

1400191448

• copia 2

Un sistema de modelado geométrico de sólidos
basado en el modelo de fronteras
y árboles octales

P. Brunet

Report LSI-89-4


E.T.S.E.I.B.
Departament de
llenguajes
i sistemes
informàtics
Diagonal, 647, planta 8.
E-08028 Barcelona


Facultat d'Informàtica
de Barcelona - Biblioteca

30 NOV. 1995

Abstract: This work presents the DMI Geometric Modelling System. It includes the structure of the representation schemes that are used for solids (boundary representation and extended octrees), together with the basic algorithms related to the main operations supported by the system -version of year 1985-.

Resumen: El trabajo presenta la estructura del Sistema de Modelado Geométrico de Sólidos 'DMI', detallando la estructura de los sistemas de representación utilizados (modelo de fronteras y árboles octales extendidos) y los algoritmos básicos de las principales operaciones soportadas por el sistema, en su versión del año 1985.

UNIVERSITAT POLITÈCNICA DE CATALUNYA
ESCOLA TÈCNICA SUPERIOR D'ENGINYERS INDUSTRIALS

DEPARTAMENT DE MÈTODES
INFORMÀTICS

UN SISTEMA DE MODELADO
GEOMETRICO DE SOLIDOS
BASADO EN EL MODELO
DE FRONTERAS Y ARBOLES
OCTALES.

Pere Brunet Crosa

Diciembre de 1985

Raport de Treball DMI03-85

UN SISTEMA DE MODELADO GEOMETRICO DE SOLIDOS BASADO EN EL
MODELO DE FRONTERAS Y ARBOLES OCTALES

Informe de Recerca DMI03/85

INDICE

	Página
1. Introducción	2
2. El Modelado Geométrico de Sólidos	3
3. Estructura del Sistema DMI	20
4. Operaciones básicas.	24
4.1 Entrada por barridos	24
4.2 Transformaciones Geométricas, traslación interactiva	26
4.3 Secciones planas	27
4.4 Visualización con dibujo de líneas	28
4.5 Visualización con imagen de facetas planas	29
4.6 Visualización mediante z-buffer y suavización de Gouraud.	30
4.7 Salidas sobre papel. Interfase con un sistema de dibujo 2-D.	31
4.8 Calculo de propiedades volumétricas.	31
4.9 Realización de operaciones booleanas, en el modelo de fronteras.	33
4.10 Conversión entre modelos: de modelo de fronteras a octtrees y viceversa.	34
4.11 Realización de operaciones booleanas en el modelo de octtrees.	35
5. Generación de superficies esculpidas	36
5.1 Generación interactiva de superficies bicúbicas basadas en B-splines.	51
5.2 Operaciones: revolución, obtención de trozos bicúbicos, offset.	52
5.3 Extensiones: tratamiento de superficies bicuadráticas; octtrees extendidos.	54
6. Ejemplos.	
6.1 Comentarios a las figuras	55
6.2 Figuras.	61
7. Referencias	73
Anexo	77

1. INTRODUCCION

El presente documento describe el sistema DMI, un Sistema de Modelado de sólidos experimental, diseñado y programado en su totalidad en el Departamento de Métodos Informáticos de la E.T.S. de Ingenieros Industriales de Barcelona*. El sistema ha sido construido en su versión actual sobre VAX 11/750 entre noviembre de 1984 y diciembre de 1985, utilizando trabajos parciales ya realizados desde 1981 que habian dado lugar a la implementación de algunas de las rutinas del sistema en el ordenador anterior de la ETSIIB (un ordenador Data General, Eclipse S-130). Los trabajos han sido financiados en parte por la CAICYT, a través del proyecto 2700/83. La construcción del sistema ha sido llevada a cabo, bajo la dirección del autor, por el equipo de Gráficos del Departamento, y fundamentalmente por los siguientes profesores y colaboradores,

Robert Juan	Enric Torres
Isabel Navazo	Albert Trill
Dolors Ayala	Ramón Vilà
Xavier Pueyo	Jordi Pascual
Pere Brunet	Josep Giralt

En el apartado 2, se presentan los principales conceptos relacionados con el modelado de sólidos, discutiendo sus prestaciones y las principales ventajas e inconvenientes de los distintos sistemas de representación no ambigua. Seguidamente, en el apartado 3 se describe la estructura interna del sistema DMI, junto con las extensiones previstas en un próximo futuro. El apartado 4 presenta, con un cierto detalle, los algoritmos correspondientes a las operaciones básicas del sistema en cuanto al modelado de sólidos poliédricos; y el apartado 5 indica el estado actual del diseño de objetos delimitados por superficies esculpidas. Finalmente, el apartado 6 presenta algunos ejemplos, analizando el tiempo de cálculo necesario para las distintas fases del proceso de modelado.

(*) con la colaboración, durante el año 1985, de Lola Vicente y Adrià Llisterri, del Centro de Cálculo de la UPC, bajo la dirección de R. Juan.

2. EL MODELADO GEOMETRICO DE SOLIDOS

La utilización de sistemas de CAD se encuentra cada vez más extendida en la industria. Entre sus principales ventajas podemos citar la interactividad y facilidad de crear nuevos diseños, la posibilidad de simular el comportamiento del sistema antes de la construcción del prototipo, modificando si es necesario sus parámetros, la generación de planos con todo tipo de vistas, detalles y secciones, y la posibilidad de conexión con un sistema de fabricación asistida por ordenador para la mecanización automática de un prototipo.

Los sistemas de CAD que permiten el diseño de objetos tridimensionales (diseño de piezas mecánicas, diseños en chapa, en plástico, diseños de obra civil, arquitectura y urbanismo, etc), pueden llegar a ofrecer al usuario las siguientes prestaciones,

- En un módulo de preproceso, se define interactivamente la forma tridimensional del objeto o conjunto de objetos a diseñar. El ordenador almacena un modelo tridimensional completo del mismo, que permite la generación de cualquier vista (diedrico, axonométrico, perspectivas) del mismo, así como secciones, detalles y planos. Asimismo, el modelo de representación tridimensional contiene la información necesaria para el cálculo de las propiedades geométricas del objeto que se está diseñando: superficie, volumen, peso, centro de gravedad, momentos de inercia, etc.
- En una segunda fase, de proceso, se utiliza el modelo obtenido para realizar cálculos y simulaciones más complejos, como pueden ser el cálculo de tensiones por elementos finitos, o la simulación del comportamiento aerodinámico en el caso del diseño de carrocerías, perfiles de avión etc.
- En una tercera fase se pueden visualizar gráficamente los principales resultados de los programas de cálculo. Si no son correctos, el usuario incidirá sobre la forma del objeto, modificando el modelo y repitiendo el proceso; si en cambio ya son aceptables, el sistema podrá generar automáticamente una cinta de control numérico para la generación automática de un prototipo del objeto diseñado. Este último proceso es conocido con el nombre de CAM (computer-aided manufacturing).

En el presente trabajo, nos concretaremos al Modelado Geométrico, que constituye la primera fase, de entrada y visualización de objetos tridimensionales. Este proceso de diseño geométrico de la forma del objeto final constituye una parte importante de los sistemas CAD actuales, que difícilmente llegan a ofrecer todas las prestaciones antes indicadas.

Desde un punto de vista histórico, los antecedentes del Modelado Geométrico pueden situarse en los años 1955-1964, [1], con la aparición de los primeros lenguajes de control numérico -el APT-, las primeras experiencias de salida gráfica en ordenadores y el estudio de algoritmos relacionados con la geometría proyectiva. Desde 1965 a 1972, aparecieron los primeros sistemas para dibujo en 2-D, junto con algunos algoritmos para el diseño de curvas y superficies: curvas y superficies de Coons y Bezier. No obstante, debe esperarse hasta el período 1973-78, [1], para disponer de sistemas completos de diseño de curvas y superficies basados, además de en las técnicas ya conocidas, en la aproximación por B-splines. En este mismo período aparecen las bases teóricas y los primeros sistemas experimentales de diseño de sólidos, que utilizan modelos de fronteras, CSG y barrido. Finalmente, entre 1979 y 1984, los sistemas de diseño de superficies curvadas -o superficies esculpidas- se amplían con la posibilidad de subdivisión y edición de las mismas, mientras que los sistemas de modelado de sólidos aparecen ya en el mercado. Estos últimos sistemas, que almacenan una información volumétrica completa del objeto, no tratan en cambio (en general) objetos delimitados por cualquier superficie curvada; se limitan a considerar modelos delimitados por caras planas cilíndricas, cónicas y en algunos casos esféricas- piezas mecánicas-. En la evolución prevista para los próximos años, es de esperar que los sistemas permitan tratar objetos delimitados a la vez por caras planas o cilíndricas y por superficies esculpidas.

Concretándonos a los sistemas de modelado de sólidos, su estructura debe ser tal que permita la entrada, por métodos simples, de la forma del objeto a diseñar, y el almacenamiento de un modelo geométrico completo que facilite las operaciones, representación gráfica y modificación.

Ahora bien, no todos los sistemas de almacenamiento de la información del sólido diseñado en el ordenador ofrecen las mismas prestaciones. Por ejemplo, si lo único que se ha hecho ha sido introducir información de planos ya existentes, el sólido quedará representado en el ordenador por un conjunto de vistas y secciones. Esta información no es completa, aunque se venga utilizando extensamente, por ejemplo, en los sistemas de dibujo asistido por ordenador (no de diseño): por una parte, puede ser ambigua (el conjunto de vistas que hemos almacenado corresponde a más de un sólido real); por otra parte, es extremadamente complejo obtener otras proyecciones a partir de las almacenadas.

En primer lugar, el sistema debe ser capaz de distinguir los sólidos geoméricamente correctos de aquellos que no lo son, y solo debe permitir almacenar los primeros (En muchos casos, el propio sistema de entrada ya asegura que el usuario solo pueda generar diseños correctos). En concreto, se define como sólido representable, [2], [3], el que cumple las siguientes restricciones,

- Debe ser rígido: su forma ha de ser independiente de la posición espacial y orientación.
- Debe ser finito, ocupando una porción finita del espacio.
- Su superficie externa debe ser cerrada y orientable: Ha de determinar, sin ambigüedad, qué parte es interior al sólido y que parte es exterior. Obsérvese que en la botella de Klein, no se cumple esta propiedad.
- Debe cumplir la ecuación de Euler: $C+V-A=2+R-2H$, [3], donde C es el número de caras, V es el número de vértices, A el de aristas, R el número de anillos interiores en las caras, y H el número de agujeros que atraviesan el cuerpo.
- Toda operación geométrica (traslaciones y rotaciones) o booleana (reunión, intersección o diferencia con otro sólido) entre sólidos representables debe producir como resultado otros sólidos representables. La definición de operadores regularizados [4], asegura el cumplimiento de esta propiedad.

Por otra parte, es evidente que no todo sistema de almacenamiento de sólidos representables da las mismas prestaciones. Para poder comparar entre sí los distintos métodos que se utilizan, se definen los siguientes parámetros [2],

- Dominio: cuanto mayor es el dominio en un sistema de representación de sólidos, mayor es el número de objetos reales que son representables.

- Ambigüedad: un sistema es ambiguo si un modelo representado en él corresponde a más de un sólido real.
- Grado de unicidad: es deseable que todo sólido real tenga un único modelo de representación interna, para poder detectar y evitar posibles duplicaciones en la Base de Datos.
- Validez: las representaciones internas deben corresponder a sólidos correctos.
- Ocupación de memoria: La cantidad de memoria necesaria para almacenar un determinado sólido real debe ser reducida.
- Facilidad de creación de objetos: las herramientas para la creación de diseños por parte del usuario deben ser simples.
- Finalmente, el sistema de representación debe permitir la generación, con facilidad, de todo tipo de visualizaciones, así como el cálculo de propiedades y las operaciones con otros sólidos.

No existe en estos momentos ningún sistema de representación de sólidos óptimo con respecto a todos estos parámetros. En lo que resta de este apartado analizaremos los más utilizados en la actualidad, comparándolos por lo que respecta a estas características.

2.1 El modelo de alambres

En realidad, sólo es posible obtener todo tipo de proyecciones del objeto diseñado si el ordenador almacena información tridimensional. Uno de los sistemas, tal vez el más sencillo, es el llamado modelo de alambres. En él, el ordenador dispone de las coordenadas en el espacio de todos los vértices del cuerpo, junto con información de qué pares de vértices se encuentran unidos mediante aristas. Mediante sencillas transformaciones geométricas de proyección, se puede obtener cualquier vista del objeto. Sin embargo, este modelo de alambres tiene el grave inconveniente de

ser ambiguo [2], y de no permitir la producción de secciones y vistas con eliminación de partes ocultas. Por todo ello, es poco utilizado en sistemas avanzados de diseño. Como muestra de la ambigüedad de este sistema, la figura 1-a presenta un modelo de alambres: dos paralelepípedos, uno dentro del otro, con los vértices homólogos unidos por aristas. Es inmediato observar que este modelo puede corresponder tanto al sólido 1-b como al 1-c.

Existen algunos sistemas interactivos de generación de modelos tridimensionales no ambiguos (apartados siguientes) a partir de vistas en diédrico [5] o del modelo de alambres [6]. En todos ellos es precisa la intervención del usuario en el caso de modelos ambiguos, para decidir entre los posibles sólidos resultantes (decisión entre 1-b y 1-c en el caso del modelo de la figura 1-a).

2.2 El Modelo de Fronteras (figura 2)

Uno de los sistemas de representación más utilizados en la actualidad, es el llamado modelo de fronteras. En este sistema, lo único que se hace es ampliar la información que almacenaba el modelo de alambres, incluyendo datos de los polígonos (un polígono es un conjunto de aristas, habitualmente plano y cerrado) y de las caras del objeto (una cara está formada por uno o más polígonos; en el segundo caso, uno de ellos constituye la frontera -cerrada- de la cara, y los demás pueden ser, por ejemplo, agujeros). El modelo de fronteras contiene toda la información tridimensional, es no ambiguo (representa un único sólido real), es completo (permite representar un gran número de sólidos, incluyendo por ejemplo la posibilidad de caras cilíndricas y cónicas) y posibilita todo tipo de operaciones y representaciones realistas del sólido.

Una de las técnicas más utilizadas en los Sistemas de Modelado Geométrico para la creación de nuevos sólidos es el llamado método de barrido (sweep). Con este sistema, el usuario genera el objeto tridimensional mediante traslaciones o rotaciones de caras planas que dibuja en pantalla. Las figuras 3 y 4 muestran las posibilidades del Sistema de Modelado con representación interna de fronteras que se presenta. En concreto, la figura 3 describe el proceso detallado de generación de una pieza mecá-

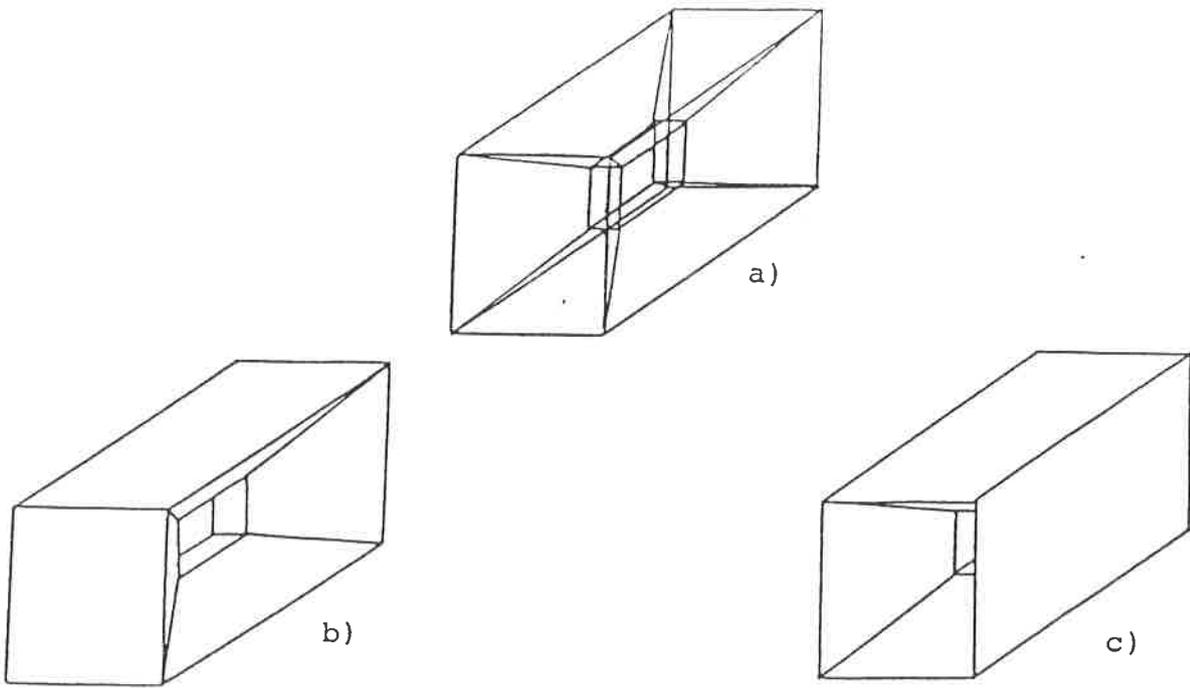


Figura 1. a), modelo de alambres. En b) y c), dos sólidos distintos que corresponden al mismo modelo.

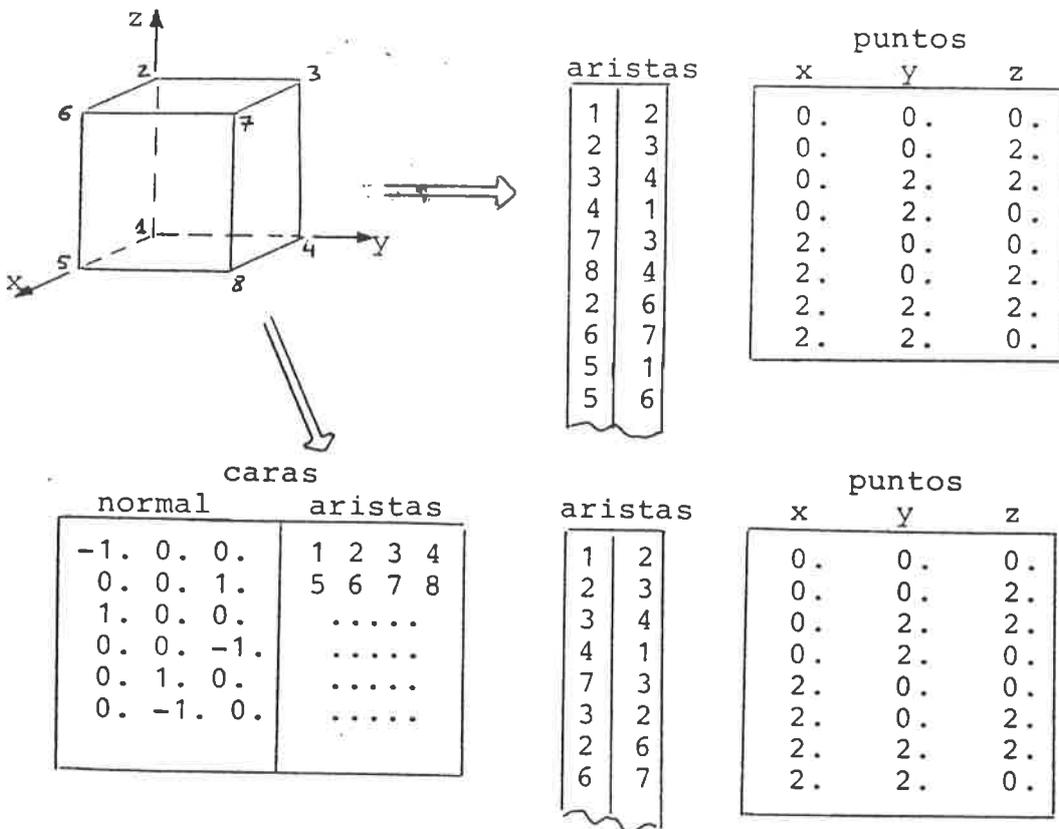
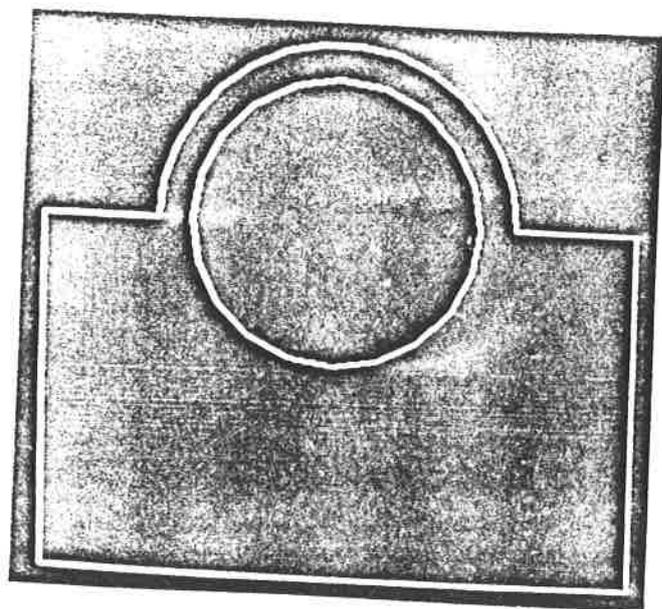


Figura 2. Arriba, información que contiene el modelo de alambres: coordenadas (x,y,z) de los puntos, junto con las aristas. Abajo, en el modelo de fronteras se ha incorporado la información de las caras. (Véase un modelo más completo, que permite caras con agujeros, en la figura 8).

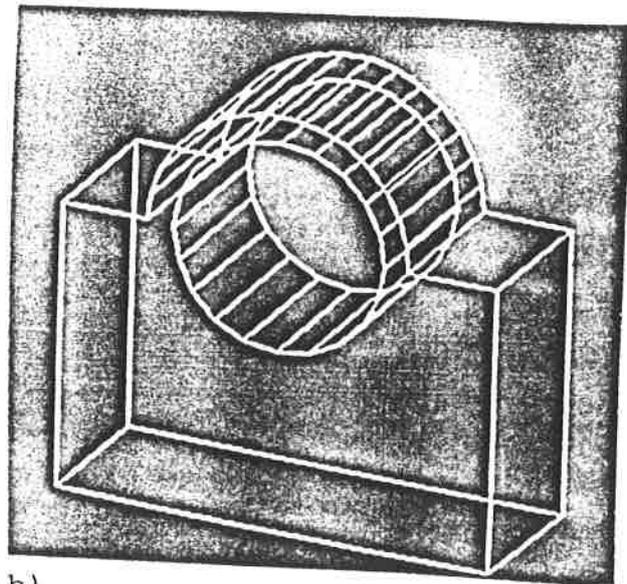
nica. En 3-a, se ha dibujado -mediante técnicas usuales de dibujo en 2-D- una de las caras del objeto. Indicando al sistema cual debe ser su espesor, éste genera un modelo tridimensional de fronteras que nos permite generar, por ejemplo, la vista axonométrica de 3-b. El barrido puede repetirse, y puede ser hacia el interior o el exterior del objeto: En la figura 3-c, un barrido vertical hacia el interior de dos polígonos hexagonales laterales, produce el "mecanizado" de dos agujeros. El usuario, en la figura 3-c, ha seleccionado ya la cara inferior -a través- para dibujar en ella un rectángulo al cual aplicará un pequeño barrido hacia el interior del objeto (figura 3-d); en esta misma figura, se ha pedido un último barrido, éste hacia el exterior del objeto, de un polígono con una concavidad semicircular dibujado en una cara lateral.

En este sistema, como se observa en las figuras 3 y 4, las caras cilíndricas se aproximan por un número suficientemente elevado de caras planas. Mientras algunos Sistemas de Modelado incluyen caras analíticamente cilíndricas, otros -ver también[9], por ejemplo- consideran preferible su aproximación por caras planas, con el fin de aumentar la velocidad en los algoritmos de tratamiento del modelo de fronteras. En todo caso, es siempre posible, en el momento de la representación gráfica final y generación de planos, la obtención de una aproximación a la superficie real dentro de cualquier tolerancia especificada, mediante interpolación por splines.

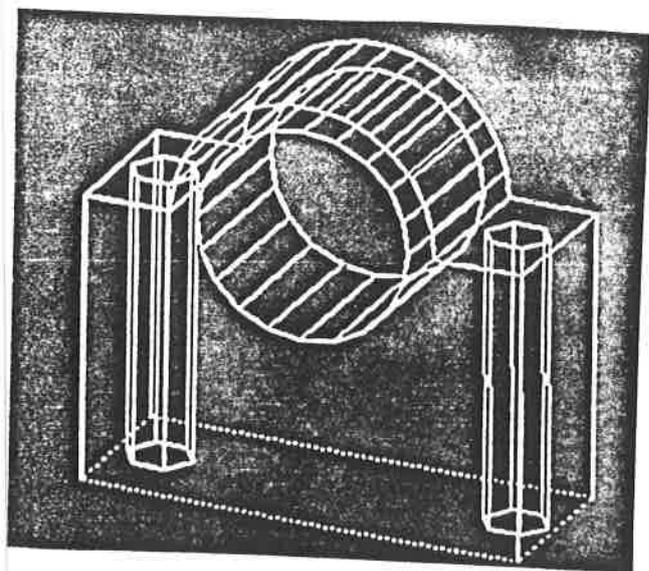
La figura 4 muestra diversas visualizaciones del objeto diseñado en la figura 3 que ponen de manifiesto la no ambigüedad del modelo de fronteras. Puede observarse un dibujo con eliminación de las líneas ocultas en 4-a (dibujo que no sería posible con un modelo de alambres, en que el sistema no posee información de los elementos opacos -las caras-); en las figuras 4-b y 4-c se presenta el objeto seccionado por varios planos (el resultado de la sección es otro sólido, con información tridimensional completa almacenada con el modelo de fronteras). Finalmente, una doble sección del sólido, en la figura 4-d, puede representarse con simulación de iluminación en una pantalla de barrido.



