

## Capítulo VI

### Resultados y conclusiones

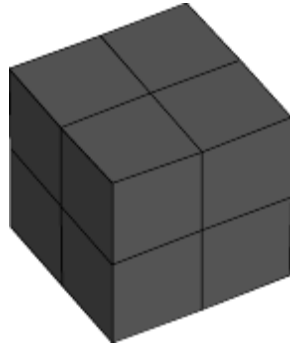
#### 6.1 Objetos utilizados para las pruebas

Con la implementación del modelo de los BSP-OctTrees cuyas características se encuentran descritas en [ApéndiceA], se realizaron diversas pruebas para verificar todas las características, ventajas y desventajas, detalladas a lo largo del presente trabajo. Dichas pruebas emplearon 7 diferentes objetos de creciente complejidad [Tabla6.1], realizándose estudios comparativos tanto en el modelo de los BSP-OctTrees como en el modelo clásico, Extendido y BSP. En [Figura6.(1-7)] se muestran dichos objetos representados mediante BSP-OctTrees. En cuanto al modelo Extendido, se consideraron todos los tipos de nodo (Cara, Arista, Vértice y Casi-Vértice).

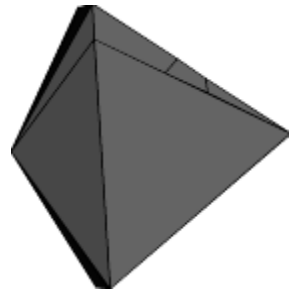
**TABLA 6.1**

Descripción de los objetos utilizados en las pruebas (elaboración propia).

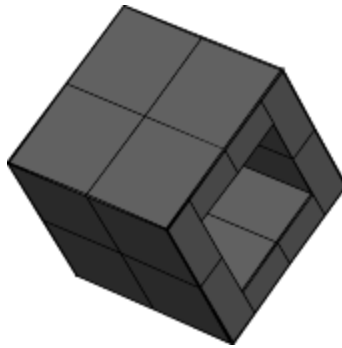
	# Caras	# Aristas	# Vértices	Volumen aproximado (u <sup>3</sup> )
Cubo	6	12	8	1,728,360
Octaedro irregular	8	12	6	1,366,108
Cubo con un agujero	10	24	16	10,920,840
Cubo con tres agujeros	30	72	40	5,801,920
Silla	20	54	36	3,065,050
Venus del Milo	1,418	2,127	711	2,477,902,892
Volcán Popocatépetl	3,485	10,216	6,733	383,882,580



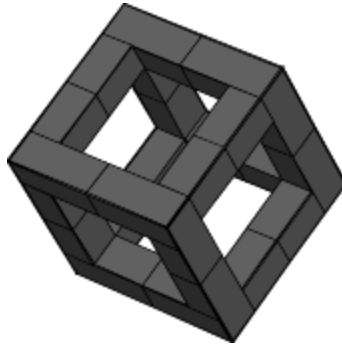
**FIGURA 6.1**  
Cubo (elaboración propia).



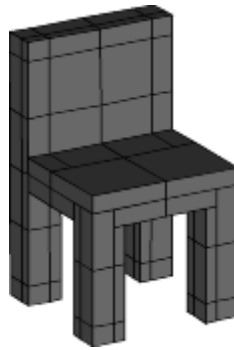
**FIGURA 6.2**  
Octaedro irregular (elaboración propia).



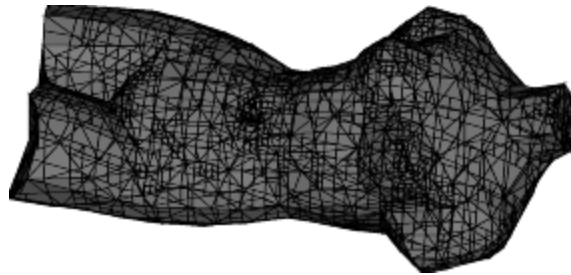
**FIGURA 6.3**  
Cubo con un agujero (elaboración propia).



