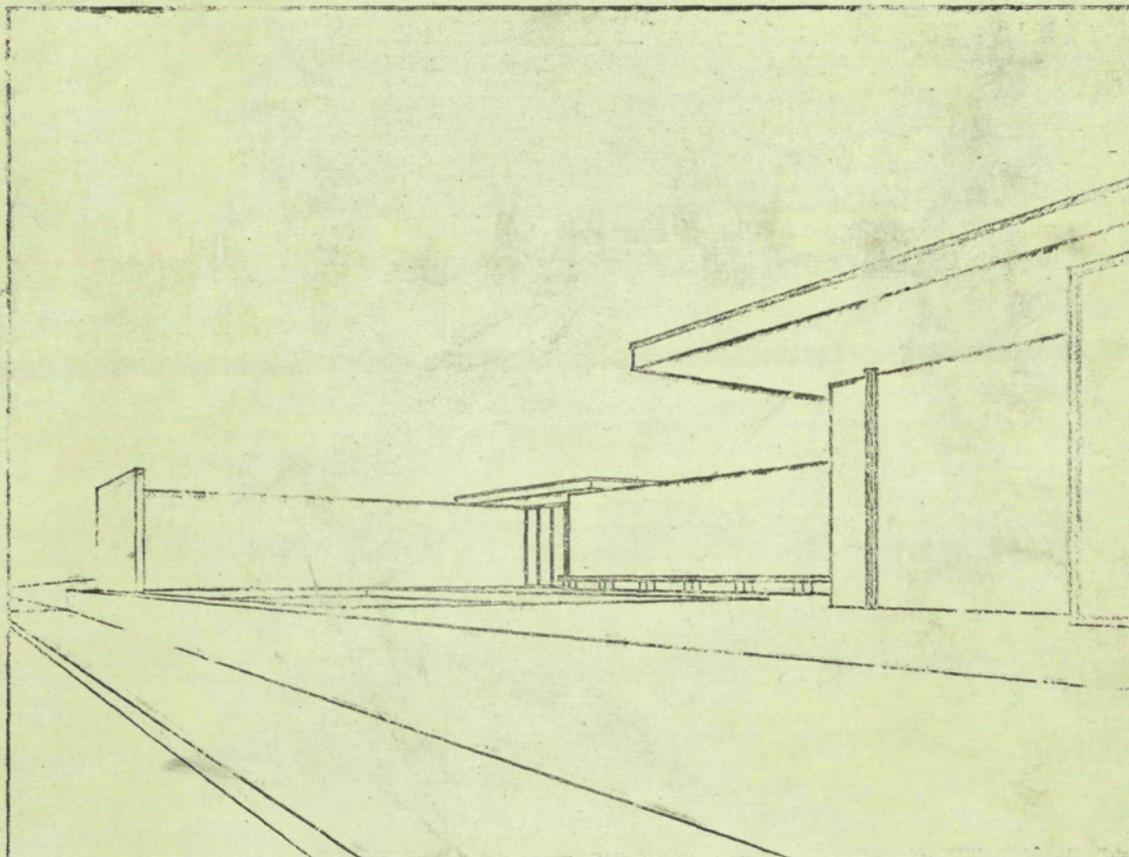


M M M M  
M M M M  
M M M M

TESIS DOCTORAL

BASES PARA LA UTILIZACION  
GRAFICA DEL ORDENADOR



AUTOR

MANUEL JOSE LUQUE GONZALEZ

DIRECTOR

MANUEL BAQUERO BRIZ

TESIS

86/2

1400069111

UNIVERSITAT POLITÈCNICA DE CATALUNYA  
ADMINISTRACIÓ D'AFILIACIÓ D'ACADÈMICS

Aquesta Tesi ha estat enregistrada  
a la pàgina 3 amb el número 22

Barcelona, 30 de juliol de 1986

L'ENCARREGAT DEL REGISTRE,

M. Lluís - Bay

*Luque*  
*12*

TESIS DOCTORAL

BASES PARA LA UTILIZACION  
GRAFICA DEL ORDENADOR

AUTOR

MANUEL JOSE LUQUE GONZALEZ

DIRECTOR

MANUEL BAQUERO BRIZ

a Isabel

## AGRADECIMIENTOS

a D. Manuel Baquero Briz, Doctor Arquitecto y catedrático de Dibujo I en la E.T.S.A.B., quien ha actuado como Director de esta Tesis.

a D. Juan Antonio Soler Renales, Arquitecto con quien el autor ha compartido una gran parte de las vicisitudes que han acompañado la elaboración de esta Tesis.

## INDICE

Introducción .....	6
Parte I	
I - Descripción del modelo utilizado .....	10
II- Transformaciones en tres dimensiones .....	27
III- Proyecciones .....	44
IV - Recortes (Clipping) .....	54
V - Visualización .....	78
VI - Desarrollo de un programa de visualización	95
Parte II	
I - Generación del modelo de un objeto .....	109
II- Descripción general del sistema propuesto .	117
III-Utilización del sistema .....	137
IV - Ejemplos de utilización .....	154
Resumen a modo de conclusiones .....	213
Referencias bibliográficas .....	221

Anexo. Listados.

GenPrVis .....	228
VisOp .....	232
RutVar .....	243
FichOp .....	256
CÍipDep .....	264
DibOp .....	296
GenOp .....	315

## INTRODUCCION

Tradicionalmente el arquitecto ha venido sirviéndose de un amplio repertorio de instrumentos para la realización de las representaciones gráficas que necesitaba en el ejercicio de su profesión. La forma de trabajar con estos instrumentos, junto con la utilización o no de algunos de ellos, puede crear diferencias de resultado, pero el proceso de ejecución no es sustancialmente distinto. Sin embargo el avance tecnológico ha permitido la presencia del ordenador en el campo de la realización de gráficos, lo cual comporta una forma diferente de afrontar el problema, pues el equipo informático que permite esta tarea constituye un instrumento tan eficiente que no puede ignorarse su existencia, siendo en algunos casos simplemente insustituible, pero exigiendo a su vez una preparación específica para su utilización correcta de forma que permita obtener del mismo todas sus inmensas posibilidades.

La dedicación del autor, durante más de una década, a la docencia del dibujo en la Escuela de Arquitectura, junto con su gran interés por los temas informáticos, le ha hecho consciente de la necesidad de incorporar el estudio de las posibilidades gráficas del ordenador, tanto desde el punto de vista pedagógico como del de la investigación, a las tareas del departamento en que se desarrolla su actividad universitaria, y constituye la intención y la justificación de la presente tesis formar parte de dicha tarea.

## Introducción

Se ha trabajado en los dos temas de mayor interés en el área de estudio mencionada, dedicándose cada una de las dos partes de esta tesis a uno de ellos, incluyéndose, en el anexo situado al final de la misma, los programas realizados, como parte importante y significativa de la tarea emprendida, y a su vez utilizados para la obtención de la práctica totalidad de las figuras incluidas, para ilustrarla.

La primera parte se ocupa de los procesos que es necesario realizar para conseguir obtener la visualización de un objeto tridimensional poliédrico, desde una posición cualquiera del observador, con la depuración de las partes del objeto que permanezcan ocultas, así como, en caso necesario, prescindiendo de la porción del mismo situada en la zona del espacio que no se encuentra incluida dentro de la apertura de visión considerada. Es importante resaltar que los primeros capítulos de esta parte figuran como necesario preámbulo al desarrollo de los algoritmos de recorte (clipping) de caras en 3D, que figura al final del capítulo IV, y del de depuración de líneas ocultas, analizado en el capítulo VI, que constituyen la contribución del autor dentro del tema de visualización de objetos tridimensionales.

La segunda parte del trabajo expone el desarrollo de un sistema de generación de sólidos tridimensionales que propone el autor como alternativa, en el campo aquí contemplado, al modelo de fronteras, permitiendo la manipulación de sólidos en 3D con una mayor eficiencia y comodidad que el mismo. Mediante este sistema se posibilita la fácil generación posterior de los modelos de fronteras necesarios para representar en el interior del ordenador temas arquitectónicos, con la intención de realizar posteriormente su exploración visual mediante los algoritmos de visualización analizados en la primera parte.

## Introducción

Los condicionantes que puede imponer el hardware disponible no deben convertirse en las directrices fundamentales del trabajo, pues, debido a la rápida y constante evolución tecnológica que es característica de este campo, esto convertiría algunas decisiones, tomadas en base a determinadas limitaciones del hardware, en injustificables en un futuro relativamente cercano, al aparecer en el mercado equipos más potentes y versátiles a disposición del profesional medio. Pero es inevitable que algunos aspectos como la capacidad de memoria disponible, la velocidad de proceso y la precisión de los cálculos, sean factores clave, en la elección entre las varias alternativas posibles, en el momento de determinar la estrategia a seguir en puntos específicos del programa, debiendo tenerse en cuenta, en la medida de lo posible, la fácil modificación del algoritmo para adaptarse a condiciones más favorables cuando estas se presenten.

El equipo utilizado para la realización del trabajo está dotado de un microprocesador Intel 8086 de 16 bits, una memoria Ram de 640K, aunque el sistema operativo utilizado no admite la utilización de más de 128K como area de trabajo, destinándose el resto a ser utilizado como RamDisk. El almacenamiento externo disponible consiste en dos unidades de disco flexible de 5,25" de 720K de capacidad cada una de ellas. Además de ser el equipo de mejores características que, de una forma efectiva, ha podido utilizar el autor para la realización de esta tesis, es un equipo que puede estimarse normal en el campo de aplicación considerado, que es la actividad profesional media de un arquitecto en la actualidad. Sin que esto vincule de forma exclusiva el trabajo a un equipo de este tipo, pues es poco complejo el adaptar todo lo expuesto en esta Tesis a cualquier equipo de prestaciones similares o superiores si esto fuese aconsejable.

PARTE I  
VISUALIZACION

## 1. DESCRIPCIÓN DEL MODELO UTILIZADO

### 1.- El modelo como descripción interna del objeto real.

El desarrollo de un proceso de visualización, al consistir en operar con una realidad para obtener una imagen de la misma, debe iniciarse con el análisis de los objetos a visualizar, los cuales pueden ser todos o una parte de los existentes en el mundo real, y de acuerdo con las características de los mismos, en función de las limitaciones del proceso en estudio, será o no factible conseguir determinados tipos de visualización.

Dado que el presente trabajo presta especial interés por los temas arquitectónicos, serán estos y los que compartan sus características fundamentales, los que centrarán el área de estudio, debiendo definir las condiciones que permitan considerar un determinado objeto como apto o no para ser utilizado por el sistema de visualización desarrollado.

Serán pues los objetos sólidos, con una clara y exacta definición geométrica, unos límites espaciales precisos, y sin que quepa la menor ambigüedad respecto de ninguna de las características de los mismos usadas a lo largo del proceso de obtención de una figura bidimensional que permita su representación gráfica, el grupo formal en que nos moveremos.

## Parte I      Capítulo I

Pero por mucha precisión con que podamos conocer el objeto y sus características formales, la forma en que se encuentren definidos los datos puede no ser utilizable por el ordenador, es por tanto necesaria la traducción de la información de que dispongamos sobre el objeto en cuestión a una forma de expresión asimilable por él, que no entiende ningún tipo de información gráfica, ni puede manipular ninguna experiencia arquitectónica, sino tan solo datos de una restringida serie de tipos, valores numéricos y caracteres alfanuméricos esencialmente, estructurados de una manera específica, que permita una utilización eficiente de toda la información necesaria.

Por ello debemos recurrir al uso de modelos, es decir representaciones de una realidad que, al ser sometidas a determinadas transformaciones, tienen un comportamiento que permiten deducir el que tendría el objeto real al ser sometido a las transformaciones consiguientes.

Trabajaremos entonces con un modelo paralelo al objeto real que permita un tratamiento adecuado por parte del ordenador, lo que nos obligará a prescindir de algunos aspectos del objeto que, si bien puedan ser fundamentales en la experiencia perceptiva real del tema, no lo sean en su descripción formal, necesaria para proceder a la representación gráfica del mismo.

Pese a ello la visualización plana de temas arquitectónicos tridimensionales, es un instrumento de una riqueza y funcionalidad incuestionables, avalado por una dilatada utilización, por lo que facilitarla por medio de métodos informáticos, con la velocidad y precisión que les caracteriza es una tarea necesaria y enriquecedora, cuyos fundamentos interesa conocer a quien vaya a utilizarla en su actividad profesional.

## Parte I      Capítulo I

El modelo que utilizamos contempla con precisión y unicidad las características geométricas del objeto a visualizar. Precisión en tanto que las operaciones que realizaremos, para elaborar las distintas proyecciones del mismo, son de una eficacia más que dudosa si las posiciones absolutas y relativas de todos los elementos de trabajo no quedan perfectamente establecidas a lo largo de todo el proceso, sin sufrir variaciones significativas en la gran cantidad de operaciones matemáticas que se realizan, con los márgenes de error inevitables que, aunque de muy poca entidad, pueden llegar a adquirir valores significativos al acumularse en cada operación. Unicidad porque es fundamental que una vez conformado el modelo, este tenga una única réplica posible en el mundo real, y esta obviamente sea coincidente con el tema objeto de nuestro trabajo, así como el que un objeto real tenga una única manera de ser representado por el modelo elegido es necesario para evitar ambigüedades en el trabajo del sistema.

Consideraremos como propiedades que definen a un sólido rígido, y que por tanto deben poseer los objetos que tratemos de visualizar, las enumeradas por Aristides A. G. Requicha (ACM, Computing Surveys, Diciembre 1980):

a) Rigidez: la configuración de un sólido debe ser invariante, independientemente de su situación y orientación en el espacio.

b) Homogeneidad tridimensional: en un sólido debe existir un interior, y su frontera no puede contener partes aisladas o conectadas puntual o linealmente con el resto del cuerpo.

c) Finitud: un sólido debe ocupar una porción finita del espacio.

## Parte I Capítulo I

d) Cerrado a los movimientos rígidos y a ciertas operaciones booleanas: los movimientos rígidos (traslaciones y/o rotaciones) y las operaciones booleanas que agregan o eliminan materia deben, cuando son aplicadas a un cuerpo rígido, dar como resultado otro cuerpo rígido.

e) Descripción finita: los modelos de sólidos en un espacio tridimensional deben tener algunos aspectos finitos (ej. un número finito de caras) para permitir su representación en el ordenador.

f) Frontera definida: la frontera de un sólido debe determinar sin ambigüedad lo que es "interior" y por lo tanto comprendido en el sólido.

El utilizar únicamente objetos que cumplan las condiciones enumeradas no constituye en ninguna medida una limitación significativa del ámbito de aplicación del trabajo, ya que difícilmente pueden concebirse elementos arquitectónicos que no las cumplan.

Es preciso que al realizar transformaciones geométricas o posicionales al objeto en el mundo físico, tengamos unos instrumentos adecuados que nos permitan llevar a cabo los correspondientes cambios en el modelo, de forma que el comportamiento del mismo ante dichos procesos sea paralelo al del mundo real, lo que a su vez nos permitirá manipular el modelo simulando cambios en el objeto y visualizando posteriormente el resultado de los mismos, teniendo así conciencia de lo que los citados cambios representarían en el mundo real.

Para describir internamente en el ordenador un determinado objeto, existen diversas opciones, entre las que cabe mencionar como más utilizadas las siguientes:

a) el modelo más elemental para representar un objeto es el "modelo de alambre" que utiliza las aristas del mismo para su definición. Se puede considerar como ventaja la

rapidez de su procesado para la visualización bidimensional de un objeto tridimensional, sin embargo tiene el grave inconveniente de su ambigüedad, al poder referirse el mismo modelo a diversos cuerpos reales, no pudiendo además procesarse la eliminación de las líneas ocultas en la visualización del mismo.

b) el modelo conocido por las siglas C.S.G. (geometría constructiva de sólidos) se basa en representar un cuerpo como una secuencia estructurada de operaciones booleanas realizadas con unas pocas primitivas preexistentes. Es de una gran comodidad de manejo, así como presenta una muy interesante riqueza de posibilidades cara a la generación de sólidos tridimensionales y su fácil modificación, pero de momento necesita unas capacidades de memoria y una velocidad de procesado que exigen grandes equipos para una utilización razonable.

c) la representación mediante "octtrees" (árboles octales) que describen el objeto descomponiendo el espacio en celdas unitarias codificando con distintos valores las que pertenecen en su totalidad al cuerpo en cuestión y las que están totalmente fuera del mismo, procediendo a una nueva subdivisión de las que se encuentran parcialmente ocupadas por el cuerpo, y así hasta el límite de precisión necesario, dando lugar a una información que, organizada mediante una estructura en árbol, es eficientemente procesada por el ordenador.

d) el "modelo de fronteras", que es el que utilizaremos en el presente trabajo, describe el objeto definiendo con toda precisión su "frontera", mediante la descomposición de la misma en elementos más simples (vértices, aristas, polígonos y caras), necesitando una estructura de archivo formada por diversos grupos de datos interrelacionados, constandingo cada grupo de la información referente a uno de los elementos mencionados, incluyendo las necesarias referencias a los elementos de nivel inferior que lo configuran.

2.- El modelo de fronteras utilizado.

Debido a que nuestro interés se centra en la visualización de temas arquitectónicos, los aspectos que cuidaremos con especial atención serán los geométricos referidos a la forma, con el fin de permitir un funcionamiento más agil del equipo, pues considerar otros aspectos físicos del objeto (cálculo de masas, momentos de inercia, centros de gravedad, etc...), que podrían procesarse a partir de modelos muy similares al que utilizamos, incrementaría la complejidad del proceso, y debido a las limitaciones que imponen las prestaciones del equipo se ha creído que restaría eficacia al objetivo fundamental del trabajo.

El desarrollo del estudio se orienta a la utilización de ordenadores del tipo profesional de sobremesa, con las limitaciones de velocidad y memoria consiguientes, pero que son los que parece razonable suponer que utilizará el arquitecto en su actividad profesional privada, siendo de todas formas de validez general todo lo que se expone de aquí en adelante, aunque en el caso de disponerse de ordenadores mayores o más potentes podrían seguramente replantearse algunos aspectos de detalle, con el fin de optimizar el proceso al disponer de un instrumento con más altas prestaciones.

El objetivo perseguido es la representación gráfica bidimensional de objetos tridimensionales, o grupos de los mismos, con el mayor realismo posible, y la capacidad de repetir el proceso modificando total o parcialmente el tema considerado, su posición en el espacio o la situación del observador respecto del mismo, lo que permite la exploración visual de temas arquitectónicos, así como la

obtención de dibujos de las distintas visiones procesadas por el ordenador.

El modelo de fronteras permite el control preciso de la forma del objeto, así como las relaciones espaciales con otros objetos que conformen su entorno. Su almacenamiento en el ordenador exige una cantidad de memoria razonable, teniendo una estructura que, trasladada a la proyección bidimensional, permite actuar con independencia en cada uno de los elementos, tratando las caras con distintos tipos de acabado, trazando las aristas con líneas de características adecuadas a su significado (líneas de sección, ocultas, vistas a través de un medio transparente, etc...), y pudiendo dibujar o no determinados aspectos del tema mediante indicadores a tener en cuenta por el programa que controle el trazado del dibujo. Ha parecido pues adecuado desarrollar el procesado de la visualización a partir de un modelo de fronteras cuya estructura se detalla en el apartado siguiente.

### 3.- Estructura utilizada para desarrollar el modelo.

Toda la información que conforma el modelo se puede descomponer en unidades coherentes que se almacenan por separado en la memoria del ordenador, pero que a su vez están estrechamente relacionadas entre sí, por lo que podemos hablar de una estructura de datos vinculada al modelo, definiendo los distintos espacios de archivo para los mismos y las formas de conexión entre ellos.

Para ello procedemos al análisis del tipo de objetos que podrán ser utilizados para nuestros propósitos y las características de los mismos que deberá contemplar el modelo utilizado.

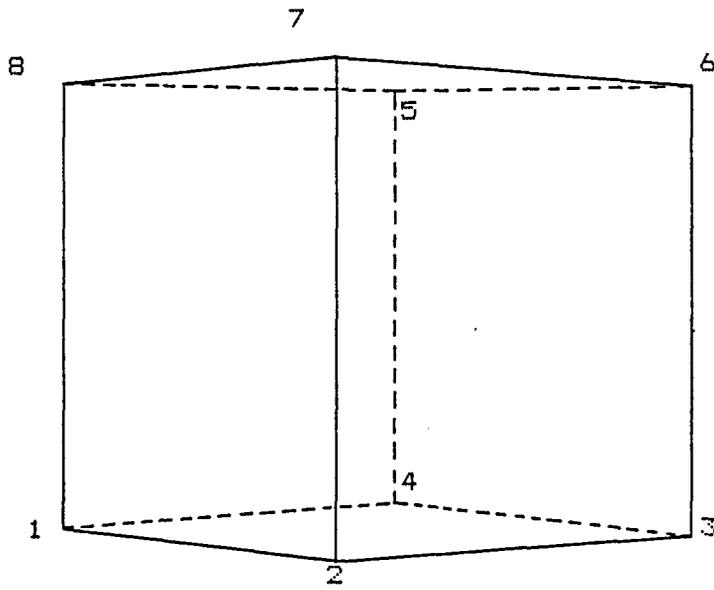
Van a ser utilizados por nuestro estudio todos los sólidos rígidos poliédricos, con las desgraciadamente inevitables limitaciones de complejidad y medida que imponga el equipo informático de que dispongamos a la hora de llevar a la práctica los planteamientos teóricos que se desarrollan a lo largo del trabajo.

Para permitir el procesado de las superficies no planas, las asimilaremos a superficies poliédricas, que se adaptaran lo máximo posible a ellas en el aspecto formal, con el fin de conseguir una visualización lo más cercana a la realidad que sea factible.

Los elementos más simples en que se puede descomponer un objeto poliédrico, de forma que permita una definición del mismo apta para nuestros fines, son las caras, las aristas y los vértices, y es esta descomposición la que, por el momento, utilizaremos para conformar el modelo con el que trabajar en el ordenador, aunque posteriormente incrementaremos la complejidad del mismo.

Veamos como ejemplo más sencillo los datos necesarios para definir de forma adecuada un simple paralelepípedo que se muestra en la figura I.1.3.1 y del que se desarrolla una estructura de datos de acuerdo a lo comentado en el párrafo anterior.

Los vértices son la unidad más elemental de información del objeto, y definen su posición en el espacio, sus dimensiones, etc..., pero no permiten conocer su forma, ni otras muchas características geométricas del mismo por si solos.



VERTICES

n.	Cx.	Cy.	Cz.
1	0	0	0
2	10	0	0
3	10	0	10
4	0	0	10
5	0	10	10
6	10	10	10
7	10	10	0
8	0	10	0

ARISTAS

n.	v1	v2
1	1	2
2	2	3
3	3	4
4	4	1
5	8	1
6	7	2
7	6	3
8	5	4
9	5	6
10	6	7
11	7	8
12	8	5

CARAS

n.	vért.
1	1, 4, 3, 2
2	3, 4, 5, 6
3	8, 7, 6, 5
4	1, 2, 7, 8
5	2, 3, 6, 7
6	1, 8, 5, 4

Figura I.I.3.1

Es preciso recurrir a la información referente a las aristas y las caras del objeto para considerarlo definido, no siendo suficiente la información de las aristas, pues no garantiza unívocamente la fidelidad del modelo a un único objeto sólido en el espacio.

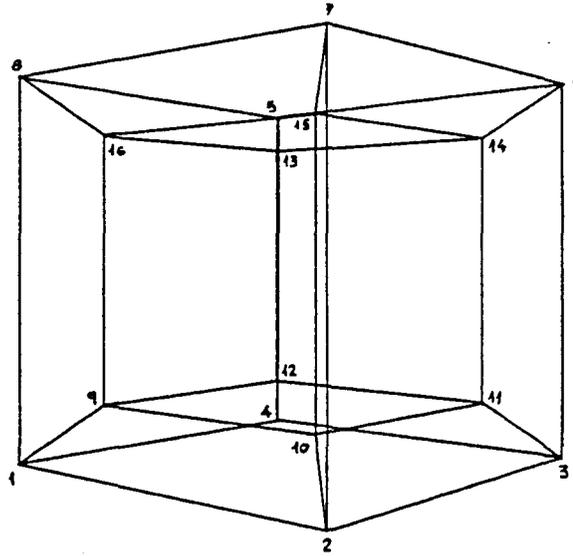
Puede observarse en el ejemplo mostrado por las figuras I.I.3.2 y I.I.3.3 como es imprescindible la información de las caras para diferenciar entre las diversas posibilidades que la figura de alambre ofrece para configurar un sólido en el espacio, pues los vértices y las aristas son coincidentes en ambos casos, como se observa en la primera figura.

Hasta ahora hemos contemplado objetos sin huecos ni agujeros que atraviesen sus caras, sin embargo sería una limitación inadmisibles no poder contemplar la existencia de caras con perforaciones en un tema arquitectónico, para ello debemos introducir el concepto de "polígono", diferenciado del de "cara", y considerar una cara como un conjunto de polígonos, cada uno de los cuales debe pertenecer a una de las siguientes categorías:

a) El polígono que forma la frontera exterior de la cara, y al que son interiores todos los demás polígonos de la misma, al cual denominaremos "polígono frontera" y será el primero en la lista de polígonos de la cara. Debe existir siempre, y es el utilizado para determinar algunas características de la cara (concavidad, ecuación, color, etc...).

b) Los polígonos que delimitan los agujeros que atraviesan la cara. Su existencia depende evidentemente de la de huecos o perforaciones en el sólido, y de, si existen, tengan una de sus aberturas situada en la cara en cuestión.

Parte I Capítulo I



VERTICES

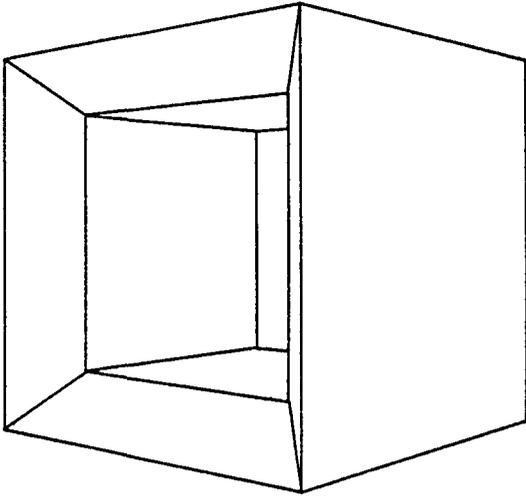
ARISTAS

n.	Cx.	Cy.	Cz.
1	0	0	0
2	100	0	0
3	100	0	100
4	0	0	100
5	0	100	100
6	100	100	100
7	100	100	0
8	0	100	0
9	20	20	20
10	80	20	20
11	80	20	80
12	20	20	80
13	20	80	80
14	80	80	80
15	80	80	20
16	20	80	20

n.	v1	v2
1	1	2
2	2	3
3	3	4
4	4	1
5	8	1
6	7	2
7	6	3
8	5	4
9	5	6
10	6	7
11	7	8
12	8	5
13	9	10
14	10	11
15	11	12
16	12	9

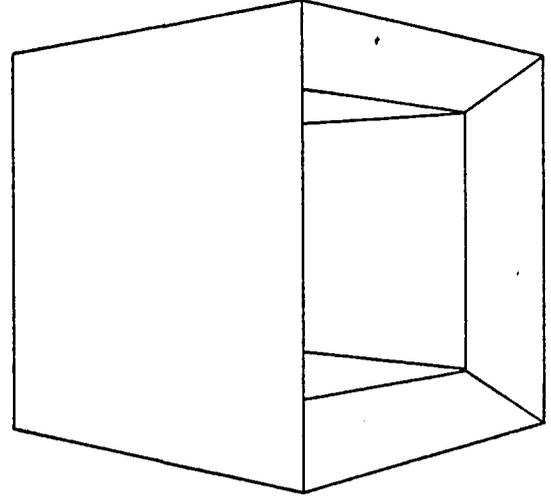
n.	v1	v2
17	16	9
18	15	10
19	14	11
20	13	12
21	13	14
22	14	15
23	15	16
24	16	13
25	8	16
26	5	13
27	6	14
28	7	15
29	1	9
30	2	10
31	3	11
32	4	12

Figura I.1.3.2



CARAS

n.	vert.
1	5, 8, 7, 6
2	2, 3, 6, 7
3	1, 4, 3, 2
4	1, 8, 5, 4
5	7, 8, 16, 15
6	2, 7, 15, 10
7	1, 2, 10, 9
8	1, 9, 16, 8
9	9, 12, 13, 16
10	9, 10, 11, 12
11	10, 15, 14, 11
12	13, 14, 15, 16
13	5, 6, 14, 13
14	4, 5, 13, 12
15	3, 11, 14, 16
16	3, 4, 12, 11



CARAS

n.	vert.
1	5, 8, 7, 6
2	3, 4, 5, 6
3	1, 4, 3, 2
4	1, 2, 7, 8
5	6, 7, 15, 14
6	3, 6, 14, 11
7	2, 3, 11, 10
8	2, 10, 15, 7
9	9, 16, 15, 10
10	9, 10, 11, 12
11	11, 14, 13, 12
12	13, 14, 15, 16
13	5, 13, 16, 8
14	1, 8, 16, 9
15	4, 12, 13, 5
16	1, 9, 12, 4

Figura I.I.3.3

c) Los polígonos que delimitan tratamientos superficiales parciales en la cara (grafismos, rótulos, etc...). Pueden o no existir y, existiendo, pueden o no ser visualizados.

El sentido de lectura de un polígono frontera es el antihorario, visto desde el exterior del sólido, en las zonas convexas de la cara que delimita. En el caso de polígonos pertenecientes a agujeros en la cara, el sentido de lectura es el contrario, siendo indiferente en los del tercer tipo, por no tener influencia en la configuración del sólido, sino tan solo en su aspecto externo.

Si ordenamos jerárquicamente los distintos elementos que hemos descrito, considerando un elemento superior a otro cuando lo contiene, tendremos en sentido descendente, el objeto, la cara, el polígono, la arista y el vértice, aunque según la estructura de datos que utilicemos no será necesario tenerlos todos en archivos independientes.

La información necesaria de cada uno de estos elementos para poder considerarlo suficientemente definido es:

- Del vértice necesitamos conocer sus coordenadas en el espacio.
- De la arista los dos vértices que la limitan.
- De cada polígono las aristas que lo limitan, ordenadas secuencialmente según el orden de lectura que corresponda al tipo de polígono de que se trate.
- De la cara, la lista completa de los polígonos que la forman, siendo el primero el polígono frontera y conociendo el tipo de cada uno de ellos.
- Del objeto, la lista de caras que lo limitan.

No es imprescindible tener almacenada toda la información anterior, pues conociendo algunos elementos pueden procesarse los demás cuando sea preciso, debiendo

considerarse que cuanto más información se almacene, de una forma más acentuada aumentará la velocidad del proceso, pero mayor será la memoria necesaria para contener la estructura de datos del modelo.

#### 4.- Almacenamiento de la información del modelo.

Quando se almacena en el ordenador la información concerniente a un elemento determinado, debe decidirse si es conveniente incorporar junto con la información puramente descriptiva del mismo, otros datos de los que pueda convenir su utilización en el proceso de visualización y no es deseable calcular cada vez que se precisan. Existe además otro tipo de información que posibilita el manejo de la estructura de datos y que por tanto debe ser incluida. Se analiza a continuación la información que se debe conocer de cada uno de los elementos del modelo de acuerdo con la estructura de datos que se utiliza.

El vértice es el elemento inferior del modelo, consta de tres coordenadas, su número de identificación, que normalmente coincide con su posición en el archivo, y por tanto no suele incluirse como información independiente, lo cual es común al resto de elementos del modelo.

De la arista precisamos conocer sus dos puntos extremos, pero al ser leídas secuencialmente como contorno de un polígono, solo necesitamos saber su origen, pues su otro extremo será el origen de la siguiente arista, y en el caso de la última, el origen de la primera del polígono. Para prever posibles modificaciones del archivo lo organizamos como lista encadenada, incorporándose la dirección de la siguiente arista en cada caso, salvo en el de la última del polígono, en que figura la dirección de la

primera con signo negativo para detectar el fin del polígono. Puede a veces ser conveniente conocer el polígono a que pertenece la arista, con lo que se incluiría como tercer campo de información; esto permite la utilización de la estructura de datos en sentido tanto ascendente como descendente, pero en nuestro caso debe valorarse el incremento en la ocupación de memoria.

En el caso del polígono ocupamos los dos primeros campos con la dirección de la primera arista y el número de las mismas, aunque podría prescindirse de esta última información, por la forma de configurarse el archivo de aristas, debido a motivos de seguridad la incluimos. El tipo de polígono y otros posibles atributos (texto, color, visibilidad, etc...) serán necesarios si se consideran modelos del tercer tipo, en cuyo caso figurará una dirección referida a un archivo independiente donde se encontrará dicha información; este campo puede ser utilizado en el caso del polígono frontera (el primero) para determinar su color, caso de ser posible el tratamiento cromático. Como en el caso de las aristas, necesitamos otro campo que contenga la dirección del siguiente polígono, utilizándose de forma similar. A la información referente a la cara a que pertenece el polígono son aplicables las reflexiones hechas en el párrafo anterior para la arista.

La cara ocupa sus dos primeros campos con la información referente al primero de sus polígonos y el número de los mismos. También contendrá la dirección de la siguiente cara y opcionalmente la del objeto a que pertenece. Puede incorporarse como información propia de la cara, su condición de cóncava o convexa y su ecuación en el espacio. Las consideraciones respecto a la ocupación de memoria son las mismas que se han realizado para los anteriores elementos.

## Parte I Capítulo I

Considerando como tema de trabajo la visualización de un conjunto de objetos sólidos, se precisará un archivo de objetos, en el que cada uno tendrá la dirección de la primera de sus caras y el número de ellas, la dirección del siguiente objeto y un campo de control que permita decidir su visibilidad o atributos que afecten al objeto en su totalidad.

En la figura I.I.4.1 se muestra un esquema que refleja gráficamente lo antedicho, en el cual quedan reflejados, no solo los contenidos de cada registro de los distintos archivos, sino también las conexiones entre ellos.

Caso de trabajar en un sistema que permita conformar un tema a partir de primitivas preexistentes, cada objeto puede ser una primitiva con información referente a las transformaciones que hace falta realizar para su incorporación al modelo, lo cual permite una agil utilización del sistema, pero esto será desarrollado en la segunda parte de este trabajo.