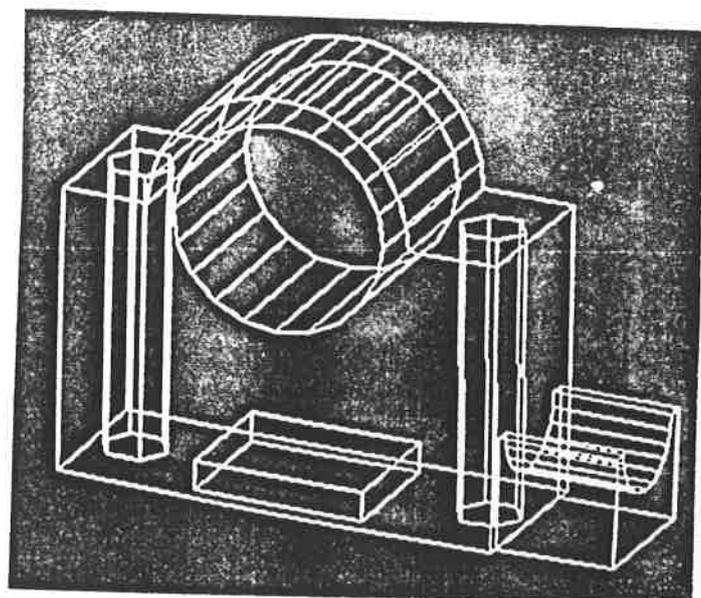
a)



b)



c)



d)

Figura 3. Proceso de diseño de un sólido por barridos sucesivos.

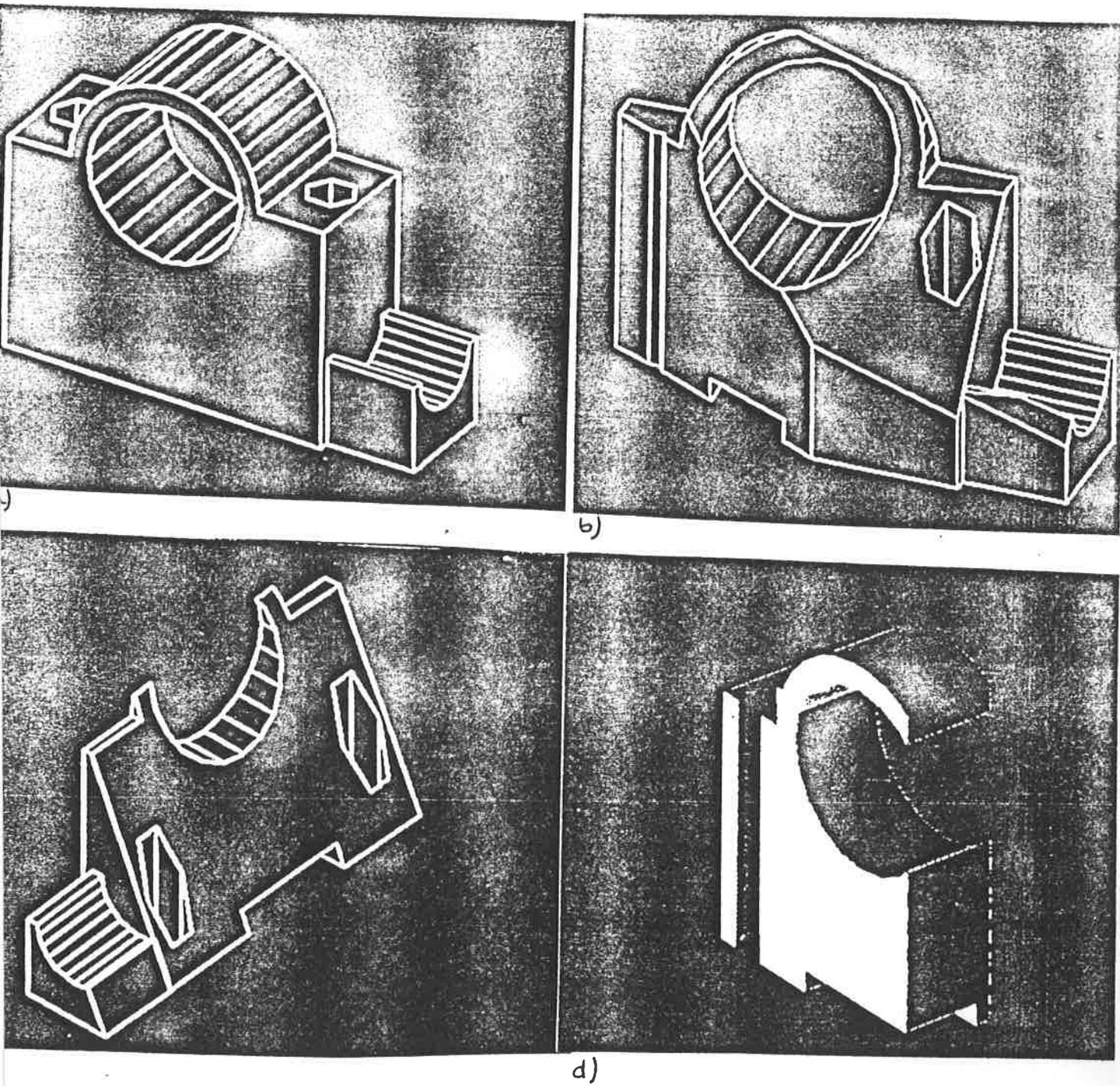


Figura 4. Diversas visualizaciones y secciones del objeto diseñado en la figura 3.

2.3 La Representación Mediante Octtrees

Aparte del modelo de fronteras, existe otra gran familia de esquemas de representación interna no ambigua, denominados métodos de enumeración espacial. En ellos, divide el espacio en una serie de celdas, y para cada una de ellas se guarda información de si es interior o exterior al objeto. Existen no obstante, diversas opciones en lo que respecta a la forma y tamaño de las celdas elementales. La elección más simple es suponer el espacio dividido en una malla de cubos idénticos en cuanto a tamaño. Si la porción de espacio en que tenemos el objeto a modelar se divide en N divisiones según cada eje, obtendremos N^3 cubos, equivalentes a un total de N^3 bits (1 si el cubo es interior, 0 si es exterior) para representar cualquier objeto. Aunque este sistema de representación es claramente no ambiguo, válido, y asegura la unicidad de los objetos modelados, la ocupación de memoria es excesiva; por otra parte el objeto no queda representado de forma exacta: las caras inclinadas, por ejemplo, quedan almacenadas como una escalera de pequeños bloques cúbicos.

Una mejora que reduce considerablemente la memoria necesaria consiste en la codificación mediante octtrees (o árboles octales). El espacio se divide en cubos, pero no todos son del mismo tamaño; solo se llega a la mínima división en la periferia del objeto, no en su interior, [10], [11]. El espacio cúbico inicial que contiene el objeto a diseñar se divide en 8 octantes, y para cada uno de ellos se analiza si es blanco (exterior al objeto), negro (interior) o gris (en parte interior y en parte exterior). Se vuelven a dividir en 8 octantes sólo estos últimos nodos grises, hasta llegar a nodos blancos o negros o bien alcanzar la mínima división. El modelo resultante tiene estructura de árbol, ya que de cada nodo gris descienden 8 nodos correspondientes a sus octantes. Existen diversos métodos para la representación del árbol en memoria, [10], [11], [12], que permiten un gran ahorro respecto a la simple enumeración espacial. Por otra parte, se tienen algoritmos simples para la visualización de un octtree; también es simple el cálculo de las propiedades volumétricas (la masa, por ejemplo, es la suma de masas de los nodos negros), y la realización de operaciones de reunión, intersección y diferencia entre objetos (únicamente deben intersectarse los nodos cúbicos negros), que conducen a algoritmos recursivos muy claros. Sin embargo, la ocupación de memoria es generalmente

mucho más alta que en el modelo de fronteras, y todavía tenemos el problema de una representación escalonada en la superficie del objeto.

En [13], [14] se presenta un nuevo esquema de representación basado en octrees en el que se permiten tipos de nodos distintos a los clásicos (blanco, negro, gris): nodos cara, que pueden contener parte de una de las caras del objeto; nodos arista, que contienen parte de una arista junto con las caras que confluyen en ella, y nodos vértice, que contienen uno de los vértices del poliedro representado. Como se observa en la figura 5, el número total de nodos y por tanto la memoria necesaria se reduce considerablemente [14]. La visualización y la posible reconversión al modelo de fronteras son ahora exactas, mientras que los algoritmos de operaciones booleanas continúan siendo simples [14].

2.4 La Representación Mediante Arboles CSG

En los sistemas que usan el modelo de Geometría Constructiva de Sólidos, (CSG), en vez de generar volúmenes a partir de caras, el usuario puede combinar sólidos elementales (prismas, cilindros, conos y esferas), moldeando con ellos la forma del cuerpo final. Las operaciones que puede realizar con estos sólidos primitivos son:

- 1) Traslación, escalado y rotación, para situar las primitivas en la posición adecuada
- 2) Unión;
- 3) Intersección;
- 4) Diferencia.

Estos sistemas almacenan únicamente las primitivas utilizadas y el conjunto de operaciones que se ha realizado con ellas. Así, en la generación del objeto en la parte superior de la figura 6, el usuario ha empezado uniendo un paralelepípedo y un cilindro; restando de este conjunto otro cilindro concéntrico al primero consigue practicar un agujero; finalmente, la parte central puede desaparecer si restamos un paralelepípedo de dimensión adecuada. La información que guarda el sistema es únicamente,

- Las dimensiones de las primitivas utilizadas
- La localización espacial de las mismas
- Un árbol con las operaciones realizadas: Unión, Diferencia, Intersección.

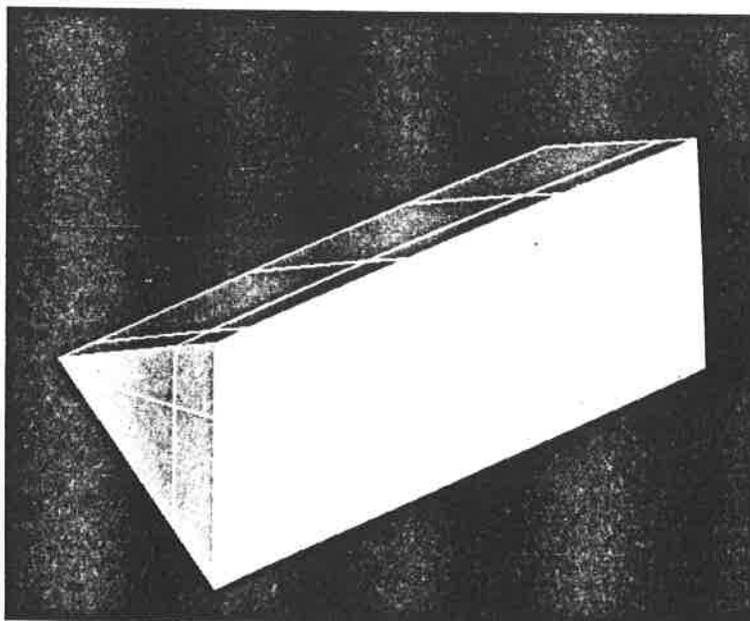
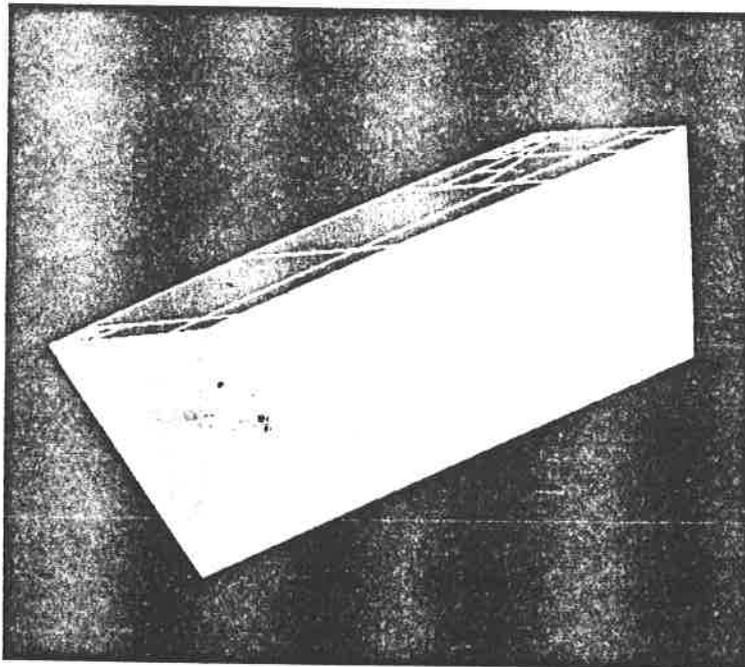
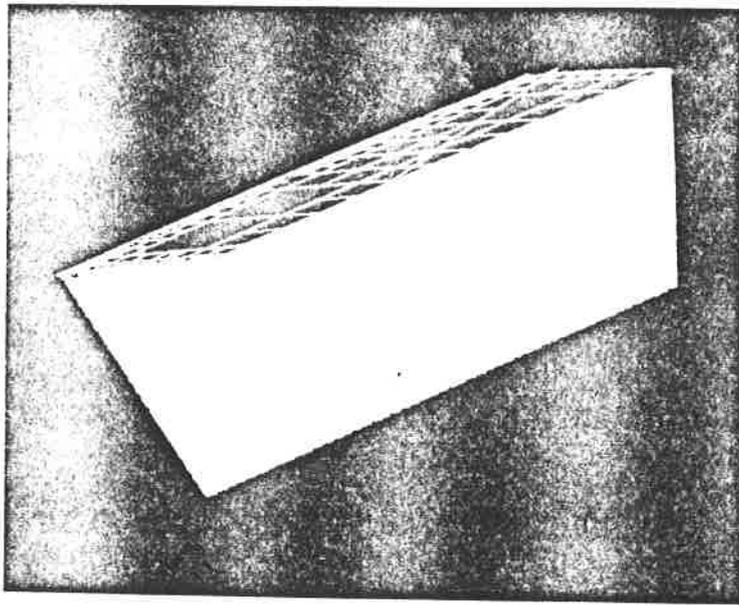


Figura 5. Representación mediante octrees. Distintas aproximaciones del mismo objeto, incluyendo nodos cara, arista y vértice respectivamente

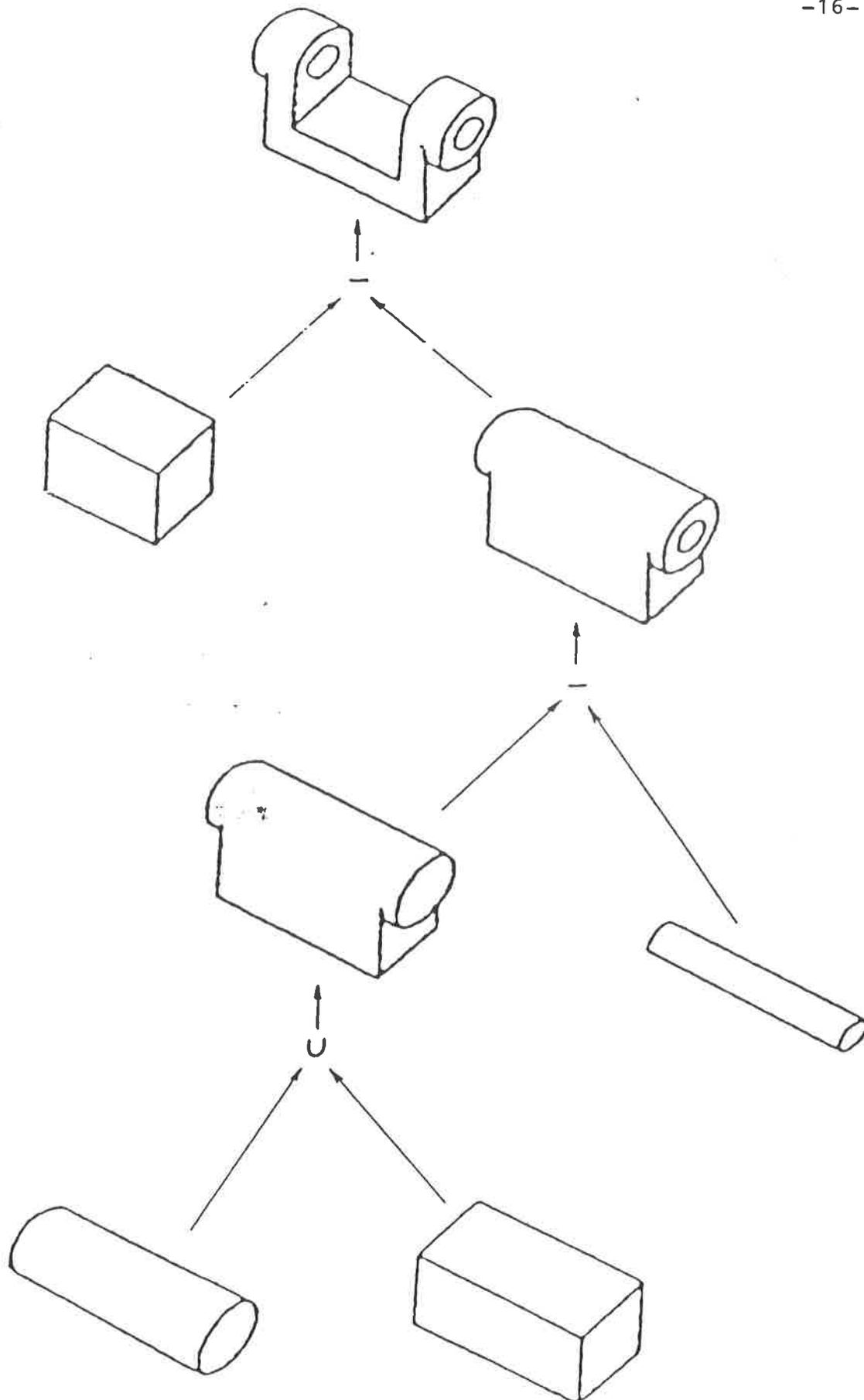


Figura 6. Geometría constructiva de sólidos: los objetos se crean y representan a partir de sólidos primitivos, que se operan una vez posicionados y dimensionados. Se almacena el árbol de operaciones booleanas realizadas.

y por tanto la representación interna es mucho más compacta que cuando se utiliza el modelo de fronteras o el de octrees. El cálculo de propiedades volumétricas no es simple, en cambio, a diferencia de los métodos de enumeración espacial: deben combinarse las propiedades de las primitivas. Por otra parte, las operaciones booleanas entre objetos se reducen a combinar los dos árboles CSG para producir el árbol final. Ahora bien, en la actualidad sólo existen algoritmos aproximados para la visualización directa de árboles CSG en tiempo real, [15], [16], [17]. Si lo que se desea es una representación exacta del objeto diseñado, para la realización de planos o control de un proceso de CAM, es preciso efectuar una conversión al modelo de fronteras, y esta conversión es costosa, [2], [18].

2.5 Conversión entre sistemas de Representación

Los sistemas de representación no ambiguos presentan prestaciones muy distintas en cuanto a los distintos parámetros definidos al principio de este apartado. Así, y como ya se ha visto, mientras el modelo de fronteras es especialmente adecuado para la generación de proyecciones y vistas, los octrees y árboles CSG son útiles para la realización de las operaciones de unión, diferencia e intersección. Algunos Sistemas de Modelado Geométrico, en consecuencia, utilizan simultáneamente más de un sistema de representación [1], [18], y poseen algoritmos para la actualización constante de todos ellos con el fin de evitar incoherencias en la Base de Datos. Los cambios de representación más utilizados son los siguientes, [18],

- Conversión de barridos (conjuntos de caras y movimientos de rotación o traslación) a modelo de fronteras. En este caso, el algoritmo únicamente ha de generar el conjunto de caras que aparecen en el movimiento de barrido.
- Conversión del modelo de fronteras a octrees. Supone estudiar para cada nodo cúbico, si es interior o no al cuerpo, utilizando los algoritmos clásicos de determinación de puntos interiores [18]. La conversión puede ser útil para la realización de operaciones booleanas.
- Conversión de árboles CSG a octrees. Como en el caso anterior, se utilizan algoritmos de detección de puntos interiores.

- Conversión de octrees a modelo de fronteras. Como ya se ha mencionado en el apartado 2.3, solo es posible realizarla de forma exacta si los tipos de nodos permitidos incluyen nodos que contienen una o más caras.
- Conversión de árboles CSG a modelo de fronteras. Es imprescindible para la visualización y dibujo preciso del modelo diseñado. Uno de los posibles algoritmos, [18], intersecta -en una primera fase- todos los objetos primitivos entre sí. Los trozos de caras de primitivas a que da lugar este proceso se almacenan y en una segunda fase se analizan con respecto al árbol CSG, eliminando las caras interiores o exteriores al objeto representado por él.

En cambio las conversiones de modelo de fronteras o de octrees a modelo CSG no son únicas, y prácticamente no se utilizan en los Sistemas de Modelado.

2.6 Prestaciones de los sistemas de modelado

La fase final de todo proceso de modelado, una vez diseñada interactivamente la forma del sólido y modificada con las operaciones booleanas y de sección, debe generar salidas numéricas y gráficas que permitan el análisis de sus características y la posible fabricación de un prototipo. Entre este tipo de salidas se encuentran,

- Las propiedades volumétricas -o integrales- del mismo: volumen, masa, momentos y productos de inercia. Estos valores se calculan mediante algoritmos conocidos [19], [20], directamente a partir del modelo de representación. En concreto, el cálculo a partir del conocimiento del modelo de fronteras es particularmente simple, [19] ya que se reduce a la integral de determinadas funciones a lo largo de la superficie exterior del objeto. Lógicamente, los cálculos son también inmediatos en cualquier sistema de enumeración espacial.
- La representación del objeto mediante proyecciones bidimensionales. Como ya se ha indicado, el modelo de fronteras es el más adecuado para este tipo de salidas. El proceso de representación consta de dos etapas, en la primera de las cuales se obtiene -mediante una transformación geométrica, [21]- la proyección de todos los puntos del modelo. En una segunda fase, puede procederse a una eliminación de las líneas -aristas- que quedan ocultas por las caras del objeto, o bien a una elimina-

ción de superficies ocultas, [21], [22]. En el primer caso, se genera una vista o dibujo, figura 3-a, mientras que en el segundo se obtiene una imagen con simulación de iluminación, figuras 3-d y 4.

- La conexión del objeto con otros elementos de una Base de Datos, [23], [25]. Estos elementos pueden ser objetos diseñados por el mismo sistema, con lo que es posible generar conjuntos con estructura de Base de Datos Gráfica jerárquica -figura 4-, [7]. En ellos el diseño está formado por un cierto número de subconjuntos, y estos a su vez por elementos cada vez más detallados). No obstante, puede conectarse la información gráfica diseñada con elementos de una Base de Datos no gráfica; en este caso, el conocimiento de estos atributos numéricos permitirá la generación, por ejemplo, de presupuestos y listas de materiales.
- La generación de las cintas de control numérico [24], para la mecanización automática del objeto diseñado.

A continuación se presenta una tabla comparativa de los principales sistemas de modelado existentes. En ella se comparan el sistema de representación, los métodos que puede utilizar el usuario para entrada de objetos, y el lenguaje de programación. Se indica asimismo una referencia bibliográfica correspondiente a cada uno de ellos.

SISTEMA	Entrada	Sistemas de Representación	Lenguaje	País de Origen	Referencia
EUCLID	CSG, Barrido	Fronteras	Fortran	Francia	26
ROMULUS	CSG, Barrido, OPE	Fronteras	Fortran	G.B.	27
SYNTAVISION	CSG	CSG(Primitivas) Fronteras (2°)	Fortran	USA	28
MEDUSA	CSG, Barrido	Fronteras	Fortran	G.B.	29
PADL-2	CSG	CSG(Primitivas) Fronteras (2°)	Fortran	USA	30, 31
TIPS GEOMAP	CSG	CSG (semiespacios)	Fortran	Japón	32
GM Solid	CSG, Ba-	CSG y Fronteras	PL/I	USA (General Motors)	33
BUILD	CSG, OPE	Fronteras	Algol	G.B.	34

Por lo que respecta a los sistemas de entrada, puede observarse que oscilan entre las técnicas de barrido y la entrada directa de las operaciones entre primitivas que definen el árbol CSG. No obstante, tanto ROMULUS como BUILD poseen un sistema adicional (OPE, Operadores de Euler) que permite la inclusión y eliminación de caras, aristas y vértices individuales; el sistema comprueba en todo momento la validez del modelo sólido interno. En cuanto al sistema de representación, se utiliza el modelo de fronteras y el CSG. SYNTAVISION y PADL contienen un esquema de representación primario basado en CSG sobre primitivas geométricas, junto con un modelo secundario de fronteras. En cambio, en GM Solid, los dos modelos son coexistentes. Tanto EUCLID como MEDUSA aproximan las superficies no planas -cilíndricas- por un número suficientemente elevado de facetas planas. Finalmente, el esquema CSG utilizado en TIPS utiliza como primitivas, semiespacios en vez de objetos geométricos sencillos.

3. ESTRUCTURA DEL SISTEMA DMI

La estructura del sistema DMI se representa esquemáticamente en la figura 7. Todas las operaciones en el sistema se apoyan sobre una representación no ambigua de los sólidos en Modelo de Fronteras, que para determinadas operaciones (fundamentalmente las booleanas) puede ser convertido a un modelo secundario de octtrees extendidos con nodos cara, arista y vértice.