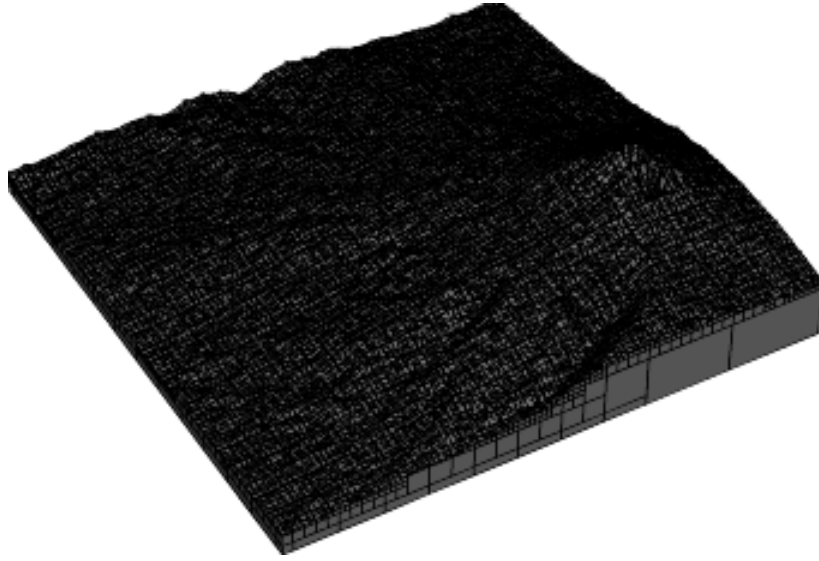
**FIGURA 6.4**  
Cubo con tres agujeros (elaboración propia).



**FIGURA 6.5**  
Silla (elaboración propia).



**FIGURA 6.6**  
Venus del Milo (elaboración propia).



**FIGURA 6.7**  
Volcán Popocatépetl (elaboración propia).

## 6.2 Análisis de resultados obtenidos

Las pruebas realizadas se enfocaron, principalmente, a realizar estudios comparativos entre el espacio requerido para codificar cada uno de los objetos en los diferentes modelos de representación mencionados, así como el tiempo de procesamiento requerido para obtener dichas codificaciones, visualizarlas empleando diferentes técnicas, y manipularlas mediante las diferentes operaciones Booleanas descritas en este trabajo. En lo que respecta al cálculo del tiempo de procesamiento, se utilizó simplemente el reloj disponible en el lenguaje de programación empleado [ApéndiceA.2], cuya resolución se encuentra alrededor de los 10 milisegundos, por lo que cualquier resultado en el presente capítulo que refleje un porcentaje igual a 0.00% únicamente implica que el tiempo de procesamiento fue inferior a los 10 milisegundos por lo que no pudo ser medido con más

precisión. Sin embargo, el objetivo del presente trabajo no era el profundizar en este tipo de mediciones ni el de desarrollar un sistema completo de interrupciones o cualquier otra variante para alcanzar exactitudes superiores, así que esta resolución se consideró como suficiente.

Así pues, en lo que respecta al tiempo de procesamiento requerido para obtener las distintas codificaciones de los objetos probados, en [Tabla6.2] se muestran los resultados obtenidos. Como puede observarse, básicamente en el modelo clásico el tiempo depende directamente del volumen del objeto, mientras que en el resto de los modelos está más relacionado con la complejidad del mismo, es decir, por ejemplo, con el número de vértices. Ciertamente, el modelo BSP-OctTrees resulta ser únicamente más rápido que el modelo clásico, puesto que la detección y creación de los nodos BSP es generalmente más compleja que en el caso de los nodos Extendidos. Sin embargo, este es un proceso que requiere realizarse una única vez, por lo que su peso no es tan nocivo.

**TABLA 6.2**

Tiempos de creación comparativos entre las diferentes codificaciones (elaboración propia).

	<b>OctTree Clásico</b>	<b>OctTree Extendido</b>	<b>BSP- OctTree</b>	<b>BSP</b>
Cubo	100.00%	6.41%	15.56%	0.00%
Octaedro irregular	100.00%	7.49%	14.42%	0.00%
Cubo con un agujero	100.00%	1.36%	3.04%	0.04%
Cubo con tres agujeros	100.00%	1.25%	2.82%	0.07%
Silla	100.00%	1.90%	4.75%	0.06%
Venus del Milo	100.00%	0.32%	0.37%	0.14%
Volcán Popocatépetl	100.00%	<b>N/A</b>	1.71%	<b>N/A</b>

