

UNIDAD 5

5.1 Análisis de dos Ejemplos

Las siguientes pruebas fueron realizadas en una computadora personal Pentium III a 450 MHz y con 64 Mega Bytes de RAM. Con un software desarrollado en Java. Algunas de las subrutinas fueron realizadas en estrecha colaboración con el Dr. Antonio Aguilera.

Los objetos están definidos en un dominio de 256 por 256 por 256. En el software los objetos se pueden guardar en disco, únicamente se almacena la expresión Booleana y la información de las primitivas que forman al objeto, no se almacena ningún árbol octal.

5.1.1 Unión de 8 esferas disjuntas

A continuación se muestra un ejemplo de un objeto que esta formado con la unión de 8 esferas con la información que se muestra en la Tabla 5.1

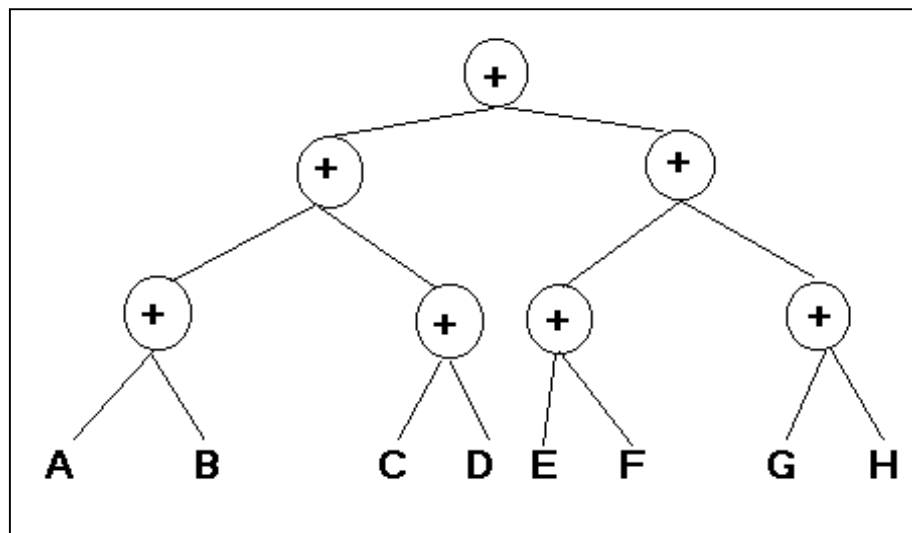
Tabla 5.1: Información de 8 esferas en un dominio de 256 por 256 por 256.

Esfera	Origen en X	Origen en Y	Origen en Z	Radio en el eje X	Radio en el eje Y	Radio en el eje Z
A	-55	-55	-55	55	55	55
B	-55	-55	55	55	55	55
C	-55	55	-55	55	55	55
D	-55	55	55	55	55	55
E	55	-55	-55	55	55	55
F	55	-55	55	55	55	55
G	55	55	-55	55	55	55
H	55	55	55	55	55	55

"Elaboración propia"

Las 8 esferas son idénticas y están disjuntas, la información de cada esfera puede verse en la tabla 5.1 tienen un radio de 55. Cada esfera tiene su nombre en este caso son A, B, C, D, E, F, G y H. La expresión Booleana es: $((A+B)+(C+D))+((E+F)+(G+H))$. El Árbol CSG de la unión de 8 esferas se muestra en la figura 5.1.

Figura 5.1: Árbol CSG de la unión de 8 esferas llamadas A, B, C, D, E, F, G y H

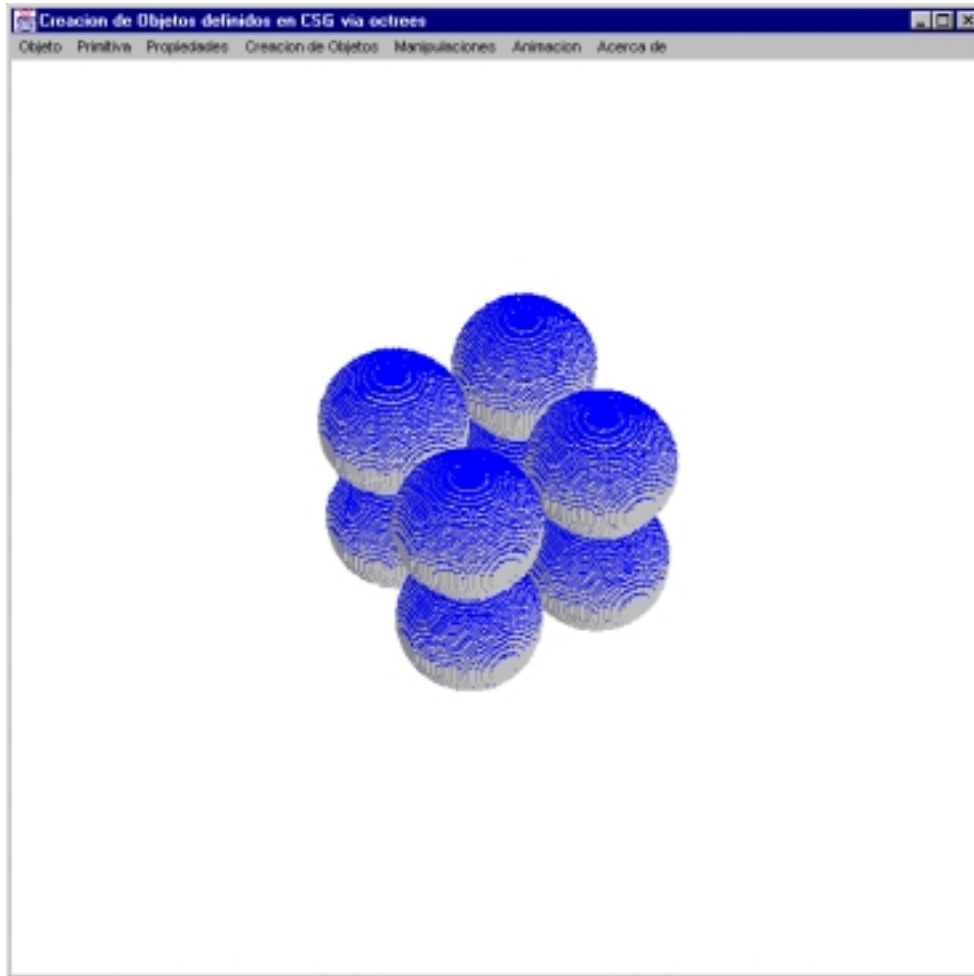


"Elaboración propia"

La esfera A se une con la esfera B, la esfera C y la D se unen, la unión de las dos uniones anteriores forma la tercera unión, la cuarta unión es E con F, la quinta unión es G con H, la sexta unión es la cuarta unión con la quinta, la última unión es la tercera unión con la sexta unión.