Los distintos procesos que actúan sobre los modelos de fronteras y octtrees, pueden agruparse según se utilicen para la entrada, operación o visualización de sólidos:

a) Operaciones de entrada

- Entrada de sólidos poliédricos, por técnicas de barrido.
- Generación de superficies esculpidas de tipo B-splines
- Obtención de superficies de revolución, y generación de sólidos mediante técnicas de creación automática de grosor (offset)

b) Operaciones entre sólidos

- Transformaciones geométricas, traslación interactiva
- Secciones planas
- Operaciones booleanas, en el modelo de fronteras

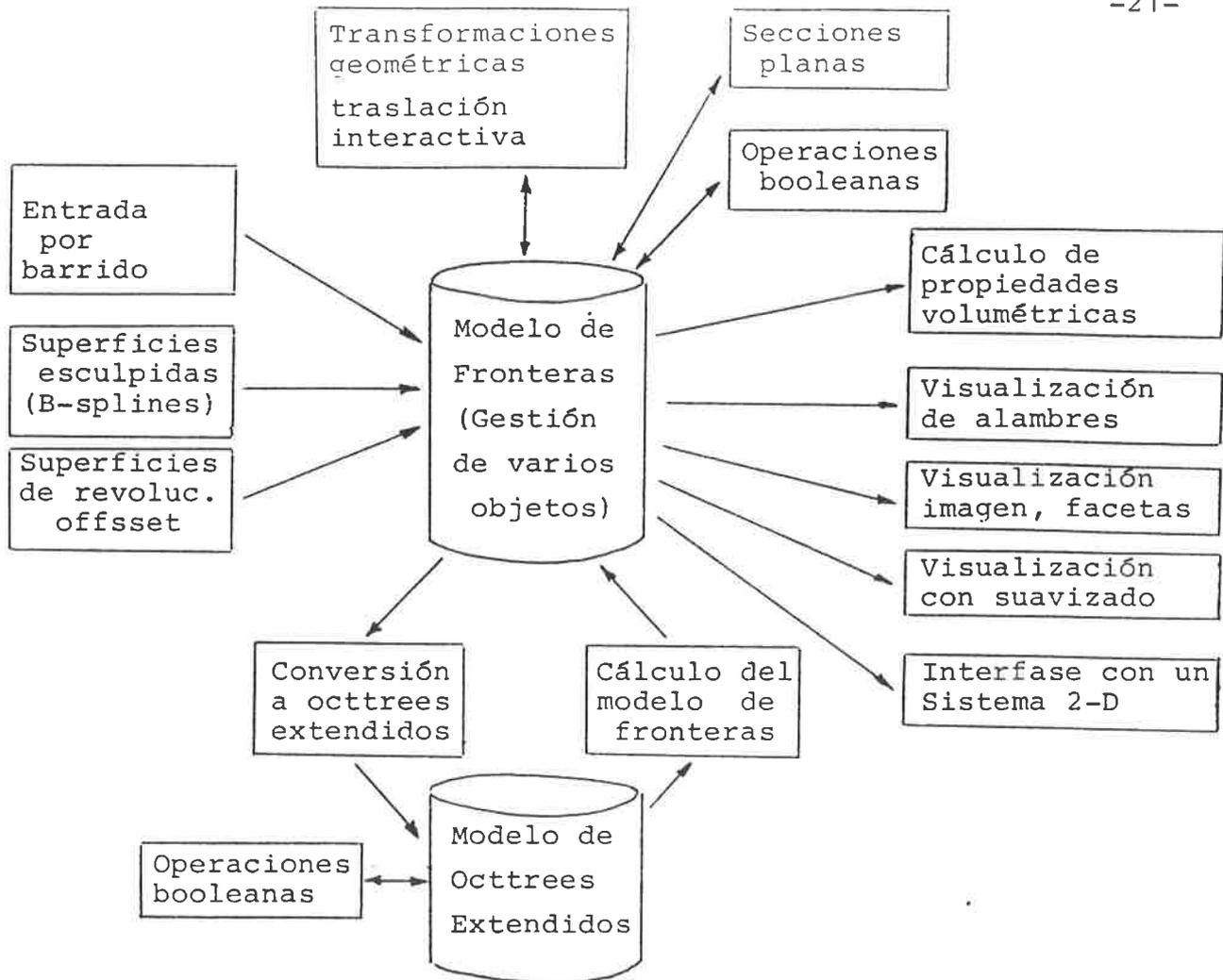


Figura 7. Estructura del sistema DMI.

- Conversión entre modelos
- Realización de operaciones booleanas y secciones con el modelo secundario de octrees extendidos.

c) Operaciones de visualización y salida

- Obtención de propiedades volumétricas.
- Visualización de alambres
- Visualización con eliminación de líneas ocultas
- Visualización con imagen de facetas planas aproximadas
- Visualización con suavización de las partes curvadas
- Obtención de dibujos en plotter.
- Conexión con sistemas de dibujo 2-D.

Todas estas operaciones se detallan en los apartados 4.1 ... 4.11 , 5.1 y 5.2 .

La mayoría de operaciones actúan directamente sobre la estructura de datos central, y son seleccionadas por el usuario a través de menú. Sin embargo, en este momento, los módulos de entrada de superficies y los correspondientes al modelo secundario de octrees extendidos forman parte de programas independientes que se comunican con el paquete general a través de archivos. En los próximos meses, cuando se disponga de versiones definitivas de los algoritmos, se incorporarán al resto del sistema, trabajando directamente con los modelos en memoria.

En un próximo futuro, se desea ampliar el sistema actual con los siguientes módulos,

a) Entrada

- Generación de objetos simples por instanciación de primitivas
- Obtención directa del modelo de fronteras a partir de la digitalización de planos en diédrico.
- Generación de superficies de unión, superficies dirigidas por un eje, y realización de operaciones con trozos bicúbicos y bicuadráticos. Obtención de superficies por subdivisión recursiva (Doo y Sabin, [63])

b) Sistema de representación y operaciones.

- Operación de pegado de cuerpos
- Sistema de representación de superficies esculpidas por árboles octales extendidos.

c) Salida y visualizaciones.

- Visualización por el algoritmo de Warnock
- Visualización por técnicas "ray-tracing", con sombras
- Algoritmos de visualización de trozos bicúbicos.
- Conexión del Modelo de fronteras con el paquete Modulef, de análisis por elementos finitos.

Pasamos a describir el Modelo de Fronteras utilizado, que contiene una estructura de apuntadores en forma de árbol, sin apuntadores hacia atrás.

Según se observa en la figura 8, el modelo consta de cuatro estructuras: caras, polígonos o anillos, aristas, y vértices. Esquemáticamente, la estructura correspondiente a las caras posee su ecuación matemática, junto con un apuntador al primer polígono de la cara. Dado que en el modelo solo se admiten caras planas, se almacena únicamente el vector normal unitario de dirección hacia el exte-

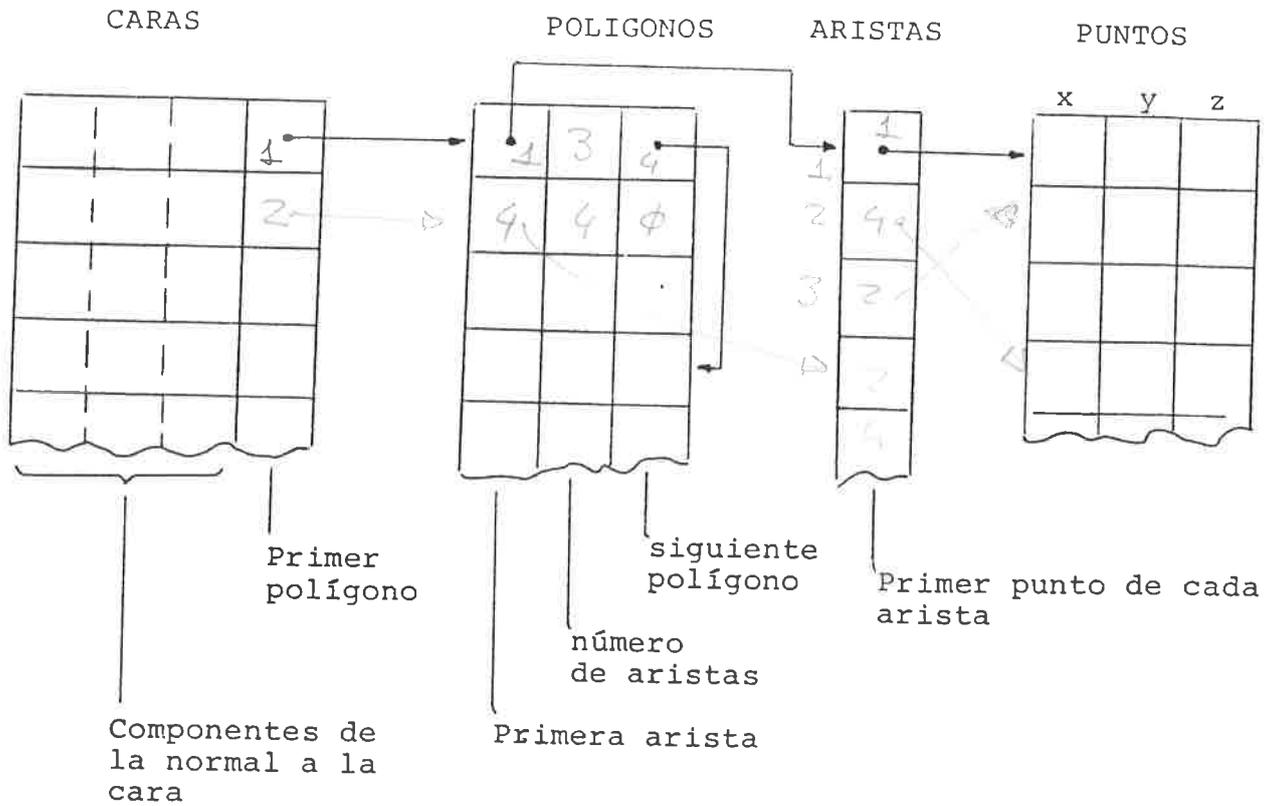


Figura 8. Estructura esquemática del modelo de fronteras utilizado

rior del objeto; la ecuación del plano de la cara puede deducirse inmediatamente a partir de esta normal y de las coordenadas de uno de sus puntos.

El primer polígono de cualquier cara está formado por un conjunto cerrado de aristas que forman su frontera. No obstante, una cara puede poseer más de un polígono si contiene agujeros o protuberancias del sólido. En este caso, los distintos polígonos de la cara se encadenan, figura 8. La estructura de polígonos contiene además el apuntador a su primera arista, y el número de aristas que lo forman. Las aristas de un mismo polígono se almacenan de forma contigua, y contienen únicamente el primero de sus puntos. Finalmente, para cada punto se almacenan sus coordenadas 3-D. Esta última información, junto con la de los vectores normales a las caras, es la única que se modifica en las operaciones de transformación geométrica (apartado 4.2).

4. OPERACIONES BASICAS

4.1 Entrada por barridos

En el proceso de creación de nuevos objetos por barrido, el usuario realiza, esquemáticamente, un proceso con las siguientes etapas,

- a) Especificación de parámetros generales: tolerancia para la aproximación de cilindros por facetas planas, plano en el que va a trabajar en el espacio
- b) Dibujo de una primera cara del objeto, con las técnicas clásicas de los sistemas de dibujo 2-D (Anexo y ref. 61). En esta cara, pueden dibujarse agujeros.
- c) Realización de un primer barrido (de traslación, rotación o de conificación). A partir de este momento, el sistema muestra un conjunto de 4 vistas del objeto, incluyendo el axonométrico, en representación de alambres. Es posible modificar en todo momento los ángulos que definen el axonométrico. Ver figuras 9-a, 9-b, 11-a y 11-b, 13-a y 13-b.
- d) El usuario decide si quiere terminar el proceso (saltando al paso h), eliminar un barrido erróneo, o continuar complicando el objeto con nuevos barridos. En este último caso, se continúa con la etapa e) . Para eliminar un barrido erróneo, se señala una cara en cualquier vista, y desaparece automáticamente el barrido que ha producido esta cara. Se sigue con el paso e), o con el f) si no se ha deseado eliminar el polígono inicial que ha generado el barrido.
- e) Se selecciona, con el cursor, una cara del objeto en cualquiera de las vistas. La cara aparece en verdadera magnitud, y es posible dibujar polígonos en ella, con las técnicas de dibujo 2-D (anexo). En todo momento se conocen las coordenadas x-y-z del cursor.
- f) Se realiza un nuevo barrido con los polígonos introducidos en la fase anterior. El barrido puede ser de traslación, rotación o conificación. Si es hacia el exterior del cuerpo, se producirá una protuberancia, figuras 11-c y 12-a . Si en cambio, se dirige al interior del objeto, se produce un entrante o una perforación, figuras 9-d y 12-a.
- g) Se vuelve al paso d).
- h) Si el objeto es correcto, puede almacenarse en la base de datos, para lo cual el sistema nos pregunta su nombre.

en el caso que, en vez de crear un nuevo objeto, se desee modificar un objeto ya existente, el proceso empieza directamente en el paso d).

El algoritmo del proceso de entrada asegura que en todo momento se dispone de un modelo de fronteras no ambiguo del objeto que estamos diseñando, modelo de fronteras que en la fase h será directamente transmitido a la Base de Datos general del sistema. En concreto,

- En la fase b, el modelo de fronteras almacena una única cara, junto con sus posibles polígonos agujero, aristas y coordenadas de puntos.
- En la fase c, el modelo de fronteras se complementa con la información de todas las caras que crea el proceso de barrido y que cierran el cuerpo. Se calculan las coordenadas de los nuevos puntos, y el vector normal a todas las caras. Al finalizar esta etapa, el modelo de fronteras es siempre cerrado y no ambiguo.
- En la fase e, se pasa a un modelo temporalmente incompleto, que almacena algunos polígonos agujero en la cara seleccionada.
- En la fase f, al igual que en la c, el modelo de fronteras vuelve a contener información coherente, ya que se complementa el modelo de fronteras con la información proveniente de las caras producidas en el proceso de barrido.

El cálculo de la normal a una cara cuyo polígono frontera contenga los puntos (x_i, y_i, z_i) , $i=1\dots n$ puede realizarse a partir de las siguientes ecuaciones,

$$n_x = \sum_{i=1}^n (y_i - y_j) (z_i + z_j)$$

$$n_y = \sum_{i=1}^n (z_i - z_j) (x_i + x_j)$$

$$n_z = \sum_{i=1}^n (x_i - x_j) (y_i + y_j)$$

donde se ha supuesto que $x_{n+1}=x_1, \dots, z_{n+1}=z_1$. Las ecuaciones anteriores se basan en la orientación de la proyección de la cara sobre cada uno de los tres planos coordenados.

4.2 Transformaciones Geométricas. Traslación interactiva.

El sistema permite la traslación, escalado, giro o simetría de cualquier objeto en la Base de Datos. Esta operación afecta únicamente a las coordenadas de los puntos del objeto y a los vectores normales a las caras, que deben ser modificados de acuerdo con la transformación geométrica realizada, [36]. Las transformaciones de escalado y giro se realizan entrando los parámetros que las definen por teclado; en cambio, la traslación y simetría pueden realizarse también gráficamente. En concreto, si se desea trasladar un objeto A respecto a otro B, aparece en pantalla el conjunto de los dos objetos, con representación en modelo de alambres y tres vistas (planta, alzado, axonométrico). En este momento, el usuario debe realizar los siguientes pasos,

- Seleccionar, con el cursor, una de las vistas.
- Seleccionar un punto del objeto a trasladar.
- Señalar la nueva posición de este punto. Puede optarse por depositar directamente el punto seleccionado en esta nueva posición, o situarlo en las mismas coordenadas del punto más cercano del objeto de referencia.

Existe también la posibilidad de traslación geométrica con copia del objeto, que en lo sucesivo denominaremos instanciación. En este caso, la traslación, giro, simetría o escalado del objeto A produce un nuevo objeto B, sin modificar la situación de A. De cara al usuario, el objeto B tiene los mismos atributos que A, y pueden realizarse las mismas operaciones con uno y otro. Sin embargo, el sistema almacena únicamente la información geométrica de A; el cuerpo B contiene solamente la transformación geométrica aplicada y un apuntador a A.

Las figuras 13-c y 15 muestran el efecto de la instanciación de objetos. En 14-a y 14-b aparecen las posibilidades de la traslación interactiva.

4.3 Secciones planas.

Una vez especificado un objeto (o conjunto de objetos) y un plano de corte, el sistema produce automáticamente la sección plana. Se visualiza en distinto color las dos partes del objeto seccionado, y el usuario puede escoger,

- guardar una de las dos partes en la Base de Datos
- guardar las dos partes en la Base de Datos
- no guardar el resultado de la sección.

En los dos primeros casos, el resultado de la sección es un nuevo objeto con modelo de fronteras completo. En este sentido, es posible utilizar la operación de sección como una herramienta de modelado de la forma durante el proceso de diseño.

La figura 10-3, así como la 16, muestran distintos resultados del algoritmo de sección.

El algoritmo sigue básicamente las siguientes etapas,

- 1- Clasificación de todos los puntos del objeto respecto al plano de sección. Se considera que el punto P_k es positivo si formará parte del resultado de la sección.
- 2- Para cada cara del objeto a seccionar,
 - 2.1- Si todos sus puntos son positivos, almacenarla en la lista de salida. En caso de poseer puntos positivos y negativos,
 - 2.2- Cortar la cara por el plano. Almacenar la porción o porciones que permanecen, en la lista de salida.
 - 2.3- Almacenar además, las nuevas aristas obtenidas en una lista de aristas en el plano de corte.
- 3- Analizar la lista de aristas en el plano de corte. Detectar bucles, y formar caras y polígonos agujero. Asignar agujeros a caras.
- 4- Formar el modelo de fronteras del objeto seccionado, a partir de las caras existentes en la lista de salida, y de las creadas en el paso 3.

4.4 Visualización con dibujo de líneas

Para visualizar cualquier objeto o conjunto de objetos mediante representación de alambres, el algoritmo es muy simple, ya que debe recorrerse únicamente parte de la información almacenada en el modelo de fronteras. Los pasos a realizar son los siguientes,

- 1- Transformación geométrica de todos los puntos del objeto u objetos a visualizar, [21] , [36]. Este proceso, como es sabido, se reduce al producto del vector de coordenadas por una matriz de 4 x 4, calculada al principio en función de los parámetros de visualización.
- 2- Recorrido por todos los polígonos de los objetos a representar, dibujando cada una de sus aristas con los puntos ya proyectados.

El usuario tiene la posibilidad de modificar los siguientes parámetros de visualización,

- Angulos horizontal y vertical de la dirección de observación (aparecen indicados por dos manecillas en punta en los indicadores inferiores de la parte derecha de la pantalla, ver por ejemplo las figuras 14-a y 14-b)
- Angulo de rotación de la cámara alrededor de su propio eje de visualización (indicador superior en la parte derecha de la pantalla; su valor es cero en todas las figuras).
- Angulos que determinan la dirección del foco de iluminación (válidos solamente para las representaciones que se exponen en los apartados 4.5 y 4.6). Se representan, en los indicadores horizontal y vertical, por dos manecillas rectangulares.
- Zoom: varía la ampliación de la imagen en la pantalla.
- Pan: permite modificar la posición del centro de la imagen
- Separación: es la distancia del usuario al objeto. Si es muy elevada, se obtiene una vista en axonométrico.
- Color: afecta solamente a las visualizaciones expuestas en los apartados 4.5 y 4.6.

Una vez modificados los parámetros, puede pedirse el redibujado. Los parámetros se mantienen de una visualización a la siguiente: podemos ajustar la perspectiva observando el modelo de alambres,

y luego pedir una visualización con eliminación de partes ocultas. La visualización incluye también la posición de los ejes en el espacio, figuras 13-d, 14-a y 14-b.

La representación con eliminación de líneas ocultas, figuras 10-a, 12-b, requiere únicamente la especificación de estos mismos parámetros. Sin embargo, el algoritmo es considerablemente más complejo: una vez finalizada la fase 1 de transformación geométrica de todos los puntos, se analiza cada arista del objeto con respecto a todas las caras que tiene delante, y si es preciso se recorta dejando únicamente las partes visibles. Al final, se dibuja, si existe, la parte de arista que no es tapada por ninguna cara, [65], [66] .

4.5 Visualización con imagen de facetas planas

En este caso, se obtiene una representación del objeto con simulación de iluminación desde la dirección especificada en los parámetros generales. A cada una de las caras planas del objeto se asigna una tonalidad, dentro del color también especificado por el usuario, que es proporcional al coseno del ángulo formado entre la normal a la cara y la dirección de iluminación.

El algoritmo utilizado, referencia [64], genera en una primera fase un árbol binario de prioridades de las caras del objeto, en espacio objeto (ref. [65]). Este árbol contiene información que puede ser utilizada en visualizaciones desde cualquier punto de vista, en axonométrico. En la segunda fase, se recorre el árbol en un orden que depende de la dirección de visualización, [64], y se obtienen directamente las caras orientadas hacia el observador, ordenadas desde la más lejana a la más cercana. En consecuencia, es un algoritmo especialmente adecuado para la obtención de distintas imágenes variando el punto de vista: el tiempo en la fase 2 se dedica prácticamente al redibujado de las caras en pantalla.

Las figuras 12-c, 14-c y 16 presentan visualizaciones obtenidas con este algoritmo.

4.6 Visualización mediante Z-buffer y suavización de Gouraud

Este tipo de visualización, indicada cuando se precisa una mayor calidad en la imagen, es en cambio más lenta que la expuesta en el apartado anterior. En ella se suaviza la intensidad de color entre caras adyacentes que provienen de la aproximación de una superficie cilíndrica, con lo que se obtiene una representación suave de la misma. El tiempo de proceso es del orden de un minuto en un ordenador VAX 11/750, mientras que el cálculo y visualización del árbol de prioridades correspondiente al apartado 4.5 puede consumir entre 1 y 5 segundos de CPU. Se presentan algunos ejemplos en las figuras 10-b, 10-c, 14-d, 15 y 17-c.

Al igual que en los restantes algoritmos de visualización, el usuario debe especificar únicamente los parámetros generales, que se conservan entre sucesivas visualizaciones. El algoritmo produce la imagen línea a línea, utilizando el algoritmo z-buffer, ref. [65]. Para cada línea, de 640 pixels, se realiza el siguiente proceso:

- Puesta al día de la lista de caras activas en la línea. Una cara es activa si su intersección con esta línea de barrido es no nula. Este proceso implica dar de baja a las caras que han finalizado entre la línea anterior y la presente, e incorporar aquellas que aparecen por primera vez.
- Inicializar el vector de color : $color(i)=color_del_fondo$
 $i=1...640$
- Inicializar un vector de profundidad en cada pixel de la línea:
 $prof(i)=profundidad\ del\ plano\ posterior\ de\ la\ escena$
 $i=1...640$
- Para cada cara activa en la línea de barrido,
 - Calcular sus intersecciones con la línea, junto con la profundidad y color en los extremos de estos tramos. En el método de Gouraud, el color en las caras a suavizar se interpola linealmente a partir del color en los vértices, que es promedio del que corresponde a las caras confluyentes.
 - Para cada pixel interior a los tramos de intersección de la cara con la línea, calcular (de forma incremental, por interpolación lineal) la profundidad y el color. Si la profundidad de la cara $< prof(i)$ entonces $color(i)=$ color de la cara; $prof(i)=$ profundidad de la cara.

- Finalmente, debe volcarse la línea de barrido sobre la pantalla raster. Dado que se utiliza una pantalla de 16 colores, se realiza antes un proceso de dithering con matriz de 4 x 4, referencia [65], que simula un mayor número de intensidades de color.

4.7 Salidas sobre papel. Interfase con un sistema de dibujo 2-D

En los algoritmos de visualización por líneas (representación con alambres o eliminación de líneas ocultas, apartado 4.4), es posible escribir en un archivo en disco el conjunto de aristas visualizadas. Esto permite la obtención de dibujos en plotter, así como su edición posterior en el sistema DOSDE, construido en el Centro de Cálculo de la UPC. En este último caso, el sistema DOSDE recupera las dimensiones reales del objeto, si la proyección ha sido realizada en planta, alzado o perfil.

4.8 Cálculo de propiedades volumétricas

Dado cualquier objeto o conjunto de objetos, el sistema calcula automáticamente las siguientes propiedades volumétricas:

- volumen (masa con densidad 1)
- superficie total de las caras
- perímetro total de las aristas
- coordenadas del centro de gravedad
- momentos de inercia respecto ejes paralelos a los coordenados pasando por el centro de gravedad.
- productos de inercia, para completar el tensor de inercia.

Para realizar los correspondientes cálculos, se utiliza el teorema de Stokes [20], que permite calcular propiedades de volumen en función de integrales extendidas al conjunto de caras del objeto. En concreto, y suponiendo densidad unitaria,

$$M = \frac{1}{3} \iint [x n_x + y n_y + z n_z] \, dS$$

$$G_x = \frac{1}{2M} \iint x^2 n_x \, dS$$

$$G_y = \frac{1}{2M} \iint y^2 n_y \, dS \qquad G_z = \frac{1}{2M} \iint z^2 n_z \, dS$$

$$I_{xx} = \frac{1}{3} \iint [y^3 n_y + z^3 n_z] dS$$

$$I_{yy} = \frac{1}{3} \iint [x^3 n_x + z^3 n_z] dS$$

$$I_{zz} = \frac{1}{3} \iint [x^3 n_x + y^3 n_y] dS$$

$$P_{xy} = -\frac{1}{2} \iint yx^2 n_x dS, \quad P_{yz} = -\frac{1}{2} \iint zy^2 n_y dS, \quad P_{zx} = -\frac{1}{2} \iint xz^2 n_z dS$$

siendo M la masa, G_x, G_y, G_z las coordenadas del centro de gravedad, I_{xx}, I_{yy}, I_{zz} los momentos de inercia, y P_{xy}, P_{yz}, P_{zx} los productos de inercia. La superficie total, S, se calcula directamente como integral de dS, y el perímetro sumando las longitudes de las aristas almacenadas en el modelo de fronteras.

Las anteriores integrales se extienden a toda la superficie del objeto. Si el número de caras del objeto es n_c , y el vector normal unitario a la cara i tiene componentes n_{xi}, n_{yi}, n_{zi} , puede escribirse, por ejemplo,

$$M = \frac{1}{3} \sum_{i=1}^{n_c} \left[n_{xi} \iint_{C_i} x dS + n_{yi} \iint_{C_i} y dS + n_{zi} \iint_{C_i} z dS \right]$$

donde las integrales se extienden a la superficie de la cara i. Para el cálculo de las demás propiedades volumétricas, se obtendrían expresiones parecidas.