Desde el punto de vista del espacio requerido por cada codificación, en [Tabla6.3] se muestran dichos resultados. Nuevamente, en el modelo clásico se depende directamente del volumen del objeto, mientras que en el resto se depende del número de vértices del mismo. Puede observarse que el modelo BSP-OctTrees permite obtener codificaciones más concisas que el modelo Extendido, y que esta diferencia se hace mucho más notoria a medida que la complejidad de los objetos a representar aumenta. Por otro lado, aunque ciertamente el modelo BSP demostró obtener las representaciones más concisas entre todos los modelos utilizados, diferencia que inclusive se hizo más notoria a medida que la complejidad de los objetos aumentó, también es cierto que el modelo BSP debe ser generalmente utilizado en combinación con el modelo de Fronteras para resolver los problemas mencionados en [Sección2.7], por lo que si se suman los datos de las dos últimas columnas de la tabla puede observarse que el tamaño de las codificaciones que utilizan al modelo BSP y al de Fronteras en combinación resultan ser siempre menos concisos que el modelo BSP-OctTree, el cual no requiere de ningún otro modelo auxiliar.

**TABLA 6.3**

Espacio requerido comparativo entre las diferentes codificaciones (elaboración propia).

	<b>OctTree Clásico</b>	<b>OctTree Extendido</b>	<b>BSP-OctTree</b>	<b>BSP</b>	<b>Modelo de Fronteras</b>
Cubo	100.00%	0.17%	0.17%	0.16%	0.14%
Octaedro irregular	100.00%	0.28%	0.16%	0.15%	0.10%
Cubo con un agujero	100.00%	0.07%	0.07%	0.05%	0.06%
Cubo con tres agujeros	100.00%	0.11%	0.11%	0.07%	0.21%
Silla	100.00%	0.24%	0.21%	0.13%	0.29%
Venus del Milo	100.00%	0.65%	0.36%	0.27%	0.17%
Volcán Popocatépetl	100.00%	<b>N/A</b>	4.08%	<b>N/A</b>	2.55%

Respecto a la visualización de los diferentes modelos, se emplearon cuatro técnicas diferentes para cada uno, es decir, se visualizaron tanto directamente en pantalla como empleando una imagen *offscreen*, y en ambos casos se utilizó tanto el sistema nativo del lenguaje en el que se realizó la implementación como utilizando el estándar OpenGL [OpenGL97]. En [Tabla6.4] pueden observarse los resultados obtenidos. En resumen:

- Como anteriormente, el modelo clásico dependió del volumen de los objetos representados y el resto dependió de la complejidad de los mismos.
- Respecto a las técnicas empleadas, la más eficiente resultó ser el uso de OpenGL en *offscreen*, excepto para el modelo BSP donde lo más efectivo resultó ser el lenguaje nativo directamente a pantalla (debido principalmente a que el modelo BSP fue tan conciso que se requirió un mínimo número de llamadas OpenGL, por lo que el *overhead* del establecimiento de la comunicación entre el lenguaje nativo y OpenGL provocó el mayor tiempo de ejecución).
- El empleo del lenguaje nativo en *offscreen* resultó ser peor que directamente a pantalla debido a que la transferencia de la imagen *offscreen* es relativamente lenta.
- OpenGL directamente a pantalla resultó ser la técnica menos eficiente, ya que OpenGL no está diseñado para este tipo de uso, excepto para los objetos más complejos en el modelo clásico donde esta técnica resultó ser ligeramente mejor que el lenguaje nativo en *offscreen* (debido principalmente al gran número de llamadas OpenGL que fueron requeridas).
- En cuanto a comparaciones directas entre las distintas codificaciones, nuevamente los BSP-OctTrees demostraron ser mejores que el modelo Extendido.

- El modelo clásico desde luego fue el que requirió mayor tiempo de ejecución, mientras que el modelo BSP resultó ser el más rápido, aunque desde luego ya se mencionaron previamente las desventajas de este modelo al manipular objetos complejos.

**TABLA 6.4**

Tiempos de visualización comparativos entre los diferentes modelos (elaboración propia).