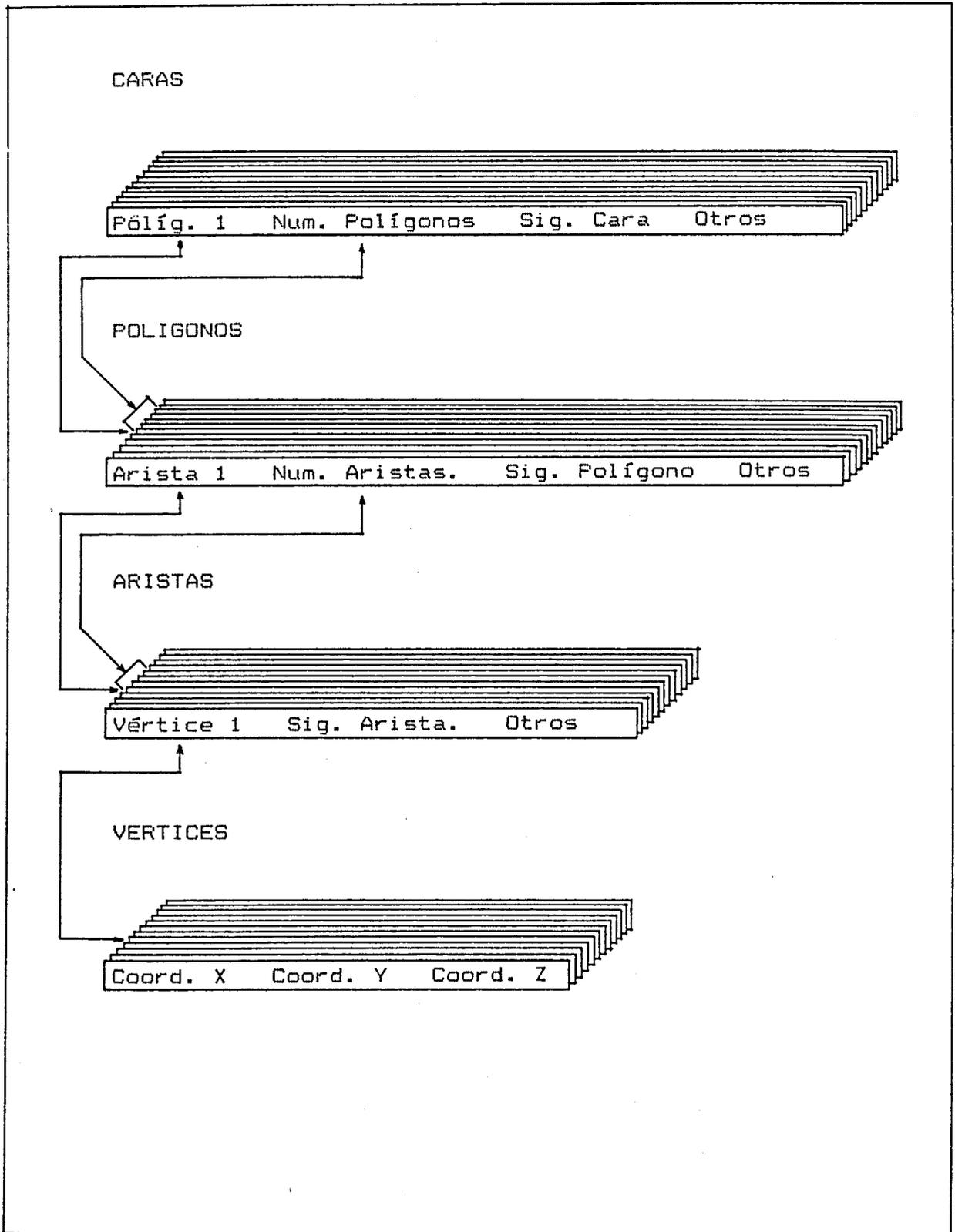


Figura I.I.4.1

## 2. TRANSFORMACIONES EN TRES DIMENSIONES

### 1.- Análisis geométrico de las transformaciones.

Consideramos que aplicar una transformación a un cuerpo consiste en realizar un determinado tipo de operación con el conjunto de sus vértices para obtener otro distinto, con el mismo número de vértices y sin afectar al resto de su estructura de datos, salvo en el caso especial de la simetría, en que cambia el sentido de lectura de los polígonos.

Las primeras transformaciones que utilizamos son la traslación y el giro, que desplazan el objeto alterando únicamente su posición en el espacio, no afectando por tanto a su forma, ni a las relaciones de sus elementos entre si, sino únicamente a sus relaciones espaciales con otros objetos que puedan formar parte del tema o a su posición con respecto al observador.

El escalado y la cizalladura por el contrario afectan a las dimensiones e incluso a la forma del cuerpo, variando en el caso de la cizalladura los valores angulares propios del objeto, por lo que son en realidad deformaciones aplicadas al mismo.

Estas cuatro transformaciones se aplican unicamente a los vértices del objeto, ubicandolos en otro lugar del espacio, pero sin alterar las relaciones entre ellos, reflejadas en la estructura de datos y que definen el sólido.

La quinta transformación es la simetría, la cual no solo afecta a la posición de los vértices en el espacio, sino al sentido de lectura de los polígonos, que debe ser cambiado al efectuar la transformación.

## 2.- Los operadores matriciales de las transformaciones.

La aplicación de las transformaciones al modelo exige una herramienta que permita su actuación de una forma rápida y flexible, la cual se encuentra en el calculo matricial, facilitando extraordinariamente el proceso, como veremos a continuación, dada su perfecta adaptación al tratamiento con el ordenador.

Aunque al trabajar en el espacio tridimensional, cada vértice viene definido por tres valores numéricos, que reflejan su posición de acuerdo al sistema de coordenadas elegido, nosotros utilizaremos coordenadas homogeneas, por lo que incorporaremos una cuarta coordenada a las anteriores.

La verdadera posición en el espacio de cada punto viene definida por los cocientes de las tres primeras coordenadas por la cuarta, por lo que la primera consecuencia que observamos es la utilidad de esta última coordenada como factor de escala, pues sustituyendo este valor por su mitad, a modo de ejemplo, automaticamente hemos doblado el valor de las verdaderas coordenadas del punto, sin afectar a los valores numéricos almacenados. No

será este, sin embargo, el aspecto en que utilizaremos fundamentalmente las posibilidades que nos ofrece trabajar en coordenadas homogneas.

Llamaremos operador de una transformación a la matriz cuadrada de orden cuatro que, debido a los valores numéricos de sus elementos, permita mediante su multiplicación con el vector-columna formado con las coordenadas homogneas de un punto, obtener como resultado el vector-columna de las del punto transformado.

Consideremos la matriz tipo

$$\begin{vmatrix} M(1,1) & M(1,2) & M(1,3) & M(1,4) \\ M(2,1) & M(2,2) & M(2,3) & M(2,4) \\ M(3,1) & M(3,2) & M(3,3) & M(3,4) \\ M(4,1) & M(4,2) & M(4,3) & M(4,4) \end{vmatrix}$$

y estudiemos el significado que tiene cada uno de sus elementos al ser utilizada la misma como operador de una transformación, en función de los efectos que produce sobre la figura a que se aplica.

Para empezar observemos que la matriz identidad, utilizada como operador, equivale a la transformación identidad, es decir aquella que deja invariante el punto al que se aplica.

Aplicaremos la transformación colocando en primer lugar la matriz operador, y en segundo como columna, el vector formado por los valores de las coordenadas homogneas del punto a transformar, en el que en general se habran hecho las operaciones necesarias para que el cuarto

valor sea la unidad. El resultado del producto será el punto transformado, con los valores de sus coordenadas homogeneas colocados en forma de columna.

Estudiemos ahora por separado cada una de las transformaciones enumeradas al principio de este capitulo:

Traslación.

Si deseamos desplazar en el espacio el sólido objeto de nuestro estudio, según un recorrido definido por el vector de coordenadas (Tx,Ty,Tz), la operación necesaria consistirá en sumar a los valores de las coordenadas reales (con la cuarta igual a la unidad) de todos los vértices el correspondiente a la traslación en su eje de referencia, lo cual es equivalente al producto matricial con el operador correspondiente.

Para obtener este operador partimos de la matriz identidad, y sustituimos los tres primeros valores de la cuarta columna, por los tres valores que corresponden a las tres componentes del vector que define la traslación segun cada uno de los ejes coordenados, quedando la operación matricial de la forma

$$\begin{vmatrix} 1 & 0 & 0 & Tx \\ 0 & 1 & 0 & Ty \\ 0 & 0 & 1 & Tz \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix} = \begin{vmatrix} x + Tx \\ y + Ty \\ z + Tz \\ 1 \end{vmatrix}$$

Observamos pues que el resultado de la operación se corresponde con los valores previstos, al incrementar el valor de cada una de las coordenadas reales del punto en la cantidad correspondiente a la traslación deseada.

Rotación.

La rotación alrededor de un eje en el espacio, es una transformación cuyo operador no tiene un cálculo inmediato, salvo en el caso de que el eje de giro sea uno de los tres ejes coordenados, para el cual se desarrollan a continuación las matrices de los operadores correspondientes.

Rotación alrededor del eje X con un ángulo de giro A en sentido horario mirando al origen desde un punto del semieje positivo.

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos A & -\operatorname{sen} A & 0 \\ 0 & \operatorname{sen} A & \cos A & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Rotación en torno al eje Y de las mismas características.

$$\begin{vmatrix} \cos A & 0 & \operatorname{sen} A & 0 \\ 0 & 1 & 0 & 0 \\ -\operatorname{sen} A & 0 & \cos A & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Igualmente para el eje Z.

$$\begin{vmatrix} \cos A & -\operatorname{sen} A & 0 & 0 \\ \operatorname{sen} A & \cos A & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Observamos fácilmente que estos operadores dejan invariantes los valores de las coordenadas correspondientes al eje de giro.

Al final del capítulo se estudia el proceso para determinar el operador de una rotación en torno a un eje cualquiera.

Escalado.

Entendemos por escalado la transformación cuyo efecto es la multiplicación de las dimensiones del objeto por un factor numérico; en general consideraremos la posibilidad de utilizar un factor distinto para cada eje coordenado, lo cual producirá un cambio en las proporciones relativas del cuerpo.

Suponiendo que los factores de escala para los ejes X, Y y Z sean  $E_x$ ,  $E_y$ , y  $E_z$  respectivamente, el operador asociado a esta transformación será

$$\begin{vmatrix} E_x & 0 & 0 & 0 \\ 0 & E_y & 0 & 0 \\ 0 & 0 & E_z & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Si deseamos un incremento homogéneo de las dimensiones del cuerpo, es decir el mismo en las tres direcciones, y de valor  $F$ , al utilizar coordenadas homogéneas son equivalentes los siguientes operadores

$$\begin{vmatrix} F & 0 & 0 & 0 \\ 0 & F & 0 & 0 \\ 0 & 0 & F & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

o bien, utilizando las propiedades, comentadas anteriormente, que las coordenadas homogéneas tienen, en lo que afecta a los valores de las coordenadas absolutas del

punto en el espacio

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/F \end{vmatrix}$$

Pudiendo por tanto utilizarse el cuarto elemento de la diagonal principal como factor de escala general.

#### Cizalladura

La cizalladura es una deformación global del cuerpo, en que los puntos ven variadas cada una de sus coordenadas en función de las otras dos, y de unas constantes prefijadas, distintas para cada combinación de coordenadas.

Dicha variación consiste en incrementar el valor de cada coordenada del punto con la suma de las otras dos, previa multiplicación de cada una de ellas por el factor correspondiente.

La ubicación de cada uno de los factores de cizalladura en la matriz operador de esta transformación, viene determinada por la coordenada transformada que fija la fila, y por aquella de que es factor que fija la columna, es decir que si deseamos variar la coordenada I, sumándole el producto de la coordenada J por un factor K, haremos igual a K el elemento M(I,J) de la matriz, consideremos como ejemplo el caso en que deseamos incrementar la segunda coordenada de cada uno de los puntos de un sólido en un valor igual al triple de la primera coordenada más la mitad de la tercera, para conseguirlo deberemos configurar el siguiente operador

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0.5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

La cizalladura es una transformación compleja de utilizar, pero muy útil e insustituible en determinados procesos de generación de formas a partir de primitivas más sencillas.

### Simetría

Utilizaremos la simetría especular respecto de uno de los tres planos del triedro trirectángulo que usamos como sistema de coordenadas, es fácilmente observable que se mantendrán invariantes las coordenadas referentes a los ejes contenidos en el plano, y cambiarán de signo las correspondientes al eje perpendicular, por lo que el operador de esta transformación será la matriz unidad, con signo negativo en el elemento de la diagonal principal que corresponde al eje perpendicular al plano de simetría.

Sirva de ejemplo el operador correspondiente a una simetría respecto al plano XZ

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

en el que se aprecia fácilmente que, al ser operado con cualquier vértice, producirá el único efecto de cambiar de signo su segunda coordenada.

3.- Concatenación de transformaciones.

Aunque las transformaciones vistas hasta el momento permiten la manipulación de sólidos en el espacio, definen actuaciones excesivamente simples, por lo que por si solas serian de poca utilidad. Es mediante la aplicación secuencial de un conjunto de las mismas cuando se obtiene un instrumento suficientemente rico y versátil para nuestros propósitos.

Es importante destacar la circunstancia de la no conmutatividad de las transformaciones que estudiamos, salvo casos muy particulares, como el de tratarse tan solo de traslaciones, lo cual si permitiría la aplicación en distinto orden de las mismas, sin que ello afectase al resultado final. Un ejemplo elemental constituido por una traslación en la dirección del eje X ( $T_x$ ), y un giro alrededor del eje Y ( $G_y$ ), se puede observar en la figura I.II.3.1 en la que se reflejan los resultados de la aplicación en distinto orden de las transformaciones antedichas. Sin embargo si tienen la propiedad asociativa, es decir, puede aplicarse cualquier grupo de transformaciones, teniendo el mismo efecto que si se aplicaran las mismas por separado, siempre que no se altere el orden naturalmente.

El producto matricial es también asociativo, pero no conmutativo, lo que establece un claro paralelo entre el comportamiento del sólido en el espacio ante una secuencia de transformaciones y el del proceso de cálculo que traslada al modelo estas transformaciones, al ser realizado este proceso mediante el producto matricial, asociando a cada transformación o grupo de ellas un operador matricial.

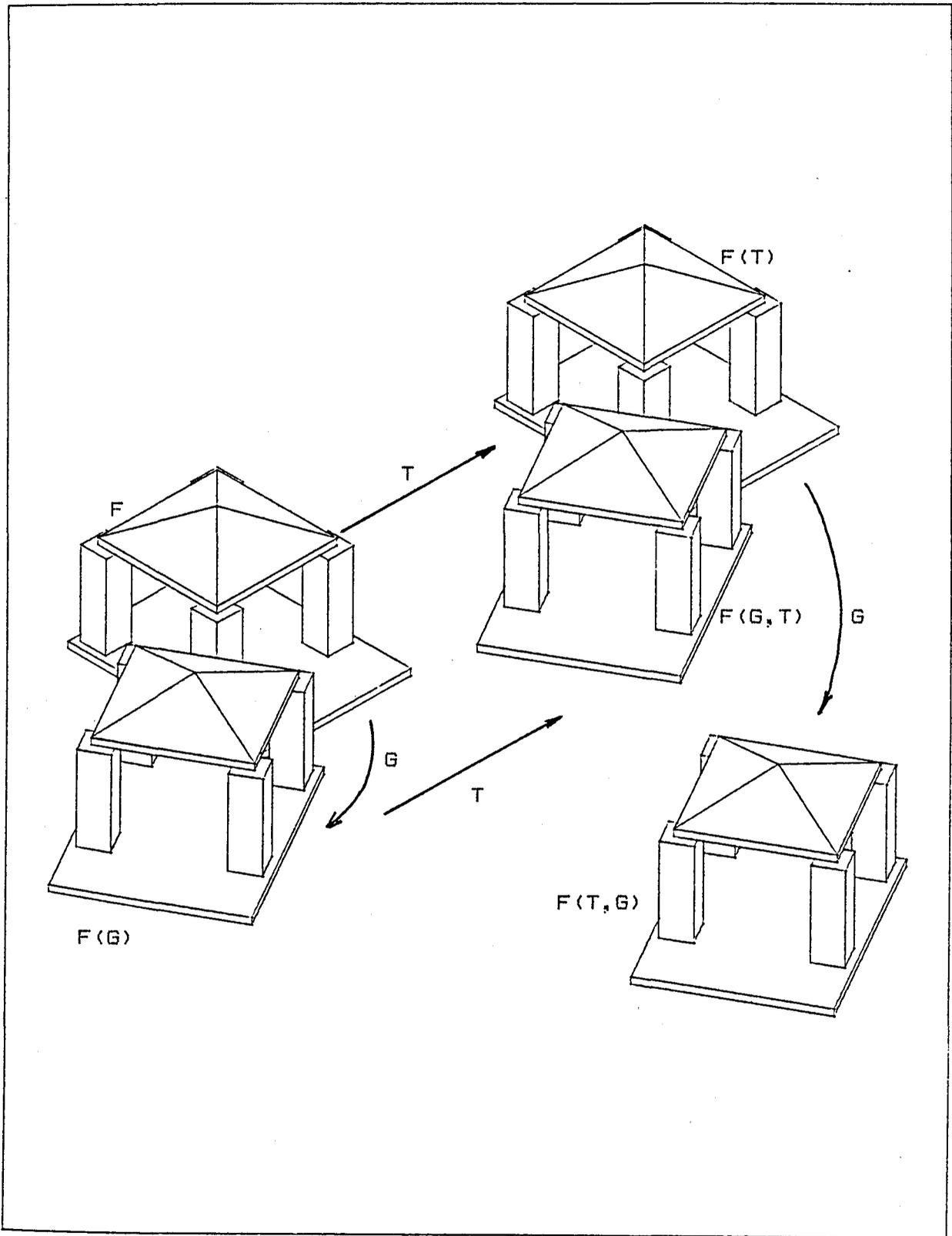


Figura I.II.3.1

Para rentabilizar el hecho de la asociatividad del producto matricial, al aplicar una secuencia determinada de transformaciones a un sólido, calcularemos primero los correspondientes operadores, pero no serán aplicados consecutivamente, sino que los agruparemos en uno solo, el cual será el que se operará con los valores de las coordenadas de los vértices del objeto, contenidas en el archivo correspondiente del modelo, lo cual no es sino aprovechar la propiedad asociativa del producto matricial para agilizar el proceso de cálculo.

En virtud de la no conmutatividad debemos aplicar los operadores sin alteración alguna en el orden que corresponde a las transformaciones que representan, por lo que, debido al tipo de producto matricial que utilizamos con los vectores (colocandolos posteriormente al operador), debemos colocar las distintas matrices en el orden inverso al de su aplicación, es decir comenzando por el operador que corresponda a la última transformación y terminando con el de la inicial, que por tanto será la contigua al vector-columna y la primera en operar. A continuación multiplicaremos todas las matrices-operador, obteniendo así una única matriz como resultado del producto, que será la utilizada para operar con ella todos los vértices del sólido.

#### 4.- Dos ejemplos significativos.

Vamos a estudiar la utilización de lo descrito en el apartado anterior, desarrollando dos casos prácticos muy comunes, como son, el giro alrededor de un eje cualquiera, y la realización de un cambio de sistema de coordenadas.

Para definir una rotación en torno a un eje, necesitamos determinar el eje, lo que haremos mediante las coordenadas de dos de sus puntos, y el ángulo de giro así como su sentido, que será el horario mirando desde el segundo punto al primero.

La estrategia para descomponer esta transformación en otras más sencillas, de las que conocemos directamente sus operadores, consiste en realizar la secuencia de transformaciones necesaria para hacer coincidir el eje con uno de los coordenados, efectuar la rotación, y retornar a la posición inicial mediante la aplicación de las inversas de las transformaciones, lo que devolverá al eje de giro a su posición inicial.

Siguiendo la obra de Newman y Sproull (Principle of Interactive Computer Graphics), veamos como se realiza esta operación, considerando  $(X_1, Y_1, Z_1)$  el primer punto del eje de rotación, y  $(X_2, Y_2, Z_2)$  el segundo, así como  $A$  el ángulo que deseamos girar. El vector dirección del eje será por tanto  $(XV, YV, ZV)$  donde  $XV = X_2 - X_1$  y similarmente las restantes coordenadas.

a) Realizamos una traslación de forma que el eje de giro pase por el origen, pudiendo suponer que su primer punto quede coincidente con el origen, lo cual se consigue mediante la utilización del siguiente operador

$$\begin{vmatrix} 1 & 0 & 0 & -X_1 \\ 0 & 1 & 0 & -Y_1 \\ 0 & 0 & 1 & -Z_1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

al que denominaremos T.

b) Giramos alrededor del eje X para abatir el segundo punto sobre el plano XZ, el ángulo de giro será igual al  $\text{Arctg}(YV / ZV)$ , por lo que, llamando K a la raíz cuadrada de la suma de cuadrados de YV y ZV tendremos el operador

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & ZV/K & -YV/K & 0 \\ 0 & YV/K & ZV/K & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

al que denominaremos R1.

c) Ahora, mediante una rotación en torno al eje Y, hacemos coincidir con el eje Z el de la rotación en estudio, llamando M al módulo del vector dirección del mismo, obtenemos el operador

$$\begin{vmatrix} K/M & 0 & -XV/M & 0 \\ 0 & 1 & 0 & 0 \\ XV/M & 0 & K/M & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

al que denominaremos R2.

d) Procedemos a realizar el giro previsto de ángulo A, alrededor del eje, teniendo como operador

$$\begin{vmatrix} \cos A & -\text{sen } A & 0 & 0 \\ \text{sen } A & \cos A & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

al que denominaremos G.

e) Mediante las transformaciones inversas a  $R_2, R_1$  y  $T$ , aplicadas en este orden, recolocamos en su posición inicial al eje de giro, ocupando de esta forma todos los puntos del espacio sus nuevas posiciones giradas un ángulo  $A$  con respecto al eje mencionado. En la figura I.II.4.1 se puede seguir gráficamente el proceso descrito, de forma que, dado el sistema de referencia formado por los ejes coordenados  $X, Y$ , y  $Z$ , y el eje de giro  $(a_0, b_0)$ , se procede a trasladar el eje mediante la traslación  $T$  a la posición definida por  $(a_1, b_1)$ , a continuación se le aplica la rotación  $R_1$  en torno al eje  $X$ , situándose en  $(a_1, b_2)$ , consiguiéndose la ubicación deseada  $(a_1, b_3)$  coincidente con el eje  $Z$ , mediante el giro  $R_2$ . A continuación se realizaría el giro deseado en torno al eje  $Z$  (actualmente el eje de giro) y se repetiría el proceso en sentido inverso.

Para analizar un ejemplo de cambio de sistema de coordenadas, vamos a elegir uno algo peculiar pero de gran interés para nosotros, pues será utilizado más adelante al hablar de la realización de proyecciones. Vamos a suponer que deseamos pasar del actual sistema de coordenadas a otro cuyo origen este en el punto  $(O_x, O_y, O_z)$ , cuyo eje  $Z$  pase por el punto  $(Z_x, Z_y, Z_z)$  y en el que las rectas paralelas al eje  $Y$  se transformen en otras paralelas al plano  $YZ$ . Llamaremos  $(D_x, D_y, D_z)$  al vector diferencia  $(O_x, O_y, O_z) - (Z_x, Z_y, Z_z)$ , y  $M$  al módulo del mismo.

La estrategia seguida, similar a la desarrollada por Angell y Jones (Advanced Graphics ...), se basa en concatenar las transformaciones necesarias para hacer coincidir los dos sistemas de ejes, para lo cual realizamos los pasos siguientes:

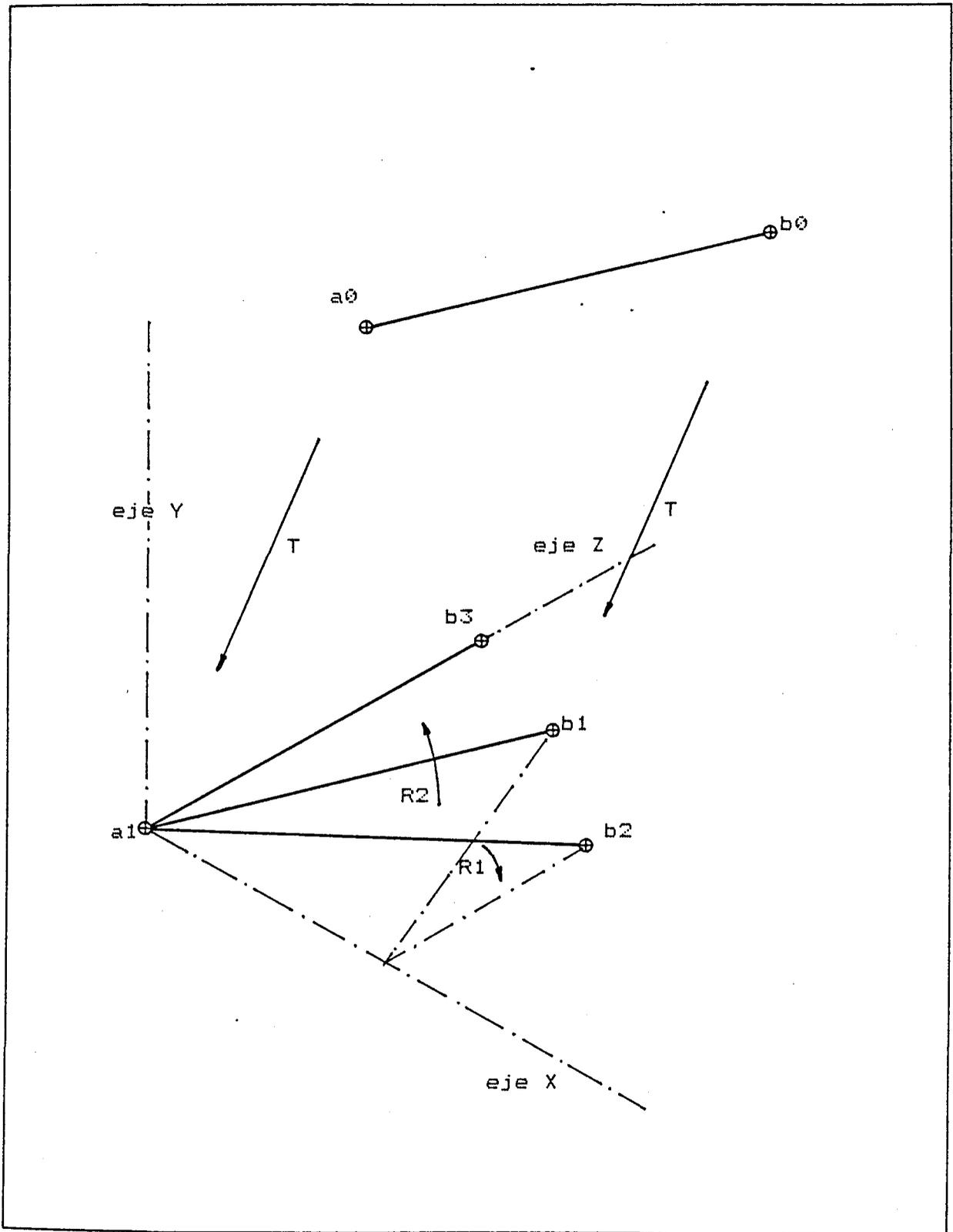


Figura I.II.4.1

a) Una traslación que sitúe el punto  $(Z_x, Z_y, Z_z)$  en el origen, siendo

$$\begin{vmatrix} 1 & 0 & 0 & -Z_x \\ 0 & 1 & 0 & -Z_y \\ 0 & 0 & 1 & -Z_z \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

el operador preciso para la realización de la misma.

b) Una rotación alrededor del eje Z de un ángulo igual a  $-\text{Arctg}(D_y / D_x)$ , de la que el operador será

$$\begin{vmatrix} D_x/K & D_y/K & 0 & 0 \\ -D_y/K & D_x/K & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

en que K representa la raíz cuadrada de la suma de cuadrados de  $D_x$  y  $D_y$ .

c) Una rotación de un ángulo igual al complementario del  $\text{Arctg}(K / D_z)$  en torno al eje Y cuyo operador será

$$\begin{vmatrix} -D_z/M & 0 & K & 0 \\ 0 & 1 & 0 & 0 \\ -K & 0 & -D_z/M & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

d) Debemos ahora conseguir que las direcciones paralelas al eje Y queden paralelas al plano YZ, para lo cual es necesario un giro alrededor del eje Z de ángulo igual a  $\text{Arctg}(-D_y * D_z / M * D_x)$ , cuyo operador es

$$\begin{vmatrix} Dx * M / J & Dy * Dz / J & 0 & 0 \\ -Dy * Dz / J & Dx * M / J & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Donde J es la raíz cuadrada de la suma de los cuadrados de (Dy \* Dz) y de (M \* Dx).

e) En este momento tenemos el nuevo eje Z coincidente con el antiguo, pero el origen lo ocupa el punto (Zx,Zy,Zz), en lugar del (Ox,Oy,Oz), por lo que debemos realizar una traslación de valor M a lo largo del eje Z para situar el punto (Ox,Oy,Oz) en su lugar, teniendo como operador para dicha traslación

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & M \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Agrupando todos los operadores descritos mediante la multiplicación de los mismos en orden inverso al de su aplicación, se obtiene una matriz que usaremos como operador para transformar las coordenadas de todos los puntos del espacio al nuevo sistema de referencia. En el capítulo siguiente veremos la aplicación de este peculiar cambio de sistema de coordenadas.

### 3. PROYECCIONES

#### 1.- Concepto geométrico de proyección.

La proyección es una transformación que convierte una figura en otra de una dimensión menos, lo cual en nuestro caso se traducirá en la obtención de una figura de dos dimensiones a partir de otra de tres.

Los elementos fundamentales que intervienen en los tipos de proyección que estudiamos son:

a) El espacio tridimensional en el cual se sitúan los sólidos a visualizar, así como el observador.

b) El plano en el cual proyectamos, es decir el plano del cuadro sobre el que se forma el dibujo resultado del proceso de visualización.

c) El rayo proyectante, que se determina a partir de la situación del observador junto con el tipo de proyección (cónica o cilíndrica), necesitándose además la dirección de la visual para definir con exactitud la vista que obtendremos del objeto.

Consideraremos el plano de proyección perpendicular al rayo proyectante, con el fin de simplificar el proceso, y por considerar que en las representaciones de elementos y espacios arquitectónicos es lo más frecuente, por ser lo más similar al funcionamiento de la visión natural. Asimismo consideraremos que el observador se mantiene vertical, es decir que en las proyecciones en que el rayo proyectante sea horizontal, se mantendrá la verticalidad.

2.- Proyecciones paralelas y cónicas.

La representación de figuras tridimensionales en un medio bidimensional ha sido extensamente utilizada a lo largo de la historia, dando lugar, las distintas formas de realizarla, a dos grandes familias de proyecciones perspectivas: las paralelas y las cónicas.

La proyección paralela tiene dos elementos clave, la dirección del rayo proyectante y la posición del plano de proyección. Nosotros consideraremos el caso en que la dirección de la normal al plano de proyección coincide con la del rayo proyectante, dando lugar a las distintas axonometrías, así como al sistema diédrico, dependiendo de la relación entre la orientación de la figura proyectada y la dirección del rayo proyectante.

El concepto de la proyección paralela es muy claro, se obtiene la proyección de cada punto del sólido haciendo pasar por el mismo una paralela al rayo proyectante, determinando a continuación la intersección de la misma con el plano de proyección, siendo este punto la proyección del primero. El proceso geométrico para su construcción, consiste en proyectar los vértices del objeto y unir estas proyecciones mediante segmentos de recta, en la misma forma en que los vértices están unidos mediante aristas, con lo que las líneas así obtenidas en el plano de proyección son las proyecciones de las aristas del objeto, quedando configurada de esta forma la visualización del mismo en el espacio bidimensional de representación.

Si lo que deseamos obtener es una proyección que ofrezca una representación más cercana a la visión real del objeto, utilizaremos la proyección cónica, que guarda una gran semejanza (aunque no identidad) con la que un

observador tendría en el mundo real. Aquí nos encontramos con la necesidad de definir tres elementos para configurar la proyección, la situación del observador que consideraremos un punto en el espacio al que llamaremos "punto de vista", y que actuará como centro de proyección; deberemos conocer también la dirección en que el observador está mirando, que determinaremos mediante un segundo punto al que llamaremos de "dirección de la visual" y que, unido con el punto de vista, forma la recta que denominaremos "visual principal", la cual es el eje del cono formado por el conjunto de líneas que surgen del punto de vista en dirección al objeto; necesitamos finalmente conocer el plano de proyección, que en este trabajo consideraremos (aunque no deba serlo necesariamente) perpendicular a la visual principal, intersectandose con ella en el punto de dirección de la visual.

A fin de establecer una relación entre este tipo de transformación y la visión real de un observador en una situación normal, nos falta determinar que el tipo de visualización que estudiaremos mediante la proyección cónica orientará el plano de proyección al proceder a su representación, de forma que las aristas que sean verticales en el sistema de referencia tridimensional, lo sigan siendo en el bidimensional que utilizamos para dibujar el resultado, siempre que la visual principal sea horizontal.

### 3.- Desarrollo de los operadores matriciales.

Para desarrollar estas proyecciones mediante el uso del ordenador, debemos estudiar el proceso necesario que permita actuar eficientemente sobre el modelo, de forma que la obtención de los datos de la proyección resultante se realice de forma ágil y precisa.

Llamaremos "coordenadas universales" a aquellas en que consideramos definido el espacio real, y por tanto en las que podemos establecer las relaciones entre los distintos elementos de la proyección (punto de vista, visual principal, plano de proyección, etc...) con el objeto del que deseamos obtener una visualización.

Utilizaremos un sistema de referencia en que el proceso de proyección se simplifica, y por lo tanto en el que se consigue optimizar el trabajo del ordenador. Utilizaremos para este sistema la denominación de "coordenadas visuales".

En el caso de la proyección paralela las características que le exigiremos al sistema de coordenadas visuales serán dos, que el eje Z coincida con la dirección del rayo proyectante, con lo que el plano de proyección será el XY, o cualquier otro paralelo, y como segunda condición que las aristas que en coordenadas universales son paralelas al eje Y (verticales) se mantengan en coordenadas visuales paralelas al plano YZ, con lo que su proyección en el plano XY, o cualquier otro paralelo, mediante un rayo paralelo al eje Z, será paralela al eje Y.

Para calcular un operador que realice la transformación equivalente a una proyección paralela, realizaremos un cambio de coordenadas universales a visuales de acuerdo con lo desarrollado en el ejemplo correspondiente del capítulo anterior.

Consideramos el punto  $(V_x, V_y, V_z)$  como aquel del que parte un rayo proyectante que pasa por el origen, entonces definimos el nuevo sistema de coordenadas como aquel cuyo origen está en  $(V_x, V_y, V_z)$ , y consideramos que el punto  $(Z_x, Z_y, Z_z)$ , de acuerdo con la nomenclatura del ejemplo

citado, es el actual origen de coordenadas universales, con lo cual tenemos como nuevo sistema de referencia el de coordenadas visuales.

Obtenemos el operador siguiendo los pasos del ejemplo y lo multiplicamos con todos los vértices del modelo, dado que la cuarta coordenada será igual a la unidad, si consideramos como plano de proyección el XY, obtenemos los puntos proyectados sin más que ignorar la tercera coordenada visual, y considerar las otras dos referidas a un sistema de referencia bidimensional con ejes de coordenadas coincidentes con los X e Y del sistema de coordenadas visuales.

Utilizando el resto de la información contenida en el modelo, referida a los puntos proyectados como vértices, podemos construir el dibujo de la imagen proyectada del sólido tridimensional, visualizando de momento la totalidad de las aristas del mismo.

En el caso de la proyección cónica las coordenadas visuales serán aquellas en que el punto de vista coincida con el origen, la visual principal con el eje Z en el sentido positivo, es decir que el semiespacio de coordenada Z positiva sea el situado frente al observador; el plano de proyección será paralelo al XY y a una distancia determinada por la posición del punto de dirección de la visual y, como en el caso de la proyección paralela se vincula el paralelismo con el eje Y de coordenadas universales con el paralelismo con el plano YZ de coordenadas visuales.

El proceso de obtención del operador para la proyección cónica es similar al de la proyección paralela, con el nuevo origen de coordenadas en el punto de vista, y el punto  $(Z_x, Z_y, Z_z)$  el de dirección de la visual; el plano

de proyección es el  $(0,0,1,-M)$  siendo  $M$  el modulo del vector cuyos extremos son el punto de vista y el de dirección de la visual, es decir el plano paralelo al  $XY$  que pasa por el punto de dirección de la visual (de ecuación " $Z - M = 0$ ").