Finalmente, el cálculo de integrales de funciones polinómicas de grado menor que 4, puede realizarse sin error mediante la siguiente regla de cuadratura, [20],

$$\iint_{C_i} f dS = \sum_{j=3}^{n_a} \text{area}(ABC) \cdot \left(-\frac{27}{48} f(G_{ABC}) + \frac{25}{48} (f(P_1) + f(P_2) + f(P_3)) \right)$$

donde se calcula la integral extendida a una cara como suma de las contribuciones de sus $n_a - 2$ triángulos (formados por los vértices 1,2,3, 1,3,4, ..., 1,n-1,n). Para un determinado valor de j, ABC indica el correspondiente triángulo, G_{ABC} su centroide, y P_1, P_2, P_3 tres puntos situados a un 40% de distancia del centroide y 60% del vértice correspondiente.

4.9 Realización de Operaciones booleanas, en el Modelo de Fronteras

Para la realización de operaciones booleanas entre objetos representados en el modelo de fronteras, se supone que éstos han sido ya escalados y trasladados adecuadamente a la posición correcta. El sistema pide únicamente el nombre de los dos objetos a operar, y la operación deseada,

- 1- Unión
- 2- Intersección
- 3- Diferencia A - B
- 4- Diferencia B - A

El resultado de la operación puede almacenarse como un objeto más, en la Base de Datos, al igual, por ejemplo, que en la operación de sección.

El algoritmo correspondiente se presenta con detalle en [60]. Realiza secuencialmente las fases siguientes,

- Cálculo de todas las aristas de intersección de las caras del cuerpo A con las de B y viceversa.
- Unión de estas aristas entre sí. Formación de nuevas caras por corte de las antiguas.
- Clasificación de la lista de nuevas caras, según la operación a realizar. Almacenamiento de las caras que formaran parte del cuerpo final, en un modelo de fronteras:

Operación	Caras a tener en cuenta
$A \cup B$	caras de A externas a B y caras de B externas a A
$A \cap B$	caras de A internas a B y caras de B internas a A
$A - B$	caras de A externas a B y caras de B internas a A
$B - A$	caras de B externas a A y caras de A internas a B.

4.10 Conversión entre modelos: De modelo de fronteras a octtrees y viceversa.

En el sistema DMI se han programado algoritmos para la conversión entre modelos, con el fin de poder realizar operaciones booleanas de forma más eficiente en el modelo de octtrees extendidos. Estas conversiones se realizan automáticamente, sin necesidad de especificar ningún parámetro. Se siguen los algoritmos expuestos en [13] y [14], y con más detalle en [62]. Se dispone asimismo de un algoritmo para la visualización directa del modelo de octtrees extendidos (Figura 5).

El algoritmo de conversión directa, de modelo de fronteras a octtrees, realiza una subdivisión recursiva de un cubo que inicialmente engloba al objeto, manteniendo una lista de caras parcialmente dentro del cubo y otra de vértices del cuerpo interiores. Cada vez que se analiza un nuevo cubo resultado de la subdivisión, deben ponerse al día ambas listas mediante un proceso parcial de recortado de las caras de la lista que proviene del nodo padre. La subdivisión termina cuando se llega a uno de los siguientes nodos terminales,

- . Blanco : nodo sin ninguna cara, exterior al objeto
- Negro : nodo sin ninguna cara, interior al objeto.
- Cara : nodo que contiene parcialmente una sola cara del objeto
- Arista : nodo que contiene dos caras, junto con una porción de su arista de contacto.
- Vértice: nodo que contiene un solo vértice del objeto, y tan solo aquellas caras que convergen en él.

o bien cuando se llega a la máxima subdivisión permitida. En este caso, si la configuración interior al nodo es admisible, [14], el nodo se codifica como un Gris Terminal.

En el algoritmo de conversión inversa, de octtrees a fronteras, se realiza un primer recorrido por el árbol, [14], [62], analizando los nodos vértice, y creando a la vez una lista para cada cara del objeto donde se almacenan los vértices en que aparece la cara junto con sus caras adyacentes en la ordenación cíclica alrededor del vértice. En una segunda fase, se analiza cada una de estas listas, concatenando los vértices para formar bucles y detectar la frontera de la cara junto con sus posibles agujeros.

4.11 Realización de operaciones booleanas en el modelo de octtrees

Las operaciones booleanas sobre el modelo de octtrees, suponen los objetos a operar ya trasladados antes de la conversión al modelo secundario de octtrees extendidos. Por ello, es preciso únicamente especificar los objetos a operar y el tipo de operación a realizar. Las operaciones básicas que pueden realizarse con el modelo de octtrees son,

- la complementación (o generación de un "molde" del objeto, delimitado por las caras del nodo cúbico raíz del árbol)
- la intersección entre dos objetos codificados mediante árboles octales.
- la sección plana de un objeto (no es más que su intersección con un octtree de un solo nodo de tipo cara)
- la reunión y diferencia, de algoritmo muy parecido a la intersección. Pueden realizarse, sin estar implementadas, con ayuda de la intersección y complementación.

El algoritmo de complementación es lineal: consiste simplemente en realizar un recorrido del árbol inicial, [14], permutando nodos blancos por negros, cambiando el sentido de las ecuaciones de los planos de las caras, y modificando la configuración de los nodos arista y vértice, [62] (la configuración permite conocer la parte de un nodo terminal que es interior al objeto modelado).

El algoritmo de intersección (así como los de las operaciones de reunión y diferencia), tienen complejidad lineal respecto la suma de longitudes de los árboles dato y el árbol resultado, [62]. Realizan un recorrido simultáneo en preorden de los dos árboles, intersectando en cada momento parejas de nodos en la misma posición espacial. La única parte compleja del algoritmo es la que se refiere a la intersección entre nodos terminales del tipo cara, arista o vértice: es preciso disponer de un algoritmo rápido, [62], que permita decidir si el resultado es asimismo terminal (blanco, cara, arista o vértice) o bien no lo es, y se debe subdividir el nodo hasta llegar a configuraciones permitidas.

5. GENERACION DE SUPERFICIES ESCULPIDAS

El proceso habitual de generación de superficies complejas en el espacio, se inicia por la generación de curvas, generalmente planas, que llamaremos perfiles, y que determinaran la forma final de la superficie. Por ello, antes de entrar a estudiar en los apartdos siguientes los métodos de diseño de superficies utilizados en el Sistema DMI, presentaremos a continuación los métodos de diseño de curvas más generalizados.

La generación de curvas de forma predeterminada se realiza en las siguientes etapas,

- 1- El usuario introduce una serie de puntos que controlaran la forma de la curva. La entrada puede realizarse mediante movimientos del cursor en la propia pantalla, por coordenadas o a través de mesa o tableta digitalizadora.
- 2- Suponiendo, por ejemplo, que los puntos se hayan entrado en en plano x-y, el sistema realiza, en una primera fase, un proceso de parametrización que asegura el carácter univaluado de las curvas resultantes. Este proceso consiste en la introducción de una nueva coordenada (el parámetro, que en lo sucesivo llamaremos u), que crece monótonamente a lo largo de la poligonal de puntos entrados, desde el primer punto al último, [38]. A partir de este momento, el sistema trabajará con dos (o tres, en el caso de curvas alabeadas) curvas univaluadas, x-u, y-u , definidas cada una de ellas por n puntos: (x_k, u_k) , $k=1..n$, (y_k, u_k) , $k=1..n$. Por lo que respecta al tipo de parametrización, existe una gran diversidad de opciones. Los métodos de parametrización uniforme asignan valores equidistanciados a los puntos entrados: por ejemplo, $u=1$ en $P_1=(x_1, y_1)$, $u=2$ en P_2 , .. $u=n$ en $P_n=(x_n, y_n)$. No obstante, en algunas aplicaciones es mejor utilizar parametrizaciones más relacionadas con el espacio recorrido a lo largo de la curva: valor del parámetro igual al espacio recorrido sobre la poligonal de puntos entrados desde P_1 , o bien igual al espacio recorrido sobre una primera curva calculada con una parametrización más primitiva. En lo que sigue, se utilizará únicamente la parametrización uniforme $u_k=u(P_k)=k$, suficientemente aceptable si no existen grandes variaciones en la separación entre puntos consecutivos.

3- El ordenador calcula la ecuación analítica de la curva o conjunto de curvas que se adapta a los puntos introducidos, en el plano paramétrico. Algunos de los métodos que veremos a continuación generan curvas que pasan por (interpolan) los puntos entrados, mientras que en otros, los puntos dato solo sirven para controlar la forma de la curva final. El sistema almacena la ecuación analítica que modela la forma de la curva, que habitualmente es de forma polinómica. En general también, y como veremos a continuación, la expresión analítica de la curva interpolada en un punto de parámetro u determinado, puede escribirse como una combinación lineal de los puntos entrados por el usuario:

$$x(u) = \sum_{i=1}^n x_i \cdot F_i(u)$$

$$y(u) = \sum_{i=1}^n y_i \cdot F_i(u)$$

(Si la curva fuese alabeada, $z(u)$ se expresaría de igual forma en función de las coordenadas $z_1 \dots z_n$). En forma vectorial, la expresión anterior puede escribirse,

$$P(u) = \sum_{i=1}^n P_i \cdot F_i(u)$$

donde $P(u) = (x(u), y(u))^T$, y P_i son los puntos entrados en el plano $x-y$. Las funciones $F_i(u)$, $i=1 \dots n$, son las llamadas funciones de forma, y constituyen una base del espacio vectorial de funciones de interpolación admisibles. Como veremos a continuación, dependen del método de diseño de curvas utilizado, y le transmiten sus propiedades geométricas.

4- A partir de las ecuaciones analíticas calculadas en el apartado anterior, se calcula un número suficientemente elevado de puntos de la curva, $P(u_i)$, $i=1 \dots np$ con $np \gg n$, y se dibuja en pantalla. Realmente, lo que aparece en pantalla es la poligonal que une estos puntos; si np es suficiente, se tendrá una buena aproximación de la curva diseñada.

5- Finalmente, los métodos de diseño de curvas permiten modificar la posición de alguno de los puntos iniciales (volviendo al paso 1) si la forma de la curva no es la deseada. En

este proceso de modificación de alguno de los puntos iniciales, cabe distinguir entre los métodos globales, en que la variación en la posición de un solo punto afecta a la forma de toda la curva, y los métodos locales, en que solo modifica la zona de curva cercana al punto trasladado.

Veamos en primer lugar una tabla comparativa de los métodos más utilizados en el diseño de curvas:

Método	Grado	Interpola los puntos?	Local o Global	$F_i(u)$	Continuidad
Int. lineal	1	si	L_2	$H_i(u)$	C^0
Int. Lagrange	n-1	si	G	$L_i(u)$	C^∞
Bézier	n-1	no	G	$J_i(u)$	C^∞
Splines cubicos	Locales	3	L_4	$CL_i(u)$	C^1
	Globales	3	G	$C_i(u)$	C^2
	B-Splines	3	no	L_4	$B_i(u)$ ajustable

Observaciones:

Todos los métodos utilizan aproximaciones polinómicas de las curvas paramétricas x-u, y-u. Se diferencian en el grado, que como vemos oscila entre 1 en la interpolación lineal y n-1 en Lagrange y las curvas de Bézier. Por lo que respecta a los B-splines, pueden ser definidos con cualquier grado, [37] , [38] , pero nosotros nos concretaremos a los cúbicos.

Los distintos métodos se comportan de forma muy variada por lo que respecta a si pasan o no por los puntos dados, el carácter local o global, y la continuidad. G significa comportamiento global, mientras que L_k indica un comportamiento local tal que la modificación de la posición de un punto afecta a un total de k intervalos de la curva. Una continuidad C^k indica continuidad de la función y k primeras derivadas en x-u, y-u entre u_1 y u_n .

A continuación, vamos a estudiar con detalle los distintos métodos de interpolación expuestos; todos ellos se utilizan exclusivamente en la fase 3 del proceso indicado anteriormente, y construyen curvas a partir de puntos en el plano paramétrico. Por ello, sólo se comentará el diseño de una de las curvas (la $x-u$), junto con las características geométricas de la curva $P(u)$ resultante.

Interpolación lineal

En este caso la "curva" $x-u$ se construye uniendo con segmentos rectos los puntos consecutivos en el plano paramétrico. En consecuencia, en el plano $x-y$ se construye la poligonal que une los puntos entrados, $P_1 \dots P_n$. La curva pasa por los puntos, tiene caracter local L_2 y continuidad C^0 , de la curva pero no de sus derivadas. Como es lógico, este sistema no se usa como método de diseño de curvas; se ha expuesto únicamente como elemento de comparación con los restantes.

Si definimos las funciones $H_i(u)$ como [37] ,

$$H_i(u) = \text{poligonal que une los puntos} \\ (h_1, u_1), (h_2, u_2), \dots, (h_n, u_n)$$

$$\text{con } u_j = j, h_i = 1, h_j = 0 \text{ si } j \neq i ,$$

es inmediato ver que la poligonal resultado de la interpolación lineal es la combinación lineal de los puntos entrados con las funciones de forma H_i , tal como se ha indicado en la fase 3 del proceso general de diseño. Las funciones H_i , "hat functions", son por tanto base del espacio de funciones interpolación lineal. Los métodos que presentamos a continuación intentan aumentar la suavidad de la curva resultante, utilizando polinomios de grado superior al primero.

Interpolación de Lagrange

En este caso, se utiliza el esquema ya descrito de ponderación de los puntos entrados mediante funciones de forma $L_i(u)$, cuyo valor depende del valor concreto del parámetro u . Las funciones de Lagrange L_i se definen de la siguiente manera,

$$G(u) = (u-u_1)(u-u_2)\dots(u-u_n)$$
$$G_i(u) = G(u) / (u-u_i) = \prod_{\substack{j=1 \\ j \neq i}}^n (u-u_j)$$
$$L_i(u) = G_i(u) / G_i(u_i)$$

Dado que la función $G(u)$ es un polinomio de grado n cuyas raíces son $u_1 \dots u_n$, los n polinomios $G_i(u)$ son de grado $n-1$; cada uno de ellos se anula en $n-1$ de las abscisas u_k correspondientes a los puntos iniciales. $L_i(u)$ se obtiene dividiendo $G_i(u)$ por el valor constante $G_i(u_i)$. Por ello los polinomios de Lagrange L_i son polinomios de grado $n-1$ que cumplen la siguiente propiedad,

$$L_i(u_i) = G_i(u_i) / G_i(u_i) = 1$$
$$L_i(u_j) = 0 \text{ si } i \neq j, \text{ ya que } G_i(u_j) = 0$$

O, abreviadamente,

$$L_i(u_j) = \delta_{ij}$$

donde δ_{ij} es la delta de Kronecker. Esta propiedad conduce de forma inmediata a la conclusión de que el polinomio de interpolación de Lagrange,

$$P(u) = \sum_{i=1}^n P_i \cdot L_i(u)$$

es de grado $n-1$ e interpola los puntos dados. El comportamiento local y la continuidad infinita son consecuencia inmediata de tener un único polinomio a lo largo de toda la curva.

El polinomio $P(u)$ es el polinomio de interpolación de los n puntos dados, que como se sabe es único si $u_i \neq u_k$ para todos i, k ; esta última propiedad se cumple siempre, debido a las hipótesis del proceso de parametrización.

Sin embargo, y a pesar de estas propiedades, la interpolación de Lagrange no es utilizada en el diseño de curvas y superficies, ya

que no es posible controlar las oscilaciones de la curva entre los puntos por los que pasa, sobretodo en el caso de n elevada.

Curvas de Bézier

Las curvas de Bézier tienen muchos puntos de contacto con los polinomios de interpolación de Lagrange: están constituidas por un único tramo polinómico, de grado $n-1$, y por tanto tienen carácter global y continuidad infinita. Simplemente se efectuado un cambio de base en el espacio de funciones polinómicas de grado $n-1$. Las funciones de la nueva base se definen como,

$$J_i(u) = \binom{n-1}{i-1} u^{i-1} \cdot (1-u)^{n-i}$$

(Se trata de la base de los polinomios de Bernstein). Habitualmente, se considera una parametrización entre cero y uno: $u_1=0$, $u_n=1$. Dado que la curva no va a pasar por los puntos intermedios, no es necesario asignarles un valor concreto del parámetro. A partir de la definición de las funciones J_i , es fácil deducir las siguientes propiedades, [35], [36], [38],

- Para cualquier valor de u , $0 \leq u \leq 1$, se cumple que

$$\sum_{i=1}^n J_i(u) = 1$$

y además, $J_i(u) \geq 0$ para todo i , u . Por ello, el cálculo de puntos de la curva,

$$P(u) = \sum_{i=1}^n P_i \cdot J_i(u)$$

puede considerarse como una ponderación de los puntos entrados, con coeficientes J_i . En consecuencia, la curva es interior a la frontera convexa definida por los puntos $P_1 \dots P_n$.

- Dado que $J_1(0) = J_n(1) = 1$, la curva pasa por el punto P_1 cuando $u=0$, y por P_n cuando $u=1$. En cambio, no pasa por los puntos intermedios.
- La dirección de salida y llegada, viene determinada por los puntos P_2 y P_{n-1} : La curva sale de P_1 en dirección a P_2 , y llega a P_n en dirección $P_{n-1} P_n$. En consecuencia, es muy sen-

- cillo gobernar la forma de la curva en sus zonas inicial y final, modificando adecuadamente la posición de cuatro puntos.
- La forma geométrica de la curva se parece a la de la poligonal $P_1P_2\cdots P_n$, aunque puede demostrarse que es más suave. Esta propiedad da la posibilidad de mejora interactiva de la forma de la curva, actuando simplemente sobre los vértices de control adecuados.
 - Es posible conectar sucesivas curvas de Bézier, con continuidad de la pendiente geométrica: sólo es preciso alinear los dos últimos puntos de la curva anterior, con el segundo de la siguiente.

En definitiva, las curvas de Bézier proporcionan una herramienta para el diseño de curvas polinómicas con un elevado control sobre la forma geométrica; en relación al método de interpolación de Lagrange, solo se ha perdido la posibilidad de pasar por los puntos.

Los splines cúbicos locales y globales

Una alternativa respecto a las curvas de Bézier, si se desea no aumentar el grado del polinomio a medida que crece el número de puntos, son los splines cúbicos. En el diseño de curvas con splines, la ecuación analítica de la curva final no es única: está constituida por un conjunto de segmentos polinómicos, que conectan entre sí de manera suficientemente suave, según las especificaciones. En realidad, la interpolación lineal puede ser considerada como un spline: la poligonal final está formada por tramos de polinomios de grado 1 - rectas- conectando entre sí con continuidad de la función, C^0 . Asimismo, la formación de una curva compuesta con tramos de curvas de Bézier de forma que la pendiente geométrica sea continua, es también un spline.

En lo que sigue, trataremos únicamente el caso de los splines cúbicos: curvas formadas por tramos de polinomios cúbicos que conectan adecuadamente entre sí.

Tanto los splines cúbicos locales como los globales, permiten la generación de curvas abiertas o cerradas (estas últimas no se podían construir de forma simétrica con Lagrange o Bézier). Los primeros tienen la ventaja del carácter local, pero en cambio

la continuidad sólo puede ser de las primeras derivadas, C^1 . En los segundos se tiene mayor continuidad a costa del carácter global que hace más difícil la modificación posterior de partes de la curva .

Veamos un primer algoritmo de construcción del spline cúbico (local o global) que interpola n puntos $(u_1, x_1) \dots (u_n, x_n)$, en que no aparecen explícitamente las funciones de la base, [39] ,

- 1- A partir de los n puntos (u_k, x_k) , se estiman las n derivadas \dot{x}_k , de la curva interpolada final. ($\dot{x}_k = (dx/du)_{u_k}$).
- 2- Una vez se dispone de los valores (u_k, x_k, \dot{x}_k) , $k=1 \dots n$, la ecuación del spline cúbico en el intervalo $u_k \leq u \leq u_{k+1}$ es, [35],

$$x(u) = (1 \ t \ t^2 \ t^3) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k+1} \\ \dot{x}_k \\ \dot{x}_{k+1} \end{bmatrix}$$

donde $t = u - u_k$, y se supone parametrización uniforme, $u_{k+1} - u_k = 1$ para todo k .

Para la estimación de las n derivadas \dot{x}_k (paso 1), en los splines cúbicos locales existen diversos métodos, [47]; en todos ellos, se utiliza únicamente la información de los puntos más cercanos. Por ejemplo, en el diseño de curvas abiertas puede utilizarse el método de Bessel [43], [47], en que, en los puntos intermedios, se estima la pendiente como la de la recta que une los puntos vecinos,

$$\dot{x}_k = (x_{k+1} - x_{k-1}) / 2 \quad , \quad k=2 \dots n-1$$

$$\dot{x}_1 = -1.5 x_1 + 2x_2 - .5 x_3$$

$$\dot{x}_n = 1.5 x_n - 2x_{n-1} + .5 x_{n-2}$$

En el diseño de curvas cerradas, no es preciso disponer de ecuaciones específicas para las pendientes de los extremos, ya que, debido al comportamiento cíclico,

$$\dot{x}_k = (x_{k+1} - x_{k-1}) / 2 \quad , \quad \text{para todo } k. \text{ Se supone } x_{n+1} = x_1 \text{ y } x_0 = x_n.$$

Tanto en el caso de diseño de curvas cerradas como en el de curvas abiertas, y sea cual sea el algoritmo utilizado para el cálculo de pendientes en el paso 1, el spline obtenido tiene continuidad C^1 , de las primeras derivadas, debido a que en los puntos de conexión éstas son función de punto y no del tramo. Por otra parte, el carácter local de esta interpolación queda asegurado por el método de cálculo del vector de derivadas, asimismo local. Como ecuación del spline, el sistema puede almacenar directamente los vectores (x_k, \dot{x}_k) .

En el caso de los splines cúbicos globales, se aprovechan los n grados de libertad que existen en el paso 1, para imponer continuidad de la segunda derivada en los $n-2$ puntos intermedios. De esta forma, se calcula el vector \dot{x}_k , $k=1..n$ que produce continuidad C^2 , y todavía quedan 2 grados de libertad en el cálculo del vector de pendientes. Existen diversas opciones para escoger el valor de estos dos parámetros, llamados condiciones de extremo del spline, [39]: Por ejemplo, se puede demostrar que si se impone curvatura nula en los dos extremos u_1 , u_n , se consigue el spline natural, que minimiza la integral del cuadrado de la derivada segunda, [37]. Uno de los inconvenientes de estos splines, asociado a su carácter global, es que es preciso resolver un sistema de n ecuaciones lineales para obtener, en el paso 1, el vector de pendientes. Este proceso es por tanto más lento, aunque la matriz del sistema es tridiagonal, [39].

Veamos el sistema a resolver para el cálculo de pendientes en el caso de curvas abiertas (condiciones de extremo de curvatura nula):

$$\begin{bmatrix} 4 & 0 & 0 & \dots & \dots & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & 1 & 4 & 1 \\ 0 & \dots & \dots & \dots & 0 & 4 & \dots \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{n-1} \\ s_n \end{bmatrix} = 6 \begin{bmatrix} 0 \\ x_3 - 2x_2 + x_1 \\ x_4 - 2x_3 + x_2 \\ \vdots \\ x_n - 2x_{n-1} + x_{n-2} \\ 0 \end{bmatrix}$$

Y luego, $\dot{x}_1 = x_2 - x_1 - \frac{1}{6} s_2$

$$\dot{x}_k = x_{k+1} - x_k + \frac{1}{3} s_{k+1} + \frac{1}{6} s_k, \quad k=2..n$$

En el diseño de curvas cerradas, debe imponerse continuidad C^2 en los n puntos, con lo que se agotan todos los grados de libertad. El sistema a resolver para el cálculo de pendientes es por tanto único:

$$\begin{bmatrix}
 4 & 1 & 0 & \dots & \dots & \dots & 0 \\
 1 & 4 & 1 & 0 & \dots & \dots & 0 \\
 0 & 1 & 4 & 1 & \dots & \dots & 0 \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & \dots & \dots & 0 & 1 & 4 & 1 \\
 1 & 0 & \dots & \dots & 0 & 1 & 4
 \end{bmatrix}
 \begin{bmatrix}
 x_1 \\
 x_2 \\
 x_3 \\
 \vdots \\
 x_n
 \end{bmatrix}
 = 3
 \begin{bmatrix}
 x_2 - x_n \\
 x_3 - x_1 \\
 x_4 - x_2 \\
 \vdots \\
 x_1 - x_{n-1}
 \end{bmatrix}$$

Veamos algunas características de los splines cúbicos, locales y globales:

- En la interpolación, de n puntos, están formados por $n-1$ tramos polinómicos cúbicos, tanto en el plano $x-u$ como en el $y-u$.
- Los splines cúbicos locales aseguran continuidad de la pendiente, y poseen carácter local L_4 . Una ventaja respecto a las curvas de Bézier es que no es preciso esperar a que el usuario haya entrado todos los puntos para dibujar la curva: esta puede dibujarse de forma automática a medida que se van introduciendo nuevos puntos y calculando pendientes.
- Los splines cúbicos globales aseguran máxima suavidad (en el caso de los naturales), poseen continuidad C^2 y comportamiento global. Al igual que las curvas de Bézier, debe esperarse a haber entrado todos los puntos para iniciar el cálculo de pendientes.
- Los dos permiten el diseño de curvas cerradas suaves.

Independientemente de la existencia del algoritmo de cálculo de los splines cúbicos que acabamos de presentar, es posible (y en muchos casos, como veremos, conveniente) calcular la base de funciones que los generan para luego, en una segunda fase, realizar su combinación lineal según las coordenadas de los puntos entrados.