	<b>Java Offscreen</b>	<b>Java a pantalla</b>	<b>OpenGL Offscreen</b>	<b>OpenGL a pantalla</b>
<b>OctTree Clásico</b>				
Cubo	89.20%	100.00%	61.18%	264.17%
Octaedro irregular	89.56%	100.00%	60.89%	264.44%
Cubo con un agujero	89.07%	100.00%	60.59%	261.24%
Cubo con tres agujeros	89.39%	100.00%	60.96%	262.44%
Silla	86.90%	100.00%	58.55%	257.10%
Venus del Milo	135.43%	100.00%	85.91%	112.09%
Volcán Popocatépetl	139.75%	100.00%	94.03%	116.61%
<b>OctTree Extendido</b>				
Cubo	0.19%	0.11%	N/A	N/A
Octaedro irregular	0.27%	0.27%	N/A	N/A
Cubo con un agujero	0.03%	0.03%	N/A	N/A
Cubo con tres agujeros	0.06%	0.04%	N/A	N/A
Silla	0.10%	0.10%	N/A	N/A
Venus del Milo	0.18%	0.16%	N/A	N/A
Volcán Popocatépetl	N/A	N/A	N/A	N/A
<b>BSP-OctTree</b>				
Cubo	0.10%	0.19%	0.10%	0.10%
Octaedro irregular	0.07%	0.07%	0.07%	0.07%
Cubo con un agujero	0.05%	0.03%	0.05%	0.06%
Cubo con tres agujeros	0.07%	0.06%	0.07%	0.11%
Silla	0.07%	0.07%	0.10%	0.14%
Venus del Milo	0.13%	0.13%	0.12%	0.14%
Volcán Popocatépetl	1.76%	1.54%	1.50%	1.83%
<b>BSP</b>				
Cubo	0.00%	0.00%	0.00%	0.00%
Octaedro irregular	0.00%	0.00%	0.00%	0.00%
Cubo con un agujero	0.02%	0.03%	0.05%	0.05%
Cubo con tres agujeros	0.04%	0.04%	0.06%	0.07%
Silla	0.03%	0.03%	0.07%	0.07%
Venus del Milo	0.42%	0.40%	0.41%	0.42%
Volcán Popocatépetl	N/A	N/A	N/A	N/A

En cuanto a las operaciones Booleanas [Tabla6.5], [Tabla6.6], [Tabla6.7], [Tabla6.8], pudo comprobarse la gran eficiencia del modelo clásico en este tipo de



operaciones. Sin embargo, a pesar de ello, el modelo BSP-OctTrees resultó ser aún más eficiente. Nuevamente, el modelo BSP resultó ser también más rápido, pero, como ya se mencionó, dadas las deficiencias que tiene en otras áreas resulta difícil utilizarlo como modelo único de representación. Sin embargo, en el caso de la operación de complemento, los tiempos son extremadamente reducidos en todos los casos, pero puede apreciarse que el modelo Extendido y el BSP resultaron ser los más eficientes, siendo que el modelo de los BSP-OctTrees, a pesar de ser mejor que el modelo clásico, requiere ligeramente más tiempo de procesamiento para complementar los nodos BSP que el requerido para los nodos Extendidos. Resulta importante mencionar que las operaciones de unión y diferencia fueron implementadas utilizando las expresiones descritas en [Sección2.4.4], es decir, combinando las operaciones de intersección y complemento, por lo que lógicamente sus resultados muestran tiempos de procesamiento ligeramente mayores a los de la operación de intersección, y resultan equivalentes al número de operaciones de intersección y/o complemento que deben realizarse para efectuar la operación deseada. En otras palabras, las cifras de [Tabla6.6] y [Tabla6.7] están relacionadas con las de [Tabla6.5].

**TABLA 6.5**

Tiempos de procesamiento comparativos para el Complemento (elaboración propia).

	<b>OctTree Clásico</b>	<b>OctTree Extendido</b>	<b>BSP- OctTree</b>	<b>BSP</b>
Cubo	100.00%	0.00%	0.00%	0.00%
Octaedro irregular	100.00%	0.00%	0.00%	0.00%
Cubo con un agujero	100.00%	0.00%	1.18%	0.00%
Cubo con tres agujeros	100.00%	0.00%	1.30%	0.00%
Silla	100.00%	0.00%	0.00%	0.00%
Venus del Milo	100.00%	0.37%	0.26%	0.04%
Volcán Popocatépetl	100.00%	<b>N/A</b>	2.11%	<b>N/A</b>

**TABLA 6.6**

Tiempos de procesamiento comparativos para la Intersección (elaboración propia).