Se multiplican todos los vértices del modelo por el operador así calculado, obteniendo de esta forma las coordenadas en el nuevo sistema de referencia. Ahora necesitamos generar la proyección sobre el plano elegido. El proceso de proyección, como ya se comentó, consiste en unir cada vértice con el punto de vista y determinar la intersección de la recta así formada con el plano de proyección, siendo éste el punto proyectado del vértice en cuestión. En la figura I.III.3.1 se puede apreciar el significado geométrico de esta operación y las operaciones necesarias para determinar los valores aritméticos de las coordenadas del punto proyección de un vértice determinado.

Vemos pues, una vez determinadas las coordenadas visuales del objeto, se realiza el proceso de obtención de las coordenadas de la proyección del mismo en el sistema de referencia que se forma sobre el plano del cuadro, con centro en el punto de dirección de la visual y teniendo como ejes  $X$  e  $Y$  los paralelos a sus homólogos de coordenadas visuales en dicho plano, sistema al que llamaremos de "coordenadas planas". Este proceso consiste únicamente en el cálculo aritmético de las nuevas coordenadas planas, para cada uno de los vértices del objeto a proyectar.

Esta última transformación la consideramos realizada con independencia del operador matricial del cambio de coordenadas universales a visuales, pues puede interesar para procesos que veremos más adelante tener por separado los archivos de vértices en coordenadas visuales, y de

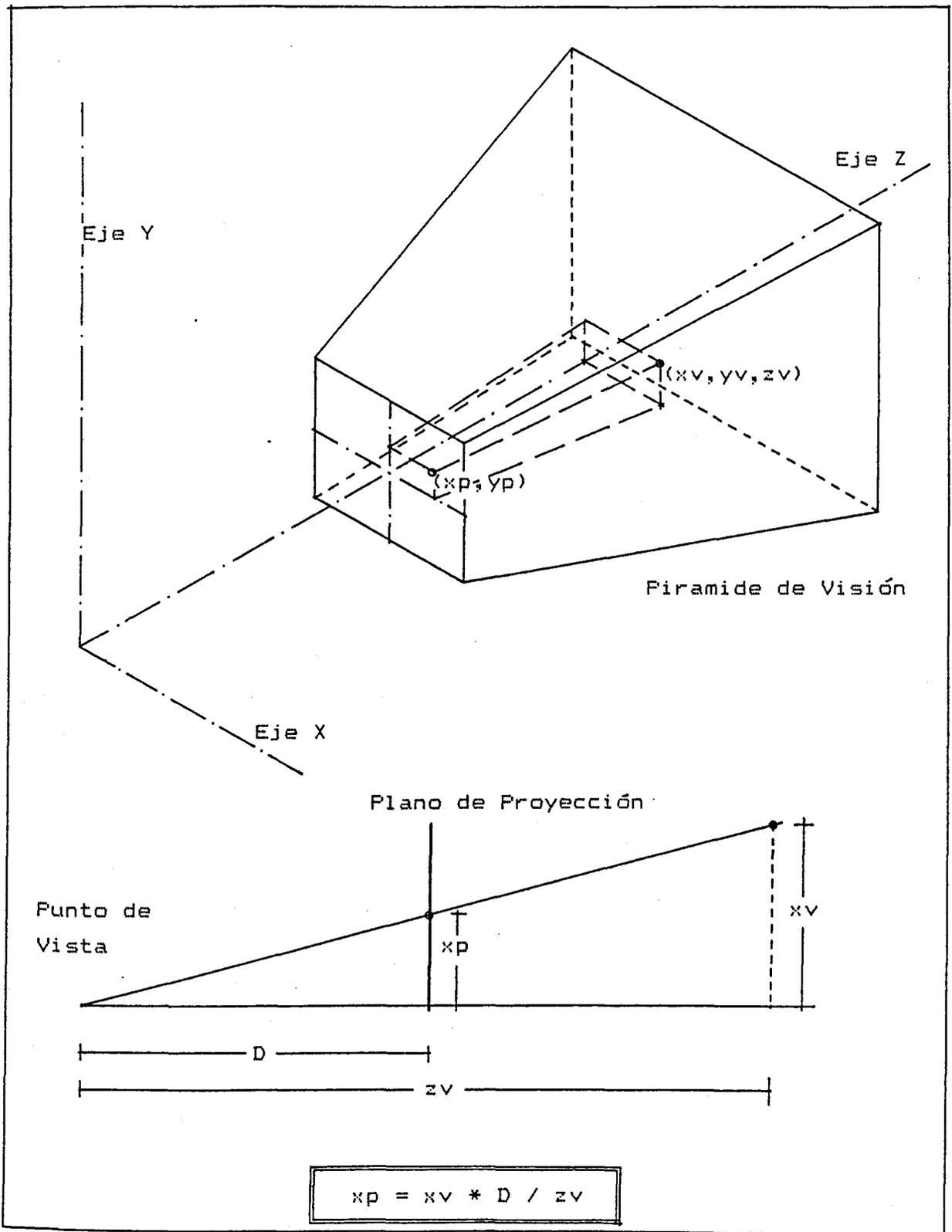


Figura I.III.3.1

puntos en coordenadas proyectadas.

Analizando la proyección cónica, y de forma similar en la proyección paralela, vemos que es preciso eliminar del proceso de proyección la parte del sólido que queda en el semiespacio posterior al punto de vista, pues el proceso que realiza el ordenador haría aparecer dentro de la figura proyectada todos los puntos, y elementos por ellos conformados, incluso aquellos que serían proyección de los que, estando situados tras el observador, al unirse con el punto de vista, dan lugar a una recta que intersecta al plano de proyección dentro de los límites de nuestra zona de trabajo, creando imágenes virtuales que confunden la visualización obtenida.

Puede parecer inmediato que una solución sea ignorar todos aquellos vértices cuya tercera coordenada visual sea negativa, pero esto representaría crear un grave problema con las aristas que tienen uno de sus extremos en estas condiciones, y el otro dentro de la zona visible. Para resolver éste y otros aspectos de características similares, desarrollaremos en el siguiente capítulo el concepto de recorte (clipping).

#### 4. Transformación de la pirámide de visión.

En la proyección cónica podemos considerar como único objeto de nuestro interés la porción de espacio comprendido dentro de los límites de visualización que fijamos para cada proceso, los cuales, como se describirá más adelante, suelen configurar un poliedro con forma de tronco de pirámide de base cuadrada, vértice de la misma en el punto de vista, y eje coincidente con la visual principal, al que llamaremos "pirámide de visión", cuyo significado se aprecia claramente en la figura I.III.3.1, en la cual se

Parte I      Capítulo III

reflejan, como ya hemos mencionado, las operaciones aritméticas a realizar para obtener las coordenadas planas de la proyección.

Simplifica el trabajo poder tratar una proyección cónica como si se tratara de una paralela, sobre todo en los procesos de depuración de líneas y partes ocultas, al tener de una forma mucho más directa la relación entre las coordenadas visuales de cada uno de los puntos en el espacio tridimensional, y las coordenadas planas de su correspondiente proyección.

Para ello, según se demuestra en la obra de Newman y Sproull (Principle of Interactive Computer Graphics), puede realizarse, una vez realizado el cambio a coordenadas visuales, la transformación a que da lugar el siguiente operador matricial:

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 / D(1-D/F) & -1 / (1-D/F) \\ 0 & 0 & 1 / D & 0 \end{vmatrix}$$

en donde D representa la distancia al origen del límite más próximo de la visualización, y F la distancia al origen del límite más lejano, quedando así limitada la profundidad del campo visual que utilizamos en la proyección que se procesa.

Queda demostrada en la obra mencionada la linealidad de esta transformación, es decir que las rectas y los planos de la figura en coordenadas visuales, se mantienen como tales en el nuevo sistema, que además transforma la pirámide visual en un paralelepípedo de aristas paralelas a los ejes de coordenadas.

Mediante esta transformación quedan las dos primeras coordenadas de cada punto en el sistema tridimensional creado, iguales a los valores que hubieran resultado de calcular la proyección cónica de acuerdo con las fórmulas utilizadas en el apartado anterior, y las terceras coordenadas quedan limitadas a valores comprendidos entre cero, para los puntos del plano límite cercano al punto de vista, y uno para los puntos situados en el plano que delimita la máxima profundidad de visualización. Podemos pues ahora proyectar este conjunto de puntos mediante una proyección paralela a la dirección de la visual, y obtendremos la figura equivalente a la que se obtendría mediante el procedimiento descrito en el apartado anterior, con la ventaja de que las referencias entre las coordenadas de un punto cualquiera de la proyección y las de aquel que se proyecta en él, son mucho más directas.

#### 4. RECORTES (CLIPPING)

##### 1. Concepto de recorte (clipping).

Si nos atenemos a lo desarrollado hasta el momento, al realizar una proyección cualquiera aplicando los operadores correspondientes, dicha proyección afectará a todos los puntos del espacio, independientemente de su localización. Como ya mencionamos, esto puede hacer inútil el proceso, al afectar a puntos situados tras el observador, y para evitarlo deberemos considerar como tema a visualizar, no el sólido en su totalidad, sino el resultado de sustraer al objeto original todas las partes que no nos interesa proyectar, por su situación fuera del campo de visión considerado.

Al aplicar el operador determinado según el procedimiento descrito en el capítulo anterior, en realidad realizamos un cambio de coordenadas que afecta a todos los puntos del espacio, pero al realizar la proyección del sólido sobre el plano, tenemos la necesidad de delimitar qué parte del espacio será afectada por dicha proyección. La configuración de la zona del espacio considerada visible se realiza mediante planos que recortan la figura dejando como resultado el fragmento de la misma destinado a ser utilizado en los procesos posteriores.

Trataremos de la proyección cónica en las reflexiones que siguen, aunque casi todo lo que comentemos será de aplicación, con mínimas variaciones, en general de simplificación, a la proyección paralela.

El motivo más común para que partes del tema no sean tomadas en consideración, y por lo tanto deban ser "recortadas", consiste en considerar a dichas partes como no visibles, bien por encontrarse situadas tras el observador, o bien por estar fuera de los límites de su visión, dichos límites vendrán fijados por la abertura que deseamos suponer al ángulo de visión, lo que determina un cono visual (pirámide visual en nuestro caso, al delimitarse la abertura visual mediante planos); también puede interesar prescindir de las partes del tema que, estando frente al observador y dentro de la pirámide visual, se encuentren excesivamente próximas o demasiado lejanas, delimitando así la profundidad de campo, con lo que finalmente obtenemos un tronco de pirámide que delimita la porción de tema a ser procesada.

Esta pirámide que delimita la zona a visualizar queda definida al cuantificar el ángulo de visión y las distancias mínima y máxima al punto de vista que deseamos considerar. Constituyen las caras de la pirámide de visión los cuatro planos que, conteniendo al eje X dos de ellos y al eje Y los otros dos, forman con la visual principal un ángulo igual al de visión. Los otros dos planos que conforman los límites del volumen a visualizar, son los planos paralelos al XY, en el semiespacio de Z positiva, que están situados a unas distancias del origen iguales a los límites superior e inferior de la profundidad del campo visual, cerrando así el volumen a considerar como visible.

Pueden existir otras circunstancias que aconsejen limitar la visualización de forma que no abarque la totalidad del tema, como pueden ser la simulación de observarlo a través de una abertura, o el deseo de representar únicamente una parte significativa del mismo. También es útil poder determinar una zona de la proyección para representarla con una ampliación que permita apreciarla con mayor detalle.

Existen diversas formas de plantear la operación descrita anteriormente, dando lugar a distintos tipos de recorte, en el espacio tridimensional, por ejemplo, podemos recortar los elementos de dos dimensiones, caras y polígonos, reconstruyendo la información referente a los mismos, y obteniendo por tanto un nuevo conjunto de datos que constituyan el modelo de fronteras completo de una figura equivalente al objeto recortado por un plano. Debido a la complejidad de este tipo de recorte en el espacio, en determinadas circunstancias puede no ser preciso disponer de la información sobre las caras recortadas, en cuyo caso será suficiente con recortar las aristas cara a obtener la visualización de los fragmentos de las mismas que se encuentran dentro del campo visual. Consideramos también el interés en realizar recortes limitados al plano que, a partir de la delimitación de una zona de la imagen proyectada, procedan a eliminar los segmentos que no pertenezcan al área considerada, con la intención de trazar posteriormente el dibujo ampliado de la zona en cuestión.

Estudiamos a continuación los algoritmos desarrollados para la ejecución de los tres tipos de recorte mencionados, los realizados en el espacio tridimensional con caras y con aristas, y el que en un dibujo bidimensional recorta los trazos del mismo, analizándolos en orden inverso al expuesto.

2. Algoritmo de recorte bidimensional de trazos.

El recorte de trazos sobre un plano es el más sencillo de los tres que se utilizan en este trabajo, debiendo establecerse como datos de partida, la figura proyectada de la que deseamos visualizar un fragmento, y el polígono utilizado para delimitar ese fragmento, al que denominaremos "marco" del recorte.

Consideramos la figura definida por un archivo secuencial de trazos, el cual es leído ordenadamente del primero al último, dando lugar a otro archivo en que quedan almacenados los trazos interiores al marco, y los fragmentos de los que lo atraviesan una vez recortados y eliminada la parte exterior de los mismos. La información que utilizamos de los trazos será la formada por las coordenadas de sus dos extremos, en el sistema de referencia bidimensional del plano de proyección, y que para cada línea llamaremos  $(LX1,LY1)$  y  $(LX2,LY2)$ . Una vez determinado el trazo a representar por estar dentro de la zona visible, se conocerán las coordenadas  $(TX1,TY1)$  y  $(TX2,TY2)$  que delimitan el mismo, según se puede apreciar en la figura I.IV.2.1 con mayor claridad.

El polígono que delimita la zona a representar gráficamente, lo consideramos como un rectángulo de lados paralelos a los ejes de coordenadas, llamando VI y VS a los valores inferior y superior de las ordenadas de los puntos en que las líneas que contienen a los lados horizontales cortan al eje de ordenadas, así como HI y HS a sus homologos respecto de los lados verticales.

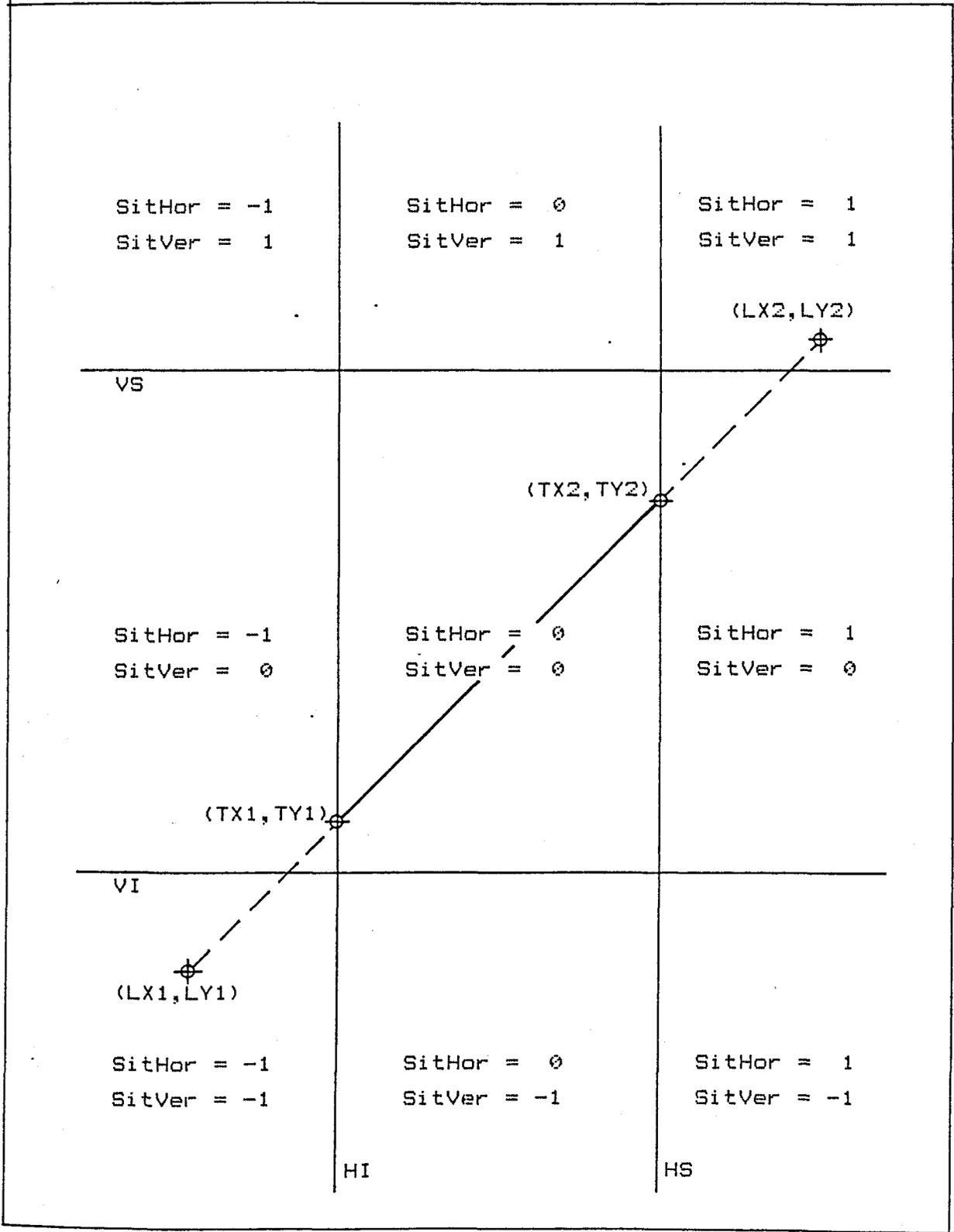


Figura I.IV.2.1

Analizamos ahora las operaciones a realizar para proceder al recorte de una línea cualquiera, teniendo en cuenta todos los casos que pueden presentarse, y utilizando la nomenclatura reflejada en la figura I.IV.2.1 a fin de seguir mejor los distintos pasos en que descomponemos el proceso.

Dividimos el plano en tres franjas horizontales delimitadas por los lados del rectángulo, realizando la misma operación verticalmente. Nos servimos de estas franjas para ubicar cualquier punto del plano, utilizando dos variables que adjudicamos a cada punto, llamandolas SitHor y SitVer, adquiriendo los siguientes valores: si la abcisa del punto es mayor que HS entonces SitHor vale 1, si es menor que HI entonces el valor de SitHor es -1, asignandole 0 en el caso de ser igual a uno de los valores limite, o estar comprendido entre ambos; el mismo criterio se utiliza para asignar valores a SitVer con la ordenada del punto. Tendremos así a SitHor y SitVer como dos pares de valores que determinan la situación, horizontal y vertical respectivamente, de los extremos de la línea.

A continuación analizamos los pasos necesarios para efectuar el algoritmo que permite realizar el recorte en estudio. Llamando línea al elemento de la figura a recortar, que obtenemos del archivo de la proyección de la misma, determinamos si genera un trazo dentro de la zona visible y, caso de hacerlo, que segmento de la misma forma este trazo.

a) Determinamos los valores de SitHor y de SitVer para cada uno de los extremos de la línea, considerando que si los dos valores de SitHor, o bien los dos de SitVer, son iguales entre si y distintos de cero, entonces ningún punto de la línea pertenece a la zona visible, por lo que esta línea no genera trazo alguno a graficar, y pasamos a

considerar la línea situada a continuación en el archivo de la figura.

b) Procedemos a recortar la línea en el sentido horizontal, de acuerdo con los límites impuestos HI y HS, sabiendo que los dos valores de SitHor deben ser distintos, salvo que sean ambos iguales a cero, veamos los tres casos que pueden presentarse en función del valor de SitHor para el primer punto de la línea (LX1,LY1):

- Si este valor es -1, entonces la línea corta al primer lado vertical del marco, procedemos al cálculo de esta intersección, y el punto determinado por la misma pasa a constituir el primer extremo de la línea, siendo su abscisa igual a HI y su ordenada la obtenida en el cálculo de la intersección, sustituyendo los valores iniciales de LX1 y de LY1 por los actuales. Si el valor de SitHor del segundo punto de la línea es 1, entonces debemos calcular la intersección con el segundo lado vertical del marco, procediendo de forma similar a lo explicado para el primero, obtenemos unos nuevos valores de LX2 y LY2. Recalculamos los nuevos valores de SitVer para el o los extremos que en el proceso descrito tengan una ordenada distinta a la inicial; si en esta nueva asignación de valores a SitVer pasan a ser iguales entre si y distintos de cero los de ambos extremos, interrumpimos el proceso al no generarse ningún trazo dentro del marco de visión, y lo reiniciamos con la siguiente línea.

- En el caso de ser cero el valor de SitHor del primer punto, operamos en función de la situación del segundo, el cual intersectará con el primer lado vertical del marco si su valor asignado en SitHor es -1, y con el segundo lado vertical si es 1. Una vez calculada esta intersección (caso de ser procedente) asignaremos los nuevos valores a las coordenadas del segundo extremo y, de forma similar al caso anterior, variaremos el valor de SitVer del mismo, y caso de coincidir con el del primer extremo siendo distinto

de cero, daremos por terminado el proceso de esta línea, como en el caso anterior.

- Por último consideramos el caso en que el valor de SitHor del primer punto sea igual a 1, procediendo entonces de forma similar a lo explicado en el primer caso, es decir que procesamos en primer lugar la intersección de la línea con el segundo lado vertical del marco, trasladando los valores de las coordenadas a LX1 y LY1, a continuación, si el valor de SitHor para el segundo punto vale -1, calculamos la intersección con el primer lado vertical, y el punto obtenido pasa a ser el (LX2,LY2). Recalculamos ahora los valores de SitVer para el o los extremos cuya ordenada sea distinta a la inicial, y en el caso de ser iguales entre si y distintos de cero, pasamos a procesar la siguiente línea, al no quedar ningún trazo comprendido dentro del marco de visión.

c) Caso de no haberse interrumpido el proceso en el apartado anterior, tenemos una línea, o segmento de la misma, incluida en la zona comprendida entre los valores HI y HS de las abscisas. Debemos ahora determinar la porción de la misma situada entre los lados horizontales del marco de visión. Para ello, siguiendo el mismo criterio que en el paso anterior, analizaremos los tres casos que pueden presentarse en función del valor de SitVer para el primer punto:

- Si este valor es -1 se debe hallar la intersección de la línea con el lado horizontal inferior del marco, adjudicándose los valores de las coordenadas de este punto a (LX1,LY1). Si el valor de SitVer para el segundo punto es 1, se recorta la línea con el lado horizontal superior del marco, determinándose así el punto (LX2,LY2).

- Cuando el valor de SitVer del primer punto sea cero, ya serán inalterables los valores de LX1 y LY1, actuando en función del valor de SitVer del segundo punto, intersectaremos la línea con el lado horizontal inferior del

marco caso de ser -1, o con el superior caso de ser 1 dicho valor, calculando así los nuevos valores para las coordenadas del punto (LX2,LY2).

- Finalmente analizamos las acciones a realizar cuando el valor de SitVer para el primer punto sea 1, en cuyo caso (LX1,LY1) será la intersección de la línea con el lado horizontal superior del marco de visión, procediéndose a continuación a comprobar si el valor de SitVer del segundo punto es -1, en cuyo caso la intersección de la línea con el lado horizontal inferior del marco nos dará los nuevos valores de LX2 y de LY2.

d) Podría darse la circunstancia de que el recorte de la línea redujera el segmento de la misma incluido dentro del marco de visión a un solo punto, lo que significaría que la línea pasaba por un ángulo del mismo sin atravesarlo, o bien tenía inicialmente uno de sus extremos en el marco de visión, y el resto de la línea era exterior al mismo. Debemos detectar si se presenta esta posibilidad comprobando si LX1 es igual a LX2, y así mismo LY1 lo es a LY2, considerando la igualdad, al trabajar en números reales, como una diferencia de valor absoluto menor que una cantidad preestablecida, en cuyo caso no consideraremos procedente aumentar en un trazo el archivo de trazos visibles, pasando a considerar la línea que se encuentre a continuación en el archivo de la figura.

e) Una vez llegados a este punto, si tenemos un trazo incluido dentro de la zona interior del marco de visión, delimitado por los puntos cuyas coordenadas son (LX1,LY1) y (LX2,LY2), cuyos valores solo serán los iniciales en el caso de que la línea objeto de recorte estuviese totalmente incluida dentro de la zona visible, igualaremos estos valores a (TX1,TY1) y (TX2,TY2), y los guardaremos en el archivo de la figura recortada a continuación del último trazo procesado.

Hemos visto el proceso de recorte de una línea, dando lugar, en su caso, a un trazo que representa el segmento de la misma incluido en la zona delimitada por nosotros como fragmento de la figura a visualizar. El proceso de recorte de la figura será pues la repetición secuencial del descrito, para todas y cada una de sus líneas, dando lugar a un archivo de trazado que permita la representación gráfica del resultado del recorte por medio del dispositivo adecuado.

### 3. Algoritmo de recorte tridimensional de aristas.

Después de haber estudiado como recortar una figura plana actuando secuencialmente sobre todas sus líneas, pasamos a analizar ahora un proceso similar, pero en este caso recortamos el objeto tridimensional, o mejor dicho, las aristas del mismo, obteniendo así una serie de segmentos de recta en el espacio que proyectados en el plano del cuadro, sin formar una estructura coherente que permita considerarla modelo de un sólido, generan un dibujo coincidente con la proyección del fragmento de objeto que se encuentra en el interior del poliedro que configuran los límites de recorte establecidos.

En el caso de que el poliedro que limita el recorte fuese un paralelepípedo, el proceso a realizar sería muy similar al desarrollado para el recorte en el plano, por lo que vamos a analizar el caso en el que dicho poliedro forma un tronco de pirámide con las características explicadas al principio del capítulo para la pirámide de visión.

Procedemos en primer lugar a definir los planos que conforman el poliedro límite del recorte, y a los que denominamos planos "límite" de la visión, estableciendo

además para ser usada a lo largo del proceso una variable lógica (boolean) denominada NoTrazo, la cual se inicializa para cada arista con valor "falso", y que adquiere valor "verdad" en el momento en que se detecta que la totalidad de la arista considerada está situada fuera de la pirámide de visión, interrumpiéndose en este punto el proceso de recorte de la arista y reiniciándose para la arista siguiente a considerar. De forma similar a lo descrito en el apartado anterior, realizamos la lectura secuencial del archivo correspondiente del modelo, obteniendo así los valores de las coordenadas de los dos extremos de la arista en el espacio, que llamaremos V1 y V2, siguiendo a continuación el procedimiento cuyos pasos se explican seguidamente.

a) Dado que la información contenida en el archivo de aristas consiste en la dirección de los extremos de las mismas, procedemos a obtener del archivo de vértices las coordenadas correspondientes a dichas direcciones, y asignamos valores a las tres componentes de V1 y V2, también inicializamos con el valor de "falso" a NoTrazo.

b) Procedemos, previa comprobación del valor "falso" de NoTrazo, a recortar la arista con los seis planos límite de la visión, haciéndolo por separado con cada uno de ellos, en un orden predefinido que, a modo de ejemplo, puede ser el siguiente: derecho, izquierdo, superior, inferior, anterior y posterior; antes de proceder a cada recorte comprobamos la posición del segmento de arista resultante hasta el momento, y si queda totalmente exterior al poliedro de recorte damos el valor "verdad" a NoTrazo, con lo que para esta arista se interrumpe el proceso, volviendo a iniciarse para tomar en consideración a la arista siguiente en el archivo del modelo. Analicemos el procedimiento mediante el cual realizamos el recorte de la arista con cada uno de los planos límite de visión:

- Consideramos el espacio dividido por el plano en dos zonas, siendo una de ellas aquella en la que se encuentra el resto del poliedro que delimita la zona de recorte, a la cual denominamos "visible", y procedemos a determinar en cual de ellas se encuentra cada uno de los extremos de la arista, pudiendo presentarse diversas situaciones que estudiamos por separado.

- Los dos extremos de la arista situados en la zona no visible, determinan la no existencia de puntos de la misma en el interior del poliedro que delimita el recorte, por lo que hacemos igual a "verdad" el valor de NoTrazo, lo cual interrumpe el proceso de esta arista, no generando ningún elemento a incluir en el archivo de segmentos de arista resultantes del recorte.

- Si los dos extremos se encuentran en la zona visible, ello significa que la arista no corta al plano que consideramos, por lo que pasamos a estudiar su recorte con el plano situado a continuación en el orden prefijado.

- Si  $V_1$  se encuentra en la zona visible, y  $V_2$  en la opuesta, procedemos a calcular la intersección de la arista, y sustituimos las coordenadas de  $V_2$  por las del punto así obtenido, pasando con los valores de las coordenadas de los nuevos extremos a determinar el recorte del segmento restante con el siguiente plano límite.

- En el caso inverso en que  $V_2$  se encuentra en la zona visible y  $V_1$  en la opuesta, el proceso es el mismo que en el anterior, pero sustituyendo con las coordenadas del punto de intersección los valores de las de  $V_1$ , prosiguiéndose en igual forma que en el caso anterior.

c) Comprobamos ahora que no se dé la circunstancia de coincidir, o diferir en una cantidad inferior al límite de precisión prefijado, las coordenadas de  $V_1$  y  $V_2$ , lo que significaría que el resultado del recorte de la arista la había dejado reducida a un solo punto, es decir que pasaba por un vértice del poliedro sin introducirse en él, o bien

partía de un punto de su superficie hacia el exterior del mismo. Caso de producirse la coincidencia de coordenadas, hacemos "verdad" el valor de NoTrazo e interrumpimos el proceso de esta arista.

d) Si al finalizar el recorte con los seis planos límite el valor de NoTrazo se mantiene como "falso" procedemos a incorporar un nuevo elemento en el archivo de aristas recortadas. Como la información que incluimos consiste en la dirección de las coordenadas de sus dos extremos en el archivo de vértices, entonces debemos comprobar si estos se han mantenido inalterados a lo largo del proceso del recorte, lo que significaría que la arista se encontraba en su totalidad en el interior del poliedro que limita el mismo; en caso contrario, procedemos a la inclusión en el archivo de vértices del o de los nuevos puntos aparecidos por causa del recorte, y son estas nuevas direcciones las que constituyen la información a guardar en el archivo de la nueva arista recortada.

Una vez concluido el proceso descrito, se encuentran recortadas todas y cada una de las aristas del objeto, dando lugar a un archivo formado por los fragmentos de las mismas comprendidos en el interior del poliedro formado por los seis planos límite del recorte. Proyectando estas aristas obtenemos un conjunto de trazos sobre el plano de proyección que representan exactamente el dibujo de la proyección del sólido obtenido al recortar el objeto por el poliedro mencionado.

#### 4. Algoritmo de recorte tridimensional de caras.

Dentro del proceso de visualización de un objeto, y no solamente por esta causa, puede presentarse la necesidad, o simplemente la conveniencia, de recortarlo con un plano,

obteniendo como resultado una estructura de datos similar a la de partida, que incluya toda la información referente al sólido resultado del recorte, es decir que al recortar una cara se consiga la información completa correspondiente a la nueva, o las nuevas caras formadas, y no solamente, como en el recorte de aristas estudiado en el apartado anterior, la del resultado de recortar sus aristas.

El algoritmo que permite realizar esta operación, es de una mayor complejidad que los desarrollados anteriormente en este capítulo pues, a partir de la estructura de datos del objeto a ser recortado, debe proceder a construir otra similar, que puede superponerse, situarse a continuación de la primera, o bien configurar una estructura completamente nueva que debe contener el resultado del recorte, utilizando para el proceso una estructura de datos temporal, en la cual se guardan los fragmentos parciales del recorte de una cara, para ser posteriormente reagrupados, generando así una o más caras a incorporar al modelo del sólido resultado del proceso.

Utilizamos para el recorte un plano paralelo al XY, situado a una distancia del mismo que denominamos "límite", estando normalmente, necesariamente en todo recorte motivado por un proceso de visualización, situado en el semiespacio positivo de las Z, pudiendo ser un plano coincidente con el que actúa como límite anterior de la visión, que utilizabamos en el apartado anterior al hablar de la determinación del poliedro para el recorte tridimensional de aristas. Por tanto todo punto cuya tercera coordenada sea menor que el límite, será considerado como no incluido en el resultado del recorte; los que la tengan igual, o diferente en un valor menor al límite de precisión prefijado, se encontrarán en el plano de corte; y los que tengan mayor su tercera coordenada se encontrarán incluidos en el objeto resultante de la

operación de recorte que efectuamos.

De forma similar a lo realizado en los algoritmos descritos anteriormente, obtenemos del archivo de la figura a recortar, la información referente al elemento objeto del recorte, en este caso la cara, mediante una lectura secuencial y completa de dicho archivo; al ser la cara el elemento a recortar, la información obtenida nos remitirá al archivo de polígonos, a fin de obtener todos los que forman dicha cara, los cuales a su vez precisarán la utilización del archivo de aristas para la definición de su perímetro, el cual solo tendrá una ubicación concreta en el espacio previa consulta en el archivo de vértices; vemos pues la necesidad de manejar la estructura de datos del modelo en su totalidad, y no parcialmente como en los casos estudiados anteriormente.

Analizamos a continuación el algoritmo que permite realizar el recorte tridimensional de caras, mediante el desarrollo de los pasos necesarios para efectuar el recorte de una cara, y obtener así el resultado del mismo, que puede consistir en una o varias caras, creando la estructura de datos necesaria para realizar la operación, así como para conservar el resultado. Llamaremos "anterior" a la posición de un punto, o cualquier elemento más complejo, que esté situado en el semiespacio de coordenada  $Z$  menor que el límite, es decir, en la zona desechada por el recorte; llamaremos entonces "posterior" a la posición opuesta, es decir con una tercera coordenada mayor que el límite, y por tanto en la zona que contiene al resultado de la operación de recorte.

a) Con la información existente en el archivo de caras respecto de aquella que vamos a recortar, procedemos a obtener la correspondiente a los vértices de su polígono frontera, analizando la posición de cada uno de ellos

respecto del plano de corte, pudiendo darse el caso de que ninguno de los mismos se encuentre situado en la zona posterior, supuesto en el cual no se incorporaría esta cara a la estructura de datos que recoge el resultado del recorte, y se procedería a reiniciar el proceso con la siguiente cara.

b) En el supuesto de que al menos un vértice del polígono frontera se encuentre en la zona posterior, se comprueba la posibilidad de que lo estén todos, con lo que, al no estar ningún punto de dicha frontera en la zona anterior, no es preciso realizar ninguna intersección con el plano de corte, y se traslada la cara en su totalidad a los archivos que almacenan toda la información referente al resultado del recorte, procediéndose a reiniciar el proceso desde el principio con la cara que se encuentra a continuación en el archivo de la figura a recortar. En el caso contrario, es decir, cuando se da la circunstancia de que existen vértices de la cara situados a ambos lados del plano de corte, se continua el proceso para determinar que parte de la cara en estudio queda en la zona posterior al plano que delimita el recorte, y configurar la información referente a la misma.

c) El estudio del corte lo realizamos en el plano XZ para todas las caras no paralelas al eje Y, detectando si la cara que estudiamos lo es, en cuyo caso procedemos al análisis del recorte en el plano YZ. Para ello creamos una variable de control, que indique la coordenada a utilizar en el proceso junto con la tercera, por lo que se le asigna el valor de 1 en el caso de que el valor absoluto de la segunda componente del vector normal a la cara sea mayor que el límite de precisión prefijado, asignándole un valor igual a 2 en caso contrario. Con esto se evita analizar la cara proyectada de perfil, lo que haría inútil el proceso.

d) Introducimos en este momento del proceso el concepto de "quebrada", llamando así a un nuevo elemento de información del modelo, intermedio entre la arista y el polígono, siendo su significado geométrico el de un conjunto de aristas coplanarias, colocadas ordenadamente, de forma que el primer vértice de cada una de ellas es el segundo de la anterior, a excepción de la primera arista, cuyo primer vértice no pertenece a ninguna otra de la quebrada, y de la última, cuyo segundo vértice tampoco pertenece a ninguna otra arista de la quebrada. Supongamos que operamos con la proyección "en planta" sobre el plano XZ de la cara que analizamos, y que procedemos a su intersección con la traza del plano de recorte sobre este mismo plano XZ, eliminando los fragmentos de la figura que quedan situados en el semiplano correspondiente al espacio anterior al plano de corte, tendremos como resultado una serie de quebradas, y tal vez algún polígono completo que, perteneciendo a la cara recortada, haya quedado situado en su totalidad en la zona del plano cuyos puntos tienen la tercera coordenada mayor que el límite.

e) Procedemos pues a determinar, y almacenar en el archivo temporal abierto a tal efecto, la información correspondiente a cada una de las quebradas que, como resultado del corte realizado, son fragmentos de los polígonos de la cara que poseían vértices a ambos lados del plano límite. Aquellos polígonos cuyos vértices se encuentren en su totalidad situados en la zona posterior al plano de corte se almacenan en un archivo temporal independiente, para ser incorporados posteriormente a la nueva cara resultante del corte en que se encuentren incluidos. Para definir una quebrada es necesario conocer los vértices que contiene, así como el orden en que están situados, lo que nos permitirá determinar sus aristas, y el sentido de lectura correspondiente a la misma que, como

veremos, es fundamental a la hora de reconstruir las caras resultantes del recorte. Veamos ahora el proceso de obtención de las quebradas generadas al cortar una cara.