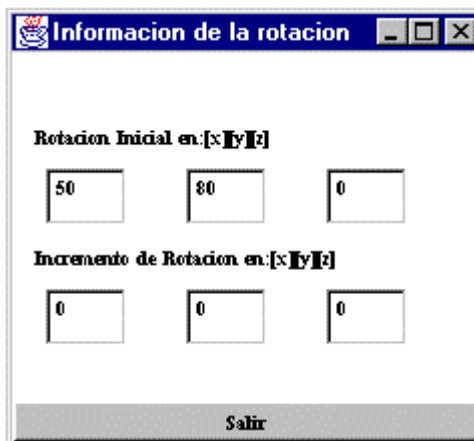
En la figura 5.2 se muestran las 8 esferas unidas, y se puede apreciar que están disjuntas, las esferas visualizadas tienen una rotación de 50 grados en X, 80 grados en Y y 0 grados en Z. Vea la Figura 5.3.

Figura 5.2: Unión de 8 esferas disjuntas



"Elaboración propia"

Figura 5.3: Rotación en X, Y y Z



"Elaboración propia"

En la Figura 5.4 se muestra los resultados del tiempo que se tarda cada método, el método clásico hace la unión de las 8 esferas en 28 segundos, mientras que el método Top Down se tarda en hacer la unión de las 8 esferas 12 segundos, esto se debe a que el tiempo en que tarda el programa para hacer el Árbol Octal de cada primitiva, es grande, por lo que el método Top Down tiene la ventaja de construir un objeto sin la necesidad de crear el árbol Octal de cada primitiva.

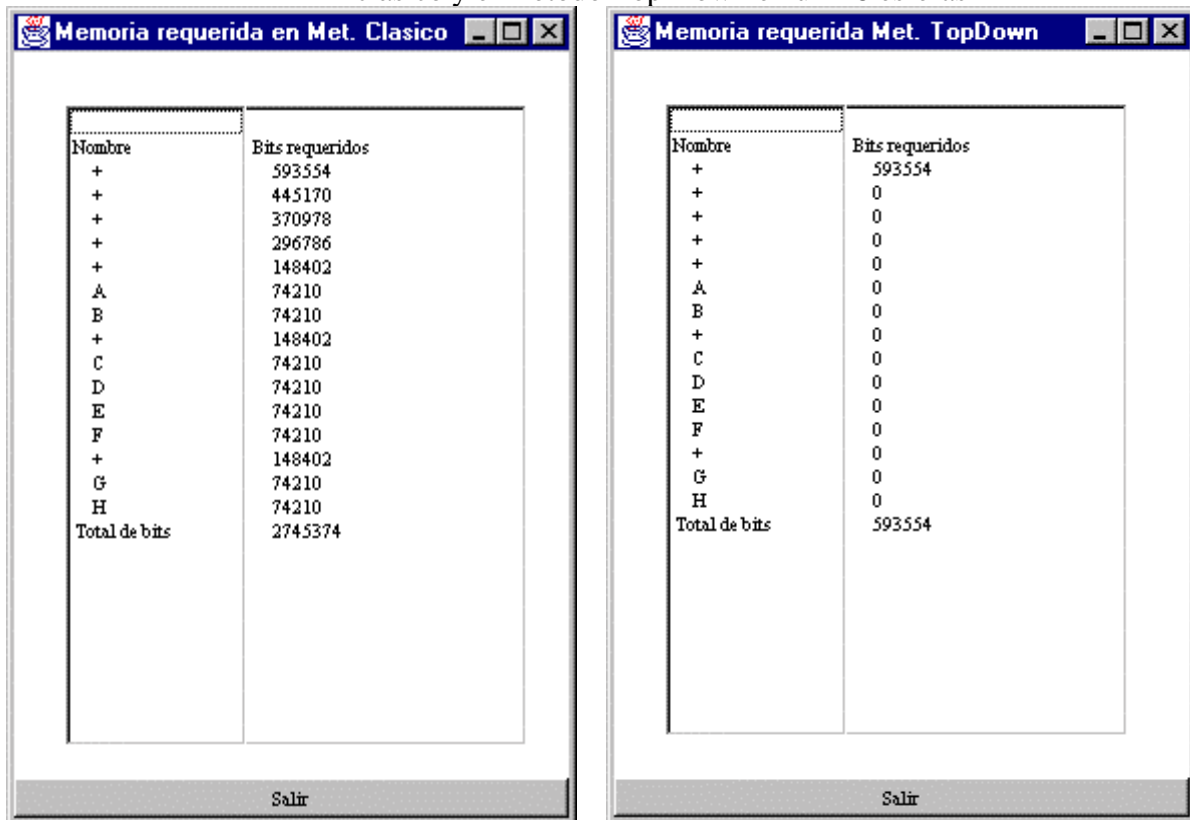
Figura 5.4: Resultados del tiempo que se tarda el método clásico y el método Top Down en unir 8 esferas



"Elaboración propia"

Otra ventaja del método Top Down sobre el método clásico es que gasta menos memoria, en la Unidad 4 se mencionó que la cantidad de memoria se mide por los octantes que se utilizan para crear cada primitiva, el programa almacena en 2 bits cada octante. En la figura 5.5 se muestra la memoria requerida por cada método para el ejemplo de las 8 esferas disjuntas, el método clásico ocupó 2,745,374 bits que equivalen a 1,372,687 octantes y el método Top Down ocupó solo 593,554 bits que equivalen únicamente a 296,777 octantes. La diferencia es muy grande y equivale a 4 veces más memoria en el método clásico. Es importante decir que la precisión con la que se ejecute influye tanto en tiempo como en memoria, si la precisión es baja la cantidad de octantes disminuyen y por lo tanto se requiere menos memoria y también se requiere menos tiempo.

