion:=falso;
end;
function ocho_cuatro.condicion(x,y : word) : boolean;
VAR
    c : boolean;
begin
    c:=falso;
    if self.puntoInterior(x,y,Mcolor^[x,y]) and
        self.conexionInterior(x,y) and self.conexionBordeInterior(x,y)
        then c:=cierto;
    condicion:=c;
end;
function ocho_cuatro.conexionBordeInterior(x,y : word) : boolean;
VAR
    n : byte;
    a : boolean;
begin
    a:=cierto;
    for n:=0 to 7 do
        if Mborde^[self.Cx(x,n),Self.Cy(y,n)] then
            case (n+1) of
                1,3,5,7 : if not (matricilla[(n+1) mod 8] or
                    matricilla[(n+7) mod 8]) then
a:=falso;
                2,4,6,8 : if not (matricilla[(n+1) mod 8] or
                    matricilla[(n+7) mod 8] or
                    matricilla[(n+2) mod 8] or
                    matricilla[(n+6) mod 8]) then
a:=falso;
            end;
        conexionBordeInterior:=a;
    end;
function ocho_cuatro.conexionInterior(x,y : word) : boolean;
VAR
    inte : boolean;
    n,m : byte;

```

APENDICE

```
begin
  n:=0;
  for m:=0 to 7 do
    if matricilla[m] then
      case (m+1) of
        1,3,5,7 : if (not matricilla[(m+1) mod 8]) and
                  (not matricilla[(m+7) mod 8])
then inc(n);
        2,4,6,8 : if (not matricilla[(m+1) mod 8]) and
                  (not matricilla[(m+2) mod 8])
and
                  (not matricilla[(m+6) mod 8])
and
                  (not matricilla[(m+7) mod 8])
then inc(n);
      end;
    if n>1 then conexionInterior:=falso else conexionInterior:=cierto;
  end;

function ocho_cuatro.puntoInterior(x,y,c : word) : boolean;
VAR
  a : boolean;
  n : byte;
begin
  a:=falso;
  for n:=0 to 7 do
    begin
      self.matricilla[n]:=falso;
      if c=Mcolor^[self.Cx(x,n),self.Cy(y,n)] and
        (not (Mborde^[self.Cx(x,n),self.Cy(y,n)])) then
        begin
          self.matricilla[n]:=cierto;
          a:=cierto;
        end;
    end;
  puntoInterior:=a;
end;

function ocho_cuatro.Cx(x,c : word) : word;
begin
  inc(c);
  case c of
    1,7,8 : Cx:=x-1;
    2,6   : Cx:=x;
    3,4,5 : Cx:=x+1;
  end;
end;

function ocho_cuatro.Cy(y,c : word) : word;
begin
  inc(c);
  case c of
    1,2,3 : Cy:=y-1;
    4,8   : Cy:=y;
    5,6,7 : Cy:=y+1;
  end;
end;
```

APENDICE

```
end;
procedure Topologia.borde(x,y : word);
begin
end;
procedure seis_seis.borde(x,y : word);
VAR b : byte;
begin
    b:=punto(x,y);
    if b>punto(x-1,y-1) then b:=punto(x-1,y-1);
    if b>punto(x,y-1) then b:=punto(x,y-1);
    if b>punto(x-1,y) then b:=punto(x-1,y);
    if b>punto(x+1,y) then b:=punto(x+1,y);
    if b>punto(x,y+1) then b:=punto(x,y+1);
    if b>punto(x+1,y+1) then b:=punto(x+1,y+1);
    if b<punto(x,y) then
        begin
            ponBorde(x,y,b);
        end;
end;
procedure ocho_cuatro.borde(x,y : word);
VAR b,c : integer;
begin
    b:=punto(x,y);c:=-1;
    if (b>punto(x,y-1)) {and (c<punto(x,y-1))} then b:=punto(x,y-1);
    if (b>punto(x-1,y)) {and (c<punto(x-1,y))} then b:=punto(x-1,y);
    if (b>punto(x+1,y)) {and (c<punto(x+1,y))} then b:=punto(x+1,y);
    if (b>punto(x,y+1)) {and (c<punto(x,y+1))} then b:=punto(x,y+1);
    if b <> punto(x,y) then
        begin
            ponBorde(x,y,b);
        end;
end;
procedure adelgazarPrevio;
VAR x,y : word;
begin
    for x:=1 to 158 do
        for y:=1 to 198 do if Mborde^[x,y] then
            begin
                if tipoTopologia^.condicion(x,y) then
                    Mborrable^[x,y]:=cierto;
            end;
end;
procedure adelgazar;
VAR x,y : word;
begin
    adelgazarPrevio;
    for x:=1 to 158 do
        for y:=1 to 198 do if Mborrable^[x,y] then
            begin
                Ncolor^[x,y]:=MbordeC^[x,y];
            end;
end;
end;
```

APENDICE