	<b>OctTree Clásico</b>	<b>BSP- OctTree</b>	<b>BSP</b>
Cubo, Octaedro irregular	100.00%	80.16%	0.66%
Silla, Cubo con tres agujeros	100.00%	68.71%	2.05%
Silla, Cubo con un agujero, Octaedro irregular	100.00%	69.47%	0.79%

**TABLA 6.7**

Tiempos de procesamiento comparativos para la Unión (elaboración propia).

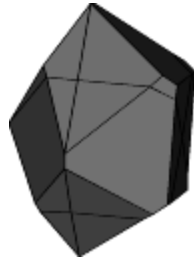
	<b>OctTree Clásico</b>	<b>BSP- OctTree</b>	<b>BSP</b>
Cubo, Octaedro irregular	154.96%	88.76%	15.21%
Silla, Cubo con tres agujeros	285.66%	77.42%	13.82%
Silla, Cubo con un agujero, Octaedro irregular	305.49%	77.61%	14.45%

**TABLA 6.8**

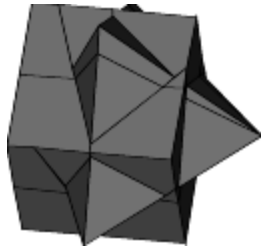
Tiempos de procesamiento comparativos para la Diferencia (elaboración propia).

	<b>OctTree Clásico</b>	<b>BSP- OctTree</b>	<b>BSP</b>
Cubo, Octaedro irregular	147.02%	84.72%	14.55%
Silla, Cubo con tres agujeros	168.20%	74.35%	13.31%
Silla, Cubo con un agujero, Octaedro irregular	147.61%	75.25%	12.61%

En [Figura6.(8-16)] se muestran los resultados de la realización de las operaciones Booleanas de intersección, unión y diferencia que corresponden a [Tabla6.6], [Tabla6.7], [Tabla6.8], para el modelo de los BSP-OctTrees.



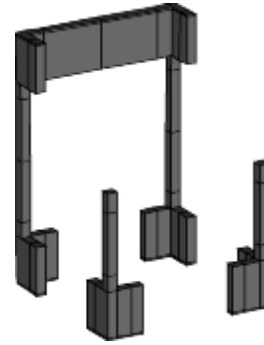
**FIGURA 6.8**  
Intersección de Cubo con Octaedro irregular (elaboración propia).



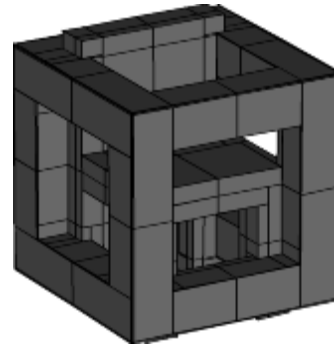
**FIGURA 6.9**  
Unión de Cubo con Octaedro irregular (elaboración propia).



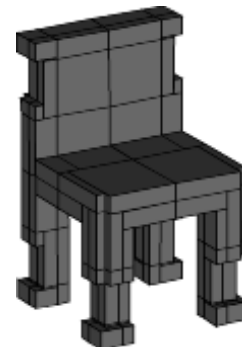
**FIGURA 6.10**  
Resta de Cubo con Octaedro irregular (elaboración propia).



**FIGURA 6.11**  
Intersección de Silla con Cubo con tres agujeros (elaboración propia).



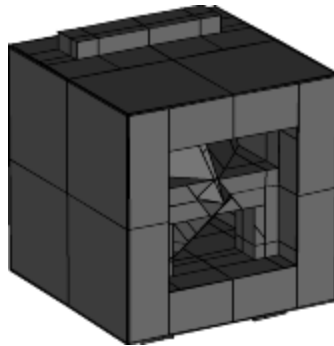
**FIGURA 6.12**  
Unión de Silla con Cubo con tres agujeros (elaboración propia).



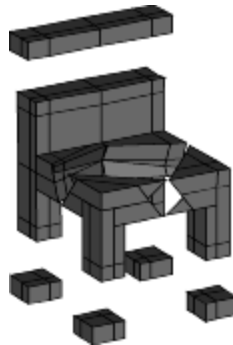
**FIGURA 6.13**  
Resta de Silla con Cubo con tres agujeros (elaboración propia).

**FIGURA 6.14**

Intersección de Silla con Cubo con un agujero con Octaedro irregular (elaboración propia).

**FIGURA 6.15**

Unión de Silla con Cubo con un agujero con Octaedro irregular (elaboración propia).