Figura 5.5: Resultados de la cantidad de bits que utiliza el método clásico y el método Top Down en unir 8 esferas



"Elaboración propia"

5.1.2 Creación de un tren con 8 primitivas

El ejemplo 2 esta formado de 8 primitivas: 5 Paralelepípedos A, B, E, F y G; y de 3 Cilindros C, D y H. La información de cada primitiva se puede ver en la tabla 5.2

Tabla 5.2: Información de 5 Paralelepípedos definidos en un dominio de 256 por 256 por 256.

Paralelepípedo	Origen en X	Origen en Y	Origen en Z	Final en X	Final en Y	Final en Z
A	-128	-64	-64	128	64	64
B	-128	-30	-30	128	30	30
E	-128	-128	-64	128	-64	64
F	90	0	-64	-60	30	64
G	60	0	-64	90	30	64

"Elaboración propia"

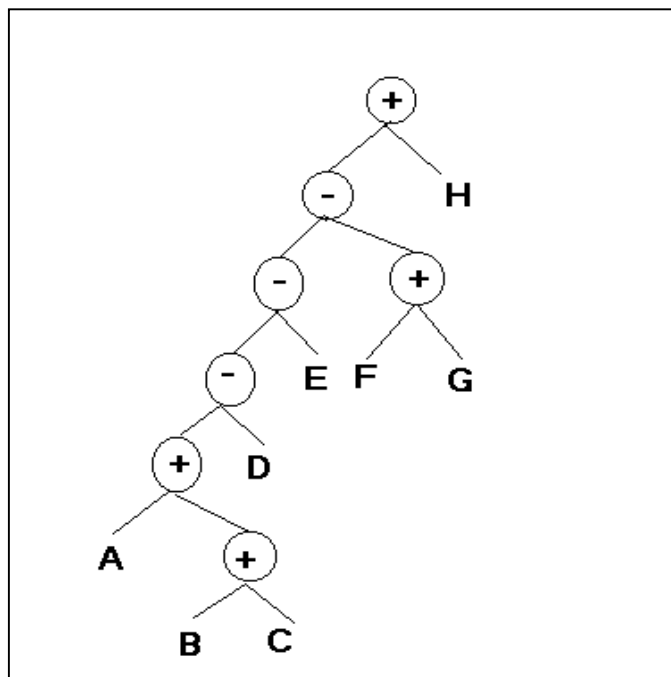
Tabla 5.3: Información de 3 cilindros definidos en un dominio de 256 por 256 por 256.

Cilindros	Origen en X	Origen en Y	Origen en Z	Radio	Tope	Eje
C	-80	-70	-80	40	80	Z
D	80	-70	-80	40	80	Z
H	-70	64	0	30	100	Y

"Elaboración propia"

La ecuación Booleana regularizada esta dada por $((((A+((C+D))-B)-E)-(F+G))+H)$ y el árbol CSG del objeto se muestra en la Figura 5.6

Figura 5.6: Árbol CSG de un tren con Formado por 5 Paralelepípedos A, B, E, F y G y tres cilindros C, D y H

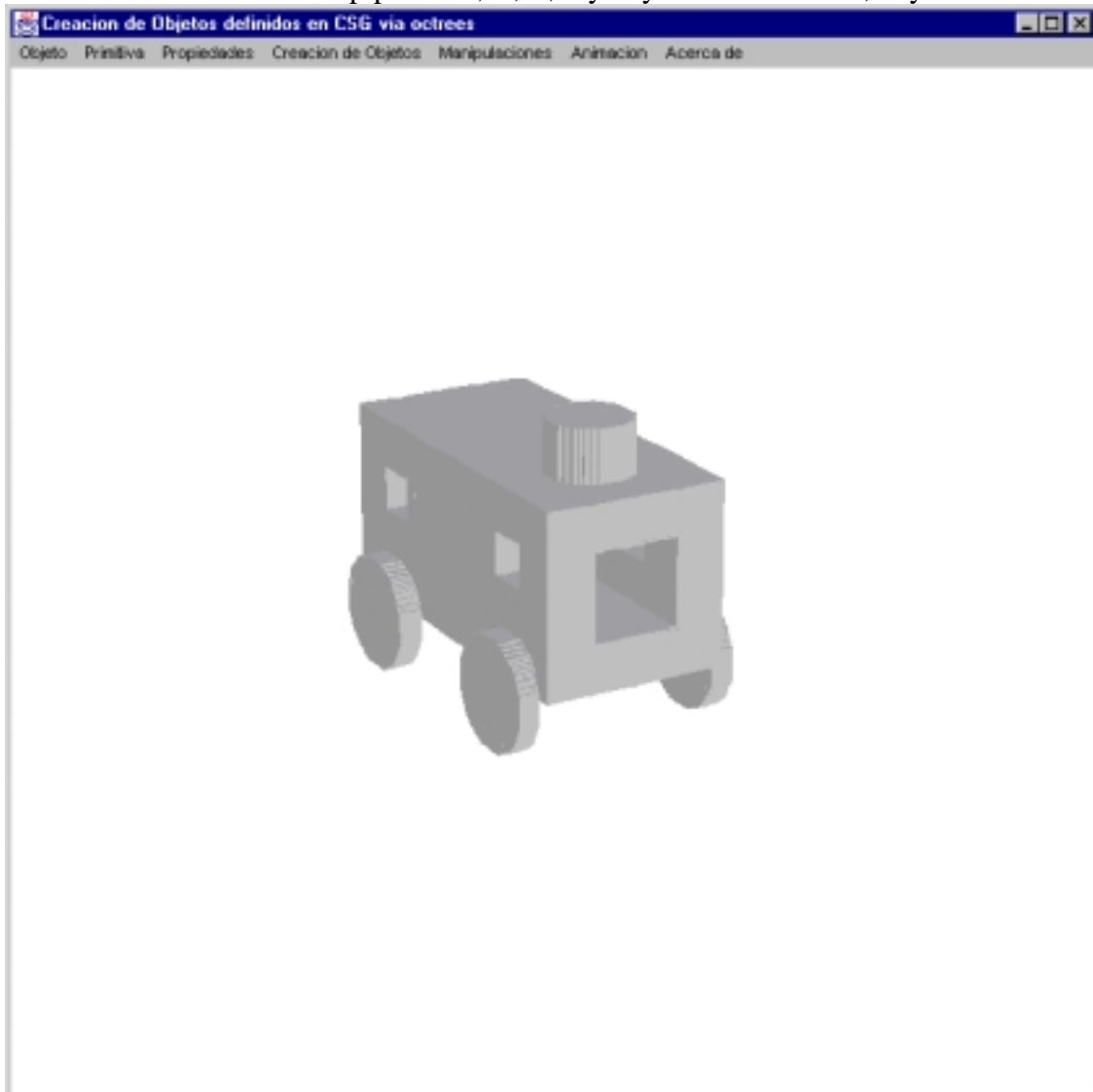


"Elaboración propia"