- Se realiza el corte de cada uno de los polígonos de la cara, pudiendo darse tres casos, que todos sus vértices se encuentren situados en el semiplano anterior a la recta de corte, que todos se hallen en el semiplano posterior, o bien que existan vértices situados a ambos lados de dicha recta. Repetiremos secuencialmente el corte con cada uno de los polígonos, iniciando el análisis con el que delimita la frontera de la cara, determinando a lo largo del proceso en cual de los tres casos comentados se encuentra cada polígono actuando en consecuencia según se explica a continuación.

- Partiendo del primer vértice del polígono en estudio analizamos cada uno de los que conforman su perímetro, para lo cual determinamos su posición referida a la recta de corte, así como la de los vértices anterior y posterior al mismo en el sentido de lectura propio del polígono. Para ello restamos al límite el valor de la tercera coordenada del vértice, considerando que la posición del mismo es igual a cero si el valor absoluto del resultado es menor que el límite de precisión prefijado, y en caso contrario que el valor de la posición es la unidad afectada del signo que corresponda a la mencionada diferencia; podremos hablar pues de posición negativa, cuando nos refiramos al semiplano de los puntos con coordenada Z menor que el valor límite, y por tanto la zona desechada por el recorte, de posición positiva cuando se dé el caso contrario, y de posición sobre la línea de corte en caso de ser su valor igual a cero.

- En función de la posición del vértice en estudio, pueden darse tres casos, que analizamos por separado:

--- Si esta posición es positiva, incorporamos el vértice a la quebrada que nos encontramos generando en este momento. Caso de ser negativa la posición del vértice

siguiente, procedemos a determinar la intersección de la recta que contiene a ambos puntos, e incorporamos el resultado como punto de dicha quebrada situado a continuación del anterior, introducimos asimismo un indicador que nos permita detectar que este nuevo punto no pertenecía previamente al polígono, sino que ha sido generado por intersección con la línea de corte, encontrándose por tanto sobre ella; asimismo detectamos mediante una variable testigo la circunstancia de haberse atravesado la línea de corte del lado positivo del plano al negativo.

--- Caso de encontrarse sobre la línea de corte, actuaremos en función de la posición del vértice siguiente, de forma que si su valor es negativo o bien se encuentra situado sobre dicha línea, y la ubicación del vértice anterior al que procesamos corresponde al semiespacio positivo, incorporamos el vértice en estudio a la quebrada, indicando su situación sobre la línea de corte y detectando, de igual forma que en el caso anterior, haber "atravesado" la línea de corte. Si la posición del punto siguiente es positiva, entonces la acción a desarrollar depende de la posición del punto anterior, que si es positiva implica incorporar el vértice en estudio a la quebrada, sin más acciones, pero que en los otros casos, además de producir esta incorporación, indica la posición del punto sobre la línea de corte, así como el hecho de haberla "atravesado."

--- Si el vértice en estudio se encuentra en posición negativa, solo actuaremos cuando el punto siguiente se encuentre en posición positiva, en cuyo caso calcularemos la intersección con la línea de corte, iniciando con el punto así determinado la generación de una nueva quebrada, dando por terminada la anterior e indicando la posición sobre la línea de corte de dicho punto.

- Una vez procesados todos los puntos del polígono, comprobamos si se han generado quebradas, lo que significa que se ha atravesado la línea de corte, en cuyo caso debemos ordenarlas, de forma que las quebradas a que el polígono haya dado lugar se inicien en un punto situado sobre la línea de corte, y terminen en otro de las mismas características, de acuerdo con el sentido de lectura que tenga cada una de ellas; o bien si, habiéndose incorporado puntos a la quebrada, no se ha atravesado la línea de corte, lo que se tiene es un polígono completamente interior a la zona positiva, hecho que hará preciso no considerar la quebrada como tal y proceder a almacenar la información resultante en el archivo de polígonos interiores creado para estos casos, que son polígonos que delimitan agujeros en la cara del recorte, y que mantienen esta característica posteriormente al mismo.

- Al finalizar el procesado de todos los polígonos de la cara, situamos un indicador en el archivo de los vértices de quebradas generadas por esta cara, de forma que nos permita reconocer el final del mismo.

f) Una vez generadas todas las quebradas producto de la intersección de la cara con la línea de corte, así como detectados todos aquellos polígonos que, permaneciendo invariables por no ser atravesados por ella, se encuentran situados en posición positiva, debemos proceder a ordenar las quebradas según el procedimiento cuyos pasos se explican a continuación:

- En primer lugar se realiza una lectura de la información obtenida en el paso anterior, referente a los vértices situados en posición positiva, o bien sobre la línea de corte, así como los extremos de las quebradas producto de intersecciones con dicha línea. De esta lectura tendremos para cada quebrada los datos referentes a su número de identificación, la dirección -en el archivo de vértices generado en el paso anterior- que ocupa el primero

de sus vértices, y el número de vértices de que consta.

- Determinamos ahora los valores extremos de contacto de cada quebrada con la línea de corte, así como el máximo valor del alejamiento del punto de vista para cada una de ellas, información que será imprescindible para proceder a la reconstrucción de las caras generadas por el corte.

- Ordenamos a continuación las quebradas en función de su menor coordenada de referencia en la línea de corte, es decir que usaremos para esta ordenación la menor X o la menor Y según que el plano de referencia que utilicemos para el proceso sea el XZ o el YZ respectivamente.

g) En este momento tenemos ya la información necesaria para realizar la reconstrucción de la cara, o las caras, resultado del corte de aquella con la que iniciamos el proceso. Para ello es necesario proceder a la lectura secuencial del archivo donde tenemos almacenada la información referente a las quebradas generadas por el corte, que ha sido creado y ordenado en el paso anterior. Al realizar dicha lectura, y en función del desarrollo de la misma, realizamos los siguientes pasos:

- Con la información de la primera quebrada, iniciamos la generación de la primera cara, o más exactamente del polígono frontera de la misma. Para ello incorporamos todos los puntos de la quebrada, siendo el inicial y el final puntos de la línea de corte, al polígono frontera, en el orden de lectura en que se encuentren almacenados. Colocamos el indicador correspondiente de la quebrada conforme su información ha sido ya utilizada, y obtenemos la información necesaria para determinar el sentido horario o anti-horario de lectura de la misma. Si no existen más quebradas en el archivo, procedemos a cerrar la frontera con la arista que, a lo largo de la línea de corte, une los extremos de la quebrada, en caso contrario proseguimos de la siguiente forma:

- Detectamos la presencia de otras quebradas interiores a la procesada, caso de no haberlas cerramos la frontera y buscamos la siguiente quebrada no utilizada para generar una nueva cara; en el supuesto contrario, consideramos el sentido de lectura de la quebrada utilizada en el paso anterior, y en función del mismo actuamos como sigue:

--- En el supuesto de ser anti-horario dicho sentido de lectura, incorporamos los vértices de la quebrada siguiente a la frontera de la cara, colocando el indicador de utilización de la quebrada de forma que lo refleje. Proseguimos la lectura del archivo de quebradas buscando la existencia de alguna interior a la primera, pero exterior a la última utilizada, en cuyo caso la incorporamos a la cara en proceso de reconstrucción, con las operaciones pertinentes al caso. Una vez comprobada la inexistencia de quebradas que satisfagan nuestras condiciones, procedemos a cerrar la frontera con los fragmentos de la línea de corte que unan los extremos de las quebradas utilizadas.

--- Cuando el sentido de lectura de la primera quebrada coincide con el horario, detectamos en primer lugar el número de quebradas que se encuentran en su interior, realizamos a continuación la lectura, en sentido contrario, del archivo de quebradas a partir de la última quebrada interior detectada, incorporando esta a la frontera de la cara en formación, así como todas las que encontremos en la lectura del archivo, siempre que sean interiores a la primera pero exteriores a la última utilizada. Todas las quebradas utilizadas tendrán el indicador correspondiente de forma que refleje esta condición, y cuando no quedan quebradas que considerar, se procede a cerrar la frontera con los fragmentos de la línea de corte que unen los extremos de las quebradas utilizadas.

- Se incorpora la frontera así generada al archivo de polígonos, y se crea un nuevo elemento en el de caras, cuyo primer polígono es el que acabamos de crear como frontera, y

cuyo numero de polígonos es de momento igual a uno.

- Se repite el proceso desarrollado hasta el momento, generando cuantas fronteras, y por tanto nuevas caras, sea necesario, hasta que todos los indicadores de utilización de las quebradas se encuentren marcando esta situación, con lo que sabremos que todas las que fueron generadas por el corte, han sido incorporadas a la frontera de alguna de las nuevas caras reconstruidas por el proceso.

- Debemos ubicar ahora los polígonos que, procedentes de la cara original que recortamos y al no intersectar la linea de corte, no formaron quebrada, y por tanto se encuentran situados en el interior de alguna de las caras generadas a lo largo del proceso. Para ello necesitamos detectar, con la información conocida de cada uno de ellos, dentro de qué polígono frontera de nueva formación se encuentra, y entonces incorporamos el polígono interior en cuestión a la cara cuya frontera lo contiene. El proceso de determinar en que cara se encuentra incluido un polígono interior, comprende los siguientes pasos:

--- Realizar una lectura secuencial de las fronteras de las caras que se han generado en el recorte de la cara que procesamos, estudiando si cada uno de dichos polígonos frontera contiene o no al polígono interior que nos ocupa, interrumpiendo el proceso cuando hayamos encontrado al que lo contiene. El proceso con cada uno de los polígonos frontera se inicia comparando su máxima Z con la máxima Z del polígono interior en estudio, caso de ser mayor la máxima Z del polígono interior, descartamos la posibilidad de que esta frontera contenga al mismo, realizando esta comparación con la que se encuentra situada a continuación; en el supuesto de ser mayor la maxima Z de la frontera que la del polígono interior ejecutamos el paso siguiente.

--- Considerar la intersección de la semi-recta con origen en un punto cualquiera del polígono interior, y paralela al eje Z en el sentido positivo del mismo, con toda la frontera considerada. Si el número de cortes que

Parte I      Capítulo IV

se producen es impar, entonces el punto es interior a la frontera, y por tanto lo es el polígono interior en su totalidad.

--- Una vez hallada la cara que contiene al polígono interior en estudio, se incorpora la dirección del mismo al último polígono de la cara, como elemento a considerar a continuación cuando necesitemos obtener la información referente a los polígonos que conforman la cara en cuestión; procedemos también a incrementar en una unidad el dato, guardado en el archivo de caras, que hace referencia al número de polígonos de la cara.

- Una vez realizado el proceso de ubicación de todos los polígonos interiores en sus caras correspondientes, damos por finalizado el recorte de la cara considerada, procediendo a reiniciarlo con la siguiente cara del archivo de la figura a recortar que lo precise, dando por finalizado el recorte de la figura una vez terminada la lectura de dicho archivo.

## ALGORITMO DE CLIPPING DE CARAS EN 3D

Proceso General:

Para N = 1 hasta Número\_de\_caras

  Con Cara[N]

    Posición contorno (Anterior,Posterior)

      Si Posterior Entonces

        Visibilidad Cara[N] (Vista)

        Si Vista Entonces

          Si Anterior Entonces

            Genera Quebradas

            Ordena Quebradas

            Configura Caras

          Si No Entonces

            Incorpora Cara[N]

        Fin Si

      Fin Si

    Fin Si

  Fin Para.

Genera Quebradas:

Repita (para todos los polígonos de la cara)

  Nueva\_Quebrada

  Agujero = Si

  Repita (para todos los vértices del polígono)

    Posición (Vértice)

```
Caso Posición
1/Incorpora Vértice a Quebrada
  Si Posición (Vértice_Siguiente) = -1 Entonces
    Calcula Intersección
    Incorpora nuevo Vértice a Quebrada
  Fin Si
0/Si Posición (Vértice_Siguiente) = 1 Entonces
  Si Posición (Vértice_Anterior) = 1 Entonces
    Incorpora Vértice a Quebrada
  Si No Entonces
    Agujero = No
    Incorpora Vértice a Quebrada
  Fin Si
Si No Entonces
  Si Posición (Vértice_Anterior) = 1 Entonces
    Agujero = No
    Incorpora Vértice a Quebrada
  Fin Si
Fin Si
-1/Si Posición (Vértice_Siguiente) = 1 Entonces
  Calcula Intersección
  Incorpora nuevo Vértice a Quebrada
Fin Si
Fin Caso
Hasta No Vértices en el Polígono
Si Agujero Entonces
  Guarda Agujero
Si No
  Reordena Vértices Quebrada
Fin Si
Hasta No Polígonos en la Cara
```

Configura Caras:

Repite (para todas las Quebradas)

Sentido\_de\_lectura (Directo)

Si Directo Entonces

Nueva Cara

Lee Quebrada

Repite

Incorpora Quebrada a frontera de Nueva Cara

Lee Quebrada

Hasta Mínimo\_Quebrada>Máximo\_Quebrada\_Anterior

Si No Entonces

Lee Quebrada

Nueva Cara

Determina quebradas interiores a Quebrada

Invierte\_Orden quebradas interiores

Repite

Incorpora Quebrada a frontera de Nueva Cara

Lee Quebrada

Hasta No quebradas interiores

Fin Si

Incorpora Nueva Cara

Hasta No Quebradas

Si Agujeros Entonces

Repite

Lee Agujero

Determina Cara que lo contiene

Incorpora Agujero a Cara

Hasta No Agujeros

Fin Si

## 5. VISUALIZACION

### 1. Concepto de figura de alambre.

Aunque en el primer capítulo aparecía un concepto similar, al tratar de los distintos tipos de estructura de datos a utilizar mencionamos el "modelo de alambre", en aquel caso nos referíamos a un modelo para guardar la información del objeto, mientras que en este caso buscamos una forma sencilla y rápida de representar la información existente en un modelo de fronteras, que al tener una mayor complejidad permite representaciones mucho más elaboradas y de mayor utilidad y realismo.

La figura de alambre consiste en la representación gráfica de la totalidad de las aristas del tema, lo que configura la visualización de un objeto tridimensional formado con elementos lineales rectos y extraordinariamente delgados (alambres) colocados en lugar de las aristas del cuerpo.

Para acceder a la información que contiene el modelo acerca de las aristas que existen en el cuerpo por él representado, debemos realizar una lectura secuencial del archivo de polígonos, y a partir del mismo determinar las aristas que forman el contorno de cada uno de ellos, de esta forma cada arista aparecerá dos veces, una en cada polígono de los dos que con su intersección la definen; para evitar trazarla por duplicado, el procedimiento más

sencillo será dibujarla solo en aquel caso en que la dirección del primer vértice sea menor que la del segundo. En el caso de desear una visualización parcial, se procederá al recorte de cada una de las aristas, previa eliminación de la duplicidad de las mismas, en el espacio tridimensional o en el plano una vez se haya procedido a realizar la proyección, en función del tipo de recorte establecido, para ser entonces dibujado el resultado de dicho recorte.

Este tipo de representación es, sin la menor duda, el más rápido e inmediato que podemos tener de un objeto definido mediante un modelo de fronteras, pero a su vez no es más que una aproximación a la visión real del mismo, por lo que su utilidad consiste en poder saber con rapidez el aspecto del objeto en un momento intermedio de su construcción, si estamos procediendo al manejo de un sistema de generación de volúmenes, o si la visualización en unas condiciones determinadas nos satisface, antes de realizar el más lento proceso de representarlo con la depuración de las partes ocultas, para obtener una imagen más realista. En el caso de figuras de una cierta complejidad, este tipo de representación puede ser tan confuso que obligue a prescindir de la misma, y se tenga que proceder a la depuración de las partes ocultas para hacer comprensible el dibujo.

La representación de la figura de alambre de un modelo, visualiza en su totalidad la estructura formal del objeto cuya información contiene. Si se da la circunstancia de necesitar actuar sobre un elemento determinado, independientemente de su posición, que pudiera hacerlo aparecer oculto al observador, es este tipo de representación la más adecuada para tal fin, así como en el caso de querer estudiar la interrelación de distintas partes de un tema, que no permitan conseguir un punto de

vista adecuado para visualizarlas simultáneamente de forma más realista. En éstas y otras situaciones en las que pueda interesar visualizar la estructura formal completa del objeto será conveniente la utilización este tipo de representación.

## 2. Técnicas de eliminación de partes ocultas.

En toda visión de un objeto nos queda oculto un fragmento del mismo, que en el caso de objetos convexos consiste en las caras orientadas en sentido contrario al observador, pero que en el caso de un objeto cóncavo pueden ser caras, o partes de las mismas, que por su orientación serían percibidas por el observador, pero otras caras más próximas al mismo las ocultan impidiendo su visión. Es obvio que si el tema a visualizar consiste en un conjunto de objetos, independientemente de la forma que tengan, siempre existirán posiciones del observador para las que unos objetos oculten, total o parcialmente, a otros situados a mayor distancia.

Para conseguir un mayor realismo en la visualización que procesamos, debemos conseguir que las partes ocultas no aparezcan en el dibujo obtenido como resultado, por lo que la eliminación de dichas partes, constituye un tema del mayor interés en cualquier proceso de visualización que pretenda ofrecer resultados que den la mayor similitud posible con la observación real del objeto. De esta forma ganaremos realismo y claridad en la representación, pudiendo apreciarse la importante mejora de percepción que se obtiene con la eliminación de las partes ocultas en la figura I.V.2.1 en la cual se muestran dos representaciones de un mismo objeto, correspondientes a las mismas condiciones de visualización, pero graficándose en la primera la figura de alambre, y en la otra el resultado de

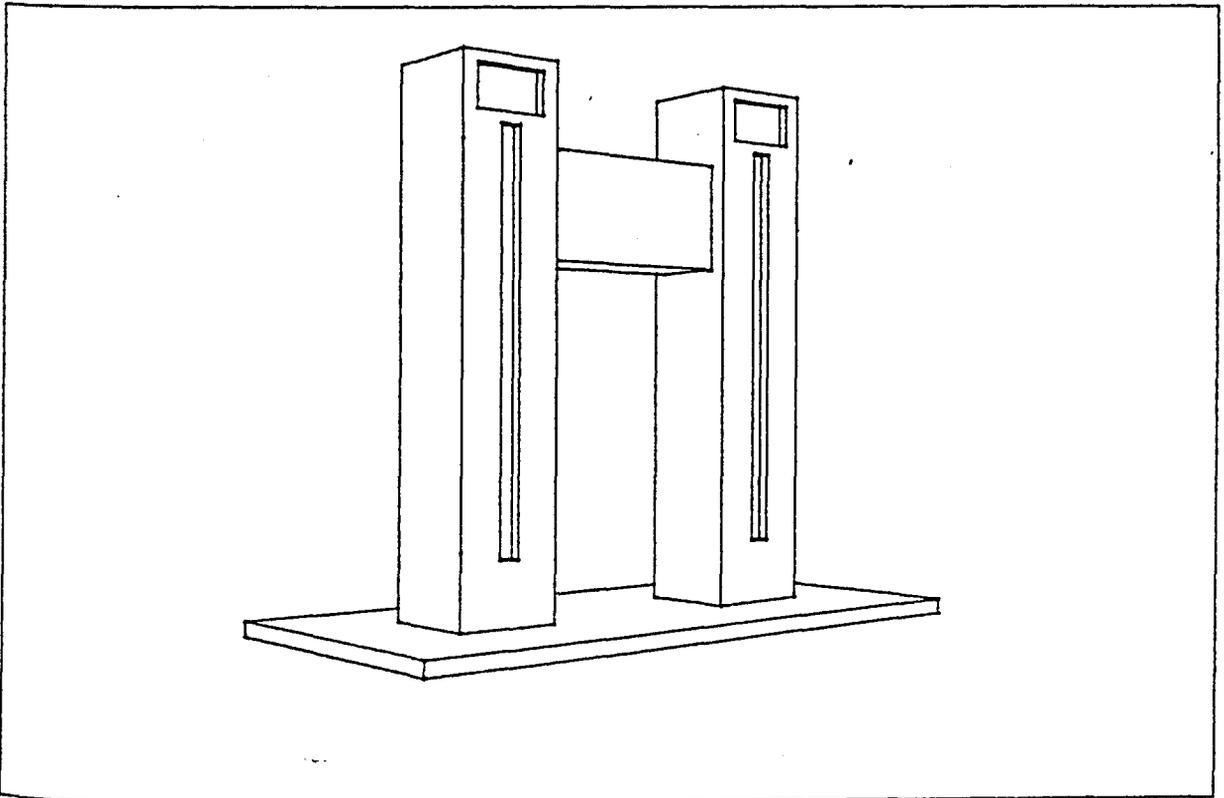
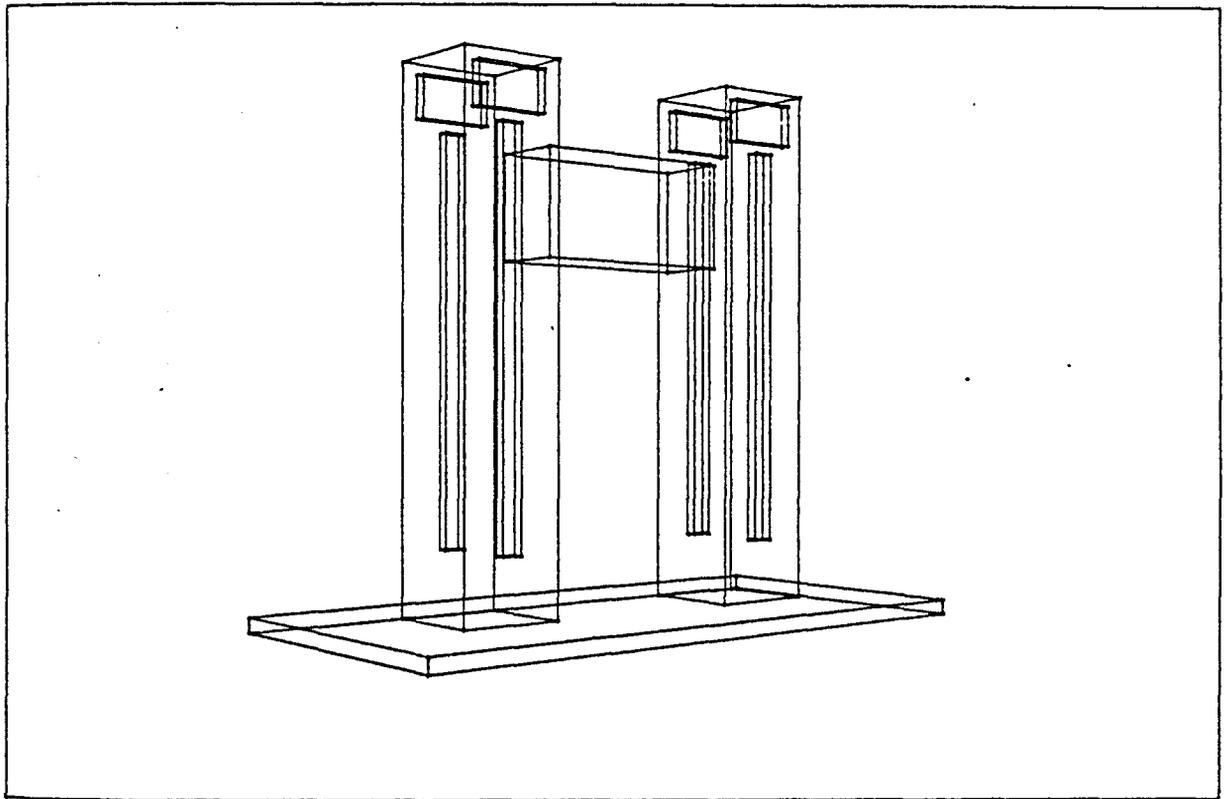


Figura I.V.2.1

haber procedido a la eliminación de las partes que permanecen ocultas para el observador.

Se hace patente el interés de la mejora que representa la eliminación de las partes ocultas del objeto, en la representación plana de cualquier visualización del mismo. El proceso para conseguirlo consiste en la exploración de cada uno de los elementos del modelo, con la intención de determinar para cada uno de ellos si existen otros que se interponen en el camino de un rayo visual que, partiendo del punto de vista, llegue hasta el elemento en cuestión, permitiendo por tanto su visión por parte del observador, y por ello su representación gráfica.

La resolución de este problema por métodos informáticos, no es en modo alguno inmediata, requiriendo algoritmos complejos y con tiempos de procesado importantes, en función evidentemente del objeto y del equipo utilizado. En un principio, estos algoritmos realizaban la depuración de las "líneas ocultas", pues los dispositivos de visualización de que se disponía permitían únicamente la representación de dibujos (como ya dijimos, figuras trazadas mediante líneas), y por tanto el proceso debía afectar a dicho elemento de representación. El más destacable de los primeros algoritmos de este tipo es el de L. G. Roberts ("Machine Perception of Three Dimensional Solids", MIT Lincoln Lab. TR 315, Mayo 1963), el cual solo contemplaba sólidos convexos y daba unos tiempos de procesado muy altos. Más adelante se dispuso de pantallas "raster", es decir aquellas en que la visualización se realiza explorando secuencialmente todos los puntos de la pantalla y variando sus condiciones de iluminación de acuerdo con las instrucciones dadas por el programa, pudiendo accederse en cualquier momento a la información referente a las condiciones de un punto determinado de la pantalla, teniendo la posibilidad de variarlas si así

conviene.

Se han desarrollado al amparo de esta tecnología, algoritmos que determinan las condiciones de visualización de cada punto de la pantalla, en función de las condiciones propias y de iluminación de la superficie en que se encuentra situado el punto del objeto que es representado por él, recibiendo el nombre de algoritmos de depuración de "partes ocultas" o "superficies ocultas", consiguiendo, con equipos especialmente adaptados para este fin, muy altas velocidades de proceso.

De todas formas no existe una solución universal a este problema, por el contrario, son muchos los algoritmos existentes, adecuándose cada uno de ellos a las características específicas del tema a tratar, del equipo a utilizar, o incluso del tipo de representación que se desea obtener. Unos algoritmos estudian las relaciones entre los distintos elementos del objeto a representar, y como consecuencia de ello, determinan la ocultación de unas partes y la visibilidad de otras, procediendo a la visualización de las últimas; otros, sin embargo, estudian la imagen proyectada con el fin de determinar en que forma se visualiza cada punto de la misma, lo cual es función de la posición y orientación en el espacio del punto del objeto que es representado por él, por lo que es necesario recurrir a la información tridimensional del sólido, para determinar las condiciones de visualización del punto en cuestión.

Existen procesos que, independientemente del algoritmo de que se trate, influyen favorablemente en su efectividad, y por tanto es conveniente contemplarlos para optimizar el tiempo de ejecución. En la determinación de la visibilidad de un elemento del objeto, este debe compararse con los demás, con la finalidad de determinar si alguno se

interpone entre el observador y el elemento en estudio, ocultandolo total o parcialmente; esta comparación consume tiempo de proceso, por lo que si se puede evitar su realización entre elementos que, por su posición o características, no pueden realizar esa ocultación, se ahorra el tiempo de proceso que se hubiera empleado en la comparación, y por tanto se optimiza el algoritmo. La tarea de realizar para cada elemento el "descarte" de aquellos que no pueden afectar a su visibilidad, es una de las que todo algoritmo de depuración de partes ocultas debe contemplar, siendo más completo cuantas más condiciones de descarte puedan ser consideraras. Para facilitar el descarte y hacerlo mas eficiente, es útil la "ordenación" de los elementos a considerar, pues así la selección de aquellos que, al no ser descartados, serán utilizados en la comparación, es más directa, interrumpiendose el proceso de descarte a partir de un determinado elemento en adelante. Debe, no obstante, considerarse que en procesos de figuras no excesivamente complejas, en ordenadores con poca capacidad de memoria interna y discreta velocidad de proceso (como el utilizado por el autor), puede no ser económico (en términos de tiempo de procesado) realizar esta ordenación, pues el ahorro de tiempo en las comparaciones posteriores para efectuar el descarte, no compensa el tiempo empleado en efectuar la ordenación, ya que el tema de ordenar elementos geométricos en el espacio tridimensional, según criterios posicionales, no es tarea fácil ni rápida de resolver.

Mencionaremos por último el concepto de "coherencia", que consiste en considerar que, en el entorno cercano de un punto de la imagen, las características de visualización guardan, por regla general, una estrecha relación con las de dicho punto, por lo que puede aprovecharse esta circunstancia para acortar el proceso, estudiando la visualización de cada punto en función de los más próximos

ya considerados, siendo esto de especial aplicación cuando se trata de la generación de imágenes, pero no tanto en el desarrollo de representaciones gráficas por medio del trazado de dibujos.

Para tener una imagen más global del panorama existente en este campo, conociendo los aspectos generales más relevantes de algoritmos de distintos tipos que se han desarrollado para resolver este problema, existe un muy interesante artículo titulado "A Characterization of Ten Hidden-Surface Algorithms" publicado en el num. 1 del Vol. 6, de Marzo de 1974 de "Computing Surveys", cuyos autores son Ivan E. Sutherland, Robert F. Sproull y Robert A. Schumacker. Se analizan en este artículo las características básicas de diez algoritmos de distintos autores, que resuelven el tema de la eliminación de superficies ocultas de distintas formas, realizando al final estudios comparativos de los mismos. Dividen los autores del artículo los distintos algoritmos en tres grupos, los que estudian el objeto para depurar a partir del mismo las partes ocultas y representar gráficamente las visibles, los que analizan directamente la imagen resultante a configurar para determinar las características de visualización de cada uno de los puntos de la misma, y un grupo que, configurando la imagen a partir del análisis de cada uno de sus puntos, estudia previamente el objeto, con el fin de establecer una lista de prioridades que facilita el procesado posterior.

Mencionaremos escuetamente las características de los cuatro algoritmos del primer grupo analizados en el citado artículo, ya que son los que guardan una cierta relación con nuestro trabajo. Todos ellos actúan sobre los elementos del objeto a visualizar, sean vértices, aristas o caras, determinando las condiciones de visibilidad total o parcial de los mismos según diversos métodos.

L. G. Roberts realiza, en el algoritmo por él desarrollado, la comparación entre cada arista pertinente con cada cuerpo de la escena que pueda ocultarla, considerando cuatro casos posibles: que sea ocultada en su totalidad, que no sea ocultada en absoluto, que quede visible una parte de la misma, o bien que sean dos porciones de la arista las que resulten visibles. Una restricción importante de este algoritmo consiste en que todos los cuerpos tratados deben ser convexos, siendo necesario descomponer los cuerpos cóncavos que puedan existir en un conjunto de cuerpos convexos, con los problemas de procesado que esto comporta. El tiempo de procesado es elevado, aumentando con el cuadrado del número de cuerpos que contenga la escena.

El algoritmo desarrollado por A. Apple analiza las intersecciones de las aristas proyectadas, definiendo la invisibilidad cuantitativa de cada segmento de las mismas, en función del número de ocultaciones que le afectan, siendo visible en caso de ser nula dicha cantidad. Partiendo de un extremo de la arista, el cual tiene una determinada invisibilidad, se estudia cada intersección que sufre con otra arista de contorno, aumentando o disminuyendo una unidad el valor de la invisibilidad de la arista en estudio, caso de ser procedente, para el segmento de la misma que en la intersección tiene su comienzo. Uno de los temas importantes es el conocimiento de la invisibilidad cuantitativa del vértice inicial de cada arista que se estudia, por lo que adquiere un aspecto relevante la utilización del concepto de coherencia, que mencionábamos anteriormente, utilizando la información obtenida al procesar una arista, respecto de la visibilidad cuantitativa del vértice final, para iniciar el proceso de aquella arista en que dicho vértice sea el inicial.

El trabajo de F. P. Loutrel consiste en la realización de un proceso muy parecido, denominando "orden de invisibilidad" a la invisibilidad cuantitativa de Apple, y utilizando el término de "arista frontera" para las aristas de contorno. A partir de la intersección de dos aristas proyectadas en el plano de proyección, se determina cual de ellas oculta a la otra, ayudándose de la información referente a sus profundidades respectivas, considerando su posición en el espacio tridimensional.

De forma similar a los dos anteriores, desarrollan su algoritmo R. Galimberti y U. Montanari, los cuales determinan el conjunto de caras que ocultan a un punto determinado, denominándolo la "naturaleza" del mismo, usando esta información como ayuda al proceso, que se realiza mediante criterios muy parecidos a los dos anteriores.

El algoritmo que se explicará detalladamente en el capítulo siguiente, desarrollado por el autor como ejemplo de aplicación de todo lo dicho, estudia las relaciones entre las aristas y las caras proyectadas, utilizando para ello la información referente al objeto en el espacio, conjuntamente con la de la figura proyectada, determinando, a partir de esta información, qué elementos son ocultados por otros, y cuales son total o parcialmente visibles. Sus restricciones cualitativas, consisten en la no admisión de superficies no planas, trabajando únicamente con sólidos rígidos poliédricos, con caras que pueden ser cóncavas o convexas, y contener o no agujeros, pero que deben ser planas y con un contorno definido mediante aristas rectas. Sus restricciones cuantitativas, se basan en la capacidad de memoria disponible, así como en el tiempo límite de proceso que se considere admisible.

3. Representación gráfica del resultado.

La representación gráfica del resultado es inmediata cuando se trata de algoritmos del tipo que hemos denominado como generadores de imágenes, ya que, al trabajar a partir del análisis de cada punto de las mismas, estas se van configurando conforme avanza el proceso. Sin embargo, los algoritmos que generan dibujos dan como información de salida un archivo, o conjunto de archivos, que definen la proyección en función de las coordenadas de los puntos que la forman y de la información referente a los trazos que unen dichos puntos, e incluso puede obtenerse como resultado información de los polígonos que, en la proyección, representan las partes visibles de las caras del objeto, con la finalidad de dar distintos tratamientos gráficos superficiales de llenado de área de estos polígonos.