**FIGURA 6.16**

Resta de Silla con Cubo con un agujero con Octaedro irregular (elaboración propia).

Por último, en [Tabla6.9], [Tabla6.10] se muestran los resultados de aplicar un caso muy específico de la operación de intersección, es decir, la realización de cortes planos de un objeto y de cortes cilíndricos (la intersección de un cilindro vertical con un objeto). Los resultados de estas dos tablas son equivalentes a los de las operaciones Booleanas generales..

**TABLA 6.9**

Tiempos de procesamiento comparativos para el Corte Plano (elaboración propia).

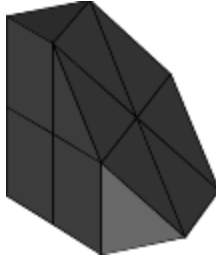
	<b>BSP- OctTree</b>	<b>BSP</b>
Cubo	100.00%	0.00%
Octaedro irregular	100.00%	0.00%
Cubo con un agujero	100.00%	0.90%
Cubo con tres agujeros	100.00%	0.49%
Silla	100.00%	0.56%
Venus del Milo	100.00%	23.61%
Volcán Popocatépetl	100.00%	<b>N/A</b>

**TABLA 6.10**

Tiempos de procesamiento comparativos para el Corte Cilíndrico (elaboración propia).

	<b>BSP- OctTree</b>	<b>BSP</b>
Cubo	100.00%	0.00%
Octaedro irregular	100.00%	0.41%
Cubo con un agujero	100.00%	0.79%
Cubo con tres agujeros	100.00%	0.37%
Silla	100.00%	0.47%
Venus del Milo	100.00%	42.44%
Volcán Popocatépetl	100.00%	<b>N/A</b>

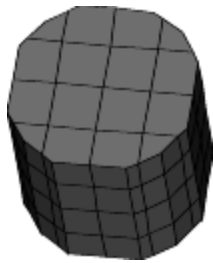
En [Figura6.(17-30)] se muestran los resultados de la realización de los cortes planos y cilíndricos que corresponden a [Tabla6.9], [Tabla6.10], para el modelo de los BSP-OctTrees.



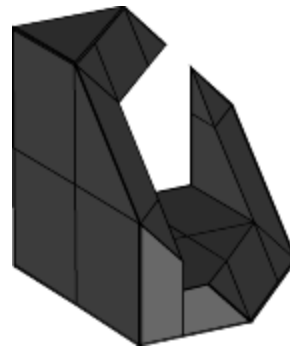
**FIGURA 6.17**  
Corte plano de Cubo (elaboración propia).



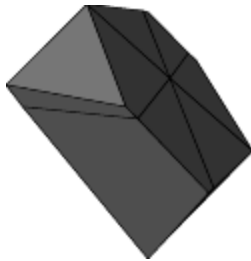
**FIGURA 6.20**  
Corte cilíndrico de Octaedro irregular (elaboración propia).



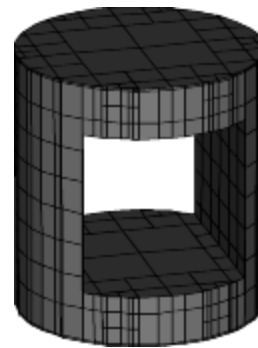
**FIGURA 6.18**  
Corte cilíndrico de Cubo (elaboración propia).



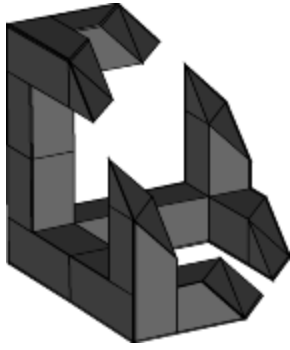
**FIGURA 6.21**  
Corte plano de Cubo con un agujero (elaboración propia).



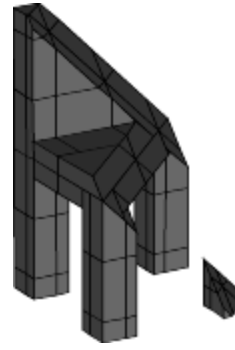
**FIGURA 6.19**  
Corte plano de Octaedro irregular (elaboración propia).