El paralelepípedo B se une con el cilindro C, y este resultado se une con el paralelepípedo A, a este último resultado se le resta el cilindro D, inmediatamente después se le resta el paralelepípedo E a este resultado lo llamaremos resultado 1, dos paralelepípedos

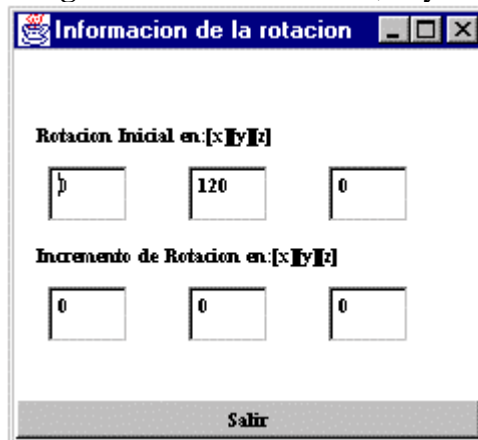
llamados F y G se unen y a este resultado lo llamaremos resultado 2. Se obtiene la diferencia entre el resultado 1 y el resultado 2 y a este último resultado se le une el cilindro H. La figura 5.7 es el tren que se forma por medio de las 8 primitivas antes mencionadas y el objeto se encuentra rotado 20 grados en X, 120 grados en Y y 0 grados en Z, como se muestra en 5.8.

Figura 5.7: Un tren con Formado por 5 Paralelepípedos A, B, E, F y G y tres cilindros C, D y H



"Elaboración propia"

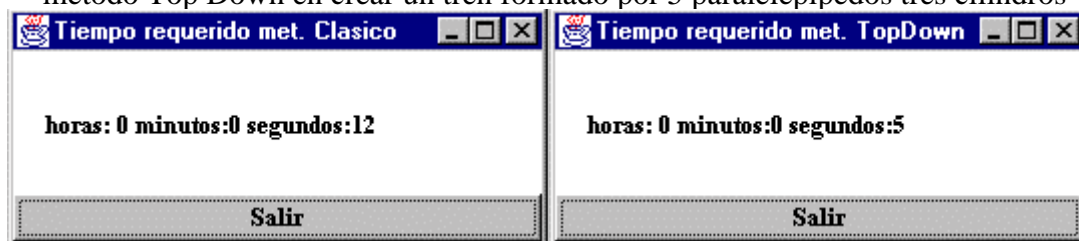
Figura 5.8: Rotación en X, Y y Z



"Elaboración propia"

En las figuras 5.9 se muestran los tiempos que se tarda cada método en obtener un tren formado de 8 primitivas: 5 paralelepípedos y 3 cilindros. Como puede observarse el tiempo que se tarda el método clásico es de 12 segundo, mientras que el método Top Down se tarda 5 segundos, lo que corresponde a un ahorro de tiempo de 7 segundos.

Figura 5.9: Resultados del tiempo que se tarda el método clásico y el método Top Down en crear un tren formado por 5 paralelepípedos tres cilindros

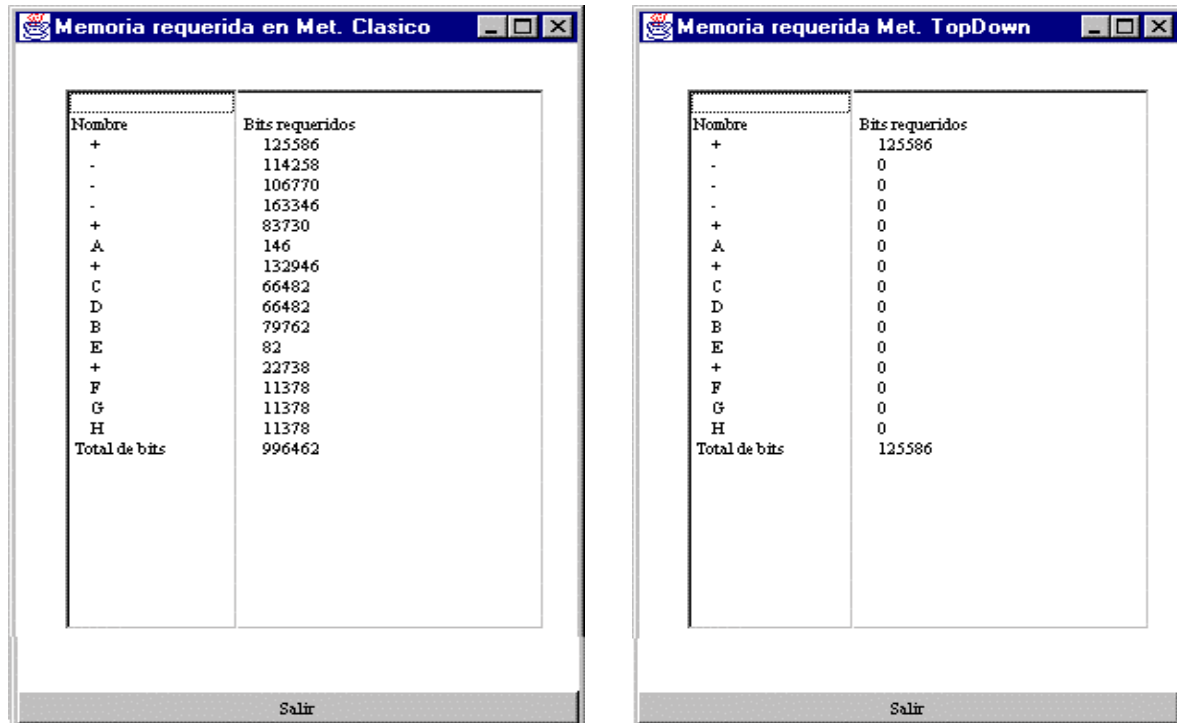


"Elaboración propia"

En la Figura 5.10 se muestran los resultados de la memoria que se gastan por medio de los dos métodos al crear el tren, como puede observarse el método clásico gasta 996,462 bits, como cada octante se almacena en 2 bits el número de octantes almacenados son 498,231, mientras que en el método Top Down se requieren 125,586 bits los cuales equivalen a 62,793

octantes, la cantidad de octantes redujo 7.93 veces es decir casi 8 veces menos de memoria con respecto al método Clásico.