Es necesario por tanto incorporar un algoritmo, bastante elemental por otro lado, que recoja esa información y la traslade al dispositivo elegido para la representación, que por regla general será la pantalla del monitor o una trazadora que nos permita obtener un registro permanente del resultado.

Puede interesar, en un momento determinado, obtener la representación de un fragmento del dibujo resultante del proceso, normalmente esta operación lleva aparejado un aumento de la dimensión de la figura que se representa, lo cual permite observar mejor los detalles de menor entidad, que en la representación total tal vez no pueden apreciarse debidamente.

Para ello debemos delimitar con un polígono la zona del dibujo total que deseamos representar aisladamente, y realizar un recorte del dibujo a partir de la información que poseemos, obteniendo como resultado la información referente al nuevo trazado, que será representada con el mismo procedimiento que es utilizado para trasladar la figura completa al dispositivo de visualización empleado en cada momento.

Suponiendo que la dimensión máxima del área de trabajo permitida por el dispositivo de salida utilizado sea suficiente, también podemos delimitar una porción de la misma para realizar el trazado, con la finalidad de incluir varios dibujos simultáneamente, o bien para conseguir un determinado formato de salida. Podemos conseguir, manejando adecuadamente las dimensiones de la representación, y los fragmentos de la proyección que delimitemos para su trazado, el efecto conocido como "zoom", el cual consiste en la ampliación creciente de una zona del dibujo con centro en un punto determinado del mismo.

No debe confundirse el efecto de "zoom" con el resultado de una aproximación real al objeto, pues, aunque los efectos visuales puedan ser similares, no son en absoluto coincidentes, al variar únicamente las dimensiones de los trazos en el caso del "zoom", cambiando la posición del observador en el caso de realizarse una aproximación, como puede fácilmente apreciarse en las figuras I.V.3.1, I.V.3.2, I.V.3.3 y I.V.3.4, representando la primera de las mismas la visualización del acceso a un edificio desde la vía pública, procediéndose entonces a representar las vistas obtenidas desde tres posiciones consecutivas situadas sobre un itinerario recto dirigido hacia la puerta de entrada al edificio; las tres figuras siguientes

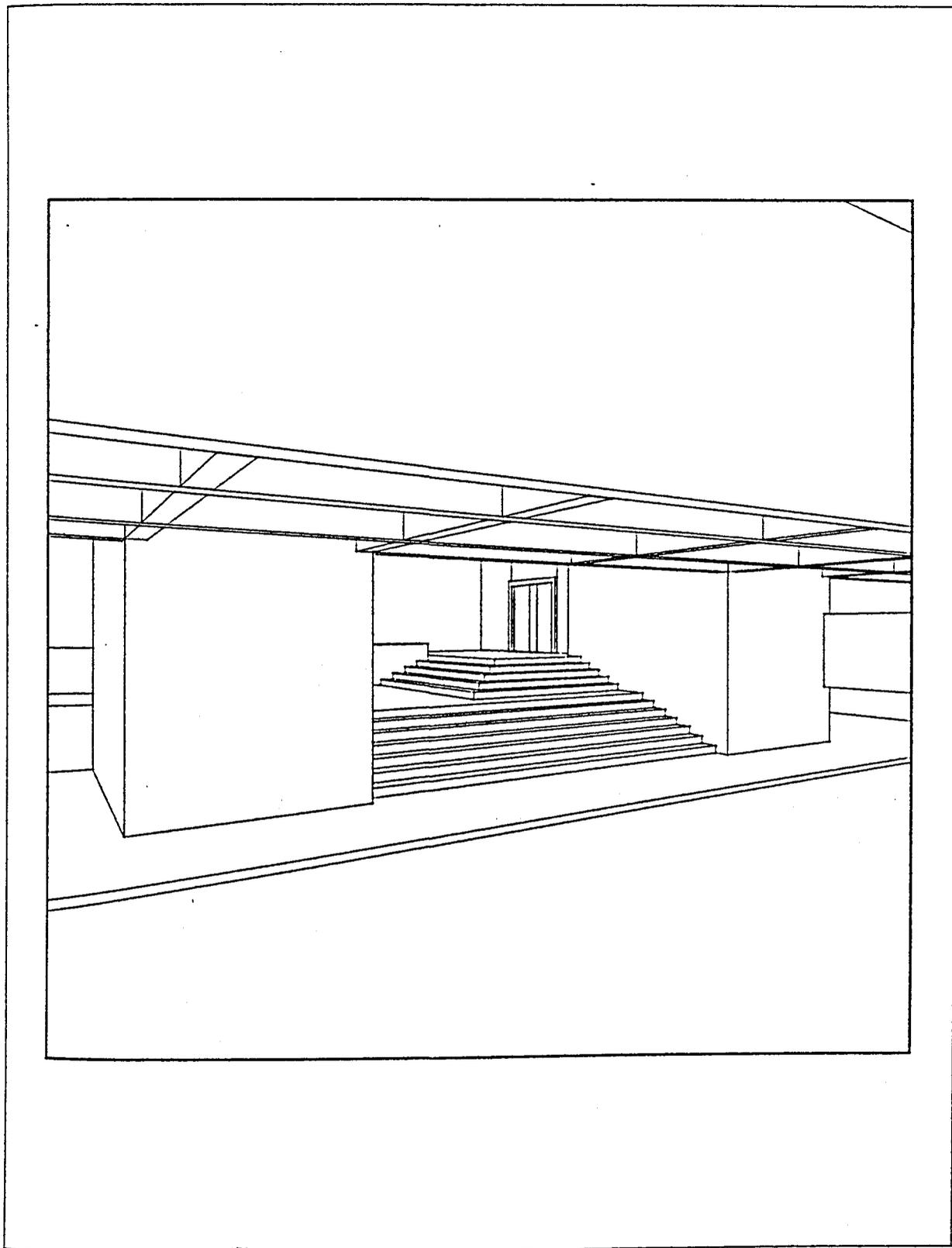


Figura I.V.3.1

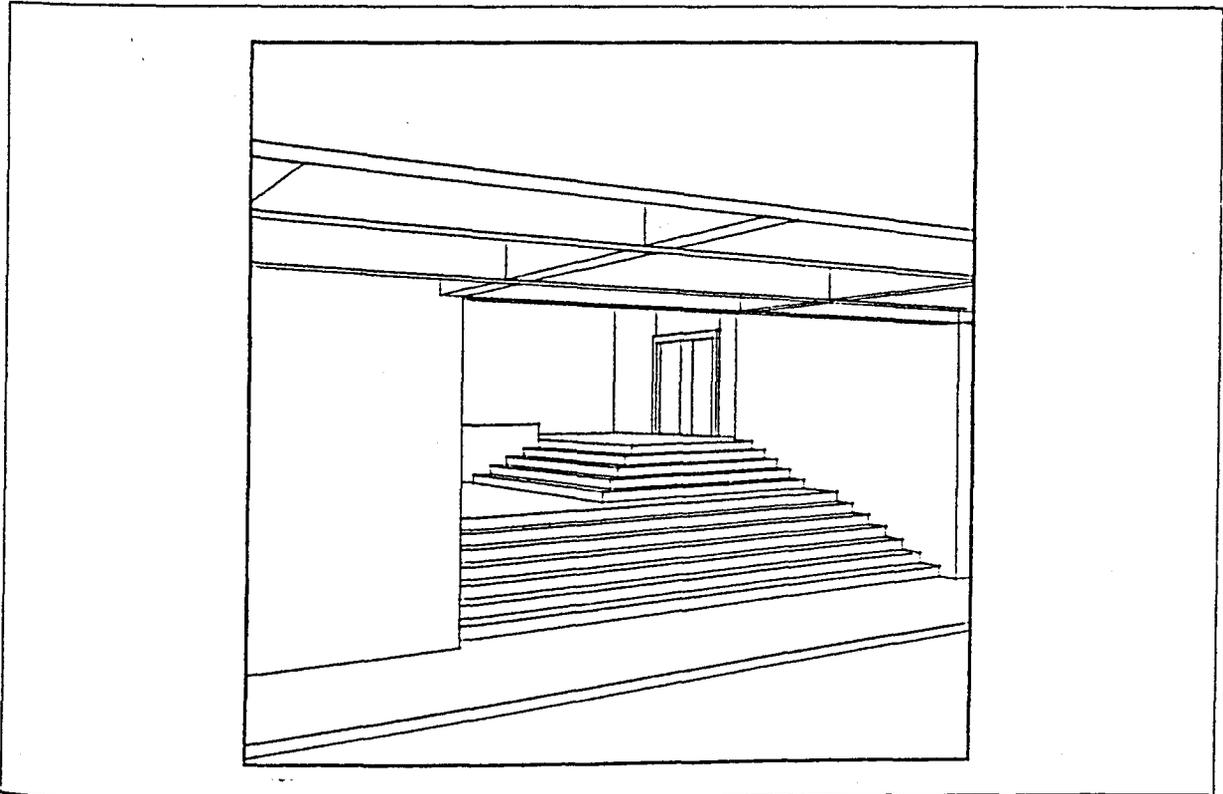
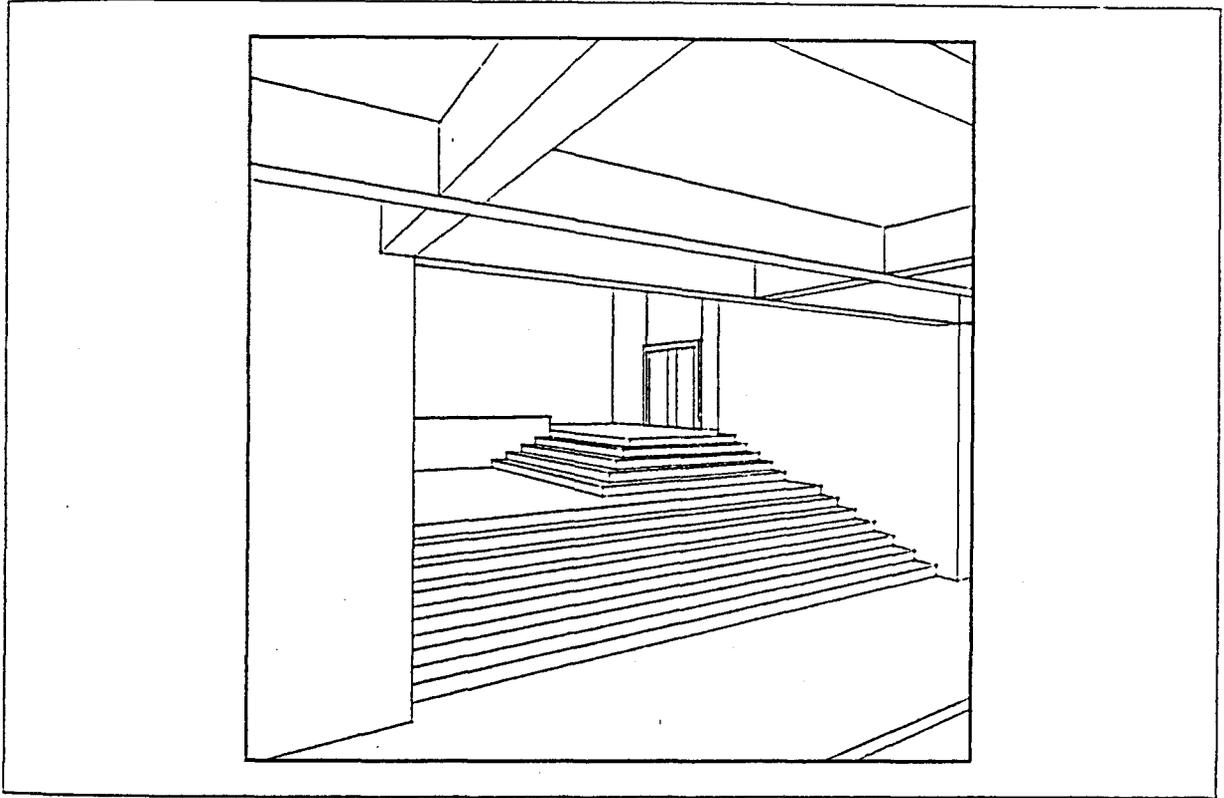


Figura I.V.3.2

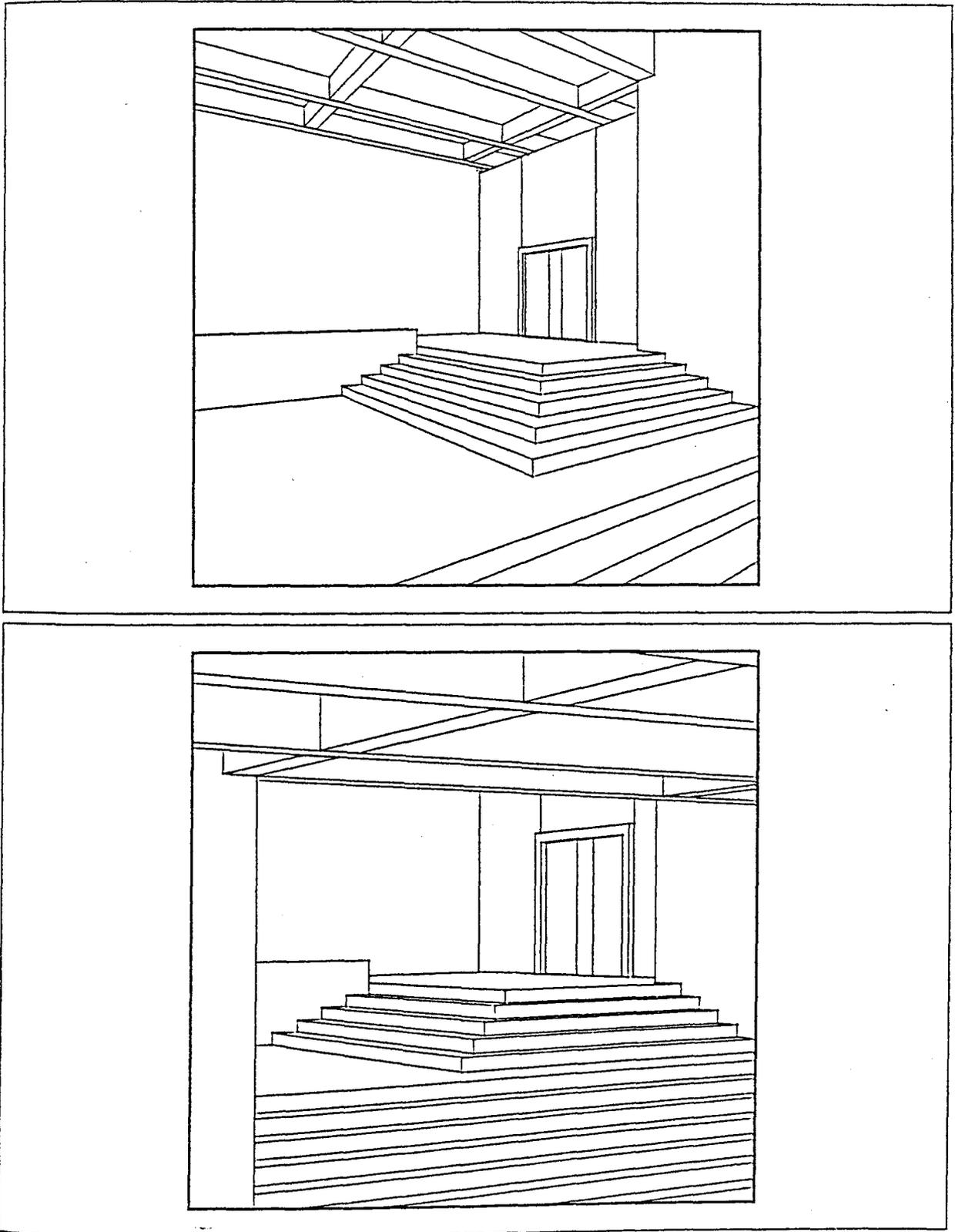


Figura I.V.3.3

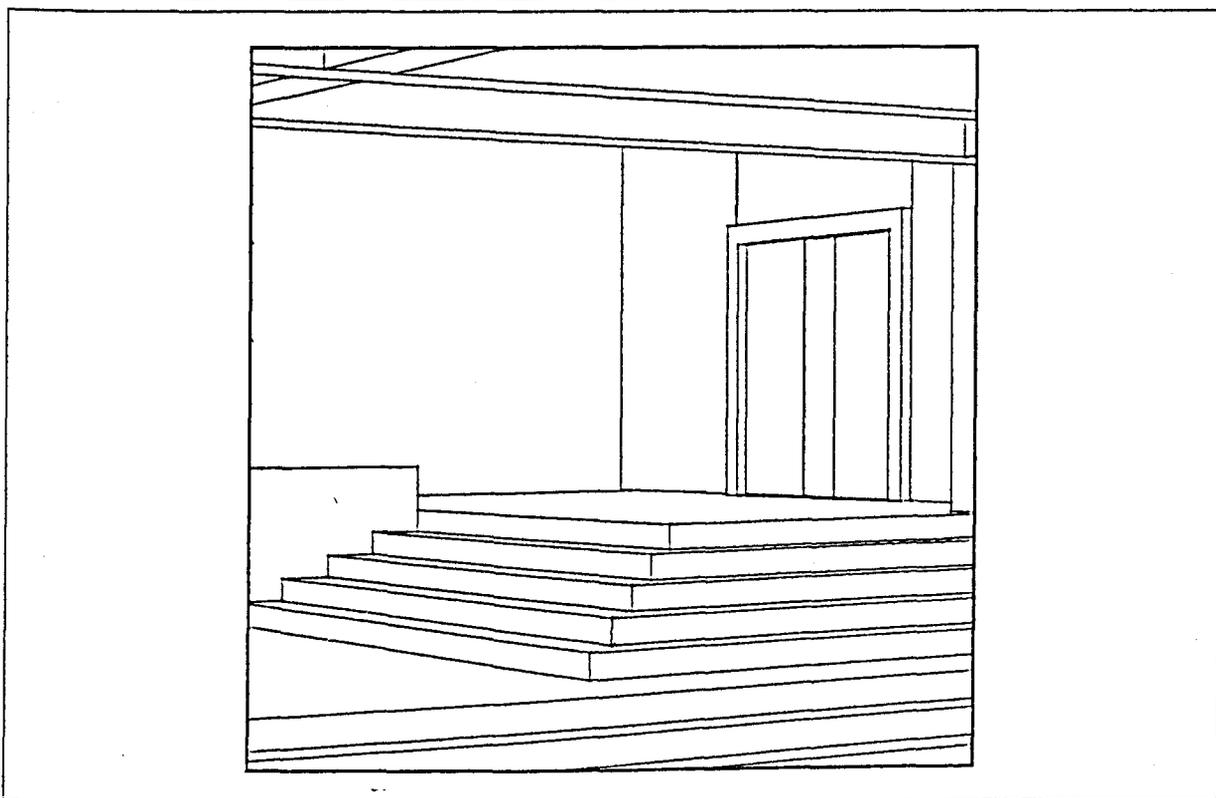
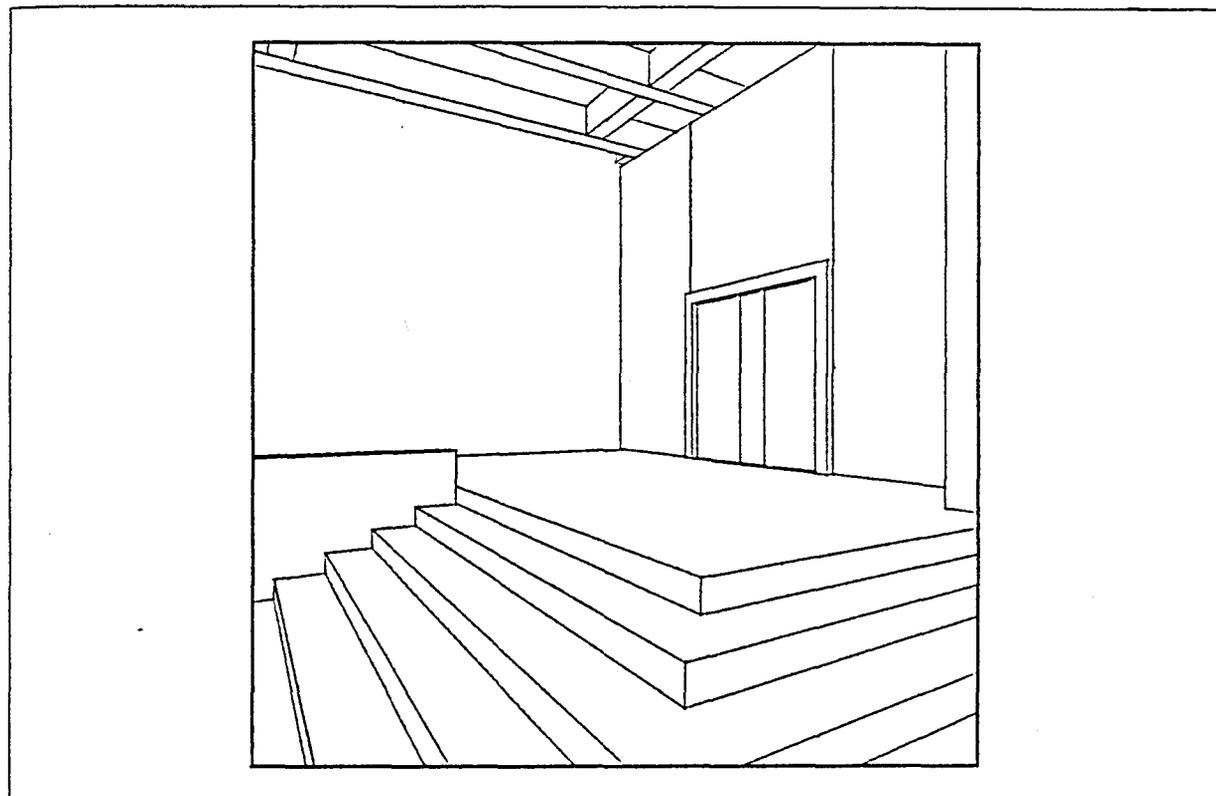


Figura I.V.3.4

Parte I    Capítulo V

muestran en su parte superior la visión desde la posición siguiente a la anterior según el itinerario prefijado, y en la parte inferior el efecto de "zoom" comentado, realizándose una ampliación de la primera figura de la serie de forma que se obtenga un efecto lo más similar posible al dibujo de la parte superior. Son evidentes las importantes diferencias entre ambas representaciones, acentuándose las mismas cuanto mayor es la cercanía al objeto, pudiendo apreciarse que no solo cambia el trazado del dibujo al acentuarse el efecto de fuga, sino que se aprecian nuevos elementos que en posiciones del observador más alejadas no podían ser percibidos por el mismo.

## 6. DESARROLLO DE UN PROGRAMA DE VISUALIZACION.

### 1. Explicación de la estrategia seguida.

En este capítulo se muestra el desarrollo de un programa concreto de visualización de sólidos en tres dimensiones, que se ha implementado en un ordenador con un microprocesador de 16 bits, y unas características de hardware que hacen razonable suponer pueda ser similar al que un profesional de tipo medio tenga a su disposición para realizar tareas de este tipo; al mismo tiempo, debe tenerse también en cuenta, que es el equipo de mejores prestaciones de que ha podido disponer el autor de esta tesis, por lo que es evidente que no pudo realizarse con otro.

Antes de iniciar la explicación del proceso que sigue el programa debemos considerar las condiciones de partida, entre las que figura la de considerar al sólido a visualizar perfectamente definido, mediante un modelo de fronteras como los mencionados en el capítulo primero, y ubicado en un espacio tridimensional con un sistema de coordenadas ortogonales de "mano izquierda". Necesita entonces conocer el programa, el punto del espacio donde se encuentra situado el observador, y otro punto de paso de la visual principal. Una vez definidos estos dos puntos, mediante sus coordenadas en el sistema de referencia inicial del objeto, determinamos otros elementos de la proyección, para lo cual consideramos que el plano del

cuadro (sobre el cual se realiza la proyección) es el perpendicular a la visual principal por el punto utilizado para definirla. Debemos también incorporar como dato inicial de puesta en marcha del programa el tipo de proyección que deseamos realizar, y si debe procederse o no a la depuración de las líneas ocultas de la misma, o por el contrario el resultado buscado es representar la figura de alambre del modelo visto en estas condiciones. Consideramos por último, y en el caso de tratarse de una proyección cónica, también como dato inicial previo al proceso el ángulo de apertura de la visión en torno a la visual principal, para poder definir la pirámide de visión que utilizaremos posteriormente.

Supondremos, en la explicación que sigue, que el proceso tiene como resultado a conseguir la visualización de una proyección cónica, con la representación únicamente de las partes vistas, por lo que deberá realizarse la parte del proceso que depura las líneas ocultas. Vamos a considerar el tema formado por un único objeto, todo lo complejo que admita el programa, pero constituyendo una unidad, caso de ser varios, esto solo favorecería el proceso, al poder descartar en determinados pasos alguno en su totalidad, pero por lo demás el proceso sería el mismo, aplicado secuencialmente al conjunto de los objetos que constituyeran el tema.

Vamos pues a analizar los pasos que es necesario realizar para conseguir la representación gráfica de la visión que el observador tendría del objeto en las condiciones establecidas en el párrafo anterior. Debido a lo complejo del proceso, y la gran cantidad de cálculos necesarios para llevarlo a cabo, es muy conveniente trabajar en un sistema de referencia que lo facilite, por ello el primer paso a efectuar es el cambio de sistema de referencia, de acuerdo con el proceso analizado en el

## Parte I Capítulo VI

capítulo segundo, a un sistema de coordenadas visuales, en que el punto de vista es el origen de coordenadas, y la visual principal el eje Z, con el sentido positivo coincidente con el de la visual principal, por lo que un mayor alejamiento en la dirección de la visual supondrá un incremento del valor de la tercera coordenada. El plano de proyección será paralelo al XY, y situado a una distancia del mismo determinada por la tercera coordenada del punto utilizado para fijar la visual principal. Este cambio de coordenadas supone determinar la matriz del operador correspondiente, y operar con ella todos los vértices del objeto, obteniendo de esta forma un archivo de trabajo de vértices, de la misma dimensión que el inicial, pero con las coordenadas de cada vértice referidas al nuevo sistema, que es en el que trabajaremos en todo lo que se expone a continuación.

Existen elementos de los que se puede detectar ya en este momento su total invisibilidad, siendo conveniente su depuración para que no consuman inutilmente más tiempo de procesado en los pasos posteriores del programa. Una cantidad importante de caras del objeto tendrán una orientación tal que permitirá detectar su invisibilidad, al determinar el ángulo que forma el vector normal a la cara, dirigido en sentido exterior al cuerpo y aplicado en uno cualquiera de los vértices de dicha cara, y el vector que, partiendo de ese vértice, se dirige al origen de coordenadas; si este ángulo es agudo, toda la superficie de la cara será visible, si por el contrario el ángulo es obtuso, la cara no permite la visión de ninguno de sus puntos, y si el ángulo fuese recto, entonces la cara se encontraría vista de perfil, lo que a efectos del proceso que estudiamos surtirá el mismo efecto que si fuese totalmente oculta. Se procede al análisis, según este criterio, de todo el archivo de caras, a partir del polígono frontera de cada una de ellas, generando así un

archivo de trabajo con las que, de acuerdo con su orientación, sean propiamente visibles, a falta de determinar si son ocultadas total o parcialmente por otras.

Con el fin de poder utilizarlos posteriormente en el proceso de descarte, se realiza la obtención de los valores límite de cada una de las caras incluidas en el archivo de trabajo, determinando para cada una de ellas los valores máximos y mínimos de las coordenadas de los vértices de su polígono frontera según el siguiente criterio: las de las proyecciones de dichos vértices para el caso de las dos primeras coordenadas, y los límites inferior y superior de las terceras coordenada en el espacio tridimensional. Tendremos así para cada cara, los valores de la máxima y mínima coordenada X (e igualmente sucederá con la segunda coordenada) de su proyección sobre el plano, así como la máxima y mínima coordenada Z de acuerdo con su posición en el espacio en el sistema de referencia de coordenadas visuales.

Mencionamos, en el capítulo tercero, el grave problema que representan las partes del objeto situadas detrás del punto de vista a la hora de realizar la proyección, pues dan lugar a imágenes virtuales que, no solo distorsionan la representación con su presencia, sino que dan lugar a ocultaciones y otros efectos indeseables sobre la parte visible. Ya dijimos que no podía ser suficiente la simple eliminación de todos los vértices con tercera coordenada negativa, y por tanto situados tras el punto de vista, pues pueden existir elementos que tengan vértices situados a ambos lados del punto de vista, y por ello deben ser parcialmente considerados. Sería rápido y sencillo realizar un recorte de aristas en el espacio con el plano XY, con lo cual el archivo de aristas quedaría constituido solamente por los segmentos de las mismas situados en el semiespacio visible, pero ello imposibilitaría procesar la

ocultación producida por las caras, al no tener la información referente a la porción de la cara incluida en la zona visible configurada en el modelo utilizado como tal cara. Debemos pues realizar un recorte tridimensional de caras, mediante el algoritmo desarrollado en el capítulo cuarto, que modifique el contenido del archivo de trabajo correspondiente, dejando únicamente en el mismo las caras contenidas totalmente en el semiespacio situado tras el plano utilizado para el recorte, e incorporando a dicho archivo las nuevas caras generadas en el proceso, fragmentos visibles de las que contenían vértices a ambos lados del plano de recorte. Utilizamos como plano de recorte el paralelo al XY situado en el semiespacio de Z positiva a una distancia del origen igual al límite próximo de visión, pues de esta forma eliminamos las partes del objeto excesivamente próximas al observador, y que por tanto tendrían una representación distorsionada respecto a su observación real, además de ocultarnos una parte considerable del objeto.

Una vez que tenemos, en los archivos de trabajo correspondientes, toda la información necesaria para definir el sólido a visualizar, podemos realizar la proyección del mismo, pero antes es interesante realizar una consideración referente al ángulo de abertura de la visión que hayamos definido como aceptable, este ángulo delimitará la porción del espacio que, situado frente al observador, es visible por éste, como ya explicamos esta delimitación es realizada mediante el recorte con cuatro planos que, pasando por el origen, contienen dos de ellos al eje Y, y los otros dos al eje X, formando con la visual principal un ángulo igual al de abertura de la visión. Consideramos un último plano paralelo al XY, que limita la profundidad máxima de la zona visible, pues a veces puede ser interesante visualizar solo partes próximas del tema, y la visión de las que se encuentran más alejadas puede crear

## Parte I Capítulo VI

confusión. La representación gráfica del resultado se realiza mediante la generación de un archivo de trazos, resultado de las proyecciones de los segmentos visibles de las aristas, que serán grafiados por medio del dispositivo de salida elegido. No necesitamos pues recortar las caras del objeto con los planos límites de visión, proceso complejo y que consume un mayor tiempo de ejecución, sino que será suficiente con recortar las aristas y proceder a la determinación de las condiciones de visibilidad de las mismas, mediante el análisis de su ocultación con las caras del archivo de trabajo. En el recorte comentado anteriormente, cuya finalidad era la eliminación de las partes del objeto situadas tras el observador, necesitábamos generar como resultado la información referente a los fragmentos de cara resultantes, y no a los segmentos de arista, pues es preciso usarlas ahora como tales caras, para determinar en que forma ocultan a cada una de las aristas; sin embargo en el caso de la pirámide visual, al estar ya todos los elementos frente al observador, ninguno de ellos produce proyecciones virtuales que perturben el proceso, por lo que las caras, al no ser utilizadas para el trazado del dibujo, sino solamente para determinar las ocultaciones, pueden sobresalir de la pirámide visual sin problema alguno, siendo suficiente con realizar el recorte de aristas, mucho más sencillo y rápido que el de caras.

Procedemos ahora a generar el archivo de trabajo que debe contener la información referente a cada una de las aristas situadas frente al observador. Para ello se realiza una lectura completa y secuencial del archivo de caras generado tras el recorte de las mismas con el plano frontal límite próximo de la visión, y asignamos valores en el lugar del archivo correspondiente para cada una de las aristas que forman los polígonos de dichas caras. En cada arista incluimos la información referente a los vértices

## Parte I      Capítulo VI

que la delimitan, junto con la que permita conocer las dos caras que se intersectan en ella. Para generar dicha información seguiremos el proceso siguiente: a partir de la cara que estamos considerando, obtenemos los datos referentes a los vértices extremos de una arista, dado que esta arista puede estar ya incluida en el archivo que estamos construyendo, leemos las aristas existentes en el mismo, si no encontramos ninguna con los mismos vértices, entonces la incluimos anotando la cara que estamos considerando como una de las dos que la contienen; si por el contrario encontramos una arista en el archivo, ya generada anteriormente, con los mismos vértices extremos, no incluiremos la que estamos considerando, para evitar duplicidades a la hora de procesar las ocultaciones, pero si incorporaremos a la información de la arista existente la cara que estamos procesando como segunda cara que la contiene.

Una vez completado el archivo de trabajo de aristas, realizamos el recorte de todas las incluidas en el mismo con los planos que delimitan la pirámide de visión, según lo expuesto en el capítulo cuarto, produciéndose una ampliación del archivo de vértices, al incorporarse los que se generan en las intersecciones con los planos de recorte, y variando la información referente a los extremos de las aristas que hayan sufrido recorte, colocando un testigo en las que hayan quedado totalmente fuera de la zona visible, para no procesar posteriormente su posible ocultación.

Tenemos pues el conjunto de aristas que, perteneciendo al objeto, se encuentran dentro de la zona de visión, con lo que procedemos ahora a la proyección de todos los vértices existentes en el archivo correspondiente, incluidos naturalmente los generados en los dos procesos de recorte. Representando gráficamente el archivo de aristas generado en el recorte, con las proyecciones de sus extremos

Parte I      Capítulo VI

aquí calculadas, tendremos la figura de alambre de la porción del objeto situada dentro de la pirámide de visión, de acuerdo con las especificaciones determinadas al inicio del proceso.

Llegamos al punto en que debemos considerar la visibilidad de cada una de las aristas que constituyen el archivo de trabajo formado a partir del recorte con la pirámide de visión. Para ello se deben analizar una a una todas las aristas de dicho archivo, comparándola con todas las caras incluidas en el archivo de trabajo correspondiente, determinando la porción de arista ocultada por cada cara, e interrumpiendo el proceso en el momento en que se detecte la ocultación total de la arista, por la cara en estudio, o por el conjunto de las estudiadas hasta el momento. El resultado de este análisis será la parte visible de la arista, que puede estar compuesta por uno o varios segmentos, los cuales son incorporados al archivo de trazado, el cual usaremos para obtener la representación gráfica en el dispositivo de salida que elijamos. Previamente a la determinación de si una cara en concreto oculta o no a la arista considerada, y en que forma se produce la ocultación, se procede a efectuar la operación de descarte correspondiente, intentando observar si se puede determinar fácilmente la imposibilidad de que se produzca dicha ocultación de acuerdo con los datos de que se dispone acerca de la arista y la cara en cuestión, caso de producirse el descarte de la cara, por la no posibilidad de ocultación, se pasa al análisis de la cara siguiente, evitando un proceso inútil, y economizando por tanto el tiempo que se hubiera empleado en el mismo, dando por supuesto que es mucho más rápido realizar el descarte que procesar la ocultación de la arista por la cara.