En este caso, y al igual que en la interpolación lineal, Lagrange y Bézier, la base obtenida será válida para la interpolación de cualquier función en la malla $u_1 \dots u_n$; en particular, servirá para interpolar la curva $x-u$, la curva $y-u$, y las curvas que pueden obtenerse por modificación de cualquiera de los puntos P_k . El proceso de cálculo de los elementos de la base es muy simple: el spline cardinal $C_i(u)$, por ejemplo, es el spline que interpola, al igual que en Lagrange, los puntos,

$$(0, u_1), (0, u_2) \dots, (0, u_{i-1}), (1, u_i), (0, u_{i+1}), \dots, (0, u_n)$$

ya que de esta forma se asegura la propiedad $C_i(u_j) = \delta_{ij}$, que asegura la interpolación de los puntos. En consecuencia, tanto si se desea calcular la base de splines cardinales $C_i(u)$ como la de los $CL_i(u)$, es preciso calcular el vector de pendientes, utilizando el paso 1 del algoritmo anterior, para n conjuntos distintos de puntos a interpolar; en cada uno de ellos, una de las alturas vale 1 mientras que las restantes valen 0. En el caso de los splines cúbicos globales, este proceso equivale a la inversión de la matriz que relaciona pendientes con valores dados. En cualquier caso, las funciones de la base son splines, que generan el espacio de splines cúbicos sobre la malla dada, por combinación lineal.

En el párrafo siguiente veremos que los splines cúbicos globales (locales) pueden también obtenerse como caso particular de los B-splines.

Los B-splines

En un principio, el método de construcción de curvas mediante B-splines, parece muy similar al basado en las curvas de Bezier: a partir de un conjunto de puntos $P_1 \dots P_n$ (conocidos habitualmente como vértices de control) que suministra el usuario, se genera una curva por combinación lineal de los P_k mediante un conjunto de funciones base o funciones de forma, $B_k(u)$.

La expresión matemática de las funciones base, [37], [40], es sin embargo distinta a la de las $J_k(u)$ que aparecían en las curvas de Bezier. Ahora, las $B_k(u)$ son funciones polinómicas a trozos, y por ello la curva final $(x(u), y(u))$ será también un conjunto de polinomios unidos convenientemente. En el caso particular de las $B_k(u)$

estén formadas por trozos de polinomios cúbicos, se obtienen los B-splines cúbicos, que son los más utilizados. En este caso:

- 1) Se puede demostrar que, con una elección adecuada de los puntos de control $P_1 \dots P_n$, se pueden generar las mismas curvas que obteníamos con los splines cúbicos globales. En este sentido, estamos ante una generalización de aquellos.
- 2) Asimismo, es posible generar las curvas de Bezier, con $n=4$; y en general, se puede demostrar que las curvas de Bezier con n puntos pueden generarse como un caso particular de los B-splines de grado $n-1$.
- 3) Se mantienen las propiedades de combinación convexa de los puntos entrados y de variación menor que la poligonal, que teníamos en las curvas de Bézier.
- 4) Las funciones base, $B_i(u)$ tienen comportamiento local (son no nulas en un máximo de 4 intervalos). Por tanto, la curva final heredará este comportamiento, a diferencia de lo que ocurría en Bézier.
- 5) Es posible generar una base $B_i(u)$ que cumpla cualquier especificación en cuanto a condiciones de continuidad en los puntos de conexión entre cúbicas; por tanto, los B-splines son una herramienta para el diseño totalmente flexible de curvas: podemos especificar continuidad elevada en algunos puntos, y discontinuidad de la pendiente, por ejemplo, en otros.
- 6) Existen algoritmos, [48], para el cálculo de los nuevos vértices de control que dan el mismo spline cuando se introduce una parametrización más fina. Estos algoritmos pueden ser interesantes cuando es necesaria una subdivisión de los trozos cúbicos, o más adelante, de la superficie.

Sin embargo, la información que debe introducir el usuario para el cálculo de los elementos de la base no es tan simple como en otros métodos, [37]. Por ello, en [41], Wu y otros proponen la utilización de un subespacio de los B-splines, en que todos los elementos de la base son idénticos. En este caso, se puede regular la continuidad geométrica de la curva final, aunque las curvas en el espacio paramétrico $x-u$, $y-u$ poseen siempre continuidad C^2 . En concreto, puede demostrarse que introducción de tres vértices de control P_k consecutivos en la misma posición en el plano $x-y$, produce una discontinuidad de la pendiente,

Tanto en el caso de diseño de curvas cerradas como en el de curvas abiertas, y sea cual sea el algoritmo utilizado para el cálculo de pendientes en el paso 1, el spline obtenido tiene continuidad C^1 , de las primeras derivadas, debido a que en los puntos de conexión éstas son función de punto y no del tramo. Por otra parte, el carácter local de esta interpolación queda asegurado por el método de cálculo del vector de derivadas, asimismo local. Como ecuación del spline, el sistema puede almacenar directamente los vectores (x_k, \dot{x}_k) .

En el caso de los splines cúbicos globales, se aprovechan los n grados de libertad que existen en el paso 1, para imponer continuidad de la segunda derivada en los $n-2$ puntos intermedios. De esta forma, se calcula el vector \dot{x}_k , $k=1..n$ que produce continuidad C^2 , y todavía quedan 2 grados de libertad en el cálculo del vector de pendientes. Existen diversas opciones para escoger el valor de estos dos parámetros, llamados condiciones de extremo del spline, [39]. Por ejemplo, se puede demostrar que si se impone curvatura nula en los dos extremos u_1 , u_n , se consigue el spline natural, que minimiza la integral del cuadrado de la derivada segunda, [37]. Uno de los inconvenientes de estos splines, asociado a su carácter global, es que es preciso resolver un sistema de n ecuaciones lineales para obtener, en el paso 1, el vector de pendientes. Este proceso es por tanto más lento, aunque la matriz del sistema es tridiagonal, [39].

Veamos el sistema a resolver para el cálculo de pendientes en el caso de curvas abiertas (condiciones de extremo de curvatura nula):

$$\begin{bmatrix} 4 & 0 & 0 & \dots & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 4 & 1 \\ 0 & \dots & \dots & 0 & 4 & \dots \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{n-1} \\ s_n \end{bmatrix} = 6 \begin{bmatrix} 0 \\ x_3 - 2x_2 + x_1 \\ x_4 - 2x_3 + x_2 \\ \vdots \\ x_n - 2x_{n-1} + x_{n-2} \\ 0 \end{bmatrix}$$

Y luego,

$$\dot{x}_k = x_{k+1} - x_k - \frac{1}{6} s_{k+1} - \frac{1}{3} s_k, \quad k=1..n-1$$

$$\dot{x}_n = x_n - x_{n-1} - \frac{1}{6} s_{n-1}$$

Finalmente, y como ya se ha dicho, los splines cardinales pueden generarse como un caso particular de los B-splines. En concreto, para el cálculo del spline cúbico con curvatura nula en los extremos que interpola un conjunto de puntos $P_1 \dots P_n$, puede utilizarse un algoritmo alternativo consistente en realizar primero un cambio de base, calculando los vértices de control del B-spline que genera la misma curva final, y luego aplicar el algoritmo que acabamos de presentar para el cálculo de puntos del B-spline a partir de los vértices de control $V_1 \dots V_n$. Esta curva no pasará por los puntos V_k , pero sí por los P_k .

En el caso de curvas cerradas, debe resolverse el siguiente sistema lineal,

$$\frac{1}{6} \begin{bmatrix} 4 & 1 & \dots & \dots & \dots & 1 \\ 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & 1 & 4 & 1 \\ 1 & \dots & \dots & \dots & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ \vdots \\ P_n \end{bmatrix}$$

Y en el caso de curvas abiertas,

$$\frac{1}{6} \begin{bmatrix} 6 & 0 & \dots & \dots & \dots & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & 1 & 4 & 1 \\ 0 & \dots & \dots & \dots & 0 & 6 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ \vdots \\ P_n \end{bmatrix}$$

En el sistema DMI se dispone de algoritmos para el diseño de curvas mediante los métodos de Bézier, Splines cúbicos locales y globales, así como mediante los B-splines de base simplificada que acabamos de presentar. Sin embargo, estos algoritmos en la actualidad no se encuentran conectados directamente al sistema central.

Como veremos en los siguientes apartados, los perfiles curvados diseñados pueden ser utilizados para la generación de superficies por traslación, rotación, o unión de perfiles. De la misma manera, es posible generalizar los métodos de diseño de curvas para el diseño de superficies a partir de una malla rectangular de puntos en el espacio. En realidad, todo algoritmo de generación de curvas por ponderación de los puntos entrados,

$$P(u) = \sum_{i=1}^n P_i \cdot F_i(u)$$

da lugar al correspondiente algoritmo de generación de superficies de producto tensorial a partir de una malla de puntos P_{ij} , $i=1..n$, $j=1..m$, donde $P_{ij} = (x_{ij}, y_{ij}, z_{ij})$, [36], [38], [46],

$$S(u,v) = \sum_{i=1}^n \sum_{j=1}^m P_{ij} \cdot F_i(u) \cdot F_j(v)$$

Como vemos, el valor (vectorial) de la superficie en un punto determinado de parámetros (u,v) , es una combinación lineal de todos los puntos de la malla que hemos entrado, ponderados con coeficientes que no son más que productos de dos de las funciones base que se utilizaban en el caso unidimensional.

La superficie hereda en este caso las propiedades del esquema de diseño de curvas: así, por ejemplo, cuando $F_i = J_i$ se tienen las superficies de Bézier, que sólo pasan por los 4 puntos extremos, y cuya dirección de salida es controlada por los puntos adyacentes; si $F_i = C_i$, se tienen las superficies bicúbicas cardinales, que interpolan todos los puntos de la malla y poseen continuidad C^2 ; y si $F_i = B_i$, tenemos las superficies B-spline, con control local, aunque en general no pasan por los puntos de control.

En los apartados siguientes, 5.1 y 5.2, se presentan los algoritmos implementados para el diseño de superficies en el sistema DMI. Otras aplicaciones y algoritmos para el diseño de superficies esculpidas, pueden ser consultadas en las referencias [49] ... [58].

5.1. Generación interactiva de superficies bicúbicas basadas en B-splines.

Si suponemos un conjunto de $n \times m$ puntos en el espacio para la generación de la superficie, existen métodos que obtienen directamente la superficie a partir de la malla, mientras que otros suponen la malla estructurada, por ejemplo, en m perfiles de n puntos en el espacio. Pasamos a describir con más detalle las distintas posibilidades, en el caso que se desee que la superficie pase por los puntos entrados al sistema,

- 1) Una vez generadas, por interpolación con splines cúbicos, las curvas correspondientes a los m perfiles en sentido longitudinal y a los n perfiles en sentido transversal, puede "llenarse" cada uno de los trozos rectangulares de la malla con una superficie que se adapte a sus fronteras. El primer algoritmo para esta interpolación fue debido a Coons, [42]. Un estudio de las condiciones que se requieren para aumentar la suavidad y la continuidad entre trozos adyacentes ha conducido a los trozos de Adini, [43] y Gregory [44], entre otros.
- 2) Puede utilizarse un método de interpolación bidimensional basado en splines cúbicos globales o locales. En este caso (camino a y b de la figura 18) la superficie cumple las mismas propiedades que teníamos en el diseño de curvas: pasa por todos los $n \times m$ puntos de la malla; en el caso de splines cúbicos globales, la superficie es suave, pero su comportamiento no es local: la modificación de uno solo de los puntos de la malla afecta a la forma de toda la superficie. En el caso de utilización de splines cúbicos locales, se consigue un comportamiento local por lo que respecta a las modificaciones, a costa de una menor continuidad -y por tanto suavidad- de la superficie diseñada. Los aspectos matemáticos del proceso de cálculo, que se obtienen por generalización inmediata del caso unidimensional, pueden encontrarse en [45], [46].

- 3) Puede utilizarse en cambio, una interpolación bidimensional con B-splines (véase también [46] para la formulación matemática de este algoritmo). En este caso, que correspondería al camino d y e de la figura 18, la superficie obtenida no pasa por los puntos de la malla que ha suministrado el usuario. Estos puntos en realidad constituyen los vértices de control, que modelan la forma de la superficie interpolante. Aunque este método permite modificaciones locales y conduce a superficies suaves, de curvatura continua, tiene el grave inconveniente de no interpolar los puntos iniciales.
- 4) Finalmente, puede utilizarse un método híbrido entre los anteriores, que queda representado por el camino a, c, e en la figura 7. El usuario introduce la matriz de puntos de paso en el espacio (normalmente, el proceso de entrada de esta matriz será perfil a perfil e interactivamente en un terminal gráfico). En el siguiente paso, el sistema calcula automáticamente, (en [45] resolviendo un sistema lineal de $n \times m$ ecuaciones con $n \times m$ incógnitas, y en [46] mediante dos productos matriciales), la matriz de vértices de control, tal que, interpolando con B-splines, dará lugar a una superficie que pasa por la malla inicial de puntos introducida por el usuario. Dado que ahora disponemos a la vez del conjunto de puntos de paso y del de vértices de control, es posible, en la fase posterior de modificaciones interactivas, cambiar alguno de los primeros o de los segundos. Evidentemente, si lo que se modifica es la posición de un vértice de control, el cambio en la forma de la superficie quedará localizado.

En el sistema DMI se dispone de la implementación de este último método, según el algoritmo presentado en [46]. La figura 19 muestra algunos de los pasos necesarios para la generación de una superficie, con modificación posterior de su forma.

5.2 Otros métodos de generación de superficies. Operaciones

Dados dos o más perfiles curvados, $P_1(u), \dots, P_m(u)$, es posible obtener superficies por interpolación entre ellos, que denominaremos superficies de unión. En el caso de dos perfiles, se genera una superficie por interpolación lineal, donde el segundo parámetro v indica la distancia a uno u otro perfil:

$$S(u,v) = P_1(u) \cdot (1-v) + P_2(u) \cdot v$$

En el caso de más de dos perfiles, ya no es posible la interpolación lineal, que produciría cambios bruscos de pendiente en la superficie final. La solución utilizada en el sistema DMI (aplicable también al caso de dos perfiles), consiste en la unión cardinal de los m perfiles: se genera la superficie como el conjunto de curvas que interpolan puntos con el mismo valor de u en cada uno de los perfiles,

$$S(u,v) = \sum_{i=1}^m P_i(u) \cdot C_i(v)$$

(en el sentido v , las curvas son splines cúbicos, en general globales).

Un caso particular de la unión de perfiles es el de la generación de superficies por traslación, rotación o movimiento de un perfil a lo largo de un eje. En los dos primeros casos, que han sido implementados, solo es preciso sustituir los perfiles distintos a P_1 por el mismo P_1 al cual se le ha aplicado la transformación geométrica correspondiente. La figura 17 muestra el diseño de una superficie de revolución con este método.

En cualquier caso de los expuestos hasta ahora en los apartados 5.1 y 5.2, la superficie está constituida por una malla de trozos de ecuación bicúbica, que son almacenados como tales en la Base de Datos. En las superficies de revolución, la interpolación circular entre perfiles se aproxima por un número suficientemente elevado de trozos bicúbicos, con lo que el error puede hacerse inferior a cualquier tolerancia especificada.

Dado que no es posible almacenar superficies abiertas en la Base de Datos, deben existir mecanismos de construcción de sólidos a partir de superficies esculpidas. Hasta este momento, se ha implementado un algoritmo de offset, que traslada la superficie en cada punto en dirección de su propia normal, y genera una segunda cara del objeto, con trozos bicúbicos que interpolan los puntos así obtenidos. Esta operación se muestra en la figura 17.

5.3 Extensiones. Tratamiento de superficies bicuadráticas.
Octtrees extendidos.

Las posibilidades del sistema DMI por lo que respecta al modelado de superficies esculpidas, serán extendidas en un próximo futuro, fundamentalmente en los siguientes campos,

- conexión de los distintos módulos existentes en la actualidad, entre sí y con la base de datos general (diseño a partir de una malla de puntos, traslación, revolución, unión cardinal).
- inclusión de algoritmos de suavización de trozos bicúbicos, [47]
- implementación de las operaciones de unión entre superficies.
- operaciones de recortado de superficies constituidas por trozos bicúbicos, [49].
- algoritmos de visualización de trozos bicúbicos y escenas mixtas
- generación de sólidos simples (paralelepípedos o prismas) en que una de sus caras haya sido sustituida por una superficie esculpida. Realización de operaciones booleanas entre ellos y sólidos poliédricos.
- obtención de aproximaciones bicuadráticas de superficies bicúbicas.
- obtención de superficies por división recursiva de la malla de vértices de control, [51] , [63] .
- extensión del modelo de octtrees para el trabajo con superficies formadas por un conjunto de trozos bicuadráticos.

6. EJEMPLOS

En el presente apartado se muestran algunos ejemplos de procesos de diseño utilizando el sistema DMI. Aunque ya han sido referenciados en los apartados anteriores, en el apartado 6.1 se comentan con detalle las distintas figuras del apartado 6.2.

6.1 Comentarios a las figuras

Figura 9. Muestra el proceso de diseño de una pieza mecánica.

- a) Se dibuja, en este caso en el plano x-y, una primera cara del objeto, mediante el cursor y las herramientas de los sistemas de dibujo en 2-D. Aparecen los ejes y las coordenadas en el espacio de los puntos que el usuario va entrando. En este caso, como se observa por la tercera coordenada, se está dibujando sobre el plano $z=0$.
- b) Se ha dibujado un círculo dentro de la cara del objeto. El sistema permite corregir el último polígono entrado, y por ello pide confirmación.
- c) Se ha realizado un primer barrido de traslación paralela de la cara entrada, junto con el agujero. Se presentan cuatro vistas, y se selecciona la opción de modificar este objeto, complicándolo más.
- d) El objeto, después de realizar otro barrido y una operación booleana de diferencia.

Figura 10. Presenta diversas visualizaciones de la pieza anterior.

- a) Representación con eliminación de líneas ocultas. dibujo mediante líneas, que puede ser enviado a plotter o a un sistema de dibujo 2-D. El sistema indica los parámetros de visualización: inclinación de 30 grados en vertical y horizontal. En la parte inferior, aparecen los valores de los restantes parámetros de visualización.
- b) Visualización, desde los mismos ángulos, con simulación de iluminación. Algoritmo de z-buffer de línea, con suavización de Gouraud. Esta suavización elimina el efecto de facetas planas en las superficies cilíndricas.
- c) Visualización, con el mismo algoritmo, de una sección plana del objeto.

Figura 11. Proceso de diseño de un segundo objeto.

- a) Dibujo de la cara inicial, también en el plano x-y. Se ha seleccionado la opción de barrido (sweep) de traslación.
- b) Resultado de un primer barrido. El menú permite diversas operaciones, entre las cuales se ha seleccionado la de introducir un nuevo polígono para crear un resalte.
- c) Se ha dibujado ya el anterior polígono (un círculo) y se ha generado el resalte mediante barrido de traslación perpendicular.

Figura 12. Continuación de proceso de diseño de la figura anterior.

- a) El objeto de la figura 11-c, después de dos procesos de barrido. En el primero, se ha dibujado un hexágono en la cara circular exterior del resalte creado en 11-c, y se ha generado un nuevo resalte por barrido de traslación. En el segundo, el usuario ha dibujado un círculo en la cara horizontal superior del saliente lateral del objeto, y se ha producido un agujero pasante al indicar un barrido de longitud mayor que el espesor de la pieza en esta zona. En este momento se escogería la opción del menú de "fi de creación".
- b) El objeto diseñado, con eliminación de líneas ocultas. Se observan los resaltes y el agujero pasante. Angulos de visualización de 30 y 30 grados.
- c) Visualización con simulación de iluminación pero sin suavizado de las facetas planas en las superficies cilíndricas, a diferencia de las figuras 10-b y 10-c. Algoritmo BSP.

Figura 13. Diseño y ensamblaje de un conjunto de objetos simples.

- a) Dibujo, en el plano x-y, de un arco de 90 grados y un pequeño tramo recto.
- b) Efecto de un barrido de rotación de 360 grados. Se ha generado un objeto sólido, tapado por su cara inferior. Representación mediante modelo de alambres.
- c) Dibujo de un círculo en el plano horizontal z-x, y barrido de traslación vertical.

- d) Después de diseñar un tercer objeto (la base) mediante técnicas parecidas (dibujo de un rectángulo, primer barrido de traslación, y dos barridos sucesivos para generar los discos sobre él), se ha instanciado 8 veces la columna generada en 13-c, y se ha trasladado interactivamente a 8 posiciones distintas sobre la base.

Representación del conjunto mediante alambres.

Figura 14. Continuación del proceso de la figura 13.

- a) Ahora se ha instanciado también el objeto generado por barrido de rotación.
- b) Mediante el menú de traslación interactiva, se ha situado la cúpula encima de las columnas. La representación en alambres del conjunto se realiza desde los ángulos habituales, 30, 30.
- c) Representación del conjunto en axonométrico, y con facetas planas (algoritmo BSP).
- d) Perspectiva cónica con suavización con el algoritmo de Gouraud. Obsérvese la diferencia, sobretodo en la cúpula, con la representación de la figura 14-c.

Figura 15. Segundo ejemplo de ensamblaje de objetos.

- a) En esta vista aparecen, con simulación de iluminación, un conjunto de objetos que provienen de la instanciación y situación en el espacio de cuatro objetos:
- La base cuadrada, en forma de marco, que ha sido obtenida por barrido de traslación del dibujo del marco con 4 agujeros circulares.
 - La base intermedia, que contiene cuatro entrantes en las esquinas, y que ha sido obtenida asimismo por barrido de traslación perpendicular.
 - Las columnas, generadas por barrido de un cuadrado con un agujero circular.
 - El tornillo, instanciado cuatro veces.
- b) Detalle del tornillo , con suavización de las facetas. Ha sido generado mediante tres barridos que han producido la cabeza, el eje y la ranura en cruz.

Figura 16. Diversas secciones planas de un objeto.

- a) El objeto inicial
- b) Sección por un plano inclinado
- c) Sección por un plano paralelo al anterior, y trasladado con respecto a él para entrar más dentro del objeto.

Figura 17. Diseño de una superficie de revolución.

- a) Generación de una superficie por revolución de una curva B-spline plana. La rotación impartida ha sido de 270 grados.
- b) La superficie ha sido dividida en sus trozos bicúbicos y se ha calculado el vector normal (en amarillo) en los 16 puntos con $u=(0,.33,.66,1)$, $v=(0,.33,.66,1)$ en cada uno de ellos.
- c) La generación de una segunda superficie por traslación en la dirección de la normal en cada punto permite generar un sólido a partir de la superficie 17-a. En esta figura se presenta la visualización suavizada de su modelo de fronteras.

Figura 18. Generación interactiva de superficies B-spline.

- a) Entrada de la malla de puntos de paso (normalmente, mediante la entrada de sucesivos perfiles planos)
- b) Generación directa, por producto tensorial de splines cardinales, de la superficie que interpola la malla de puntos.
- c) Cálculo automático de la matriz de vértices de control a partir de la de puntos de paso.
- d) Entrada directa (o modificación) de vértices de control.
- e) Generación de la misma superficie que en b, a partir de los vértices de control y producto tensorial de B-splines.

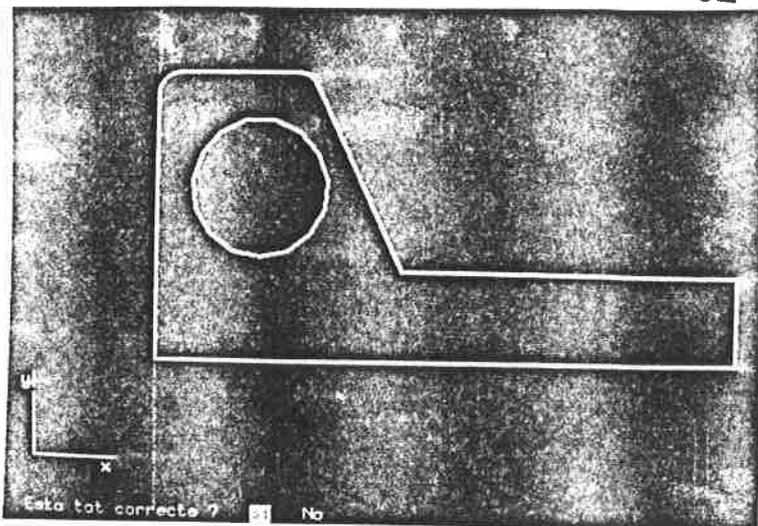
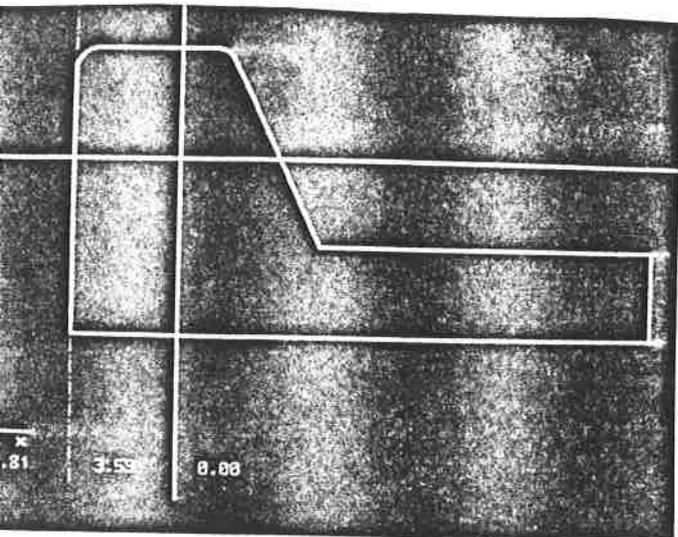
Figura 19. Proceso de diseño de una superficie.