Figura 5.10: Resultados de la cantidad de bits que utiliza el método clásico y el método Top Down para crear un tren formado de 5 paralelepípedos y 3 cilindros



"Elaboración propia"

5.1.3 Comparación de 10 objetos definidos en CSG por los métodos clásico y Top Down

En la Tabla 5.4 se muestra una comparación de 10 Objetos definidos en CSG por medio de árboles octales usando el Método Clásico y el Método Top Down. En la primera columna se describe de qué ejemplo se trata, en las figuras 5. 11 a la 5.20 se muestran los objetos para los ejemplos del 1 al 10 de la Tabla, estas figuras fueron creadas con una precisión de 100. La columna 2 indica cuánta precisión se uso en el objeto; cada objeto se evaluó con las precisiones 70, 80, 90 y 100, de acuerdo a esta precisión el tiempo y la memoria usada cambian. En la columna 3 se muestra el tiempo en que tarda el Método clásico en crear el objeto de acuerdo a la precisión requerida, la columna 4 indica cuánto tiempo tarda

Tabla 5.4: Comparación de Tiempo y memoria de 10 Objetos definidos en CSG mediante Árboles Octales usando los métodos Clásico y Top Down

Ej.	Fig.	Precisión	Tiempo en Seg. Mét. Clásico	Tiempo en Seg. Mét. Top Down	Bits ocupados en Mét. Clásico	Bits ocupados en Mét. Top Down	Octantes ocupados en Mét. Clásico	Octantes ocupados en Mét. Top Down	1/τ	1/μ
1		70	0	0	3510	786	1755	393	No captado	4.4656
		80	0	0	15030	3090	7515	1545	No captado	4.8641
		90	1	0	59574	12818	29787	6409	No captado	4.6477
	5.11	100	3	1	184374	35218	92187	17609	3	5.2352
2		70	1	0	43918	5522	21959	2761	No captado	7.9533
		80	3	1	195726	25746	97863	12873	3	7.6022
		90	10	6	853902	114322	426951	57161	1.666666666	7.4693
	5.12	100	41	23	3385742	465170	1692871	232585	1.7826087	7.2785
3		70	1	0	38250	5522	19125	2761	No captado	6.9268
		80	2	1	169384	25746	84692	12873	2	6.5790
		90	9	5	739434	114322	369717	57161	1.8	6.4680
	5.13	100	35	22	2920426	465170	1460213	232585	1.59090909	6.2782
4		70	1	0	20094	4242	10047	2121	No captado	4.7369
		80	1	1	88318	17042	44159	8521	1	5.1823
		90	4	3	378494	71698	189247	35849	1.333333333	5.2790
	5.14	100	18	9	1448190	276882	724095	138441	2	5.8659
5		70	1	0	53282	5714	26641	2857	No captado	9.3248
		80	2	1	207778	22866	103889	11433	2	9.0868
		90	9	3	791522	88658	395761	44329	3	8.9278
	5.15	100	13	5	1100450	132242	550225	66121	2.6	8.3215
6		70	0	0	2282	562	1141	281	No captado	4.0605
		80	0	0	9450	2354	4725	1177	No captado	4.0144
		90	1	0	36074	9010	18037	4505	No captado	4.0038
	5.16	100	2	0	138474	34610	69237	17305	No captado	4.0010
7		70	0	0	17360	4242	8680	2121	No captado	4.0924
		80	1	0	57562	14610	28781	7305	No captado	3.9399
		90	3	2	237914	64402	118957	32201	1.5	3.6942
	5.17	100	13	9	1010522	280210	505261	140105	1.444444444	3.6063
8		70	0	0	35690	2066	17845	1033	No captado	17.2749
		80	2	1	167146	11282	83573	5641	2	14.8152
		90	9	2	733930	53522	366965	26761	4.5	13.7128
	5.18	100	32	8	2600682	166418	1300341	83209	4	15.6274
9		70	0	0	8410	1618	4205	809	No captado	5.1978
		80	0	0	33562	6802	16781	3401	No captado	4.9341
		90	1	1	131546	27986	65773	13993	1	4.7004
	5.19	100	4	2	347546	90386	173773	45193	2	3.8451
10		70	0	0	16802	3090	8401	1545	No captado	5.4375
		80	0	1	64354	11794	32177	5897	0	5.4565
		90	4	1	250722	47890	125361	23945	4	5.2354
	5.20	100	10	5	786786	172050	393393	86025	2	4.5730

"Elaboración propia"

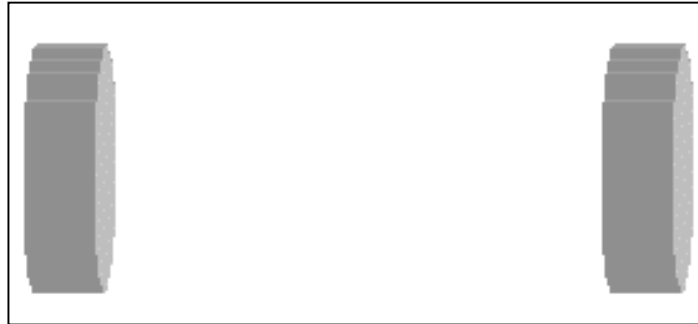
el Método Top Down para crear el objeto de acuerdo a la precisión usada, la columna 5 indica la cantidad de Bits que ocupa el Método Clásico para crear un objeto deseado de acuerdo con la precisión indicada (cada octante del árbol octal ocupa 2 bits), la columna 6 indica cuántos Bits necesita el Método Top Down para crear un objeto. Las columnas 7 y 8 nos muestran el número de octantes que utilizan el Método clásico y el Método Top Down respectivamente. Y por último las columnas 9 y 10 nos indican una comparación de tiempo y memoria entre los dos Métodos respectivamente. Nótese que se usa $1/\mu$ y $1/\tau$ que nos indica cuantas veces es más rápido el método Top Down que el clásico.