Una vez completado el archivo de los trazos resultantes que permitirán representar gráficamente la proyección de las partes vistas del objeto, se procede a su almacenamiento en un dispositivo de memoria permanente en el caso de querer conservarlos para posteriores utilidades, y a continuación, si se desea, utilizar la información contenida en este archivo para confeccionar el dibujo que nos permita visualizar la proyección obtenida, o el fragmento de la misma que delimitemos, representada en el dispositivo de visualización elegido, que en caso de tratarse de una trazadora gráfica, nos permitirá disponer de una copia del mismo sobre papel para ser utilizada con fines ajenos al proceso si así conviniera.

Como final del capítulo desarrollamos el algoritmo utilizado por el autor para determinar la parte vista de una arista determinada, mediante su comparación con las caras significativas del objeto, seleccionadas por su orientación hacia el observador.

## 2. Algoritmo de depuración de líneas ocultas.

Los datos de entrada utilizados por este algoritmo están constituidos por los archivos de trabajo de caras, polígonos, vértices en el espacio tridimensional, vértices proyectados y aristas. El algoritmo realiza el análisis de cada una de las aristas, obteniendo la información necesaria de la misma mediante una lectura secuencial y completa del archivo correspondiente. Consideramos el proceso que realiza el estudio de una arista, siendo el desarrollo completo del algoritmo la simple repetición del mismo para todas y cada una de las aristas.

Determinamos, en primer lugar, las coordenadas de los vértices que constituyen los extremos de la arista en el espacio tridimensional, así como las coordenadas en el plano del cuadro de las proyecciones sobre el mismo de dichos vértices. Definimos entonces los valores máximo y mínimo de las dos coordenadas en el plano de proyección, así como el valor de la tercera coordenada del vértice de la arista que se encuentre más alejado del plano XY.

Procedemos ahora a la lectura secuencial y completa del archivo de trabajo de caras, realizando el análisis del posible descarte de cada una de ellas según el proceso siguiente:

- Comprobamos en primer lugar si la cara es una de las dos cuya intersección determina la arista considerada, en cuyo caso descartamos la cara y pasamos a la siguiente.

- Dado que disponemos de la información referente a los límites de los valores de las coordenadas de los vértices del contorno de la cara, comparamos estos límites con los de la arista, y entonces realizamos el descarte de la cara en los casos siguientes: a) si la mayor Z de la arista es menor que la menor Z de la cara, en cuyo caso la cara se encuentra con toda seguridad situada posteriormente a la arista y por tanto no puede ocultarla. b) si alguno de los mayores valores de las coordenadas de la proyección de la arista es menor que su homólogo de los menores la proyección de los vértices del perímetro de la cara, o bien alguno de los menores valores de la proyección de la arista es mayor que su homólogo de los mayores de la cara, entonces podemos afirmar que las proyecciones de la arista y de la cara sobre el plano son disjuntas, y por tanto no puede producirse ocultación.

- En el supuesto de que no se haya producido el descarte de la cara, debemos proceder al estudio de la posible ocultación de la arista, lo cual se realiza según el proceso que se inicia a continuación:

Determinamos en el plano del cuadro las intersecciones de la recta que contiene a la proyección de la arista, con cada uno de los polígonos que constituyen la cara en cuestión, para lo cual procesamos la intersección de dicha recta con cada uno de los segmentos que forman el contorno de la proyección de cada polígono, lo cual hace necesario realizar los siguientes pasos:

- Determinar para cada uno de los puntos proyección de los vértices del polígono, en que lugar del plano de proyección se encuentra situado respecto de la recta, considerando lugar positivo uno de los dos semiplanos en que dicha recta divide al plano, lugar negativo el otro semiplano, y lugar nulo el de los puntos que se encuentran sobre la recta, o a una distancia menor que el límite de precisión prefijado.

- Con la información así obtenida calculamos las intersecciones de los lados del polígono con la recta, procesando únicamente aquellas aristas del polígono cuyos extremos están situados en lugares no nulos y de distinto signo. Para determinar si se debe calcular intersección o no, se analizan los lugares ocupados por los dos vértices consecutivos del perímetro del polígono que, en la lectura secuencial del mismo, estamos estudiando. Si los vértices en cuestión ocupan lugares de distinto signo, puede darse la circunstancia de que alguno de ellos tenga una posición de lugar nulo, si es en el segundo vértice donde se da esta circunstancia, seguiremos leyendo el contorno del polígono, hasta encontrar un vértice de lugar no nulo, en cuyo caso realizaremos el cálculo de la intersección de la arista en estudio si este vértice ocupa un lugar de signo distinto al del primer vértice, prosiguiendo posteriormente la lectura del polígono a partir de este último vértice analizado. Si el primer vértice de la arista ocupara una posición de lugar nulo no realizariamos el cálculo de la intersección.

## Parte I Capítulo VI

En el caso de haber obtenido intersecciones en el paso anterior, las introducimos en el archivo de trabajo preparado al efecto, comprobando previamente como medida de seguridad que el número de cortes sea par, pues si fuera impar detectaría algún error cometido en el proceso. Una vez introducidas, se procede a ordenarlas según su posición sobre la recta, determinando los cortes que forman pares consecutivos, de esta forma se consigue delimitar los segmentos de posible ocultación de la arista por la cara en estudio. Para cada uno de dichos pares se analiza si se produce ocultación de la arista por la cara o no, para lo cual se utiliza la porción del segmento definido por el par de cortes analizado que se encuentra entre los extremos de la arista cuya visibilidad analizamos, calculando las coordenadas en el espacio tridimensional del punto de la arista que se proyecta en el punto medio de la porción mencionada del segmento en estudio, siempre que dicha porción contenga un mínimo fragmento de la proyección de la arista, pues en caso contrario no procesamos la ocultación al no ser posible. En este momento se procede a analizar si el origen de coordenadas (que representa el punto de vista) y el punto de la arista determinado como hemos explicado, se encuentran en el mismo semiespacio (de los dos en que el plano que contiene a la cara en estudio divide al espacio), en cuyo caso no se producirá ocultación, si el punto de la arista se encontrara en el semiespacio opuesto al punto de vista, esto significaría que la cara se interponía entre ambos puntos, por lo que el fragmento de arista en estudio sería ocultado por la cara. En el supuesto de producirse ocultación, incorporamos el fragmento de la recta delimitado por los dos puntos de corte analizados como segmento oculto de la arista, creando un archivo con todos los segmentos ocultos que sean determinados de esta forma.

Parte I      Capítulo VI

Por último analizamos las porciones de la proyección de la arista que han sido ocultadas por la cara analizada, interrumpiendo el proceso en el supuesto de que haya sido ocultada en su totalidad, y modificando los límites superior e inferior que la arista tiene en este momento, utilizando estos nuevos límites para los descartes que se produzcan en adelante, con el fin de trabajar únicamente con la zona visible de la arista.

Una vez realizados los pasos descritos con todas las caras, incorporamos al archivo de trazado los segmentos de la proyección de la arista que hayan mantenido su visibilidad a lo largo de todo el proceso, y pasamos al estudio de la arista situada a continuación en el archivo correspondiente, hasta que alcancemos el final del mismo.

## ALGORITMO DE DEFURACION DE LAS LINEAS OCULTAS

Proceso General:

```
Para N = 1 hasta Número_de_aristas
  Determina valores Arista[N]
  Para C = 1 hasta Número_de_Caras
    Si Cara[C] No Contiene Arista[N] Entonces
      Descarta (Cara[C],Arista[N])
    Si No Descarta Entonces
      Oculta (Cara[C],Arista[N])
    Fin Si
  Fin Si
Fin Para
Determina Parte Visible de Arista[N]
Incorpora Parte Visible en Archivo de Trazado
Fin Para
```

Ocultar Arista con Cara

Repita (Para todos los polígonos de la Cara)

Polígono Siguiente

Repita (Para todos los vértices del Polígono)

Vértice Siguiente

Posición proyección (Vértice, PV)

Posición proyección (Vértice Anterior, PVA)

Si  $PV \neq PVA$  Entonces

Calcula Intersección

Guarda Intersección

Fin Si

Hasta No Vértices en el Polígono

Hasta No Polígonos en la Cara

Si Número\_Intersecciones  $> 0$  Entonces

Ordena Intersecciones

Repita (para todos los pares de intersecciones)

Siguiente Par

Si  $MaxPar > 0$  y  $MinPar < 1$  Entonces

Delante (Cara, Arista)

Si Delante Entonces

Incorpora Ocultación

Fin Si

Fin Si

Hasta No Intersección 0 MinPar  $> 1$

Fin Si

PARTE II

GENERACION DE OBJETOS

## 1. GENERACION DEL MODELO DE UN OBJETO

### 1. Adecuación del objeto al modelo.

Vimos, en la primera parte de este trabajo, el desarrollo de un proceso de visualización de sólidos tridimensionales poliédricos, con la eliminación optativa de sus líneas ocultas. Para ello se trabajaba a partir de una estructura de almacenamiento de la información del objeto que constituía un modelo de fronteras del mismo, al basarse en las caras planas que constituyen la "frontera" del sólido con el exterior como medio de descripción de las características formales del mismo.

Las características específicas del modelo de fronteras con que se trabaja, junto con las del equipo informático disponible, pueden dificultar la operatividad de objetos que no cumplan determinadas condiciones, normalmente cuantitativas, como puede ser el tener limitado el número de elementos, lo que imposibilita el introducir objetos con más caras, polígonos, aristas o vértices que las cantidades máximas de cada uno de ellos que admite el modelo de trabajo, lo cual viene vinculado normalmente a las condiciones del ordenador donde se encuentra implementado el programa.

Cabe la posibilidad de que, para mejorar los tiempos de proceso, sea conveniente introducir modificaciones en el modelo, y por tanto en los algoritmos que lo utilizan, como

puede ser el incorporar a la información referente a un polígono la que permite conocer directamente los vértices que lo componen, pues, si la consulta al archivo de aristas que era necesario realizar en el proceso descrito en la primera parte de esta tesis, como forma de acceso a dicha información, debe realizarse en un medio externo al programa, por no poder incluir dentro del mismo el archivo de aristas por falta de espacio libre en la memoria disponible, el tiempo que consume dicha consulta puede llegar a ser tan importante que afecte de forma sensible a la efectividad del proceso.

Vemos que deben considerarse determinadas condiciones restrictivas que acotan el ámbito de objetos procesable por el programa. Una considerable cantidad de estas restricciones desaparecerían o disminuirían de forma muy apreciable al trabajar con un equipo de mayor potencia y capacidad, otras, sin embargo, exigirían ligeras modificaciones del programa, al estar éste adaptado a las condiciones de trabajo que las prestaciones del equipo disponible imponen.

Por tanto, al desarrollar un sistema de trabajo, debemos establecer el campo de operatividad del mismo, estudiando las restricciones a imponer a los cuerpos que vayan a ser objeto de procesado por el mismo. Estas restricciones deben ser tales que no supongan limitaciones fundamentales, que pudieran limitar de forma importante la eficacia del sistema de acuerdo con el campo de aplicación a que se destine, pues unas restricciones que pueden hacer un sistema de muy poca utilidad para realizar proyectos industriales que necesiten la exacta definición de superficies curvas, por ejemplo, puede prestar servicios perfectamente aceptables en el desarrollo de proyectos de edificios, lo que conduce a la conclusión de que, a veces, debe sacrificarse la universalidad del campo de aplicación

de un instrumento, en aras a conseguir una mayor efectividad del mismo.

Una vez que el sistema se encuentra desarrollado y es operativo, si se desea utilizarlo para el estudio de un objeto, lo primero que es preciso tener en cuenta es si el mismo cumple las condiciones que lo hacen válido para ser operado por el sistema, pues, en el caso de incumplir una o varias de las restricciones del sistema, no obtendremos resultado alguno, o el resultado obtenido no tendrá la fiabilidad precisa para que sea de utilidad.

## 2. Introducción de la información.

Una vez comprobado que el cuerpo que deseamos procesar cumple los condicionantes de funcionamiento del sistema, debemos introducir en el ordenador la información necesitada por el modelo para la perfecta definición del objeto. Dicha información se compone de una enorme cantidad de datos numéricos que, aunque pueden introducirse manualmente, esto haría totalmente inoperante el proceso para objetos de una mínima complejidad. Veamos a continuación la información mínima que necesitamos introducir para poner en funcionamiento el sistema, en el caso de hacerlo sin contar con otros medios de ayuda:

- En primer lugar se debe introducir, de forma ordenada y de acuerdo con una numeración que habremos establecido previamente, para cada uno de los vértices del objeto, las coordenadas del mismo en el sistema de referencia en que lo tengamos definido.

- A continuación se define cada polígono dotándolo de un número de identificación, que normalmente será el que corresponda al del orden de su introducción, y listando ordenadamente, de acuerdo con el sentido de lectura correspondiente, los vértices de su contorno, haciendo

referencia al número que corresponde a cada vértice.

- Una vez completada la información referente a los polígonos, se procede a introducir la de las caras del objeto, con un número de identificación para cada una de ellas que, de forma similar al caso de los polígonos, coincidirá con el orden en que es introducida. Cada cara deberá contener la lista de polígonos que la componen, con la información referente al tipo y características especiales de cada uno de ellos, si fuese menester.

Como se desprende fácilmente de lo antedicho, esta introducción pormenorizada de la información necesaria para configurar el modelo, puede ser extraordinariamente tediosa cuando los elementos del objeto alcanza la cifra de unos pocos cientos, y si trabajamos con cuerpos de una mínima complejidad, o bien deseamos estudiarlos con un cierto detalle, los elementos pueden contarse por millares, teniendo en cuenta las limitaciones de un ordenador personal, pues la cifra puede crecer de forma espectacular si se dispone de un equipo con mayores prestaciones.

Al importante esfuerzo necesario para introducir toda la información mencionada en el párrafo anterior, se debe sumar el trabajo nada despreciable que supone el conocimiento, previo al procesado, de todos los datos mencionados, que en el caso de cuerpos complejos puede exigir un alarde de imaginación y de visión espacial digno de admiración.

Esta información debe tener carácter numérico, por lo que, en principio, es preciso introducirla por medio del teclado, aunque para una parte de la misma se podría disponer de otros dispositivos más eficientes, que actuaran como intermediario gráfico entre el ordenador y el operador del sistema, como puede ser una tableta digitalizadora, un lápiz óptico o un ratón, que permitirían dibujar cada cara

sobre un determinado soporte, y transmitirían esta información gráfica al ordenador el cual puede disponer del algoritmo que le permita traducirla a cantidades numéricas a incorporar a los archivos que constituyen el modelo.

Afortunadamente no es preciso realizar todas las operaciones mencionadas para utilizar el sistema, pues la información referente a determinados cuerpos, de los que se prevea la utilidad de su posterior utilización para generar otros objetos, puede ser guardada y recuperada sin necesidad de repetir el laborioso proceso de su introducción.

### 3. Concepto de primitiva.

El conjunto de la información referente al modelo de un objeto, definido en un sistema de referencia conocido y vinculado al mismo, como puede ser su centro de gravedad o cualquier punto significativo del mismo, que permita su utilización en la generación de otro cuerpo, constituye lo que denominamos primitiva, y es de importancia fundamental para la existencia de sistemas de generación de modelos de sólidos mediante ordenador, con una eficiencia que permite trabajar satisfactoriamente con ellos.

Consideremos pues que disponemos, ya que no revisten excesiva dificultad, de los algoritmos necesarios para generar los poliedros más simples, como pueden ser un prisma o una pirámide a partir de la definición de la base y de la directriz, o el vértice superior en el caso de la pirámide, desarrollándose a lo largo del proceso la información referente a los elementos de dichos poliedros y estructurándose de acuerdo con las especificaciones del modelo que utilizamos. Podremos entonces, si disponemos de los instrumentos necesarios, crear otros cuerpos que se

compongan de un conjunto de los anteriores, obteniendo sus modelos de donde estaban almacenados, y ubicándolos en el lugar correspondiente, orientándolos de distinta forma, o incluso realizando operaciones de escalado, cizalladura o simetría previamente a su utilización. Tendremos nuevos cuerpos más complejos que podremos utilizar como fin en sí mismos, o que convertiremos en primitivas de segundo orden con la intención de que puedan ser utilizadas posteriormente en la misma forma.

Vemos que ahora la generación de un cuerpo se realiza mediante la introducción de la información que define la creación de una nueva primitiva, o bien que recupera la correspondiente a una primitiva existente y, si queremos deformar, orientar o simplemente ubicar la primitiva es suficiente con proporcionar la información que define estas operaciones en la forma que nos la solicita el sistema.

#### 4. Comunicación sistema-usuario.

Para manejar un sistema basado en la utilización de primitivas, solo es necesario suministrar al mismo las instrucciones necesarias para que realice las operaciones deseadas. Esto se realiza a través de los dispositivos de que el sistema dispone para comunicarse con el exterior, emitiendo mensajes que reclaman un determinado tipo de información, y recibiendo del usuario las instrucciones que determinan la operación a realizar o el valor del dato solicitado.

La claridad en la solicitud de información por parte del sistema es un condicionante de su operatividad pues, si se duda sobre el tipo de información solicitada en un momento determinado, esto puede colapsar el sistema, al introducir una información que conduzca al absurdo o que

pueda producir resultados no deseados por el usuario. También debe tenerse en cuenta la precisión con que trabaja el sistema, pues si ésta es alta, podrán solicitarse prestaciones que de otro modo darían resultados de escasa utilidad.

La admisión de información por parte del sistema depende en gran parte de los dispositivos de que disponga pues, si el teclado constituye el único medio de comunicación del usuario con el ordenador, la información no podrá ser introducida gráficamente de una forma eficiente, pues el único medio consiste en la generación de algoritmos que admitan la generación de gráficos en la pantalla mediante el uso del cursor. Si, por el contrario, se dispone de lápiz óptico o ratón, la información podrá generarse de forma gráfica en la pantalla con mucha mayor comodidad y rapidez, teniendo como límite de precisión la definición del monitor con que se trabaje. Cuando se disponga de tableta gráfica o mesa digitalizadora, esta precisión puede aumentar, pero siempre estará limitada por el dispositivo utilizado.

La comunicación entre el sistema y el usuario se establece en forma de diálogo, en el cual el sistema ofrece una serie de opciones, y en función de la que resulta elegida por el usuario reclama los datos que necesita para poder realizar la tarea encomendada. Debe procurarse que la oferta de opciones sea lo mas clara posible, permitiendo explotar al máximo las posibilidades que el sistema posea, y al mismo tiempo es de desear que la petición de información no sea redundante, limitandose a lo estrictamente necesario para operar con la eficacia y precisión máximas que el sistema permita. Es deseable que los menús de opciones no sean excesivamente amplios, siendo preferible la organización de los mismos mediante una estructura en árbol. Es asimismo más eficiente la

presentación de las ofertas de los menús de forma icónica, si la definición de la pantalla lo permite.

Veremos en el capítulo siguiente la descripción del sistema de generación de volúmenes desarrollado por el autor, como ejemplo del funcionamiento de sistemas basados en el manejo de primitivas simples, mostrando las posibilidades de un sistema realizado con un equipo de limitadas posibilidades, y explorando lo que puede llegar a conseguirse con la utilización exhaustiva de una única primitiva básica.

## 2. DESCRIPCION GENERAL DEL SISTEMA PROPUESTO

### 1. Restricciones del sistema propuesto.

Comentamos en el capítulo anterior que todo sistema tiene una serie de restricciones, que limitan el campo de aplicación del mismo al conjunto de cuerpos que las cumplen. Haremos a continuación algunas reflexiones sobre las condiciones de trabajo del equipo utilizado, las cuales han aconsejado imponer ciertas limitaciones que permiten mejorar el rendimiento del proceso.

Dada la escasa capacidad de memoria interna utilizable por el usuario, limitada a 64K junto con otras 64K de variables internas del programa, ha sido necesario replantearse la opción de conservar como variables internas, y de rápido acceso por tanto, los archivos que contienen toda la información relativa al modelo, ubicándolos en ficheros externos al programa, ya que, en caso contrario, el número de elementos disponibles es mínimo. Esta decisión incrementa de forma considerable el tiempo de proceso, pero en cambio permite trabajar con objetos de suficiente complejidad al permitir manejar una mayor cantidad de información. Pero la posibilidad de utilizar como RAmDisk una parte importante de la memoria Ram disponible, casi 0.5M, permite tener un tiempo de acceso a la información sensiblemente mayor que si fuera memoria externa de almacenamiento masivo, aunque el dimensionado de los ficheros está más limitado.

Considerando todo lo expuesto, se ha optado por utilizar el RamDisk como lugar donde situar toda la información referente al modelo, permitiendo el procesado de temas que, aunque limitados en la cantidad de elementos de que se componen, esta limitación no es dramática, pudiéndose operar con objetos razonablemente complejos, y en un tiempo más aceptable que el necesario en el caso de trabajar con memoria externa.

De lo anterior se desprende que todo lo que permita reducir la frecuencia de consultas a los archivos, será una contribución importante a la disminución del tiempo de procesado, por lo que se han introducido algunas modificaciones a lo expuesto en la primera parte de esta tesis sobre la estructura que configura el modelo a utilizar.

Las consultas que se ha decidido evitar son las que se producen cuando se necesita conocer los polígonos que componen una cara, o bien las aristas que delimitan un polígono, lo cual, con la estructura de datos explicada en la primera parte, obliga a realizar lecturas en dos ficheros, en el del elemento de nivel superior, para conocer la dirección del primer elemento componente del mismo y el número de ellos, y en el de los componentes para realizar la lectura secuencial de los que corresponden al que estamos estudiando.

La forma de conseguir suprimir estas lecturas se basa en incorporar a la información de un elemento la correspondiente a las direcciones de todos sus componentes del nivel inmediatamente inferior, lo cual es especialmente útil en el caso de los polígonos, ya que permite acceder una sola vez al archivo de los mismos, obteniendo así las direcciones de todos los vértices que configuran su

perímetro, consultandono entonces ya directamente el archivo de vértices, sin necesitar la existencia del de aristas, y por tanto evitando su utilización.

Esto obliga a limitar el número máximo de elementos que pueden formar parte de uno de nivel superior, lo cual es una limitación muy restrictiva en general, pero que, en nuestro caso, podemos considerar admisible por facilitar extraordinariamente el proceso, sin limitar de forma inaceptable la gama de formas a procesar, pues, en el campo de los temas arquitectónicos, con formas poliédricas simples se pueden componer gran cantidad de formas habitualmente utilizadas, no dejando de ser útil como instrumento de análisis y proyecto por el hecho de estar obligado a utilizar formas geométricas de poca complejidad, pudiendo conformarse con ellas espacios de suficiente riqueza y expresividad.

## - 2. Primitiva básica del sistema.

Por todo lo antedicho, y considerando además que lo importante es desarrollar un sistema de generación de sólidos mediante la manipulación de primitivas, se ha decidido utilizar como primitiva básica un paralelepípedo de dimensiones no prefijadas y con las caras paralelas a los planos coordenados.

Para introducir la información que posibilita la definición por el sistema del paralelepípedo concreto que se desea generar, basta con suministrar como datos las coordenadas de los dos vértices extremos de la diagonal principal que tiene sus cosenos directores positivos, los cuales tienen como valores los mínimos y los máximos respectivamente de las coordenadas de todos los vértices del cuerpo. A partir de esta información el sistema genera

la correspondiente a las caras, polígonos y vértices del paralelepípedo configurando así su modelo de acuerdo a la estructura utilizada.

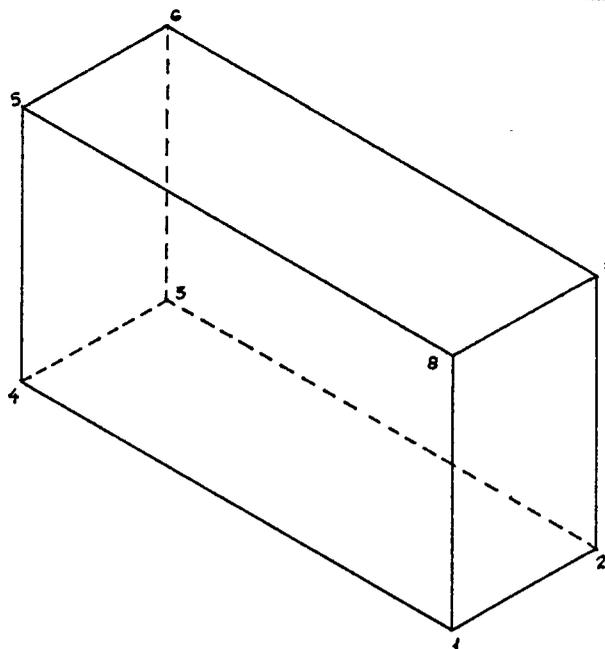
En la figura II.II.2.1 se aprecia la numeración que el sistema adjudica a todos y cada uno de los vértices del paralelepípedo, así como la relativa a los vértices que constituyen cada polígono, siendo la de cada cara inmediata, por estar constituida, de momento, por un solo polígono, que tiene su mismo número de identificación. De forma similar se refleja en la figura II.II.2.2 la información generada por el sistema en el supuesto de existir un agujero pasante que perfora el paralelepípedo básico, pudiendo observarse que, en este caso, algunas caras constan de dos polígonos.

Las posibilidades que ofrece la generación de formas arquitectónicas mediante paralelepípedos ortogonales paralelos entre si, siendo ciertamente interesantes, son bastante limitadas, por lo que es necesario intentar explotar al máximo las capacidades de manipulación de los mismos que el sistema permite y que se describen en el apartado siguiente.

### 3. Actuaciones sobre la primitiva básica.

Todas las operaciones que nos permiten modificar controladamente el elemento básico con que trabajamos, enriquecen el repertorio de formas geométricas a utilizar por el sistema, con lo cual se amplían las posibilidades que el mismo ofrece al usuario.

Vamos a considerar dos tipos de operaciones sobre la primitiva básica del sistema, aquellas que producen perforaciones en sus caras, y aquellas que deforman el



VERTICES

POLIGONOS

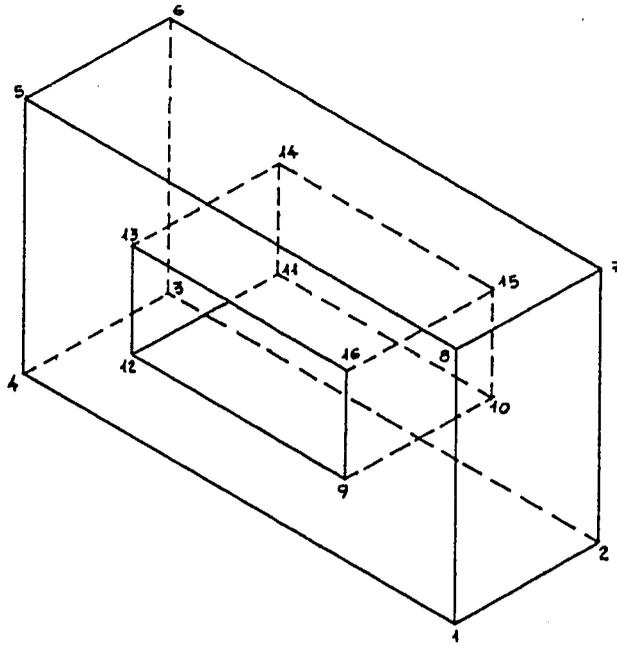
CARAS

n.	Cx.	Cy.	Cz.
1	0	0	0
2	60	0	0
3	60	0	180
4	0	0	180
5	0	100	180
6	60	100	180
7	60	100	0
8	0	100	0

n.	vért.
1	1, 8, 5, 4
2	1, 4, 3, 2
3	1, 2, 7, 8
4	2, 3, 6, 7
5	5, 8, 7, 6
6	3, 4, 5, 6

n.	polg.
1	1
2	2
3	3
4	4
5	5
6	6

Figura II.II.2.1



VERTICES

POLIGONOS

CARAS

n.	Cx.	Cy.	Cz.
1	0	0	0
2	60	0	0
3	60	0	180
4	0	0	180
5	0	100	180
6	60	100	180
7	60	100	0
8	0	100	0
9	0	30	45
10	60	30	45
11	60	30	135
12	0	30	135
13	0	70	135
14	60	70	135
15	60	70	45
16	0	70	45

n.	vért.
1	1, 8, 5, 4
2	1, 4, 3, 2
3	1, 2, 7, 8
4	2, 3, 6, 7
5	5, 8, 7, 6
6	3, 4, 5, 6
7	13, 16, 9, 12
8	11, 12, 9, 10
9	15, 10, 9, 16
10	14, 11, 10, 15
11	15, 16, 13, 14
12	13, 12, 11, 14

n.	polg.
1	1, 7
2	2
3	3
4	4, 10
5	5
6	6
7	8
8	9
9	11
10	12

Figura II.II.2.2

paralelepípedo variando sus ángulos y convirtiendolo por tanto en una figura que ya no tiene todas sus caras paralelas dos a dos.

Pueden efectuarse dos tipos de perforaciones, pasantes y no pasantes, pero con la limitación de que no se trata de un algoritmo de modelado, por lo que el sistema no contempla las posibles intersecciones que pudieran producirse, no generando las aristas y vértices que se formarían producto de las mismas, siendo el usuario quien debe controlar las perforaciones que introduce a fin de que no se produzcan intersecciones entre las mismas, ni en la superficie ni en el interior del paralelepipedo. Cualquiera que sea el tipo de perforación que realicemos, el sistema solo admite que la forma de dicha perforación sea la de un paralelepípedo con las caras paralelas a los planos coordenados, y tambien paralelas por tanto a las del paralelepípedo perforado.

Veamos ahora la información que es necesario suministrar al sistema para que el mismo complete la estructura de datos referente a la perforación y la incorpore a los datos que ya figuran en los archivos del modelo de la primitiva en proceso de creación:

- En el caso de una perforación pasante, que en adelante denominaremos "agujero", el primer dato necesario es la dirección del mismo, que se facilita dando el eje al cual es paralelo, con lo cual el sistema ya conoce una de las coordenadas de cada uno de sus vértices extremos, siendo preciso por tanto facilitar las otras dos, lo cual se realiza mediante la introducción de los valores de las mismas de forma similar al caso del paralelepípedo básico, teniendo como información redundante las coordenadas conocidas, lo que sirve de comprobación de posibles errores en la introducción de los datos. El sistema, a partir de esta información, completa el resto necesario y lo

incorpora al modelo existente.

- Cuando se trata de una perforación no pasante, que en adelante denominaremos "hueco", es preciso determinar, no solo la dirección del mismo mediante el eje de coordenadas al cual es paralelo, sino también en cual de las dos caras del paralelepípedo perpendiculares a dicho eje se sitúa el hueco, con lo cual el sistema conoce una coordenada de uno de los dos vértices extremos del paralelepípedo que determina el hueco, aunque introduciremos la información completa referente a los dos vértices extremos, considerando la repetición de la coordenada conocida como un medio de detectar posibles errores en la introducción de datos.

Analizaremos ahora las posibilidades que el sistema brinda para efectuar deformaciones en el paralelepípedo básico, teniendo en cuenta que dichas deformaciones no afectan a las perforaciones que puedan existir en el mismo, con la intención de simplificar el proceso, ya que en caso contrario la complejidad de la operación restaría eficacia a la misma. Por el mismo motivo, solo actuamos sobre la cara superior del paralelepípedo, mediante traslaciones predeterminadas de sus vértices según el tipo de operación elegido, lo cual comporta la deformación de esta cara superior y algunas o todas las laterales, dejando invariable la cara inferior. Mediante los operadores de transformación que serán descritos en el apartado siguiente, podremos variar la posición del paralelepípedo de forma que la actuación parezca haberse realizado en cualquier otra cara.

Tiene establecidas el sistema cinco opciones para esta transformación, que, a partir del paralelepípedo inicial, determinan cinco poliedros distintos, siendo necesario suministrar la información referente al tipo de deformación a aplicar, y el porcentaje de desplazamiento de los

vértices de la cara superior que definen exactamente el poliedro resultante.

Se han aplicado, a modo de ejemplo, las cinco transformaciones, con una magnitud del sesenta por ciento, a un conjunto de cuatro paralelepípedos de distintas dimensiones, para que puedan observarse claramente su efecto sobre ellos. Se muestra, en la parte superior de cada figura, el conjunto sin deformar, y en la inferior el resultado de haber aplicado la deformación. Algunos paralelepípedos se sitúan girados respecto a la posición en que les ha sido aplicada la transformación con el fin de hacer más patente el resultado. La referencia a las figuras se realiza en el párrafo correspondiente a la descripción de cada una de las deformaciones.