- a) Malla de puntos de paso entrados por el usuario.
- b) Generación de los vértices de control correspondientes a los puntos en 19-a.
- c) Superficie generada a partir de a ó de b. Se aprecia una concavidad en el techo, debida a la forma de los

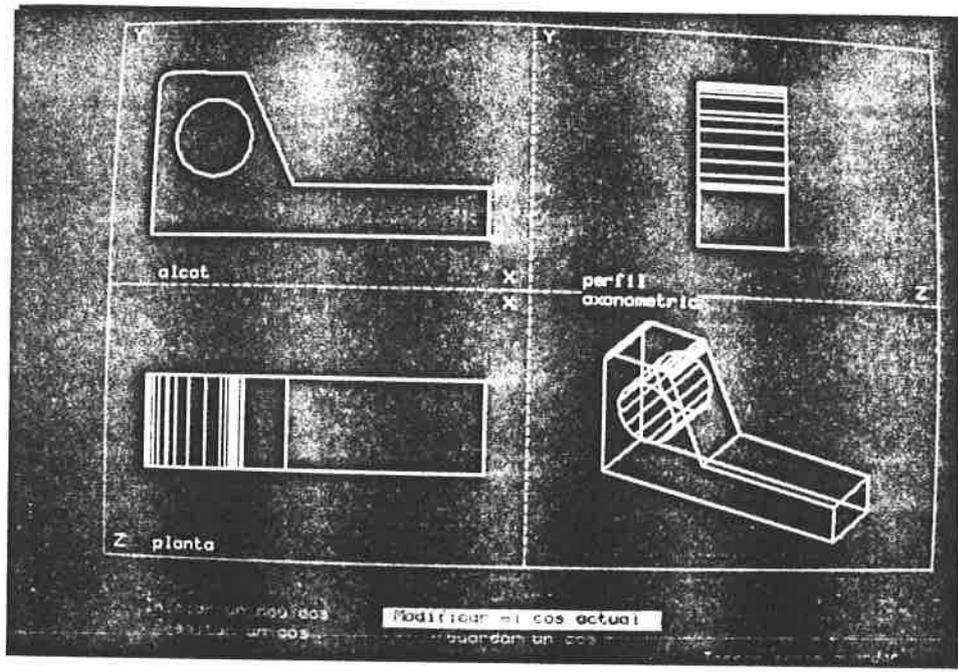
vértices de control.

- d) Modificación de algunos vértices de control, para suprimir la concavidad en c). Obsérvese la diferencia entre 19-b y 19-d.
- e) Superficie final, obtenida con B-splines bicúbicos a partir de 19-d.

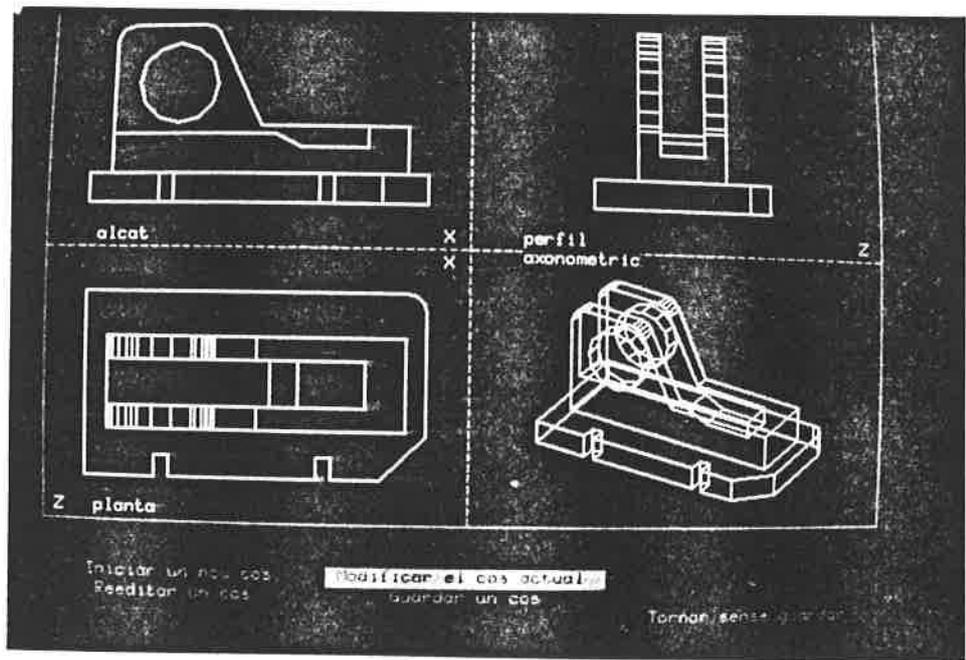
6.2 FIGURAS



b)

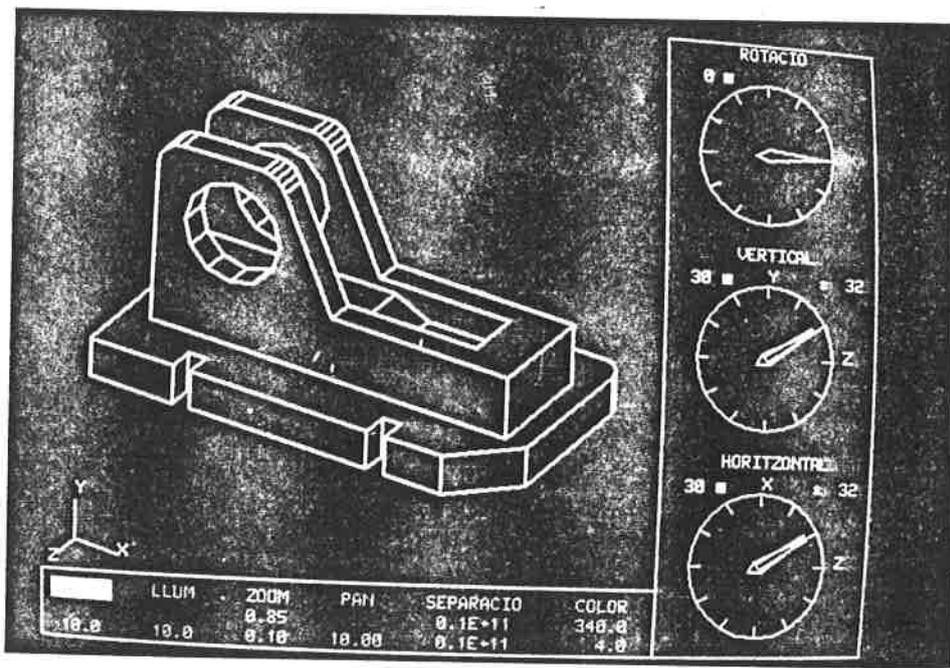


c)

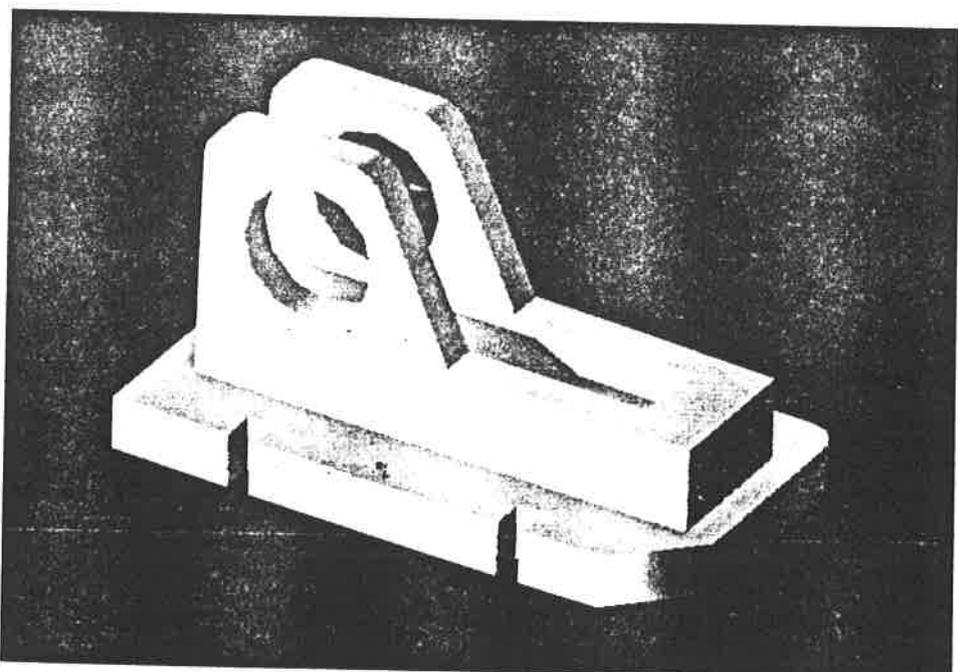


d)

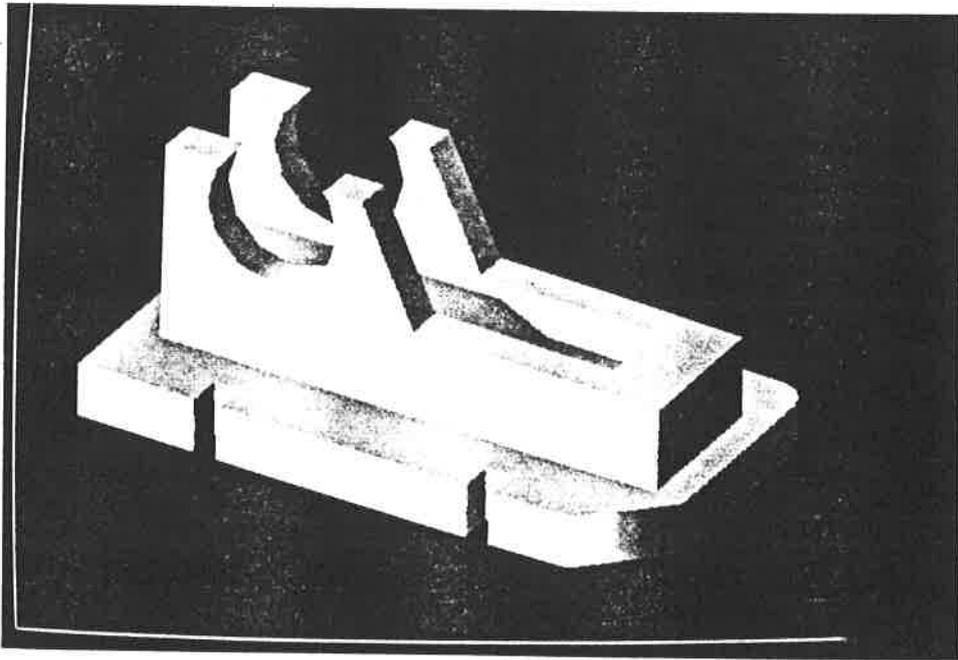
Figura 9. Proceso de diseño de una pieza mecánica.



a)



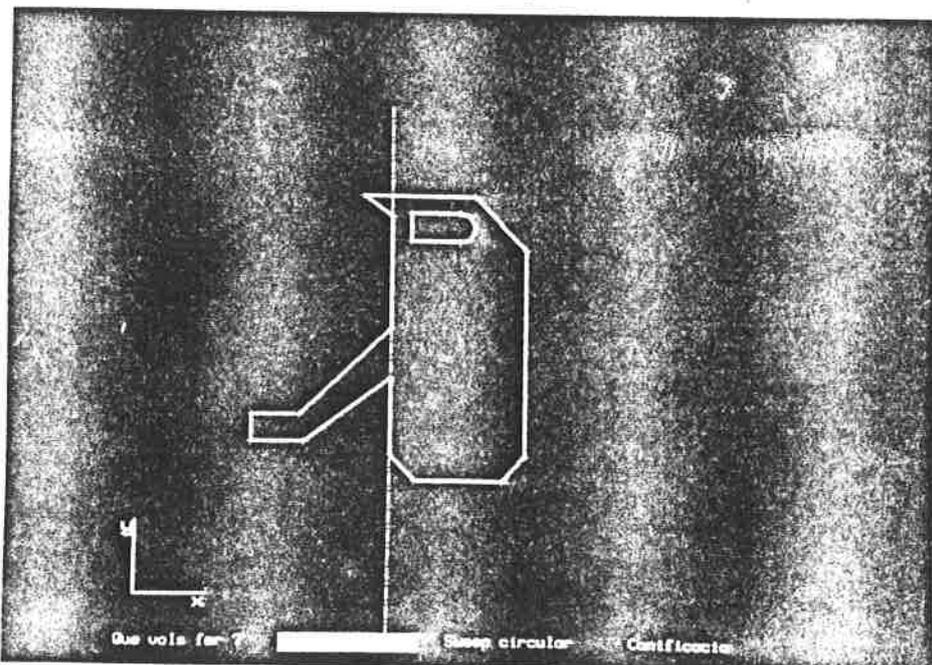
b)



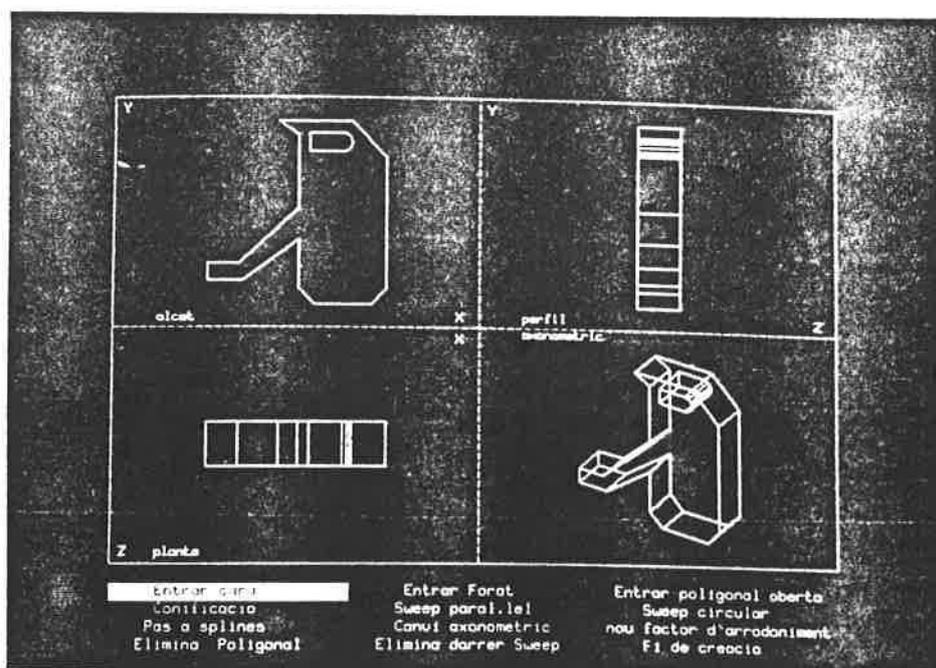
c)

Figura 10. Distintas visualizaciones y secci3n de la pieza de la figura 9.

a)



b)



c)

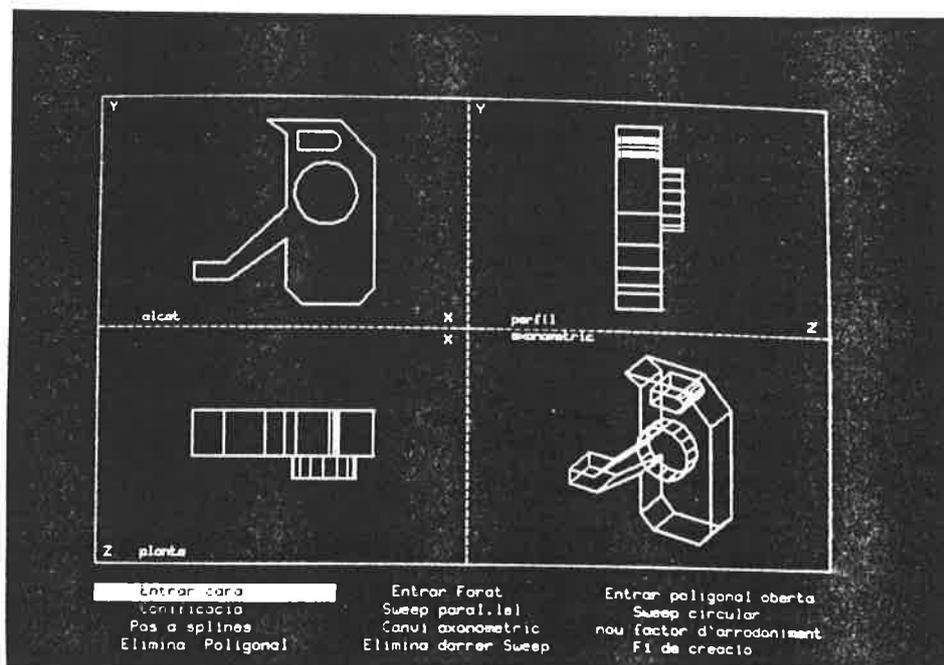
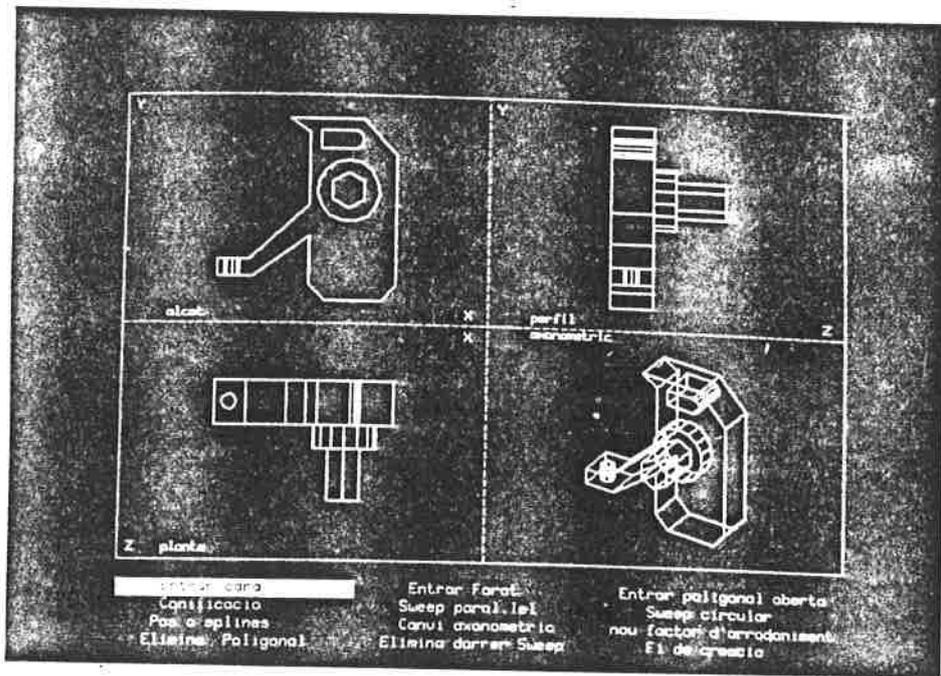
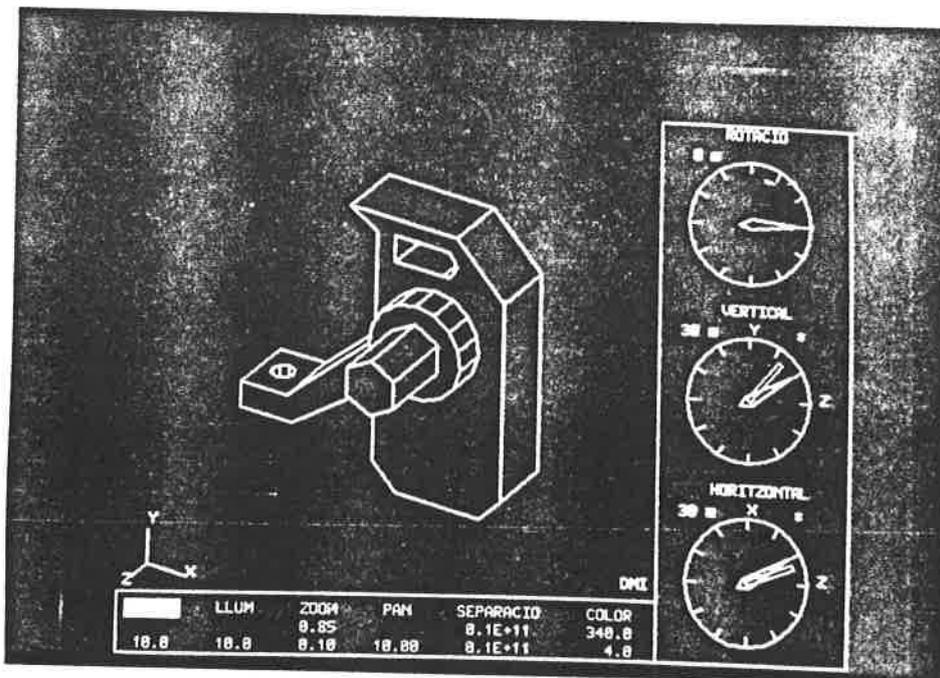


Figura 11. Diseño de un segundo objeto. Dos primeros barridos.

a)



b)



c)

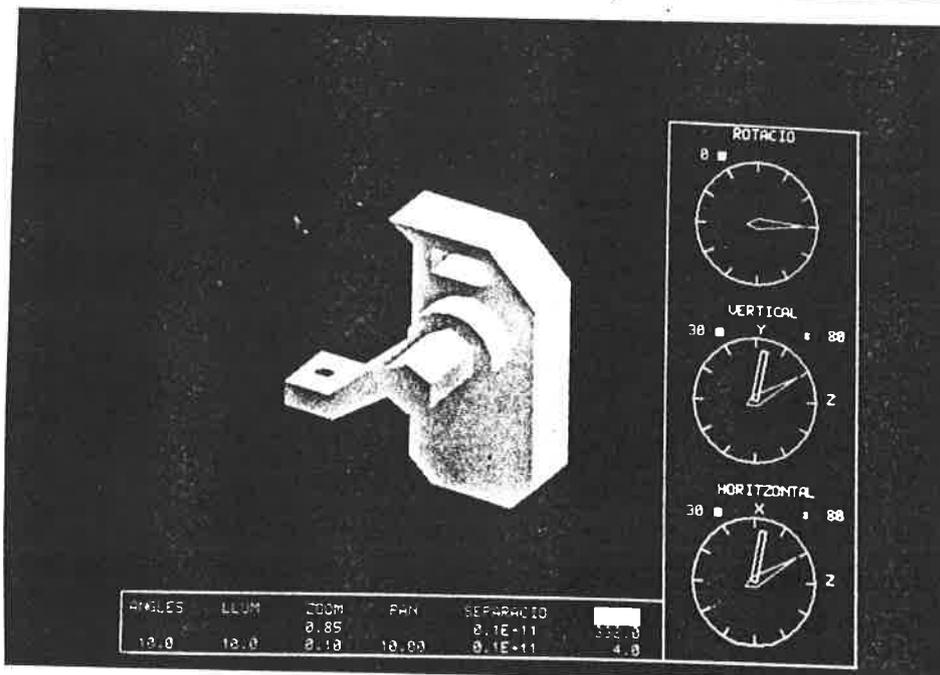
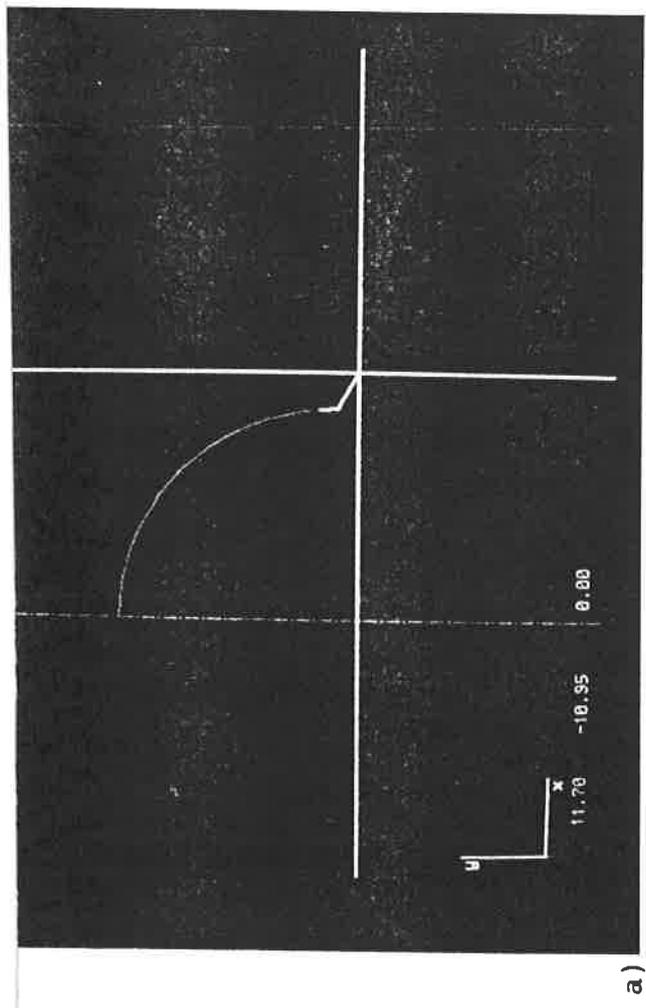
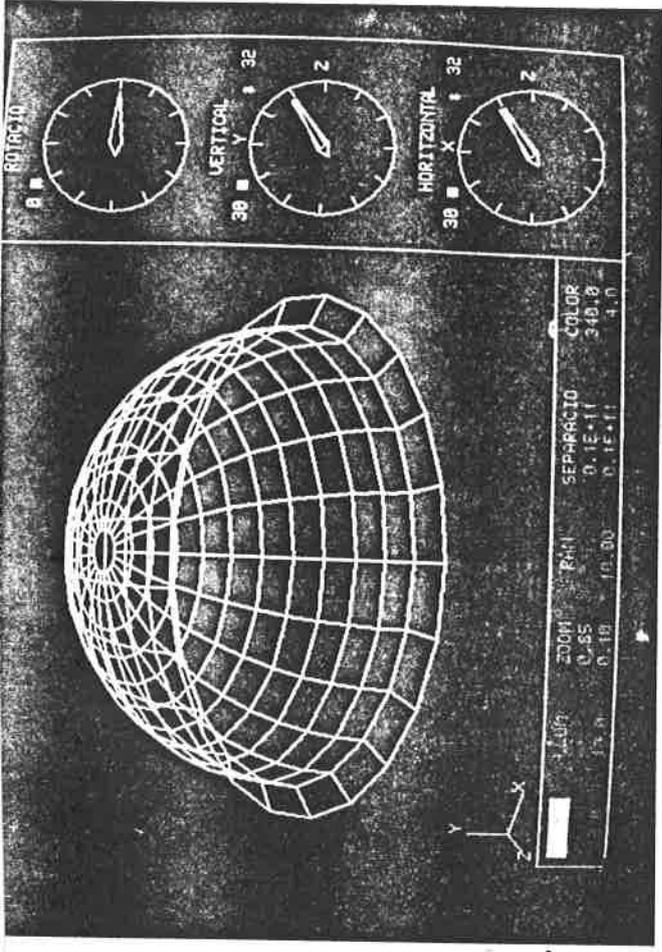


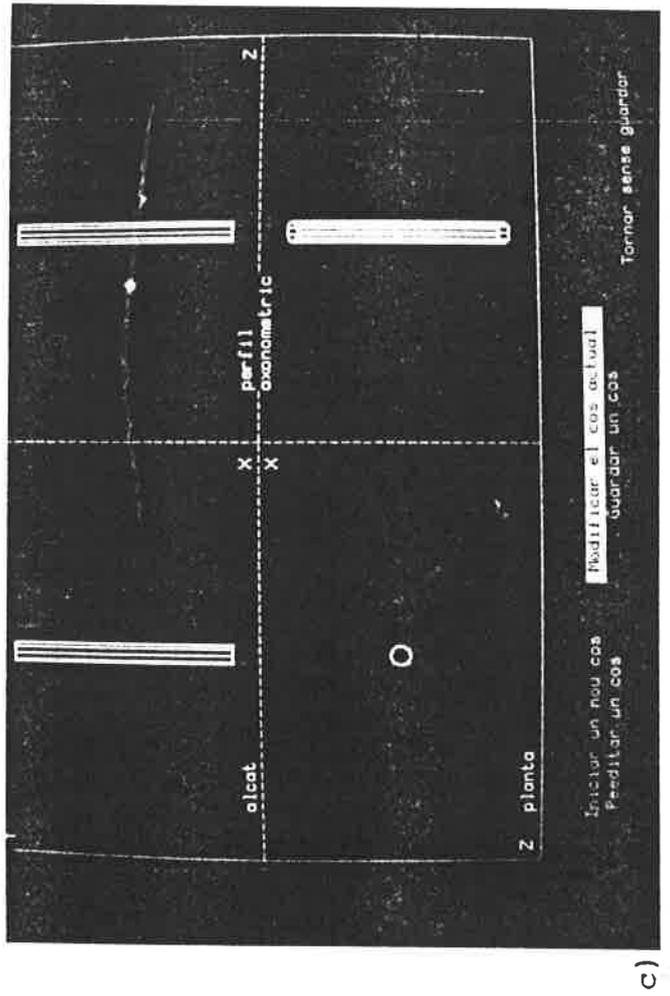
Figura 12. La pieza diseñada en la figura anterior.