Como se puede apreciar en la tabla, el tiempo requerido para crear un objeto mediante el Método Top Down es menor o igual al tiempo que utiliza el Método Clásico, en la columna 9 se tiene un cociente entre los tiempos obtenidos por los dos Métodos. La existencia de un "no captado" indica que el tiempo está dado en milisegundos y por lo tanto no fue captado el tiempo transcurrido, por la computadora. Entre más precisión se tiene en el objeto se puede obtener un ahorro de tiempo mayor, ya que como se ve en esta columna un objeto creado con cualquiera de los dos métodos con la precisión 70 el tiempo es tan parecido que no se capta la diferencia por que está dada en milisegundos, con las precisiones 80 y 90 el ahorro de tiempo se empieza a hacer notorio, mientras que con una precisión de 100 el ahorro de tiempo ya es más grande.

En la columna 10 se hace una comparación de la memoria requerida por cada método, debido a que el Método Clásico almacena cada árbol de las primitivas y de las operaciones Booleanas ocupa más memoria que el método Top Down. El ahorro de memoria en cantidad de octantes es proporcional más bien al objeto que a la precisión. Por ejemplo para la figura

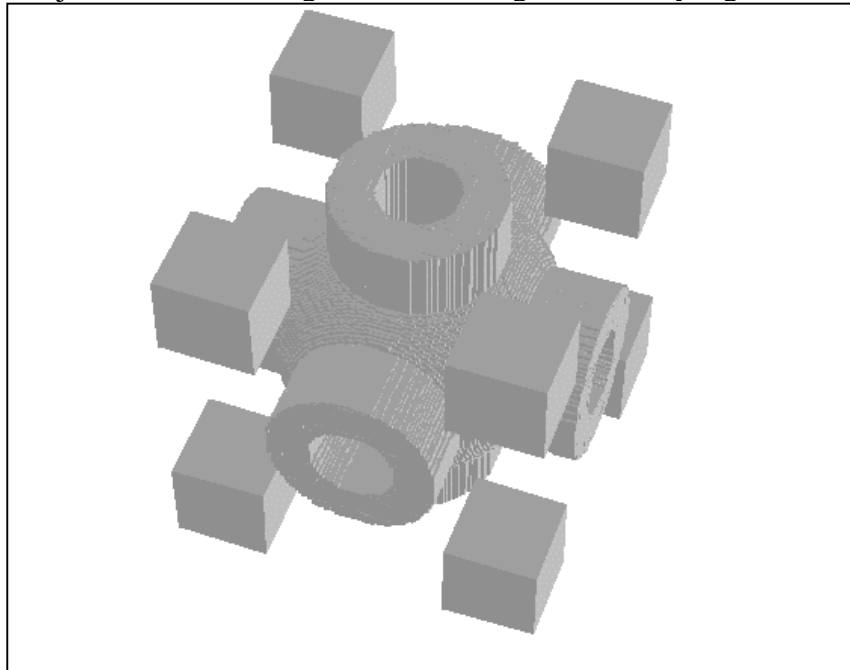
5.11 el ahorro de Memoria en octantes, en el Método Top Down es en promedio 4.8031 veces menor que en el Método clásico. Mientras que en el ejemplo 8 el ahorro de memoria en octantes es en promedio 15.3575 veces menor que en el método Clásico

Figura 5.11: Objeto correspondiente al Ejemplo 1 de la tabla 5.4
El objeto esta rotado 10 grados en X, -10 grados en Y y 0 grados en Z



"Elaboración propia"

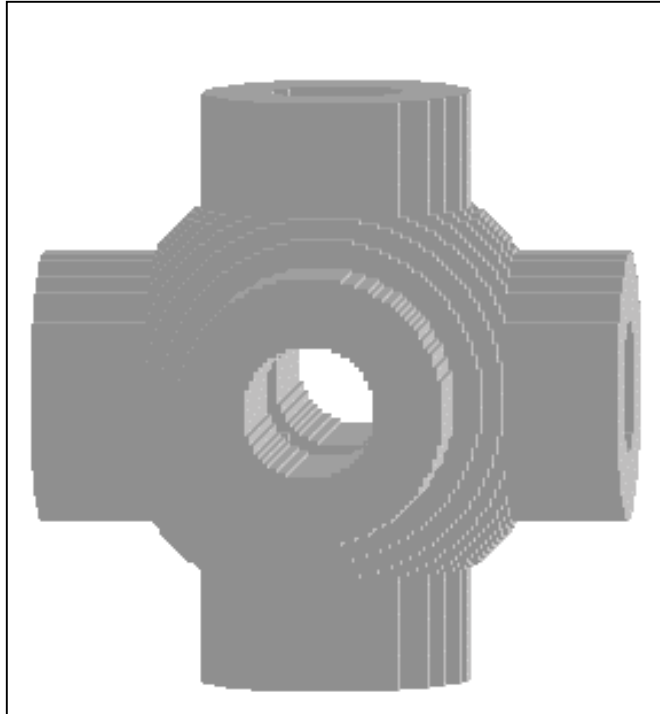
Figura 5.12: Objeto correspondiente al Ejemplo 2 de la tabla 5.4
El objeto esta rotado 50 grados en X, 60 grados en Y y 0 grados en Z



"Elaboración propia"