La primera de ellas consiste en inclinar la cara vertical paralela al plano XY y más alejada del mismo. Para ello se produce un desplazamiento hacia el citado plano de los dos vértices de la cara superior más alejados del mismo, deslizándolos sobre las dos aristas perpendiculares al plano y que los contienen, quedando por tanto más cortas, y convirtiendo las dos caras verticales de las que forman parte en dos trapecios. La magnitud del desplazamiento se determina por el porcentaje de disminución de la longitud de las aristas que se acortan, al deslizar por ellas los vértices afectados por la deformación, dejando inalterable el objeto una deformación de magnitud cero, y en el caso límite de una magnitud del cien por ciento, la cara superior quedará reducida a una línea y las laterales a dos triángulos rectángulos, al coincidir los dos vértices que se desplazan con los otros dos de la cara superior. Puede observarse gráficamente el efecto de esta deformación en la figura II.II.3.1. Esta opción, como se observa fácilmente, consiste en disminuir el valor de la tercera coordenada de los dos vértices que

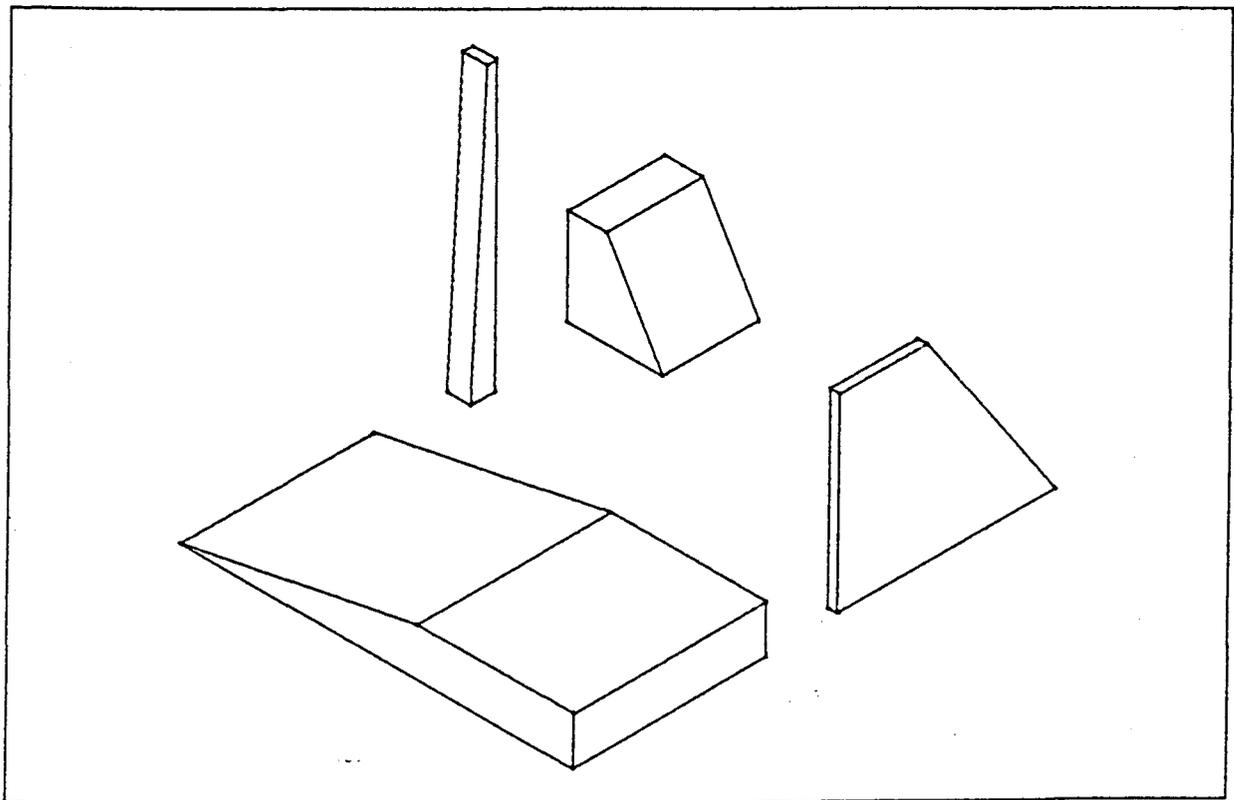
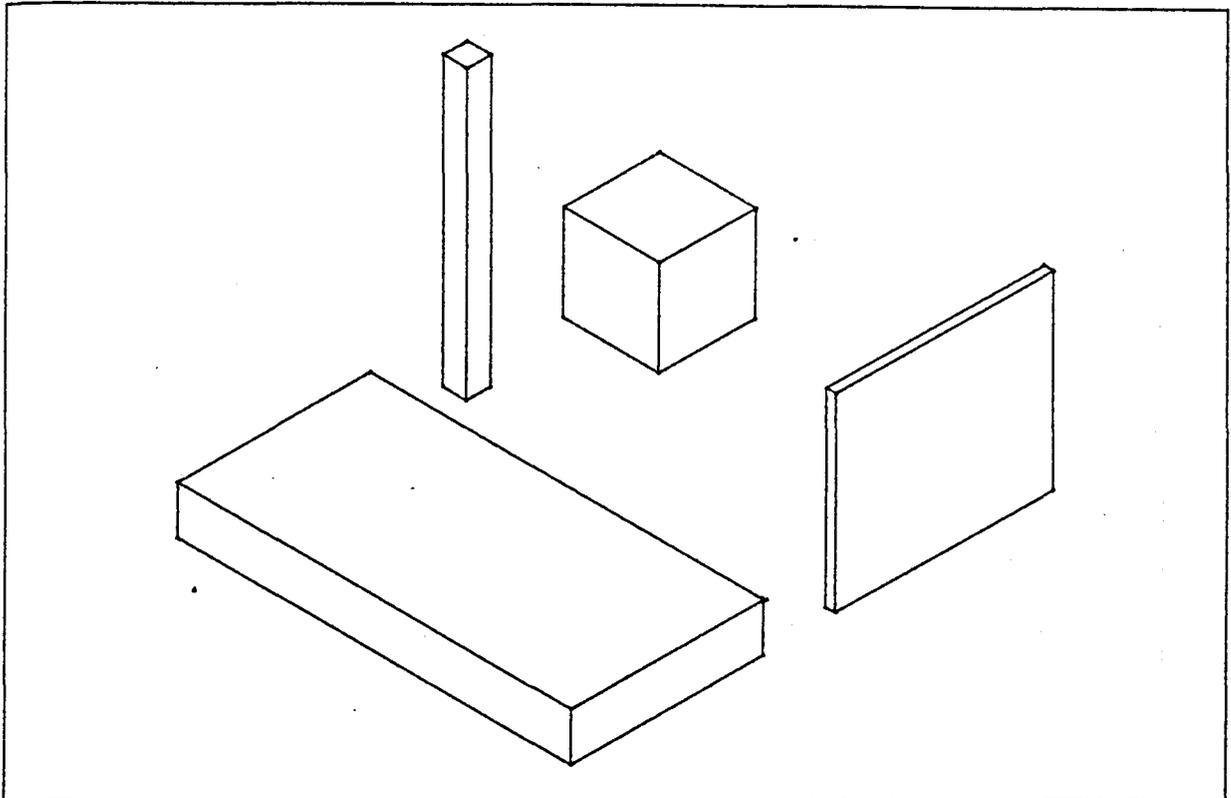


Figura II.II.3.1

se desplazan, en una magnitud determinada por la longitud de la arista y por el porcentaje fijado por el usuario.

La segunda deformación que podemos aplicar consiste en inclinar, de forma similar a lo descrito en el párrafo anterior, las dos caras del paralelepípedo paralelas al plano XY, mediante el deslizamiento, a lo largo de las aristas perpendiculares al mismo en la cara superior, de los cuatro vértices de la misma, desplazándose hacia el plano los dos vértices más alejados del mismo, y en sentido contrario los dos más próximos, afectando la deformación únicamente a las terceras coordenadas de los vértices de la cara superior, en lo que a la estructura de datos se refiere. Las consecuencias son, similarmente al caso anterior, la conversión en trapecios de las dos caras verticales paralelas al plano YZ, y la inclinación de las dos caras paralelas al XY, que por tanto dejan de serlo. Se reflejan en la figura II.II.3.2 las consecuencias de este tipo de deformación. Como puede observarse, el efecto de una deformación de magnitud máxima (cien por ciento), sería la conversión de la cara superior en una arista, situada en el centro del recorrido realizado por los vértices en la deformación, quedando las caras laterales reducidas a triángulos isósceles, con la base y altura del rectángulo inicial.

La tercera de las deformaciones disponibles en el sistema, al igual que la anterior desplaza los cuatro vértices de la cara superior en el sentido del eje Z, los dos más alejados del plano XY acercándose al mismo, y los dos más próximos en sentido contrario, pero además desplaza los dos vértices más alejados del plano YZ acercándolos al mismo, convirtiendo las caras laterales del paralelepípedo inicial en trapecios, manteniéndose vertical únicamente la cara paralela al plano YZ cuyos vértices tienen una menor coordenada X. La deformación máxima de este tipo convierte

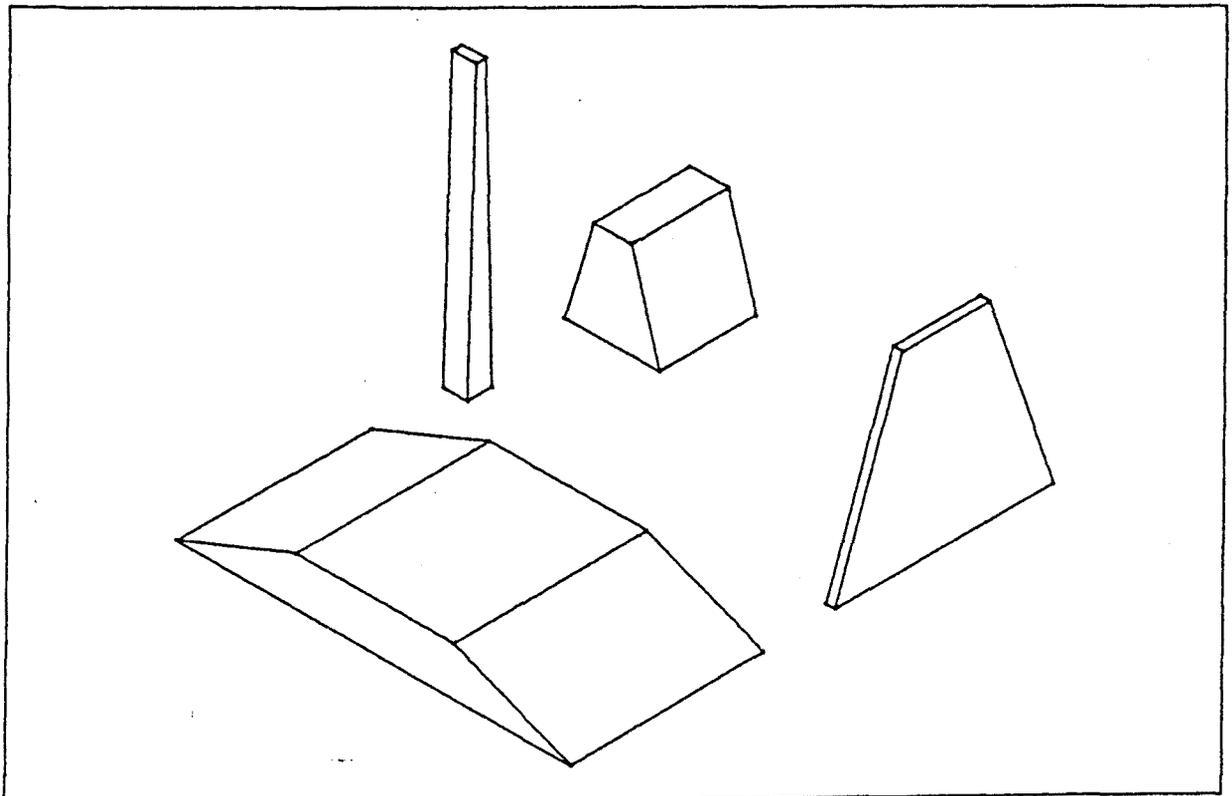
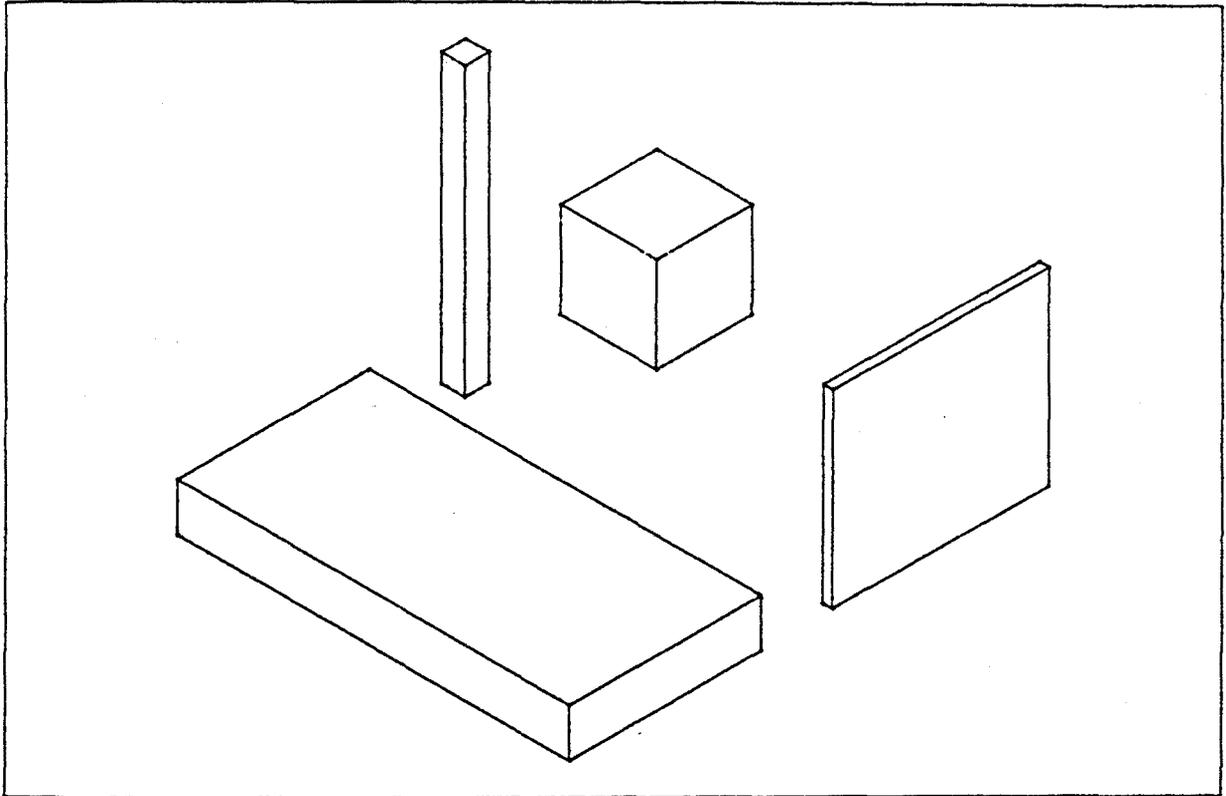


Figura II. II. 3.2

la cara superior en un punto, estando situado en el centro de la arista de la cara superior paralela al eje Z y de menor coordenada X, convirtiéndose en este caso las caras laterales en triángulos, y por tanto el paralelepípedo en una pirámide con el vértice desplazado. Puede observarse en la figura II.II.3.3 el efecto de una deformación de este tipo sobre los mismos paralelepípedos que en los casos anteriores, con el fin de comparar los efectos producidos.

La cuarta posibilidad que ofrece el sistema para deformar el volumen básico de trabajo, consiste en desplazar los cuatro vértices de la cara superior a lo largo de las diagonales del rectángulo que la configura, siendo este desplazamiento igual al producto de la longitud de una semidiagonal por el porcentaje decidido por el usuario. El resultado es un tronco de pirámide regular, cuya base es la cara inferior del paralelepípedo, y la altura la del mismo. Esta deformación, en lo que a la estructura de datos se refiere, y como todas las demas, modifica únicamente las coordenadas de los vértices de la cara superior del paralelepípedo, dejando inalterada la coordenada Y de las mismas. La magnitud máxima de la deformación aplicada a la figura básica, la convierte en una pirámide de base rectangular con el vértice centrado respecto de la misma, al convertirse la cara superior en un punto. En la figura II.II.3.4 se refleja el efecto de la deformación como en los ejemplos anteriores.

La quinta y última deformación que el sistema permite utilizar es muy similar a la anterior, pero manteniendo inalterada la arista vertical con valores mínimos de X y Z, desplazando los otros tres vértices hacia ella, dos a lo largo de las aristas y el tercero sobre la diagonal de la cara, transformándose el paralelepípedo en el tronco de una pirámide de la cual una de sus aristas verticales sería la que permanece inalterada tras la deformación. La

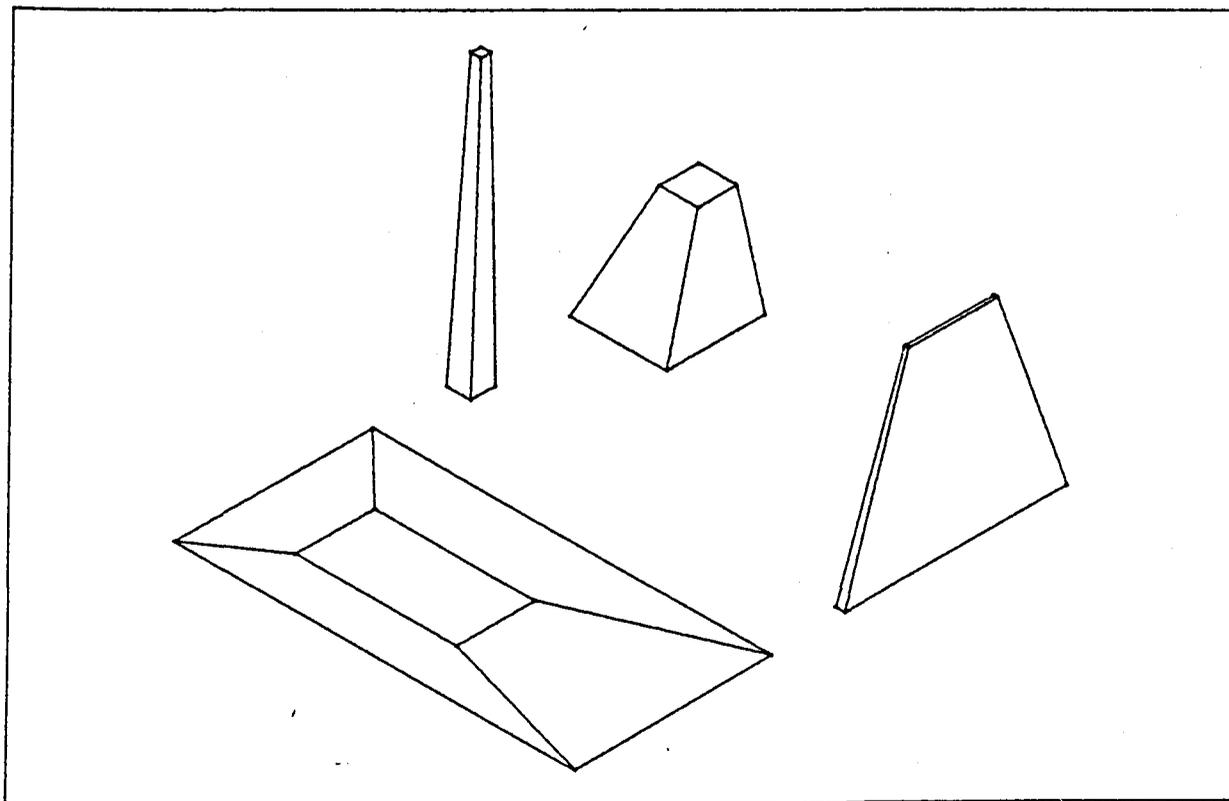
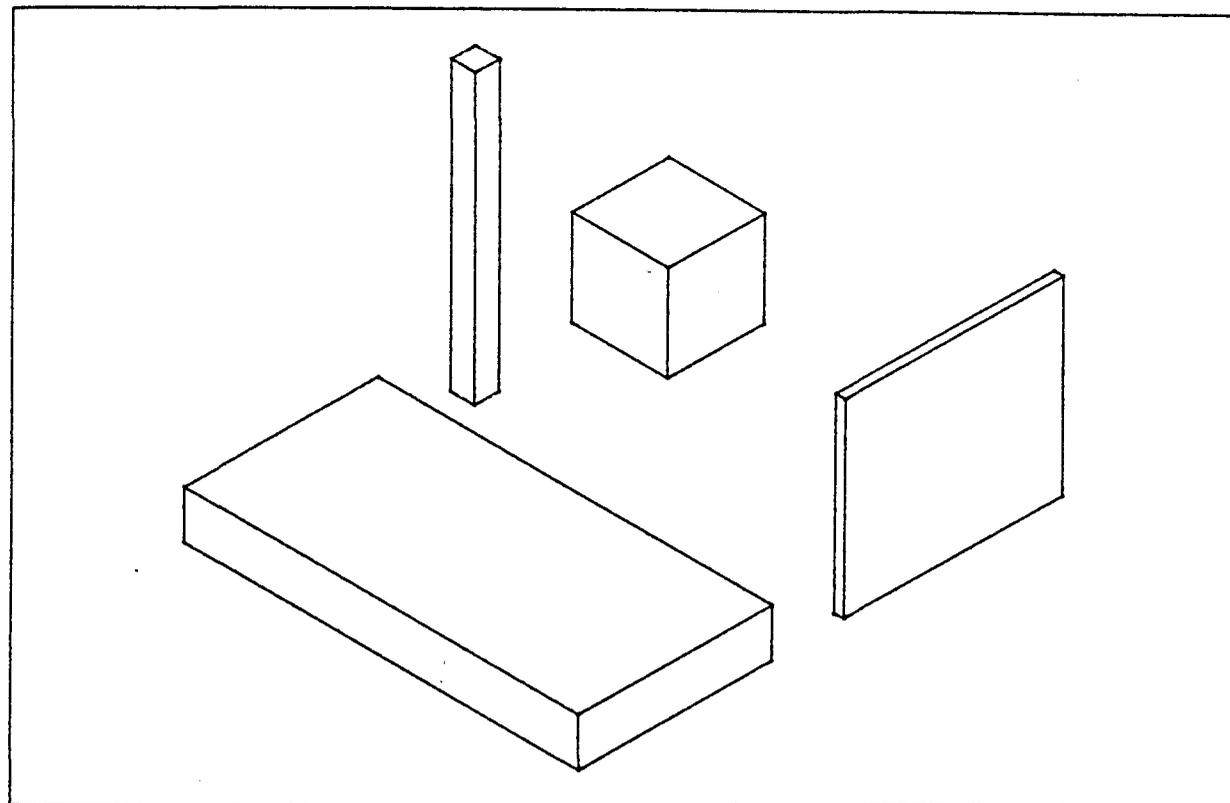


Figura II.II.3.3

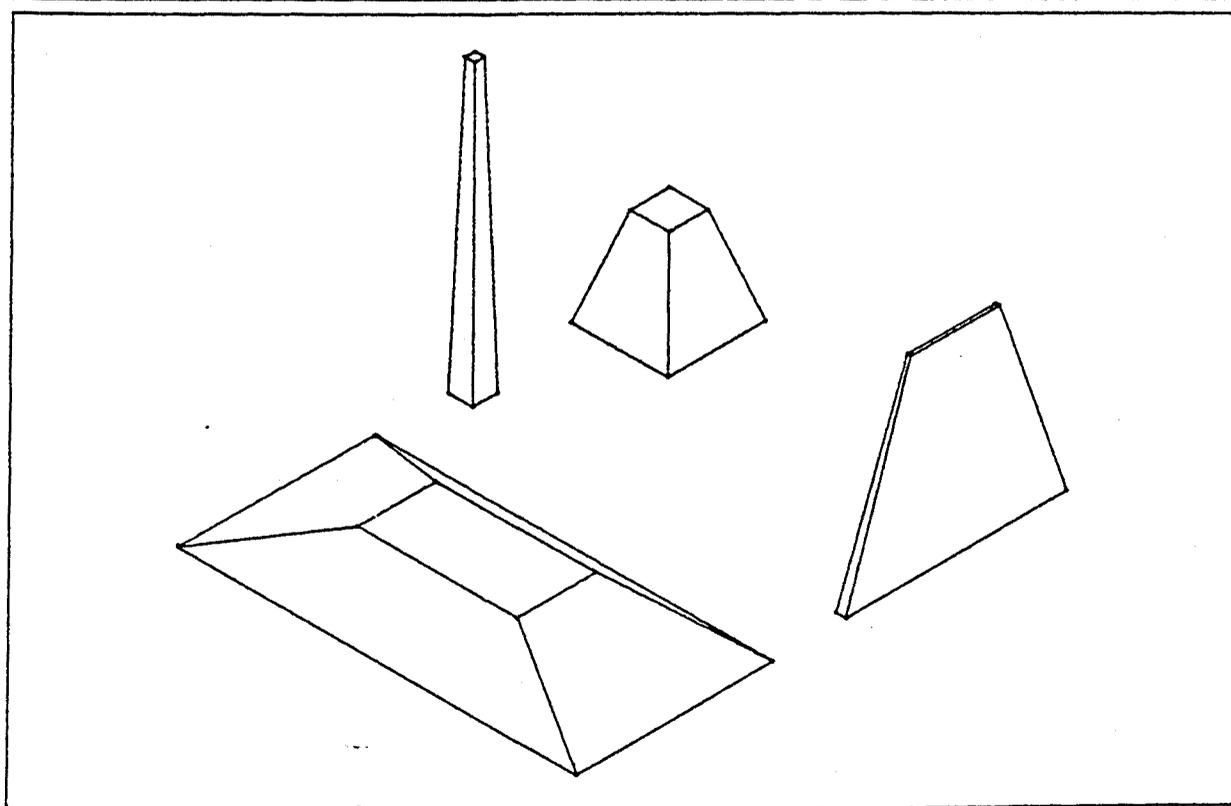
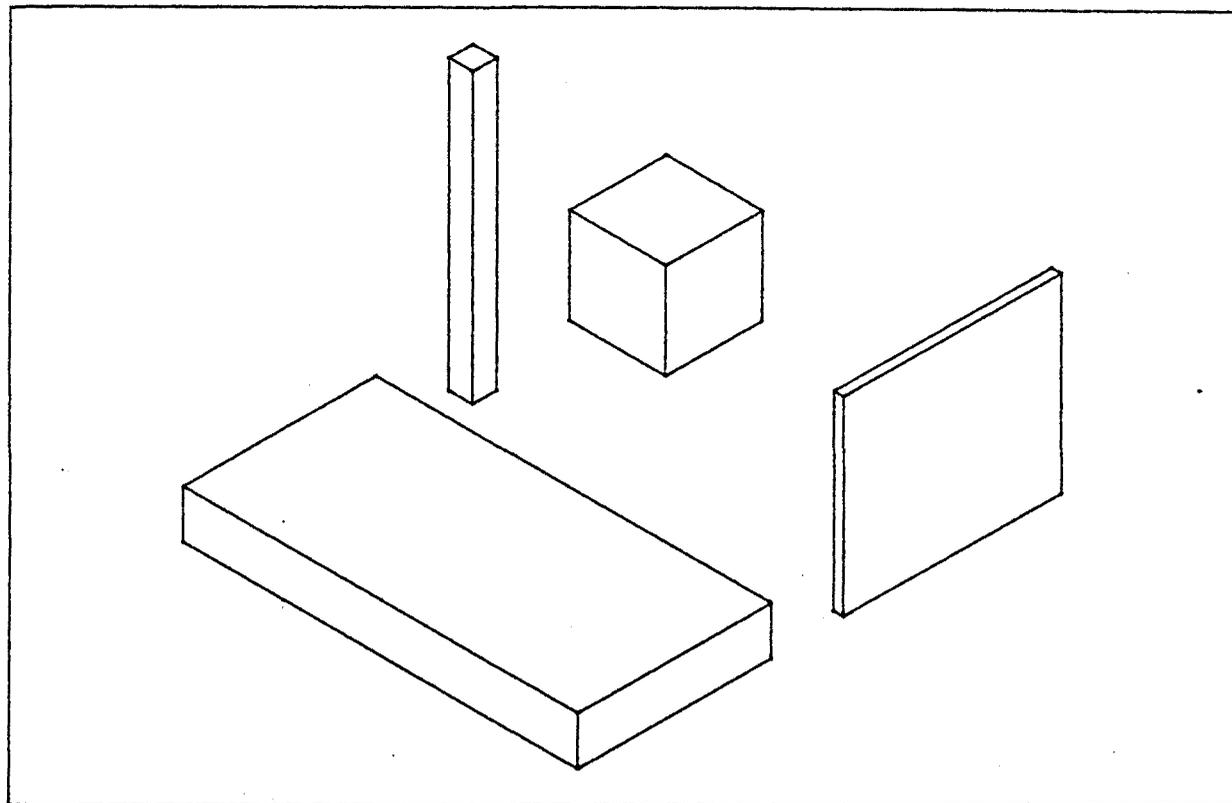


Figura II.II.3.4

deformación aplicada en su máxima magnitud nos da como resultado una pirámide con vértice superior coincidente con el del paralelepípedo que ha conservado su posición. Como en los casos anteriores, se puede observar el efecto de la deformación en la figura II.II.3.5.

Con las opciones del sistema descritas hasta el momento, el usuario puede generar un paralelepípedo ortogonal, paralelo a los planos coordenados, situado en cualquier lugar del espacio y de dimensiones no predeterminadas. Una vez generado el mismo, se ha visto la forma de introducirle agujeros y huecos, también paralelepípedos, en las caras que deseemos, con la limitación, debida evidentemente a la capacidad de memoria del equipo utilizado, de un máximo de seis perforaciones en cada cara, otra posibilidad que se nos ha ofrecido es la deformación, según cinco alternativas, del volumen básico sin afectar a las perforaciones. Pero una vez determinado de esta forma un poliedro, todavía el sistema nos permite realizar más operaciones con el mismo, las cuales se describen en el apartado siguiente.

#### 4. Transformaciones disponibles en el sistema.

Para poder configurar un tema complejo, debemos utilizar un conjunto de poliedros generados según se ha explicado en el apartado anterior, pero este proceso será más eficiente si existe la posibilidad de desplazar libremente por el espacio los objetos así generados, y mucho mejor si además se les puede aplicar determinadas transformaciones que también afecten a su forma.

Las transformaciones que el sistema utiliza para este propósito son de dos tipos: posicionales y formales, es decir, que afectan únicamente a la posición del objeto en

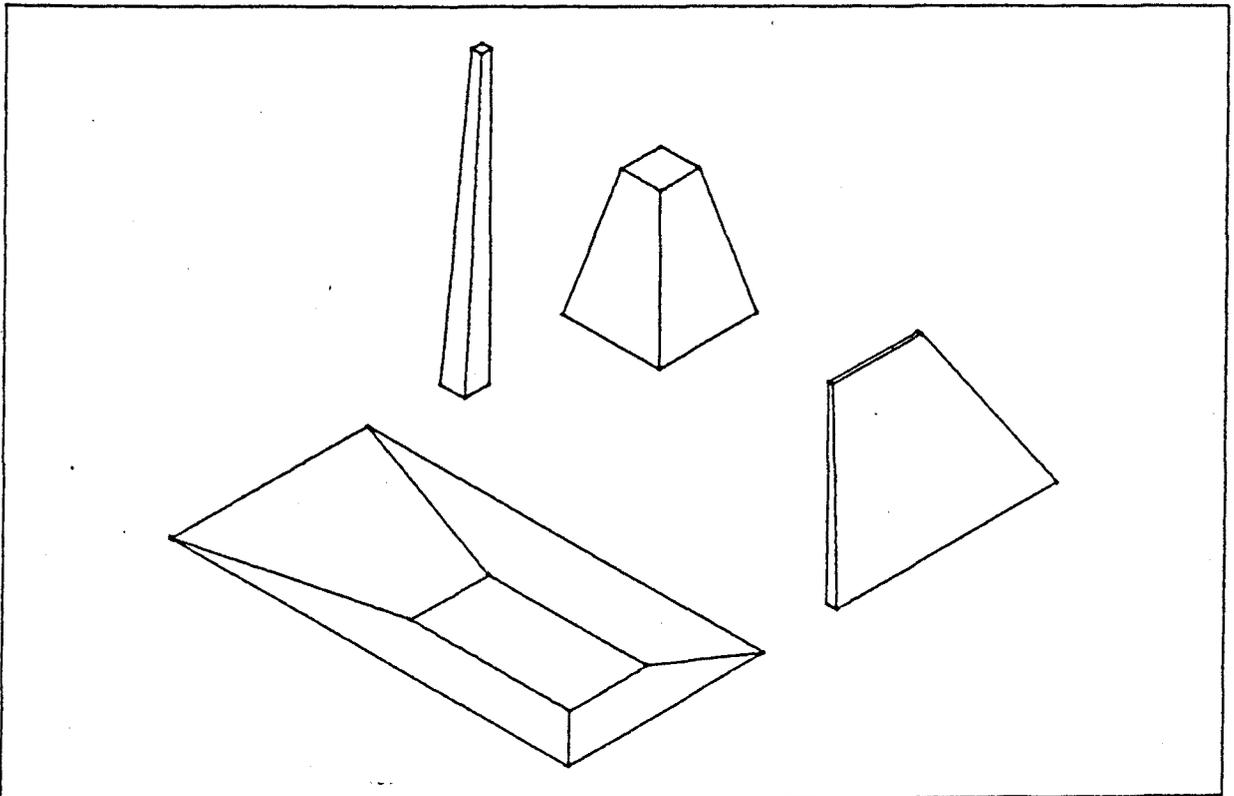
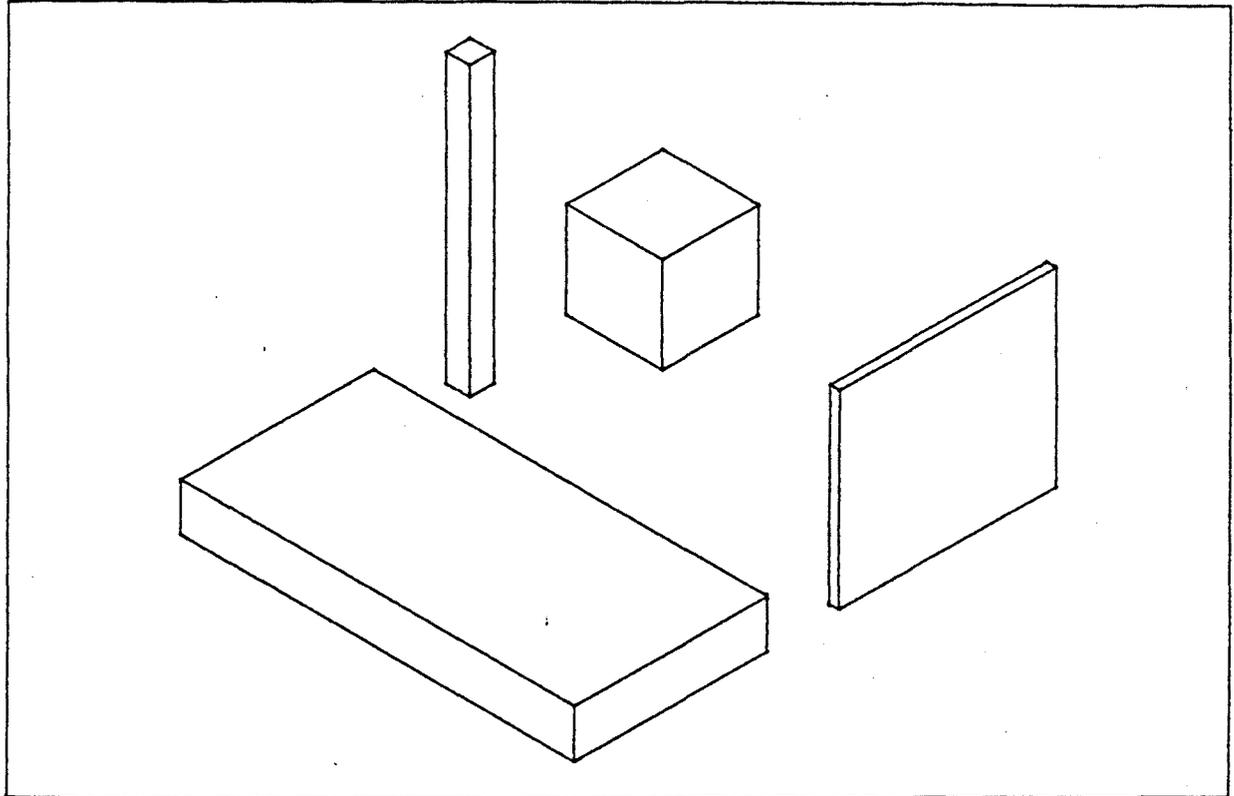


Figura II.II.3.5

el espacio, pero dejando inalteradas las relaciones entre sus partes, o por el contrario que, independientemente de si varían la posición o no del objeto, cambian las características geométricas propias del mismo.

Las transformaciones posicionales disponibles son la traslación y el giro en torno a uno cualquiera de los ejes de coordenadas. Para realizar una traslación, la única información que el sistema necesita conocer consiste en los valores de las tres componentes, según los tres ejes coordenados, del vector que la define, de esta forma el sistema sumará a las coordenadas de cada uno de los vértices pertenecientes al objeto el valor de la componente correspondiente del vector traslación, y así el objeto pasará a ocupar una nueva posición en el espacio, desplazada linealmente de la anterior una magnitud igual al vector traslación. La información precisada por el sistema para definir las características de un giro, consiste en determinar alrededor de que eje se realiza el mismo, así como el ángulo de rotación deseado, contando que el giro se realizará en sentido horario mirando, a lo largo del semieje positivo, hacia el origen de coordenadas.

Las transformaciones formales que el sistema pone a disposición del usuario son: la simetría respecto de uno cualquiera de los planos coordenados, el escalado y la cizalladura. Veamos a continuación qué información se precisa para poder determinar cada una de dichas transformaciones, de forma que pueda ser realizada por el sistema.

Para llevar a cabo una simetría, solamente se precisa determinar cual de los planos coordenados actúa como plano de simetría, para lo cual se introduce la información referente al eje perpendicular al mismo.