```
begin
writeln;
repeat
  writeln('Pon el tipo de topologia(4,6,8): ');
  readln(topo);
  until topo in [4,6,8];
case topo of
  6 : tipoTopologia:=new(Pseis_seis,incia);
  8 : tipoTopologia:=new(Pocho_cuatro,incia);
end;
writeln('Los ficheros de imágenes deben estar en C:\TRABAJO\GRAFICOS\');
write('Introduce la pantalla: ');
readln(nombre);
p:=new(Pgraficos1,incia);
p^.recuperaPantalla(nombre);
new(Mborde);new(Mcolor);new(MbordeC);new(Mborrable);
carga;
dibujaColor(160);
write(#7);
bordes;
write(#7);
repeat
  adelgazar;
  bordes;
  dibujaColor(160);
until keypressed;
p^.acaba;
end.
```

BIBLIOGRAFIA

BIBLIOGRAFIA

- [1]. B. Bollobás, *Graph Theory: An Introductory Course*, Graduate Texts in Math. , vol 63, Springer-Verlag, N.Y. 1979.
- [2]. L.Budach, B. Graw, C. Heini y S. Wack, *Algebraic and Topological Properties of Finite Partially Ordered Sets*, Teubner- Texte zur Mathematik, 1988.
- [3]. O.P.Buneman, *A grammar for the topological analysis of plane figures*, in Machine Intelligence, vol. 5, pp 383-393, Edinburgh Univ. Press, Edinburgh 1969.
- [4] T.N.Cornsweet, *Visual Perception*, Academic Press., NY. 1970.
- [5] E. Dominguez, A. Francés y A.Márquez, *A Framework for Digital Topology*, Prepublicación Dpto. de Ingeniería Eléctrica e Informática. Universidad de Zaragoza. RR-92-27.
- [6]. A. Dold, *Lectures on Algebraic Topology*, Springer-Verlag, 1972.
- [7]. R. O. Duda, P.E. Hart and J.H.Mundson, *Graphical Data Processing Research Study and Experimental Investigation*, AD650926, Marzo 1967, pp 28-30.
- [8]. W. Franz, *Topología General y Algebraica*, Selecciones Científicas, 1968.
- [9]. M.J.E.Golay, *Hexagonal parallel pattern transformations*, IEEE Trans. Comput.C-18, 1969.

BIBLIOGRAFIA

- [10]. G.T. Herman, *On Topology as Applied to Image Analysis*, Comp. Vision, Graph. and Image Procc., 52 pp 409-415, 1990.
- [11]. J.F.P. Hudson, *Piecewise Linear Topology*, Benjamin, New York, 1969.
- [12] W. Hurewicz and H. Wallman, *Dimension Theory*, Princeton Univ. Press. 1958.
- [13] E. Khalimsky, *Motion, Deformation and Homotopy in Finite Spaces*, Procc. 1987 IEEE International Conference on Systems, 87CH2503-1, pp 227-234. 1987.
- [14] E. Khalimsky, R. Kopperman and P.R. Meyer, *Computer Graphics and Connected Topologies on Finite Ordered Sets*, Topology and its Applications, 36, pp 1-17, 1990.
- [15] T.Y. Kong and A.W. Roscoe, *A Theory of Binary Digital Pictures*, Comp. Vision, Graph. and Image Procc., 32 pp 221-243, 1985.
- [16] T.Y. Kong and A.W. Roscoe, *Continuous Analogs of Axiomatized Digital Surfaces*, Comp. Vision, Graph. and Image Procc., 29 pp 60-86, 1985.
- [17] T.Y. Kong, *Digital Topology with Applications to Image Processing*, Tesis Doctoral, Universidad de Oxford, 1986.
- [18] T.Y. Kong and A. Rosenfeld, *Digital Topology: Introduction and Survey*, Comp. Vision, Graph. and Image Procc., 48 pp 357-393, 1989.
- [19] T.Y. Kong, A. W. Roscoe and A. Rosenfeld, *Concepts of Digital Topology*, Topology and its Applications, 46, pp 219-262, 1992.
- [20] T.Y. Kong, R. Kopperman and P.R. Meyer, *Guest Editors' Preface to Special Issue on Digital Topology*, Topology and its Applications, 46,

BIBLIOGRAFIA

pp 173-179,1992.

- [21] V.A.Kovalevsky, Finite Topology as applied to Image Analysis , Comp. Vision, Graph. and Image Procc., 46 pp 141-161,1989.
- [22] E.H. Kronheimer, *The topology of Digital Images*, Topology and its Applications, 46, pp 279-303, 1992.
- [23] J. T. Lopez-Raya, *A Condition on Thinnig in Digital Picture*, Aparecerá en Journal ofIntelligent and Robotic System
- [24] J. T. Lopez-Raya and A. Márquez, *A non-iterative method for computing the skeleton in binary digital picture*, Preprint nº19, Facultad de Matemáticas, Universidad de Sevilla. 1993.
- [25] J. T. Lopez-Raya and A. Márquez, *A Parallel Algoritm for 3-Dimensional Thinning of binary digital picture*, Preprint nº20, Facultad de Matemáticas, Universidad de Sevilla. 1993.
- [26] J. T. Lopez-Raya, *Transformaciones continuas en imágenes digitales no binarias*, comunicación en IV Encuentro de Geometría Computacional, Granada 1993.
- [27] W.K.Pratt, *Digital Image Processing*, 2 Ed.,Jhon Wiley & Sons, 1981.
- [28] V. Rohlin et D. Fuchs, *Premier cours de Topologie*, Mir, 1981.
- [29] C. Ronse, *A topological characterization of thinnig*, Theoret. Comput. Sci.,43, pp 31-41,1986.
- [30] A. Rosenfeld, *Connectivity in digital pictures*, J. Assoc. Comput. Mach., 17, pp 146-160, 1970.

BIBLIOGRAFIA

- [31] A. Rosenfeld, *Adjacency in Digital Pictures*, Inform. and Control, 26, pp 24-33 ,
1974.
- [32] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2^a ed, Vol 1 y Vol 2,
Academic Press, 1982.
- [33] M.T.Sancho de Salas, *Method of Conmutative Algebra for topology*,
- [34] E.H.Spanier, *Algebraic Topology*, Mc Graw-Hill, 1978.
- [35] R. Stefanelli and A. Rosenfeld, *Some Parallel Thinning algorithms for Digital
Pictures*, J. Assoc. Comput. Mach., 18, pp 255-264, 1971.
- [36] Y.F.Tsao and K.S.Fu, *A general scheme for constructing skeleton models*, Inform.
Sci. 27, pp 53-87, 1982.