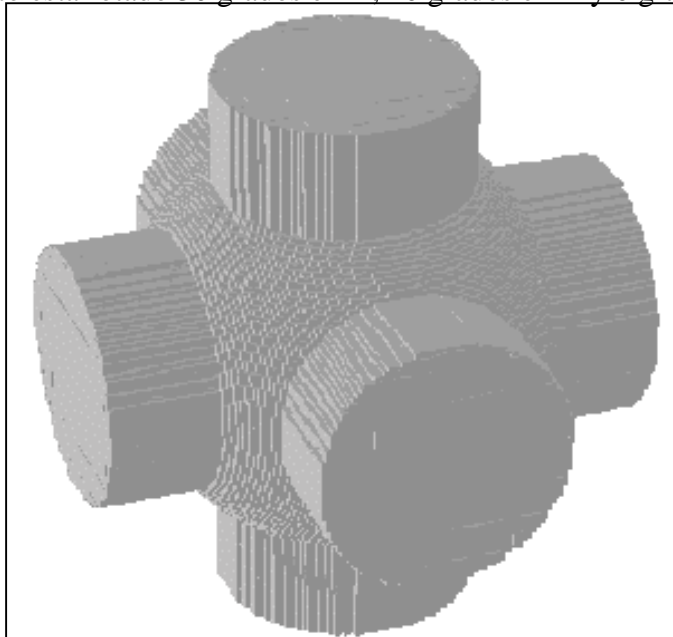
Figura 5.13: Objeto correspondiente al Ejemplo 3 de la tabla 5.4

El objeto esta rotado 10 grados en X, -10 grados en Y y 0 grados en Z



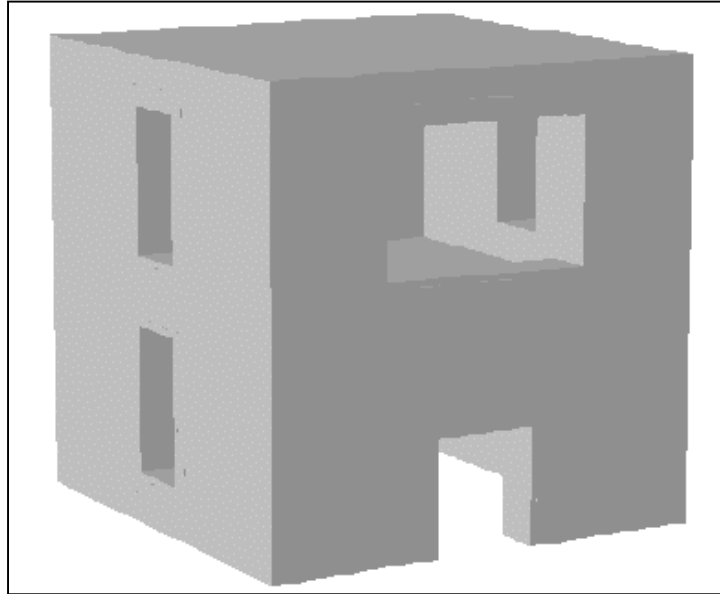
"Elaboración propia"

Figura 5.14: Objeto correspondiente al Ejemplo 4 de la tabla 5.4
El objeto esta rotado 30 grados en X, 20 grados en Y y 0 grados en Z



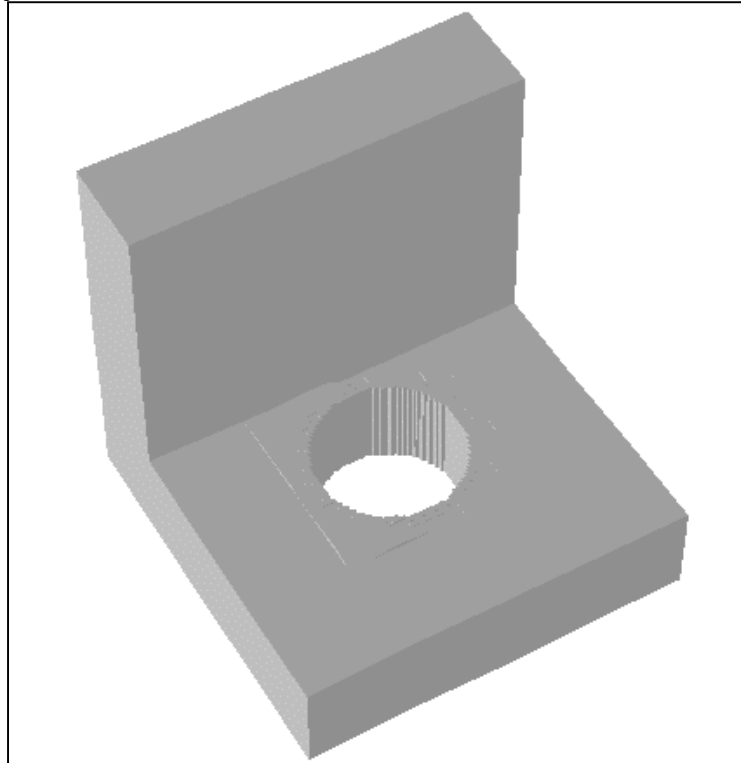
"Elaboración propia"

Figura 5.15: Objeto correspondiente al Ejemplo 5 de la tabla 5.4
El objeto esta rotado 10 grados en X, 40 grados en Y y 0 grados en Z



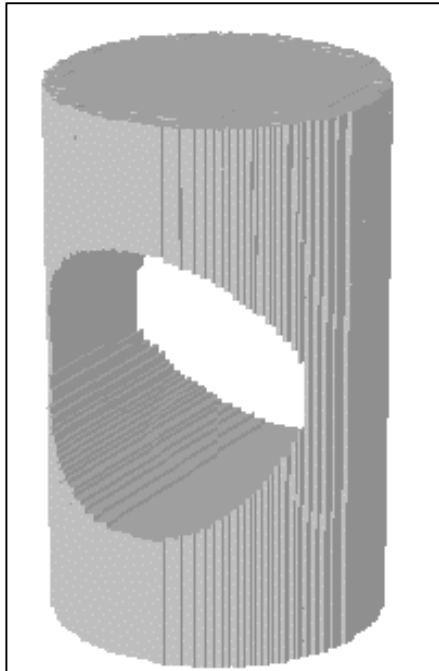
"Elaboración propia"

Figura 5.16: Objeto correspondiente al Ejemplo 6 de la tabla 5.4
El objeto esta rotado 50 grados en X, 30 grados en Y y 0 grados en Z