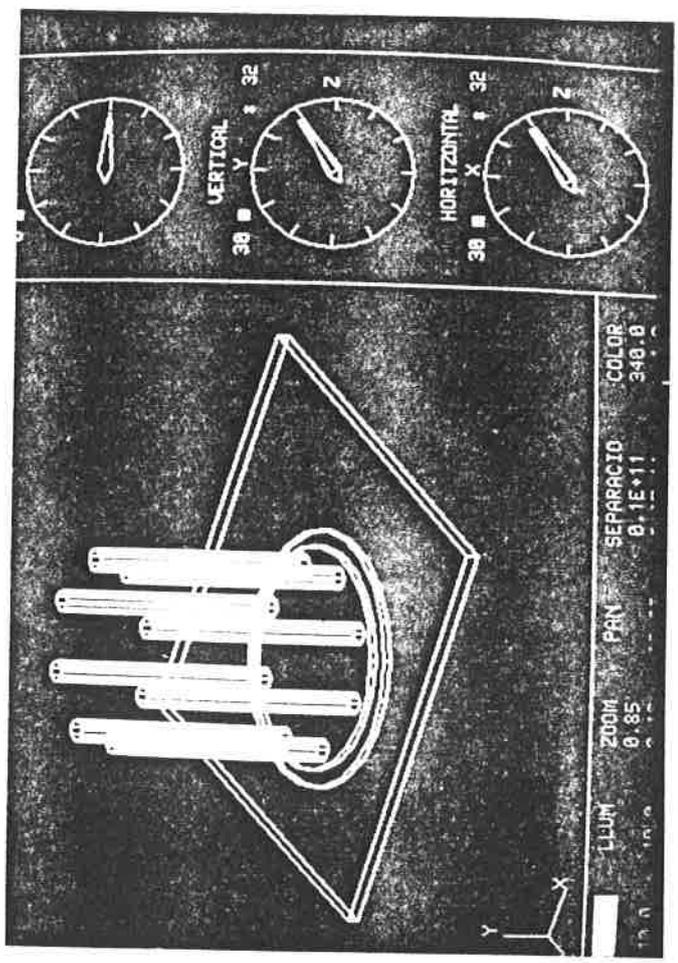
a)



b)

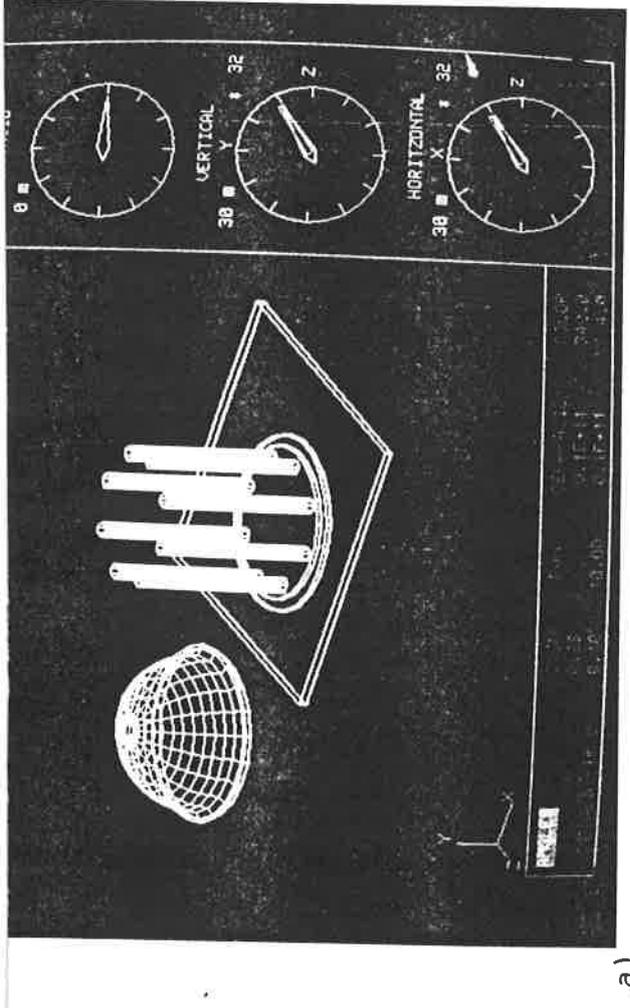


c)

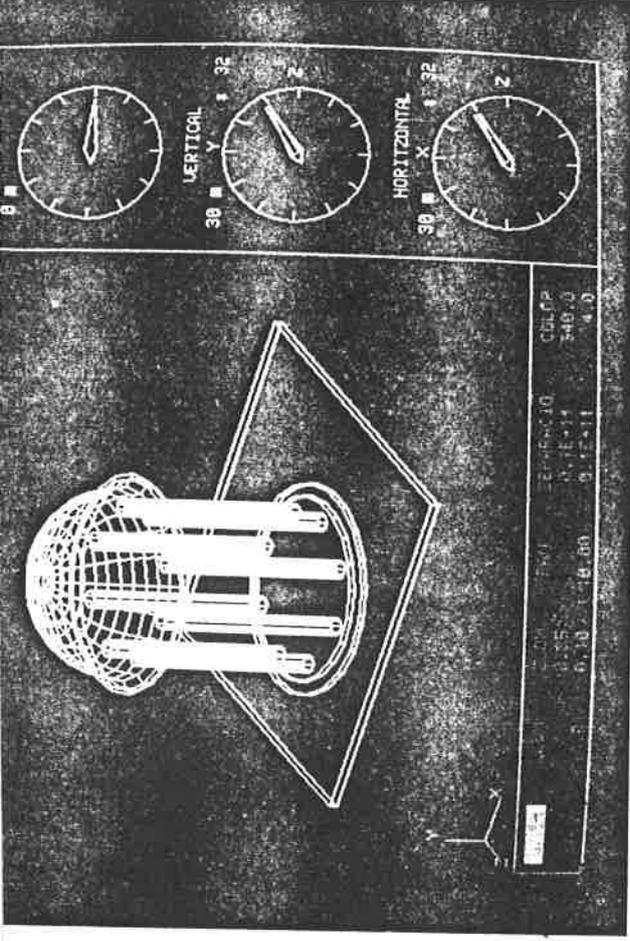


d)

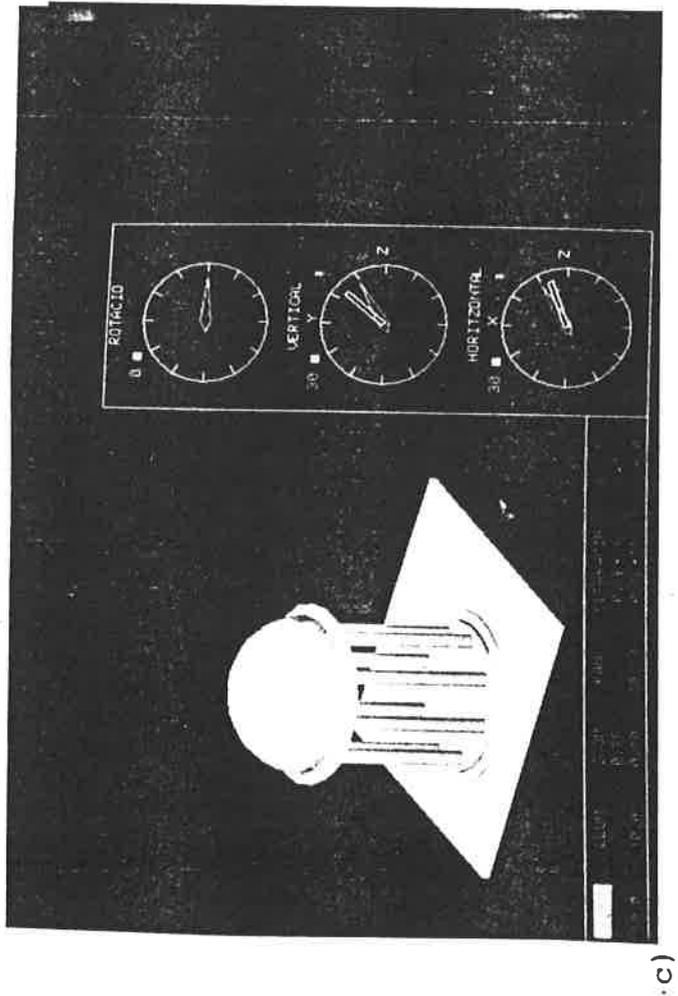
Figura 13. Diseño y ensamblaje de un conjunto de objetos simples.



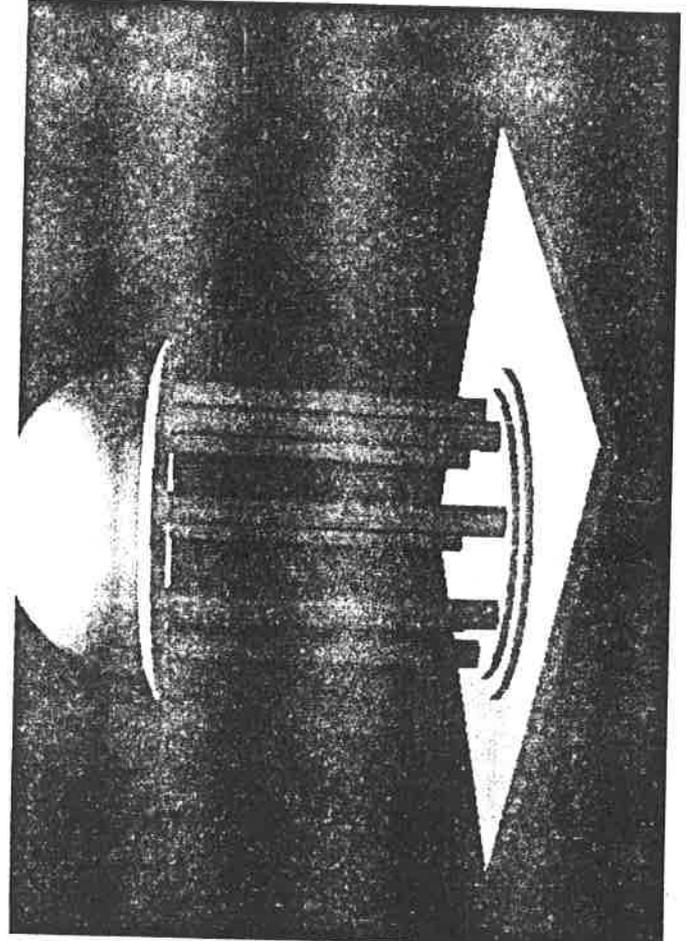
a)



b)

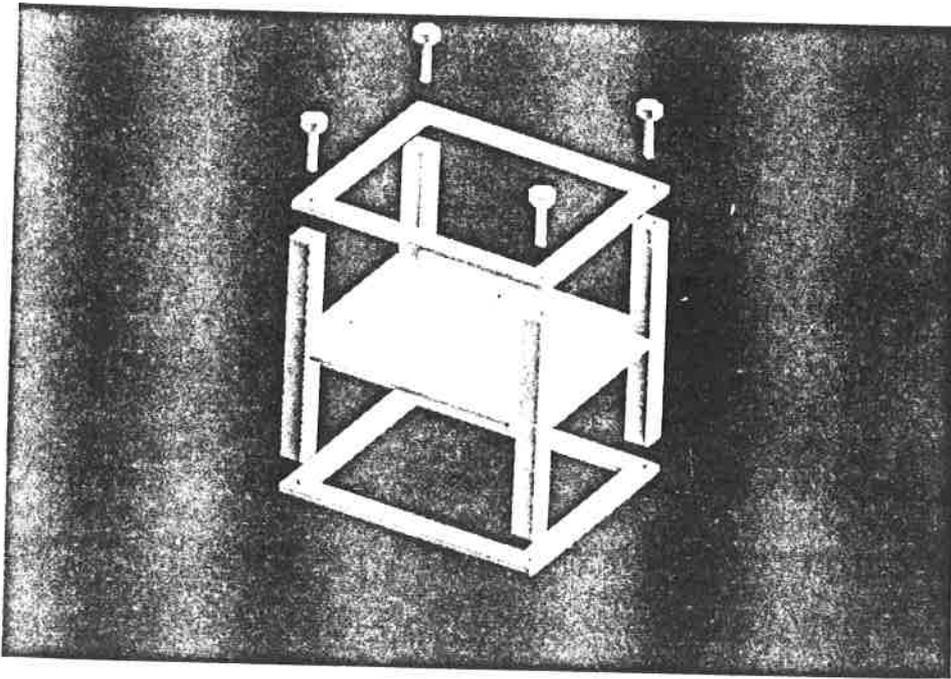


c)

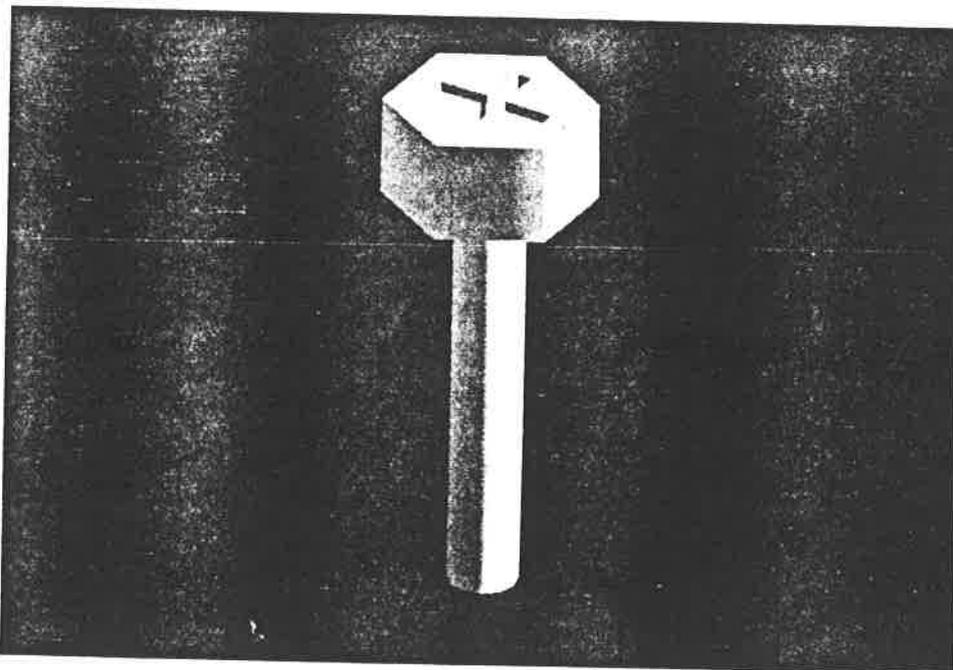


d)

Figura 14. Ensamblaje y visualización de los objetos diseñados en la figura 13.



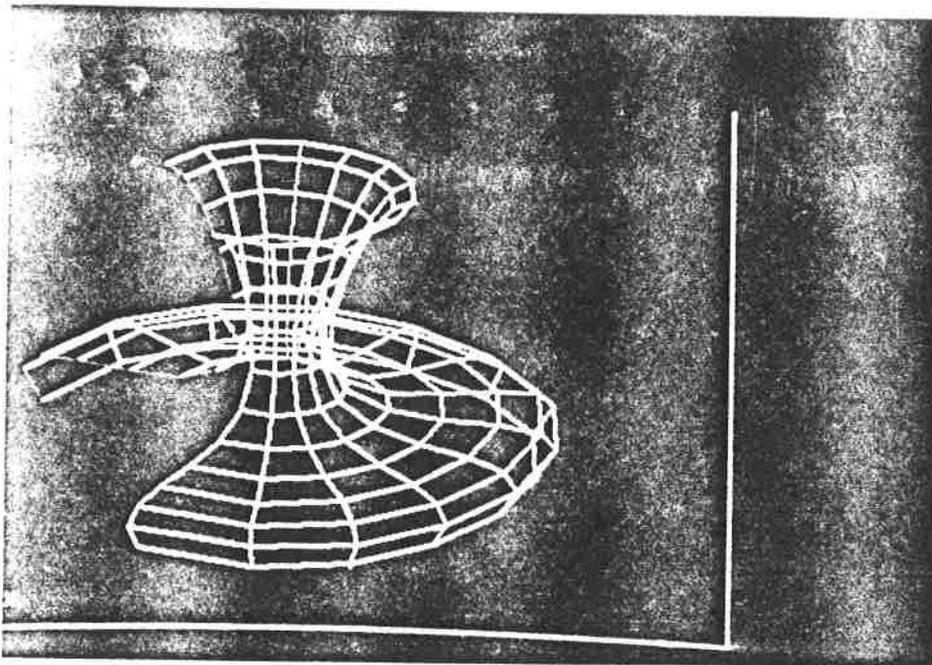
a)



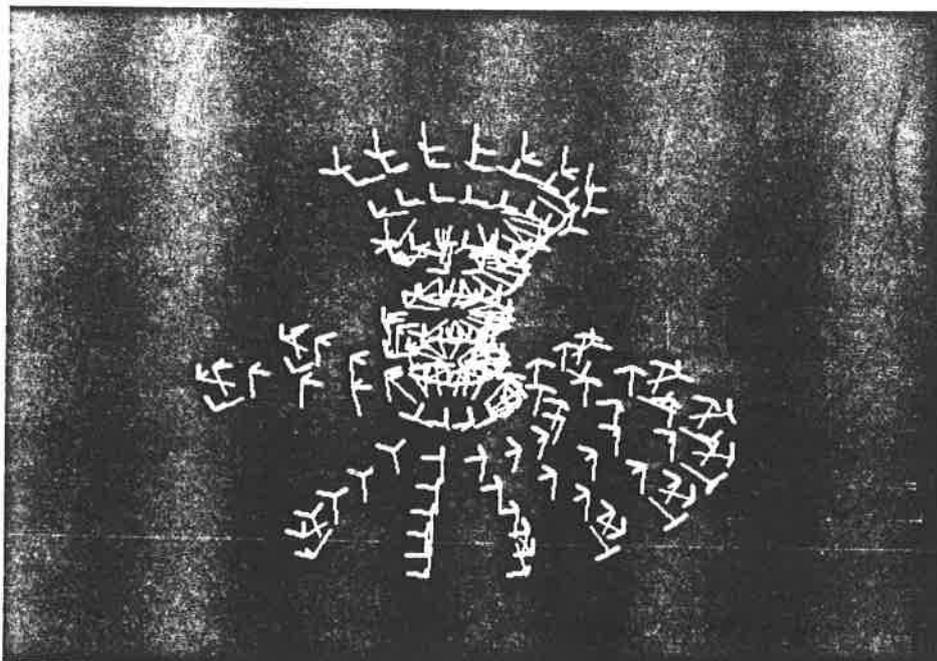
b)

Figura 15. Conjunto de varios objetos, y detalle de uno de ellos.

a)



b)



c)

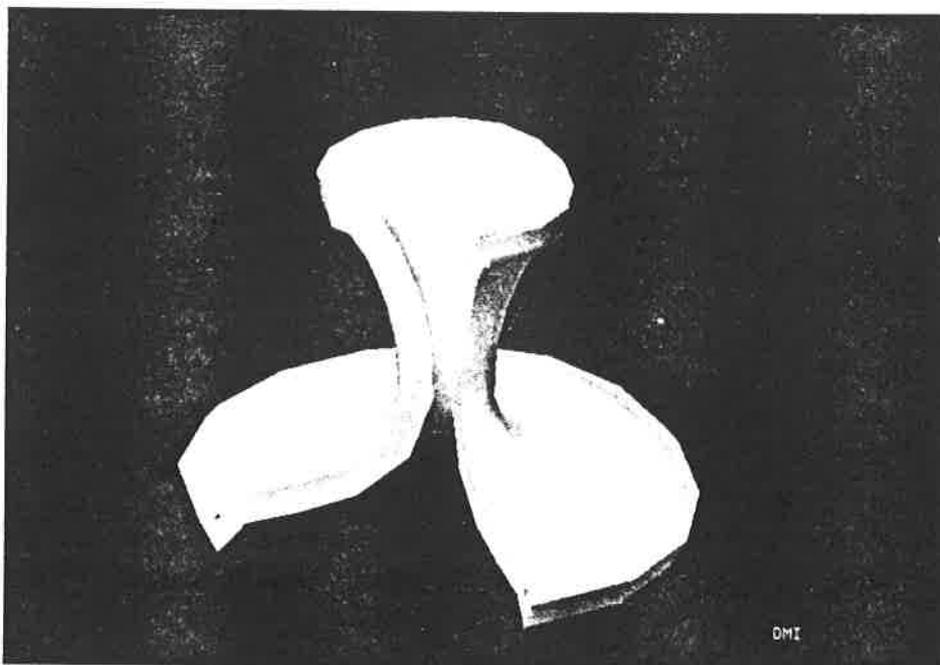


Figura 17. Diseño de una superficie de revolución y proceso de generación de un sólido (offset).