El escalado necesita definirse mediante tres valores positivos, que son los coeficientes de variación dimensional respecto de cada uno de los ejes coordenados, siendo un crecimiento de las dimensiones del cuerpo cuando el coeficiente es mayor que la unidad, y una disminución en caso contrario. Podría entenderse que un valor negativo en uno de los coeficientes, traería como resultado una simetría unida al escalado, pero el sistema no puede admitir esta forma de realizarla, pues en una simetría debe variar el sentido de lectura de los polígonos, mientras que en un escalado esta operación no es preciso realizarla.

La información necesaria para realizar una cizalladura es más compleja, pues pueden estar afectadas las tres direcciones del sistema de referencia o no. Una cizalladura, en su expresión más simple, consiste en el deslizamiento de todos los puntos del objeto, según la dirección de un eje coordenado, y con una magnitud que viene determinada por la coordenada respecto de otro eje, y por un coeficiente facilitado por el usuario. Vemos pues que pueden existir seis tipos básicos de cizalladura, es decir dos deslizamientos en la dirección de cada eje coordenado, determinados por los valores de cada una de las otras dos coordenadas. El sistema contempla la cizalladura como la actuación simultánea de un conjunto cualquiera de las seis cizalladuras básicas comentadas, por lo que al desear utilizar esta deformación nos solicitará información respecto de todas ellas, pidiendo primero el eje de deslizamiento, después el eje perpendicular a las caras que no sufren deformación, y por último el coeficiente a aplicar, cuyo signo indicará el sentido positivo o negativo del deslizamiento considerado. El usuario introduce los datos referentes a las cizalladuras básicas que componen la que desea realizar, y a continuación el sistema realiza la transformación correspondiente.

El sistema admite la concatenación de hasta catorce transformaciones para poder determinar modificaciones, tanto posicionales como formales, de una cierta complejidad, pudiendo realizarse en varias fases si esta cantidad no fuera suficiente.

Hemos visto todo lo que el sistema puede hacer con un volumen básico generado, perforado, deformado, reubicado e incluso formalmente modificado a lo largo del proceso, pero la generación de un tema de trabajo consiste en la manipulación de una gran variedad de objetos, por lo que en el capítulo siguiente analizaremos la utilización del sistema para la elaboración de objetos de una mayor complejidad, compuestos por un conjunto de sólidos como los que hemos descrito en este capítulo.

### 3. UTILIZACION DEL SISTEMA

#### 1. Niveles de trabajo.

Se ha visto hasta ahora como generar un volumen básico a partir de un paralelepípedo, vamos a proceder al análisis de la forma en que el sistema puede reunir varios de estos volúmenes para formar un conjunto más complejo, que a su vez puede ser parte de otro mayor, etc... Se definen para ello tres niveles de trabajo que, en la estructura de datos del sistema, vienen representados por tres conjuntos de archivos de dimensiones crecientes, capaces para contener la información del modelo del objeto que, en cada momento, se encuentra situado en ellas.

El nivel más amplio de trabajo constituye el tema que pretendemos generar con el sistema, y que recibirá a partir de ahora la denominación de "Objeto". Los archivos que constituyen su estructura de datos son los que van recibiendo la información procedente del nivel intermedio, y, una vez terminado el proceso, son trasladados a un soporte de almacenamiento externo, o se les aplica el programa de visualización descrito en la primera parte de esta tesis para ser representados gráficamente.

El nivel intermedio lo constituye el fragmento del tema que se está generando en cada momento, con vistas a ser incorporado al mismo una vez definido en su totalidad, lo cual se realiza trasladando el contenido de todos sus

archivos a los del Objeto, situando la información de los mismos a continuación de la existente en los del Objeto. Daremos el nombre de "Pieza" al cuerpo generado en este nivel intermedio, siendo este apelativo suficientemente significativo del papel que representa en la configuración del Objeto.

El nivel inferior de trabajo dentro del sistema, constituye la generación del modelo del sólido más elemental que utilizamos para generar el Objeto, lo cual solo significa que, dentro del proceso en que nos encontramos, es considerado como una unidad no descompuesta en partes volumétricas más sencillas. Para generar un paralelepípedo nuevo, con todas las variaciones que analizamos en el apartado anterior, se utilizan las posibilidades que el sistema ofrece para ello, y el cuerpo así formado recibe la denominación de "Bloque".

El sistema permite la recuperación y utilización de un sólido complejo generado en un proceso anterior, pudiendo ser utilizado como una Pieza sin necesidad de conformarlo de nuevo, aunque se le pueden aplicar las transformaciones posicionales y formales descritas en el apartado anterior. El sólido que es utilizado de este modo, como ya se dijo, recibe el nombre de "Primitiva".

Para hacer eficaz la utilización del sistema es preciso conocer de que Primitivas se dispone, lo cual dependerá, en la mayor parte de los casos, del trabajo anterior de generación de las mismas que hayamos realizado, aunque el sistema puede disponer de algunas muy generales por si mismo, y completar el conjunto de Primitivas puesto a nuestra disposición, caso de que éste no sea satisfactorio.

La estrategia a seguir para configurar un Objeto complejo, consiste en descomponerlo en partes que permitan su generación de forma sencilla, o bien la utilización de Primitivas que, mediante una secuencia de transformaciones adecuada, puedan ser incorporadas al Objeto como partes del mismo. Estas Piezas con que se procede a efectuar el "ensamblaje" del Objeto son, o han sido, a su vez construidas mediante un conjunto de Bloques que se generan y ubican en el lugar correspondiente hasta configurar la Pieza deseada. Puede darse el caso de construir una Pieza mediante la suma de una Primitiva existente, con la posibilidad de aplicar transformaciones a la misma, y los Bloques que sea necesario incorporarle. Se analiza mas detalladamente la generación de un Objeto en el apartado siguiente.

## 2. Proceso de utilización directa.

Ante la necesidad de generar un determinado tema, este debe ser con la intención de determinar las partes del mismo que, por un motivo u otro, interesa desarrollar independientemente para luego unirlos. Los criterios que determinan esta subdivisión del Objeto en diversas Piezas, pueden estar basados en diferentes tipos de consideraciones, las cuales no tienen porque depender de características objetivas del tema, sino que muchas veces vienen determinados por las preferencias del usuario. Veremos a continuación algunos criterios objetivos que pueden orientar sobre este aspecto de la cuestión, ya que los subjetivos no son susceptibles de análisis, al depender obviamente del usuario.

Un primer criterio objetivo de subdivisión del Objeto en Piezas consiste en considerar las características repetitivas de algunos fragmentos del tema, con lo que, creando una Pieza que coincida con uno de dichos fragmentos, se puede proceder a la generación de todos ellos, mediante la colocación de dicha Pieza en la posición ocupada por cada uno de ellos, siendo posible trabajar incluso con conjuntos parciales que agrupen a varios de estos fragmentos.

También pueden ser consideraciones de sencillez de trabajo las que aconsejen la generación de una determinada Pieza, como sucede en el caso de trabajar con un edificio que se compone de diversas partes, configuradas según una trama ortogonal cada una de ellas, pero colocadas en distintas direcciones unas de otras, con lo que lo cómodo será conformar cada una de dichas partes orientada según los ejes coordenados, y proceder posteriormente, mediante las transformaciones que ofrece el sistema, a su colocación en el lugar y con la orientación adecuada.

La existencia de partes del Objeto que tengan un cierto movimiento respecto al conjunto, unida al deseo de reflejar este movimiento en una secuencia de vistas del mismo, es otro motivo claro de la conveniencia de convertir en Primitiva estas partes, y posteriormente generar una Pieza del Objeto a partir de esta Primitiva colocada en diversas posiciones, lo que dará un conjunto de "Objetos" distintos, cuya única diferencia será la posición de dicha Pieza y, procediendo a generar la visualización de todos estos "Objetos", se obtendrán diversas representaciones en las cuales se pueda fácilmente observar el cambio de ubicación relativa del fragmento del tema que admite diversas posiciones.

Como ejemplo de lo antedicho se analiza como puede generarse un Objeto determinado mediante la utilización del sistema y la aplicación de los criterios comentados. El tema elegido es algo tan familiar como un cerramiento realizado mediante persianas de lamas orientables, una fija y la otra compuesta por dos hojas correderas.

Generamos en primer lugar una lama de la persiana, con la intención de poder utilizarla posteriormente como Primitiva. Iniciamos el proceso mediante la generación de un paralelepípedo que tenga las dimensiones de media lama, al cual se le aplica la deformación comentada en segundo lugar en el apartado correspondiente del capítulo anterior, la cual procede a inclinar las dos caras paralelas al plano XY del paralelepipedo. Sumamos a la figura resultante su simétrica respecto del plano XZ, procediendo a continuación a realizar una traslación en el sentido negativo del eje Z igual a la mitad del espesor de la lama, así como un giro respecto del eje X de sesenta grados, quedando la lama en una posición muy frecuente en situaciones reales, teniendo al eje X como eje de giro de la misma, lo cual es muy conveniente cara a su utilización posterior como Primitiva del sistema. El proceso descrito puede seguirse gráficamente en la figura II.III.2.1, reflejándose en la misma los cuatro pasos del mismo, la generación del paralelepípedo base, la deformación del mismo, la realización de su simétrico y, por último, el giro a la posición deseada.

Procedemos ahora a la creación de una hoja de la persiana, para lo cual configuramos en primer lugar el marco, generando un paralelepípedo con las dimensiones totales del mismo, y efectuandole dos agujeros en el lugar donde deben ir colocadas las lamas. Ahora es el momento de utilizar la Primitiva generada en el párrafo anterior,

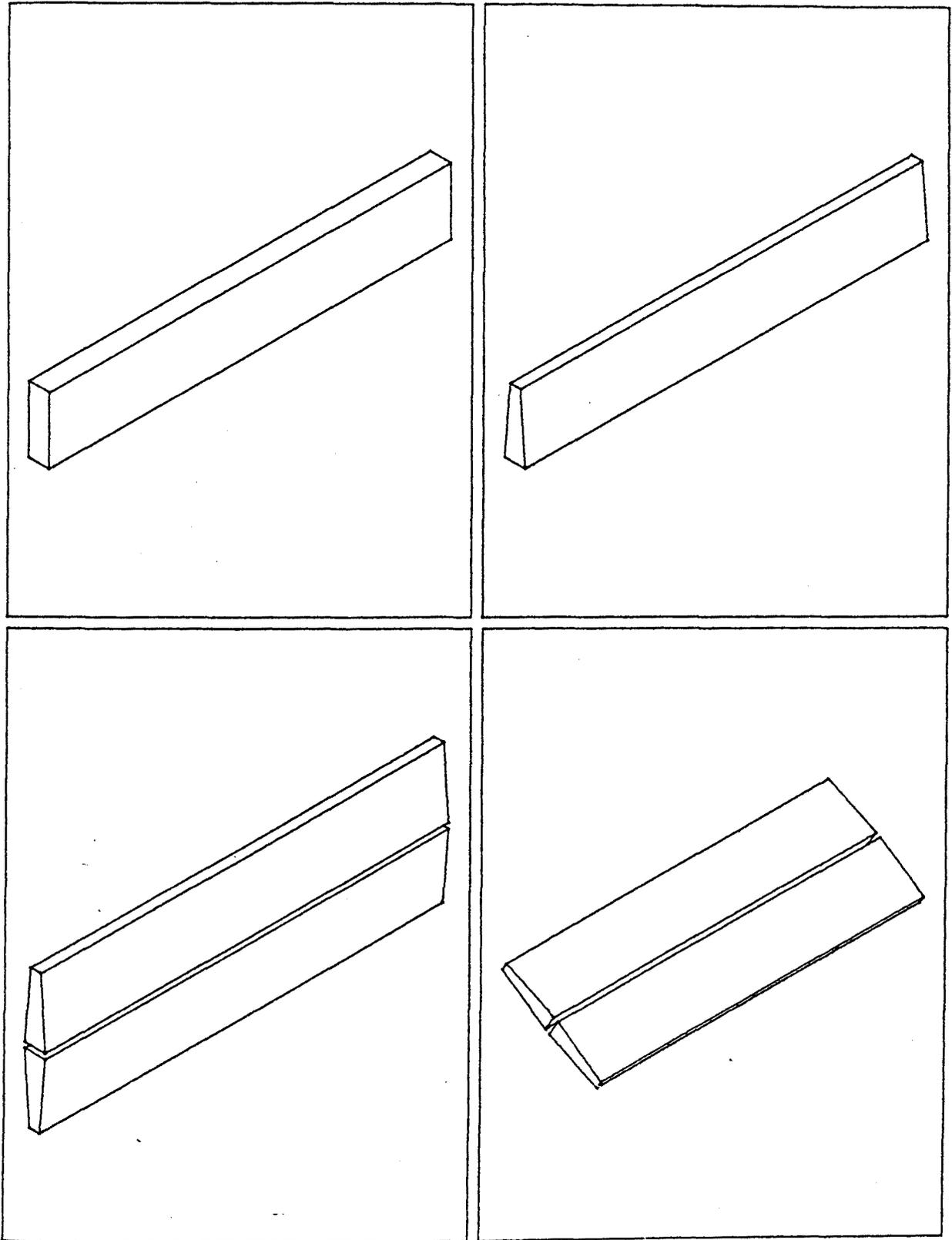


Figura II.III.2.1

escalándola a la medida adecuada para luego colocarla en la posición correspondiente y con la inclinación deseada, repitiendo el proceso las veces que sea necesario hasta haber obtenido la totalidad de las lamas que contiene la hoja de persiana. En la figura II.III.2.2 se aprecia la creación del marco, la colocación de las lamas del primer módulo con pendiente hacia el exterior, y la colocación de las lamas del segundo módulo de forma que la persiana quede cerrada.

La utilización de la hoja generada en el párrafo anterior como Primitiva, permite ubicarla cuantas veces sea preciso en las posiciones deseadas, como se refleja en la figura II.III.2.3, en la cual se presenta en la parte superior el espacio destinado a ser ocupado por una persiana compuesta de dos hojas correderas como la descrita, y otro hueco de una hoja fija situada perpendicularmente a la corredera. En la parte inferior aparecen colocadas las hojas con distintas posiciones de las lamas, representandose las dos hojas de la corredera a medio abrir.

Queda patente con lo expuesto la significación del concepto de Primitiva, así como su utilización, y la forma de generar nuevas Primitivas a partir de las existentes, proceso que solo tiene como limitación la capacidad de almacenamiento de que se disponga, la cual puede ser prácticamente ilimitada al utilizar dispositivos externos de almacenamiento masivo que, en un equipo como el utilizado por el autor, suele estar constituido por unidades de disco flexible, y en el mejor de los casos de disco duro.

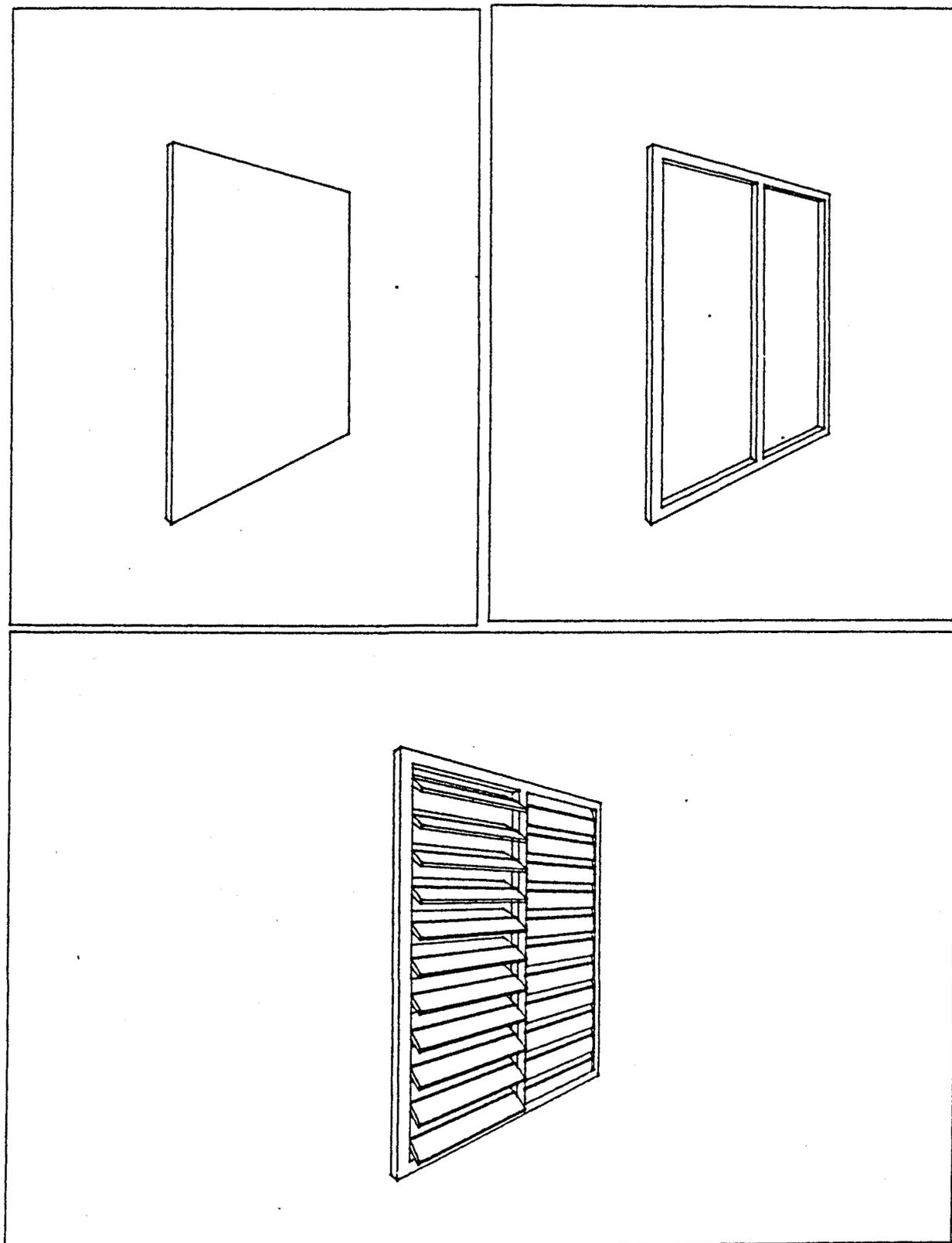


Figura II.III.2.2

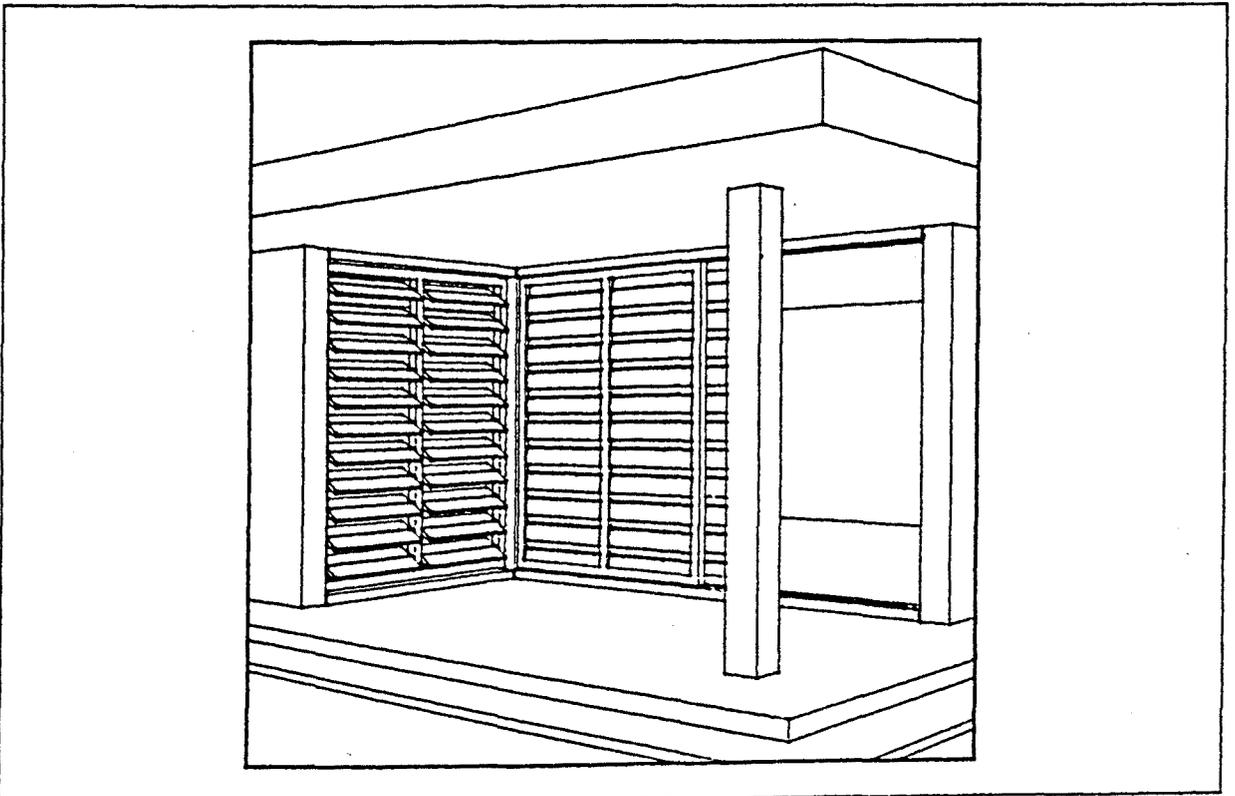
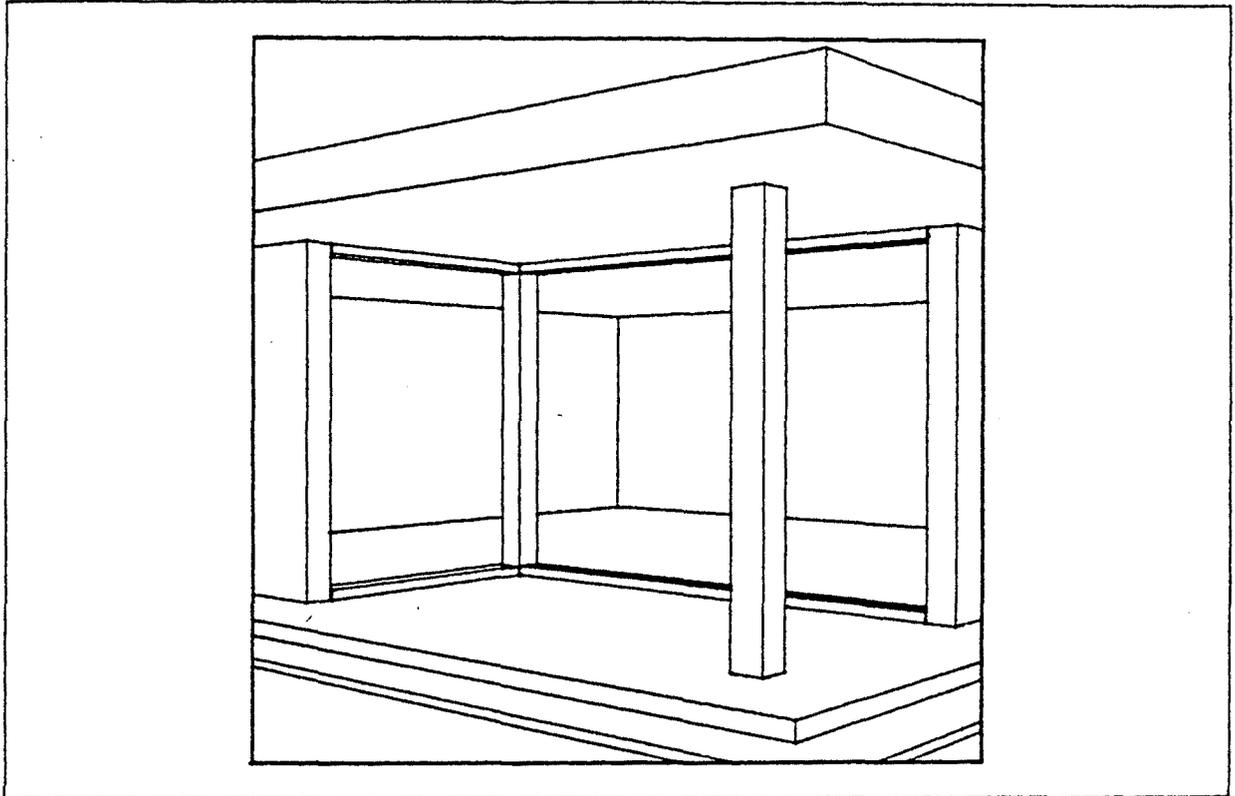


Figura II.III.2.3

### 3. Organización de una Secuencia Programada.

La forma de almacenar los Objetos y Primitivas para su posterior utilización que hemos comentado hasta ahora, se basa en el almacenamiento en disco de la información contenida en los archivos del modelo, lo cual tiene como ventaja la inmediatez de la disposición de los datos para ser utilizados en el procesado de visualizaciones, o bien en la generación de otros Objetos mediante la utilización de los existentes como Primitivas o Piezas. Sin embargo la ocupación de memoria es importante, y aunque de forma económica se pueda disponer de una gran cantidad de almacenamiento en forma de discos flexibles, a la hora de utilizar el sistema, pudiera suceder que necesitemos acceder a información grabada en diversos discos, por falta de espacio en uno solo para albergar todas las Primitivas que en un momento determinado pudiese ser preciso tener a nuestra disposición.

Un método para tener una mayor cantidad de Objetos accesibles en un solo disco, consiste evidentemente en conseguir que el espacio de almacenamiento necesario para cada uno de ellos sea menor, y lo mismo es aplicable a las Primitivas. Un sistema para conseguir esto consiste en no archivar el modelo del Objeto, sino la información referente al proceso seguido por el sistema para generarlo, y tener un programa capaz de leer esta información y, actuando en consecuencia, generar de nuevo el Objeto siguiendo los pasos indicados. Un inconveniente de esta opción radica en la necesidad de un mayor tiempo de procesado para reconstruir cada Objeto o Primitiva que se desee utilizar, pero a cambio el sistema tiene a su disposición una más amplia disponibilidad de Objetos y Primitivas para utilizar, por lo que la preferencia de una u otra forma de trabajo vendrá determinada por una serie de factores propios de cada usuario en particular, e incluso

de las características del trabajo que, en cada momento, se vaya a desarrollar.

Llamaremos Secuencia Programada a los datos que constituyen la información necesaria para reconstruir el proceso de generación de un Objeto mediante la utilización ordenada de las rutinas del sistema, y que por tanto serán guardados en memoria externa para su libre disposición posterior. Si el sistema permite acceder parcialmente a un punto concreto de una Secuencia Programada, esto facilita extraordinariamente ciertos procesos de generación de Objetos que son ligeras variantes de otros existentes, pudiendo hacer de esta forma mucho más comodo el trabajo, pese al empleo de un mayor tiempo de procesado en la reconstrucción de la estructura del modelo de fronteras, cuando esto se hace necesario para proceder a su visualización, o para efectuar cualquier otro tipo de operación que lo precise.

La organización de una Secuencia Programada se realiza en forma de estructura en árbol, para una mejor manipulación de todos y cada uno de los elementos que la componen, así como para poder introducir facilmente modificaciones de cualquier tipo que pueda interesar. El árbol utilizado está formado por tres niveles, los cuales se describen a continuación.

El nivel superior, o raíz, contiene al Objeto, y por ello la información referente a las características globales del mismo que puedan ofrecer interés para su manipulación posterior, así como la dirección del primero de sus hijos situado en el nivel intermedio.

En el nivel intermedio se encuentran situadas las Piezas en que descomponemos el Objeto, pudiendo ser Primitivas, simples Bloques, o estar a su vez formadas por

la reunión de varios Bloques, los cuales se encontrarán definidos como elementos del nivel inferior. Cada una de estas Piezas contiene la información necesaria para la estructuración del árbol, direcciones de su padre, hermano adyacente y el primero de sus hijos. También posee la información respecto de la secuencia de transformaciones necesaria para su incorporación al Objeto.

. El nivel inferior esta constituido por los Bloques que configuran las Piezas del nivel intermedio, con la información necesaria para su generación y deformación, si esto último es procedente. Como la Pieza, el Bloque tambien guarda información respecto de las direcciones del padre y del hermano inmediato, pero no respecto del hijo, pues constituye el último nivel del árbol. Asimismo consta de la necesaria información que describe la secuencia de transformaciones necesaria para poder pasar a ser parte de la Pieza que lo contenga.

Es pues otro modelo de representación interna y de almacenamiento de un Objeto el de la Secuencia Programada de generación del mismo. Como ya hemos comentado este modelo es alternativo del de fronteras, siendo necesaria la existencia de un algoritmo que, a partir de la lectura y decodificación de la Secuencia Programada de un Objeto, genere el modelo de fronteras del mismo, pues éste es imprescindible para el procesado de su visualización mediante la utilización de los algoritmos analizados en la primera parte.

#### 4. Concepto de Biblioteca.

Denominamos Biblioteca al archivo que contiene un conjunto de modelos de Objetos cuya utilización se puede realizar por un programa o conjunto de programas. Además

debe estar provisto de un sistema de búsqueda eficiente, de forma que al ser requerido un elemento de la Biblioteca para su utilización desde un programa determinado, su localización y posterior lectura desde el programa en cuestión sea fácil y rápida. Es conveniente en este sentido la existencia de diversos índices, que permitan la búsqueda de un elemento cualquiera de la Biblioteca mediante su identificación por diversos conceptos, así como la posibilidad de acceso a la consulta de dichos índices, con el fin de conocer en todo momento los elementos disponibles en la Biblioteca.

El conjunto de Primitivas que se encuentran a nuestra disposición en el sistema es muy conveniente que estén organizados en Bibliotecas, según los criterios del usuario, aunque comentaremos algunos criterios que pueden facilitar la organización de estas Bibliotecas de Primitivas por parte del usuario. Es obvio que la mayoría de las Primitivas disponibles en estas Bibliotecas pueden estar generadas por el propio usuario del sistema, debiendo ser posible y cómodo conformar Bibliotecas nuevas a partir de Primitivas existentes en otras Bibliotecas, lo cual permite reorganizar el conjunto de Primitivas disponible de acuerdo con nuevos criterios en cualquier momento.

La información existente en estas Bibliotecas respecto de las Primitivas que contienen puede consistir en el modelo de fronteras del sólido, o por el contrario en la Secuencia Programada que permita realizar su reconstrucción previamente a su incorporación al programa. Nos interesará que ésto sea de una u otra forma en función del tipo de programa que utilice dicha información, sirviendo aquí las reflexiones al respecto que hacemos en el capítulo anterior, teniendo un almacenamiento más compacto y permitiendo una manipulación más sencilla las Secuencias Programadas, pero siendo de una utilización mucho más

directa, y por tanto más rápida, los modelos de fronteras, debiendo elegir entre estas dos alternativas según el tipo de trabajo a realizar.

5. Criterios de organización de una Biblioteca.

Las características del trabajo para el cual se utiliza el sistema pueden aconsejar una u otra organización de las Bibliotecas de Primitivas. Es evidentemente decisión del usuario en cada caso concreto la aplicación de los criterios adecuados. Podemos considerar, a modo de ejemplo, el caso de trabajar en el campo de la construcción industrializada o en la realización de dibujos que describan constructivamente el edificio, otra posibilidad consiste en utilizar el sistema para obtener distintas visualizaciones del tema de trabajo, podría asimismo constituir el objetivo prioritario del usuario el desarrollo de gráficos tridimensionales que permitan estudiar el recorrido de las instalaciones en el interior del edificio, pueden darse otros muchos campos de trabajo que tendrán condicionantes específicos, pero vamos a comentar únicamente el tercero de los casos expuestos, el de la visualización del tema como lo pueda percibir un observador desde una posición determinada, con la intención de efectuar la exploración formal del mismo.

La visualización de un tema arquitectónico desde el exterior puede realizarse con una situación del observador próxima al mismo o, por el contrario, desde una posición alejada, no siendo relevantes en este caso los elementos de menor dimensión de la fachada, llegando incluso a apreciarse únicamente la volumetría del tema en el caso de estar suficientemente alejado el observador. Existe pues un criterio de diferenciación para ubicar los elementos del tema bajo dos epígrafes distintos en la biblioteca

correspondiente. Tendremos por un lado los elementos que configuran la volumetría general del tema, y por otro los que definen el tratamiento en detalle del aspecto exterior del mismo. Hay que tener en cuenta, como complemento a lo expuesto que, según el punto de vista es ineficiente procesar un objeto muy complejo si, por la distancia al observador, la representación de sus detalles se realizará con unas dimensiones que no permitirá suficiente claridad en su percepción.

Mientras las visualizaciones que interese procesar sean realizadas desde el exterior del objeto, tendremos suficiente con diferenciar los elementos a utilizar en su generación en función de su dimensión, lo cual lleva aparejado, junto con la situación del observador, su mayor o menor posibilidad de ser representados gráficamente. Pero en el momento en que interese introducir al observador en el interior del edificio, será preciso generar los elementos que componen y delimitan el espacio contenido por el edificio, por lo cual tenemos un tercer grupo de primitivas a ser consideradas, que son los elementos arquitectónicos que usualmente se encuentran contenidos en el interior de los edificios. Estos elementos no es lógico que formen parte del modelo en las vistas exteriores del edificio, salvo en el caso de desear suponer abiertas algunas aberturas practicables del mismo, lo cual hará posible la visión desde el exterior de algunos elementos interiores.

Un cuarto grupo de elementos que, en una primera clasificación, pueden constituir una unidad independiente dentro de la Biblioteca son los elementos de mobiliario, los cuales pueden o no ser considerados al realizar una visualización del interior del edificio, o bien puede ser variada su posición manteniendo inalterable el resto del modelo, e incluso en algoritmos mas sofisticados, puede

establecerse una escala de prioridades y procesarse únicamente los de determinado nivel de interés.

Es importante, cara a la eficiencia del sistema, no procesar elementos de los que se tiene la seguridad de su invisibilidad, por lo que deben utilizarse indicadores que avisen al programa de visualización según las condiciones del observador, si debe o no tener en cuenta a cada elemento o grupo de los mismos. Esto mismo es aplicable si el Objeto se encuentra definido mediante una Secuencia Programada, en cuyo caso se generará el modelo contemplando únicamente los elementos significativos en el tipo de visualización que se vaya a desarrollar posteriormente, permitiendo un considerable ahorro de espacio de memoria y de tiempo de procesado.

Si el objetivo es estudiar el efecto que la posición de un determinado edificio comporta sobre el entorno en que se situa, se utiliza el modelo que define su volumetría general y se obtiene una colección de visualizaciones desde diversas posiciones del observador, así como modificando la situación del edificio para ubicarlo en las posiciones en las que pueda considerarse interesante analizar el efecto de su relación con el entorno, del cual es obviamente necesario tener incorporado el modelo de su volumetría general.

Si, por el contrario, la comparación se establece con los edificios colindantes, deseando analizar la relación entre el tema en estudio y los mismos con un alto grado de definición, será preciso utilizar modelos detallados de los mismos, aunque esto naturalmente comporta mayores tiempos de procesado.

## Parte II      Capítulo III

En lo que al análisis de un edificio se refiere, veremos en el capítulo siguiente el desarrollo de tres ejemplos, basados en un edificio existente, un proyecto no realizado, y una obra de alto interés histórico que se encuentra en nueva construcción después de haber sido demolido. La realización de estos ejemplos permite apreciar la capacidad y las limitaciones del sistema aquí desarrollado. Es conveniente aclarar que algunas de estas limitaciones son debidas esencialmente a no disponer de un equipo de mejores prestaciones y capacidad, y por tanto no guardan relación con la efectividad del sistema que, implantado en un equipo distinto, vería ampliada de forma muy significativa su capacidad de acción.