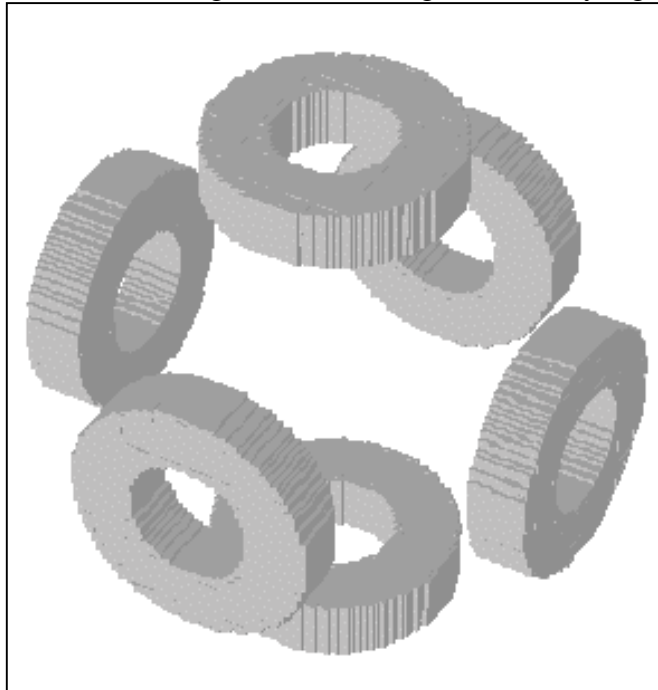
"Elaboración propia"

Figura 5.17: Objeto correspondiente al Ejemplo 7 de la tabla 5.4
El objeto esta rotado 20 grados en X, 60 grados en Y y 0 grados en Z



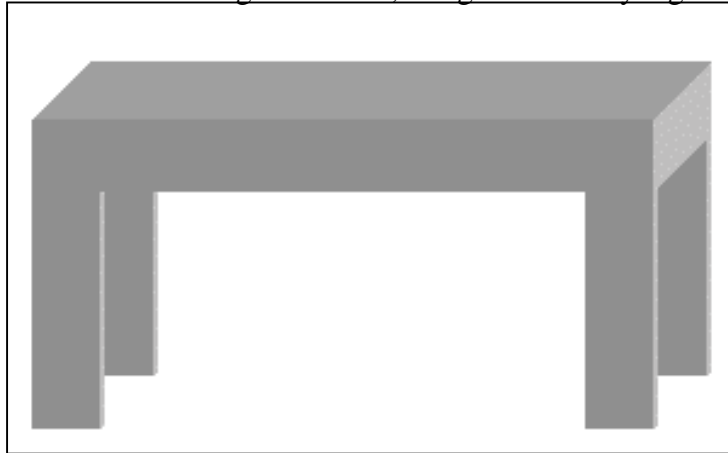
"Elaboración propia"

Figura 5.18: Objeto correspondiente al Ejemplo 8 de la tabla 5.4.
El objeto esta rotado 40 grados en X, 60 grados en Y y 0 grados en Z



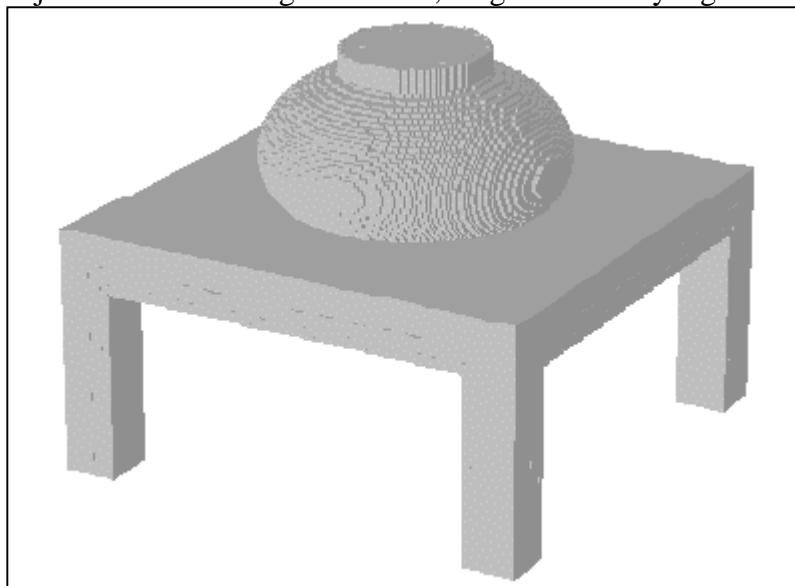
"Elaboración propia"

Figura 5.19: Objeto correspondiente al Ejemplo 9 de la tabla 5.4
El objeto esta rotado 10 grados en X, -10 grados en Y y 0 grados en Z



"Elaboración propia"

Figura 5.20: Objeto correspondiente al Ejemplo 10 de la tabla 5.4
El objeto esta rotado 20 grados en X, 60 grados en Y y 0 grados en Z



"Elaboración propia"

5.2 Conclusión

En los dos ejemplos mostrados se puede apreciar el gran ahorro de memoria cuando se utiliza el método Top Down, y se puede apreciar que entre más complejo sea el objeto el

ahorro de memoria aumenta. Es importante hacer notar que la cantidad de memoria que ocupa cada primitiva varía, generalmente una esfera ocupa muchos más octantes que un paralelepípedo, por lo tanto en los ejemplos aunque se tienen la misma cantidad de primitivas la memoria utilizada es más grande en el primer ejemplo.

En la tabla con los diez ejemplos es claro notar también, que el ahorro de memoria en el Método Top Down con respecto al Método Clásico más que deberse a la precisión es debido a como se construye cada objeto y cuantas primitivas y operaciones Booleanas intervienen al crear el objeto, ya que no nada más se crean los árboles octales de las primitivas en el método Clásico, si no también se crean los árboles octales de las operaciones Booleanas que forman parte de la definición del objeto.

También se nota en los ejemplos que el tiempo en que se tarda en construir un objeto mediante los dos métodos también es diferente, dando mejores resultados en el método Top Down, esto se debe a que nunca se crearan octantes innecesarios, por lo tanto no se analizaran ni se construirán si no es necesario. Y claramente se ve en la tabla que entre más precisión tenga un objeto más ahorro se tiene.