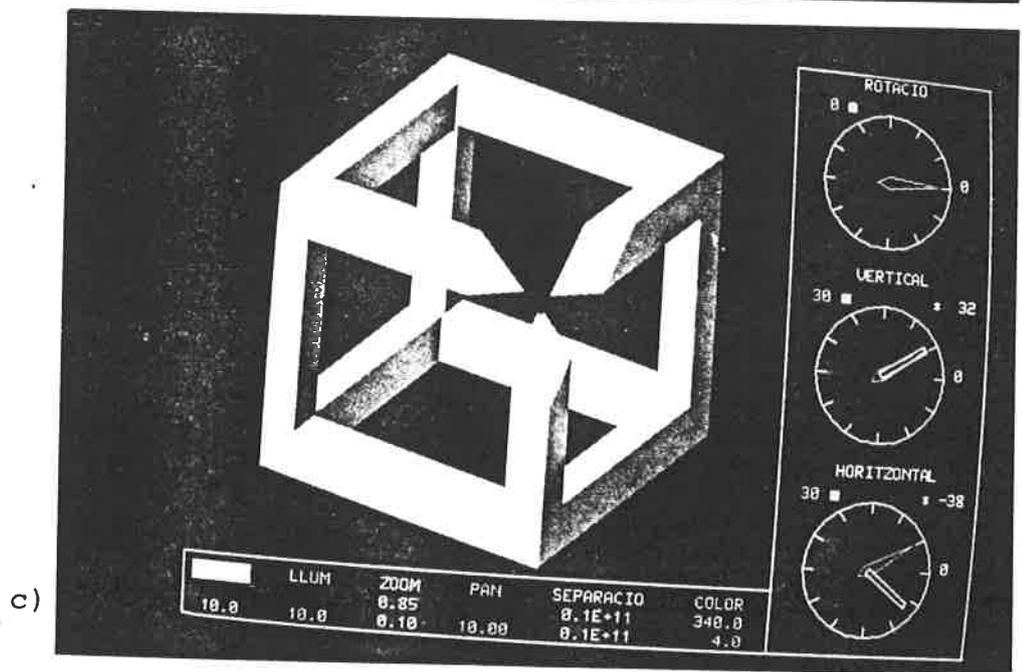
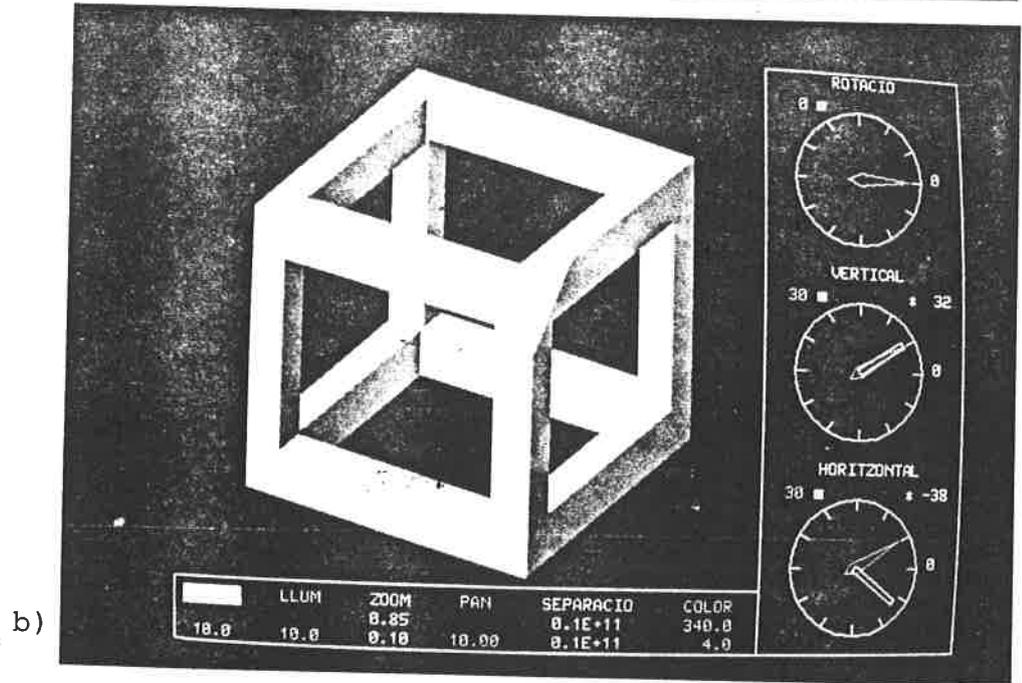
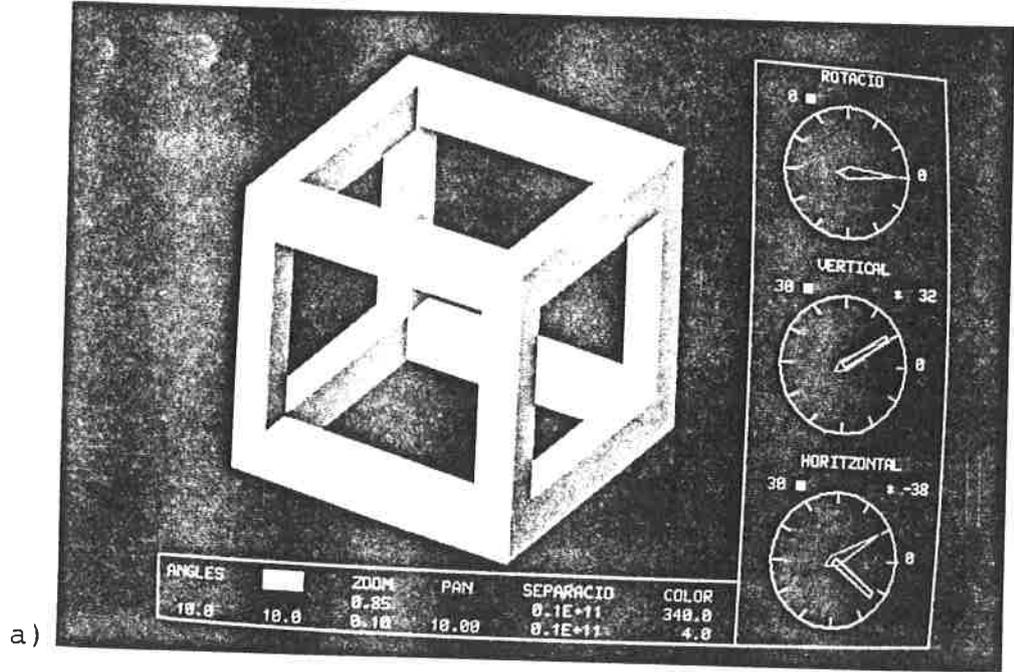


Figura 16. Diversas secciones planas de un objeto.

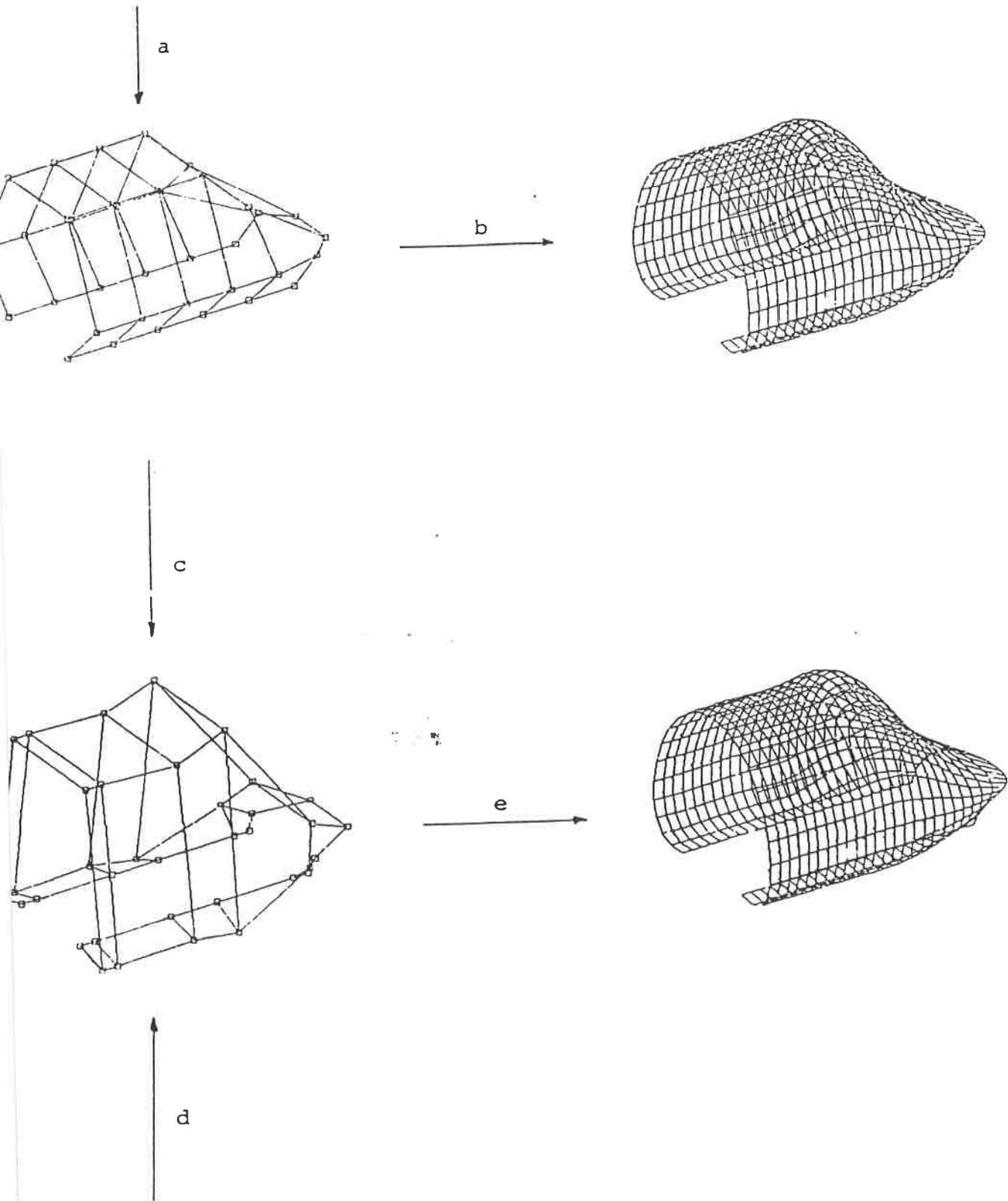
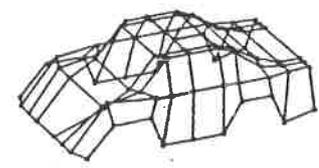
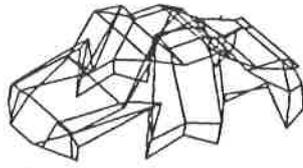


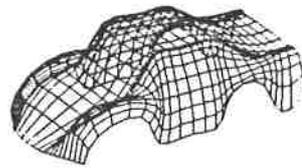
Figura 18. Utilización de puntos de paso y vértices de control en el diseño interactivo de superficies.



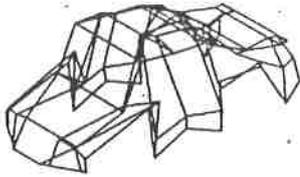
a)



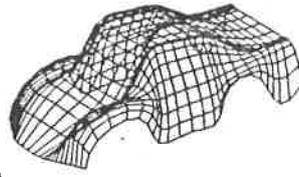
b)



c)



d)



e)

Figura 19. Proceso de diseño de una superficie.

- /1/ REQUICHA, A.A. y Voelcker, H.B. "Solid Modelling: A historical Summary and Contemporary Assessment". IEEE Computer Graphics and Applications March 1982 pg. 9-22
- /2/ REQUICHA, A.G. "Representations for Rigid Solids" Computing Surveys of the ACM Vol 12 n°4 diciembre 1980
- /3/ BAER, A., EASTMAN, C. y HENRION, M. "Geometric Modelling: A Survey". Computer-Aided Design, vol.11 n° 5, 1979 pag. 253-272.
- /4/ TILOVE, R.B. "Set Membership Classification: A Unified Approach to Geometric Intersection Problems" IEEE Trans. Computers vol. C-29, no. 10 Oc. 1980 pg. 874-883.
- /5/ ALDEFELD, B. "On automatic Recognition of 3-D Structures from 2-D Representations", Computer-Aided Design, vol 15 n°2, 1983. pg. 59-64
- /6/ WESLEY, M.A. "Construction and Use of Geometric Models", in Lecture Notes in Computer Science n°89, Springer Verlag pags. 79-136.
- /7/ BRUNET, P., LLISTERRI, A., PSCUAL, J. y PUEYO, X. "Una Base de Datos Gráfica para el Diseño Interactivo en 3-D sobre Mini o Microordenadores", Proc. de la CIL-83, Barcelona 1983, Ed. Marcombo.
- /8/ BRUNET, P. "Diseño Geométrico Asistido por Computador. Modelado de Superficies Curvadas", Mundo Electrónico n°138, 1984 pg. 73-83.
- /9/ WOLFE, R., FITGERALD, W. y GRACER, F. "Interactive Graphics for Volume Modelling", Proc. of the 18th. Design Automation Conference IEEE 1981. pg. 463-470.
- /10/ JACKINS, C. y TANIMOTO, S. "Octtrees and their use in Representing Three-Dimensional Objects" Computer Graphics & Image Processing. Vol. 14 n°3 Nov. 1980, pg.249-270.
- /11/ MEAGHER, D. "Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary Three-Dimensional Objects by computer." Tech. Report IPL-TR.80 111. Image Processing Laboratory. Rensselaer Polytechnic Inst. Oct. 1980.
- /12/ GARGANTINI, I. "Linear Octrees for Fast Processing of Three-Dimensional Objects". Computer Graphics and Image Proc., vol. 20, 1982. Pg. 365-374.
- /13/ AYALA, D. BRUNET, P, JUAN, R. y NAVAZO, I. "Object Representation by Means of Non Minimal Division Quadrees and Octrees", ACM Trans. on Graphics, Special Issue on Computational Geometry, 1985. en Prensa.
- /14/ BRUNET, P., NAVAZO I., "Geometric Modelling Using Exact Octtree Representation of Polyhedral Objects" Eurographics 85, Niza 1985.
- /15/ ATHERTON, P. "A Scan-line Hidden Surface Removal Procedure for Constructive Solid Geometry" ACM Computer Graphics, vol 17 n°3 1983. pg. 73-82.

- /16/ JANSEN, F.W.; VAN WIJK, J.J. "Fast Previewing Techniques in Raster Graphics", Proc. Of Eurographics-83, El-sevier Sci. Publishers, 1983. pag. 195-202.
- /17/ BRONSWOORT, W., VAN WIJK, J. y JANSEN, F. "Two Methods for Improving the efficiency of RAY Casting in Solid Modelling", Computer-Aided Design, vol. 16 n°1, 1984 pg. 51-55
- /18/ SABIN, M.A., "Geometric Modelling. Fundamentals", in Eurographics tutorials 83, Paul J.W. ten Hagen ed., Springer Verlag 1984
- /19/ MESSNER, A. "Solid Polyhedron Measures", ACM Transactions on Mathematical Software Vol 6 n°1, 1980.
- /20/ LEE, Y. y REQUICHA, A. "Algorithms for Computing the Volume and Other Integral Propieties of Solid Objects" Tech. Memo. No.35a. Production Automation Project. University of Rochester. Feb 1980 (también aparecido en en Comm. ACM 1982).
- /21/ NEWMAN, W. y SPROUL "Principles of interactive Computer Graphics. 2nd ed., McGraw-Hill New-York, 1979.
- /22/ FOLEY, J. y VAN DAM, A. "Fundamentals of Interactive Computer Graphics" Addison Wesley Pu. Co., 1982.
- /23/ ENCARNAÇÃO, J. y KRAUSE, F. "File Structures and Data Bases for CAD." North Holland Publ. Co. 1982.
- /24/ BESANT, C.B. "Computer-Aided Design and Manufacture", Ellis Horwood Publ. 1983.
- /25/ ENCARNAÇÃO, J. y SCHLECHTENDAHL, E.G. "Computer-Aided Design", Springer-Verlag 1983.
- /26/ BRUN, J.M. "EUCLID: A manual", Equipe Graphique du Laboratoire d'Informatique pour la Mechanique et les Sciences de l'ingenieur, LIMSI. Orsay, Francia 1976.
- /27/ VEENMAN, P "ROMULUS- The Design of a Geometric Modeller," in Geometric Modelling Seminar. W.A. Carter, Ed. P-80-GM-01 CAM-I, Inc. Bournemouth UK. nov. 1979, pg. 127-152.
- /28/ GOLDSTEIN, R. "Defining the Bounding Edges of a Synthavision Solid Model" Proc. 18th Design Automation Conf. June 29 1981, Nashville Tenn. pg. 457-461.
- 29/ Cambridge Interactive Systems/MGS, "MEDUSA: Modelling System User Guide. Cambridge, CIS/AGS, 1984.
- 30/ VOELCKER, H.B., REQUICHA, A., HARTQUIST, E., FISHER, W. METZGER, R. TILOVE, R., BIRRELL, N., HUNT, W., AMSTRONG G., CHECK, T. MOOTE, R. y MCSWEENEY, J. "The PADL-I 0/2 System for Defining and Displaying Solid Objects" Computer Graphics (Proc. Siggraph 78) Vol 12, n°3 Aug. 1978 pg. 257-263.
- 31/ BROWN, C.M. "PADL-2: A Technical Summary" IEEE Computer Graphics And Applications Vol. 2 n°2 Mar 1982, pg 69-84

- /32/ KIMURA, F. "Geometric Modelling and its Applications: A Survey".
Proc. de Eurographics-84, Copenhagen 1984, pags G-1 a G-6.
- /33/ BOYSE, J., y GILEHRIST, J. "GMSolid: Interactive Modeling for Design and Analysis of Solids" IEEE Computer Graphics and Applications, Vol. 2 n°2, mar. 1982 pg. 27-40
- /34/ HILLYARD, R. C. "The Build Group of Solid Modellers", IEEE Compt. Graphics and Applications, vol 2 n°2 1982 pg. 43-52
- /35/ FAUX, I y PRATT, M. "Computational Geometry for Design and Manufacture", Ellis Horwood Publ. 1979.
- /36/ ROGERS, D., ADAMS, J. "Mathematical Elements for Computer Graphics" Mc. Graw-Hill 1976.
- /37/ DE BOOR, C. "A Practical Guide to Splines" Springer Verlag 1978.
- /38/ BOEHM, W., FARIN, G. y KAHMANN, J. "A Survey of Curve and Surface Methods in CAGD" Computer-Aided Geometric Design vol 1 1984, pg. 1-60.
- /39/ BRUNET, P. "La interpolación con funciones Spline" Introducción automática de las Condiciones de Extremo" Questió, vol 6, n°4, diciembre 1982.
- /40/ GORDON, W. y RIESENFELD, R. "B-Spline Curves and Surfaces" in Computer Aided Geometric Design R. Barnhill and Riesenfeld, eds. Academic Press New-York 1974, pg. 95-126.
- /41/ WU, J., ABEL, D., GREENBERG, D. "An Interactive Computer Graphics Approach to Surfaces Representation" Comm. of the ACM, vol. 20, n°10, 1977.
- /42/ COONS, S. "Surfaces for Computer-Aided Design of Space Forms," Tech. Report MAC-TR'41. Massachusetts Institute of Technology Cambridge. Mass. June 1967.
- /43/ BARNHILL, R.; BROWN, J y KLUCEWICZ, I. "A New Twist in CAGD", Computer Graphics and Image Processing vol 8, 1978, pg. 78-92
- /44/ GREGORY, J. A. "C¹ Rectangular and non-Rectangular Surface Patches", en Surfaces in Computer-Aided Geometric Design, Barnhill and Boehm eds. North-Holland, 1973, pg. 25-33.
- /45/ BARSKY, B. GREENBERG, D. "Determining a Set of B-spline Control Vertices to Generate an Interpolating Surface" Computer Graphics and Image Processing, 14, 1980. pg. 203-226.
- /46/ BRUNET, P., AYALA, D. y NAVAZO, I. "An interactive Algorithm for the Generation of B-spline Surfaces" Proc. of the ICS-83. Nurnberg 1983.
- /47/ BRUNET, P. "Increasing the Smoothness of Bicubic Spline Surfaces", in Proc. of the Second Meeting on Surfaces in Computer-Aided Geometric Design, North-Holland 1985.
- /48/ COHEN, E., LYCHE, T y RIESENFELD, R. "Discrete B-Splines and Subdivision Techniques in Computer Aided Geometric Design and Computer Graphics, "Computer Graphics & Image Processing. Vol 14, n°2 Oct. 1980. pg. 87-111.
- /49/ PETERS, G. "Interactive Computer Graphics Application of the Parametric, Bi-cubic Surface to Engineering Design Problems" en Computer-Aided Geometric Design, Barnhill and Rosenfeld Eds. Ac. press 1974. pg. 259-302.
- /50/ CHARROT, P.; GREGORY, J. "A Pentagonal Surface Patch for CAGD", Comp. Aided Geometric Design, vol 1. n°1, 1984, pg. 87-94.

- /51/ CATMULL, E. y CLARK "Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes" Computer-Aided Design, Vol. 10, n°6, nov. 1978 pg. 350-355.
- /52/ BEZIER, P. "Mathematical and Practical Possibilities of Unisurf" en Computer Aided Design, Ac. Press 1974.
- /53/ FARIN, G. "Some Aspects of Car Body Design at Daimler-Benz" in Computer-Aided Geometric Design, North Holland Publ. Co. 1983. pg. 93-98.
- /54/ ELSAESSER, F. "Surfaces and their Applications at Opel" en Surfaces in Computer-Aided Geometric Design, Barhill and Boehm eds. North Holland Publ. Co. 1983. pg. 157-162.
- /55/ HOFELD, H. "Surface Description in the Application at Volkswagen" in Computer-Aided GEometric Design, North Holland Publ. Co. 1983. pg. 35-42.
- /56/ NOWACKI, H., REESE, D. "Design and Fairing of Ship Surfaces" in Computer-Aided Geometric Design, North Holland Publ. Co 1983.
- /57/ ANSON, R. y TWEEDLY, A. "Alite. An Engineering Tool for Automatic Chip Design. Proc. Of Eurographics-82, North Holland Publ. Co. 1982, pg 281-291
- /58/ DULLIN, E., OUARET, S. y MAY, P. "Computer-Aided Design Systems for three Dimensional Shape Plane Figures. Shoe Making Application" Proc. of Eurographics 82, North Holland Pub. Co. 1982, pg. 265-280.
- /59/ ELLIS, T. y SEMENKOV, O. "Advances in CAD-CAM", North Holland Publ. Co. 1983.
- /60/ AYALA, D. "Modelat Solid: Realització d'Operacions Booleanes amb Model de fronteres", Informe de Recerca DMI01/85, ETSEIB, U.P.C., 1985.
- /61/ BRUNET, P, y JUAN, R. "Dibujo versus diseño asistido por computador", Proyecto, noviembre de 1985.
- /62/ NAVAZO, I, "Contribució a les Tècniques de Modelat Geomètric d'Objectes Polièdrics Usant la Codificació amb Arbres Octals", Tesis Doctoral, UPC, 1986.
- /63/ DOO, D.V.H., y SABIN, M. "Behaviour of Recursive Division Surfaces near Extraordinary Points", Computer-Aided Design, vol 10, 1978, pag 356.
- /64/ PASCUAL, J. y VILA, R. "L'algorisme de visualització per partició binaria de l'espai", informe de recerca DMI01/86, ETSEIB, UPC, 1986 (en preparació).
- /65/ ROGERS, D. " Procedural Elements for Computer Graphics", Mc. Graw Hill, 1985.
- /66/ LUCAS, M. y GARDAN, Y. "Techniques Graphiques Interactives et CAO" Hermes publ, Francia, 1983.

A N E X O

Relación de comandos del editor 2-D, utilizados para la generación de polígonos planos en el proceso de barrido.

Entrar Poligonal. - El proceso de entrada de poligonales (conjuntos de tramos rectos consecutivos) se realiza entrando una secuencia de puntos. Los puntos pueden ser entradas de varias formas:

- Señalando con un cursor un punto en la pantalla.
- Entrando las coordenadas en valor absoluto o relativo y en coordenadas cartesianas o polares.
- Señalando un punto próximo al que se desea.
- Seleccionando el punto medio de un segmento.
- Situados en un punto y dada una recta, seleccionar el punto de la recta tal que uniendo con el primero, la recta resultante es perpendicular a la primera.
- Situados en un punto y dado un arco, seleccionar el punto del arco tal que uniendo con el primero, la recta resultante es tangente al arco.
- Seleccionar un punto de forma que la poligonal quede cerrada.

Entrar Arco. - Los arcos son aproximados por tramos rectos con una precisión variable por el usuario, con un valor de defecto correspondiente a la precisión de la pantalla de trabajo. Las posibilidades para la entrada de arcos son:

- Entrar centro, número de tramos y ángulo: a partir del último punto de la poligonal en curso se efectúa un arco con los valores especificados, así no solo se permite la entrada de arcos sino también de polígonos regulares abiertos.

- Entrar tres puntos (inicial, de paso y final): automáticamente se calcula el arco que pasa por dichos puntos.

- Entrar punto inicial, centro y final: se calcula el arco que saliendo del primer punto y con el centro especificado llega hasta el ángulo que define el tercer punto.

- Entrar radio y dos puntos (inicial y final): se enlazan así dos puntos por un arco de radio definido.

- Entrar punto inicial, punto final y ángulo: se enlazan los dos puntos por un arco de ángulo definido.

- Entrar punto inicial, centro y ángulo: a partir de un punto y de un centro se realiza un arco de un ángulo determinado.

en todas las opciones mencionadas, la entrada de puntos que determinan el arco se puede realizar con todas las posibilidades explicadas en el apartado anterior.

Entrar Círculos. - Los círculos son aproximados por tramos rectos con una precisión variable por el usuario, con un valor de defecto correspondiente a la precisión de la pantalla de trabajo. Las posibilidades para la entrada de círculos son:

- Entrar punto centro y radio.

- Entrar punto centro y diámetro.

- Entrar tres puntos del círculo.

- Entrar los dos puntos extremos de un diámetro.

Modificación De Poligonales. - Hay una serie de comandos para la edición de poligonales:

- Borrar punto: si en el proceso de entrada de poligonal existe una entrada de un punto incorrecto basta seleccionarlo y el punto es eliminado.

- Modificar punto: en cualquier momento es posible modificar las coordenadas de un punto seleccionandolo y entrando las nuevas coordenadas.

- Insertar punto: en cualquier momento es posible insertar un nuevo punto entre dos puntos cualquiera de una poligonal.

- Borrar parte de una poligonal: en cualquier momento es posible eliminar parte de una poligonal seleccionando dos puntos sobre ella.

Utilidades. - Hay una serie de utilidades para la edición de poligonales:

- Zoom: es posible disminuir y ampliar parte de una vista y restaurar la vista total.

- Copia: existe tambien la posibilidad de duplicar poligonales, obteniendo asi otra identica a la inicial.

- Traslacion: las poligonales pueden desplazarse de su posicion actual hasta una nueva situacion dentro del conjunto de poligonales.

- Rotacion: las poligonales pueden girarse alrededor de su posicion actual hasta una nueva orientacion dentro del conjunto de poligonales.

- Escalado: las poligonales pueden escalarse en su posicion actual hasta un nuevo tamaño referido al conjunto de poligonales.

- Simetria: una poligonal puede reflejarse sobre un eje de simetria obteniendo asi una nueva poligonal simetrica de la inicial.

- Spline: una poligonal de tramos rectos puede substituirse por una interpolacion por funciones polinomicas de características mucho mas suaves que la poligonal inicial.

- Guardar y recuperar poligonales: poligonales de uso frecuente o de dificil generacion pueden ser almacenadas y recuperadas para mayor facilidad de uso.

- Modificar status: existen toda una serie de características propias del proceso de generacion de poligonales que pueden ser modificadas por el usuario segun la aplicacion deseada. Dichas características son: precision de arcos y circulos, malla de gridding o redondeo, archivo de journal en on/off, paso al status general del sistema.