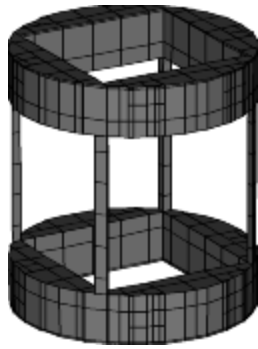
**FIGURA 6.22**  
Corte cilíndrico de Cubo con un agujero (elaboración propia).



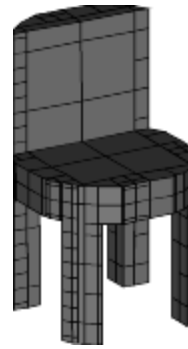
**FIGURA 6.23**  
Corte plano de Cubo con tres agujeros  
(elaboración propia).



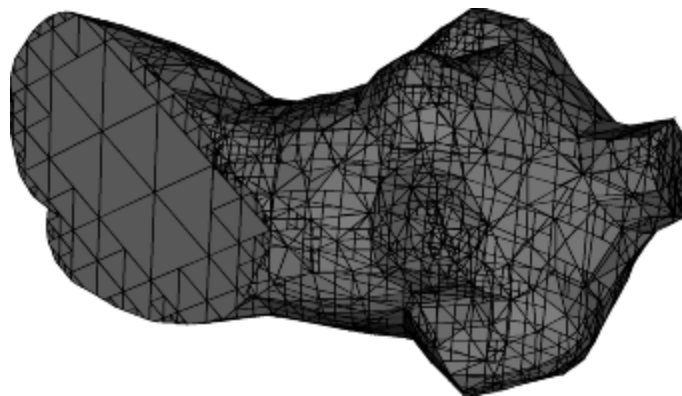
**FIGURA 6.25**  
Corte plano de Silla (elaboración propia).



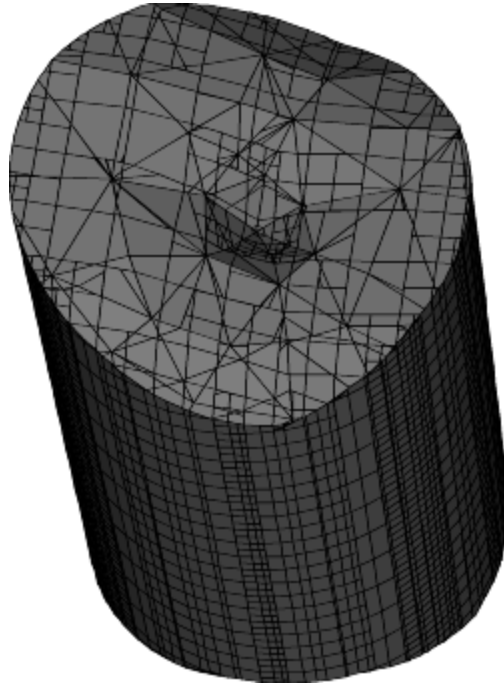
**FIGURA 6.24**  
Corte cilíndrico de Cubo con tres  
agujeros (elaboración propia).



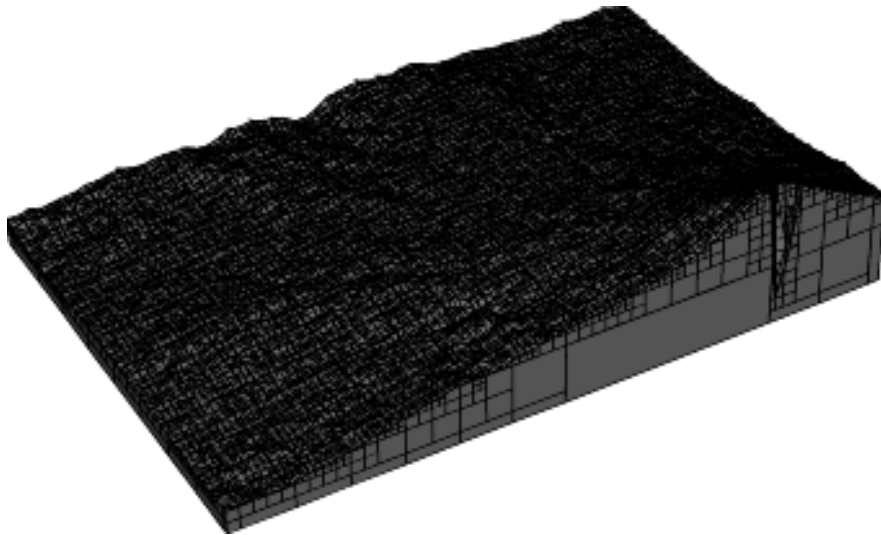
**FIGURA 6.26**  
Corte cilíndrico de Silla (elaboración  
propia).



**FIGURA 6.27**  
Corte plano de Venus del Milo (elaboración propia).

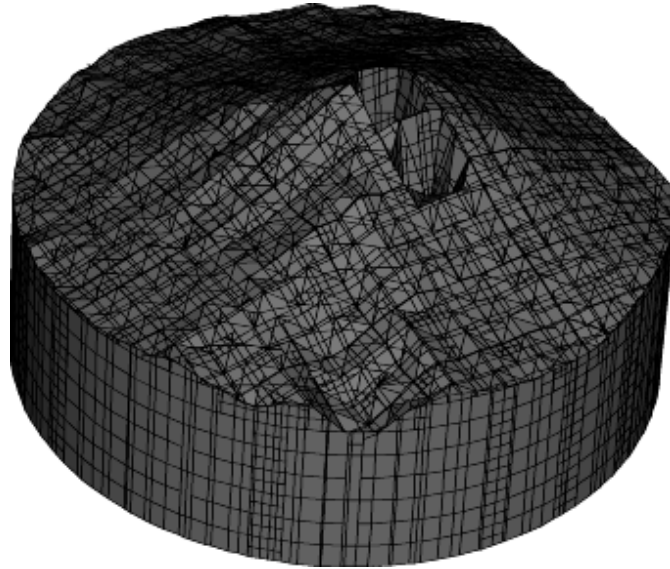


**FIGURA 6.28**  
Corte cilíndrico de Venus del Milo (elaboración propia).



**FIGURA 6.29**  
Corte plano de Volcán Popocatepetl – justo en el cráter (elaboración propia).





**FIGURA 6.30**  
Corte cilíndrico de Volcán Popocatépetl (elaboración propia).

### 6.3 Conclusiones y trabajo futuro

En conclusión, vimos que el modelo descrito en el presente trabajo, los BSP-OctTrees, son una representación sumamente eficiente tanto en la generación de su estructura [CapítuloIII], como en sus algoritmos de visualización [CapítuloIV], de reconstrucción de fronteras [CapítuloIV] y de realización de operaciones Booleanas [CapítuloV]. Además, siendo un modelo que permite obtener representaciones exactas y altamente concisas para cualquier poliedro *manifold* sin importar su complejidad [Sección3.4], y que soporta de manera natural la eliminación de partes ocultas [Sección3.4], se convierte en una codificación robusta que tiene la capacidad de manipular objetos de alta complejidad, como lo son por ejemplo los datos geográficos del volcán Popocatépetl, y se considera que este modelo tiene la fuerza suficiente para ser considerado como importante punto de estudio dentro de la graficación por computadora.

Ciertamente el presente trabajo deja algunos puntos interesantes a considerar como trabajo futuro:

- La optimización de la construcción de los árboles BSP dentro de los nodos BSP para obtener árboles más concisos, y también para utilizar algoritmos aún más robustos que eviten o reduzcan los errores de precisión numérica.
- La reconstrucción de los datos requeridos del modelo de fronteras, que fue mencionada en [Sección5.3.2] y que se emplea cuando una operación de intersección produce un nodo BSP que no cumple con la restricción mencionada en [Sección3.3.3]. En este caso no es suficiente con aplicar el algoritmo descrito en [Sección4.2.1], puesto que también es necesario “fusionar” las caras coplanares que tienen una o más aristas en común. Este proceso o similares son descritos en varias fuentes de la literatura [Thibault87], [Naylor90], [Greiner98], [Maillot92], [Toussaint85], [Vatti92], ya que puede ser fácilmente visto como un problema 2D. Sin embargo, no fue implementado en el presente trabajo debido principalmente a limitaciones de tiempo.
- La paralelización de los algoritmos de visualización y de operaciones Booleanas.
- La implementación de cortes multipoligonales.
- La implementación de más *parsers* que permitan leer otros modelos 3D de entrada.
- Implementación de las técnicas mencionadas en [Naylor90] para lograr la completa robustez numérica en el algoritmo de fusión de árboles BSP [Sección5.2.2].
- Implementación de alguna técnica para poder calcular el volumen exacto de un nodo BSP.