

CAPÍTULO 5

Resultados

Este capítulo condensa los principales resultados obtenidos durante algunas de las fases de prueba del modelo de vértices extremos implementado en Java. Se comienza describiendo la interfaz principal de la aplicación para después continuar con una breve descripción del editor de pseudo poliedros ortogonales incorporado en la misma, las características con las que éste cuenta así como sus principales cualidades, para después seguir con la descripción de los resultados referentes a las operaciones definidas en el modelo EVM, como las operaciones booleanas, la partición de sólidos y otras características involucradas en un sistema modelador de sólidos.

5.1 Interfaz principal del software EVM.

El objetivo general del presente trabajo de tesis era el de contar con un software que implementara las características del modelo de vértices extremos, así como el de que este mismo software fuera capaz de visualizar los pseudo poliedros ortogonales que maneja dicho modelo.

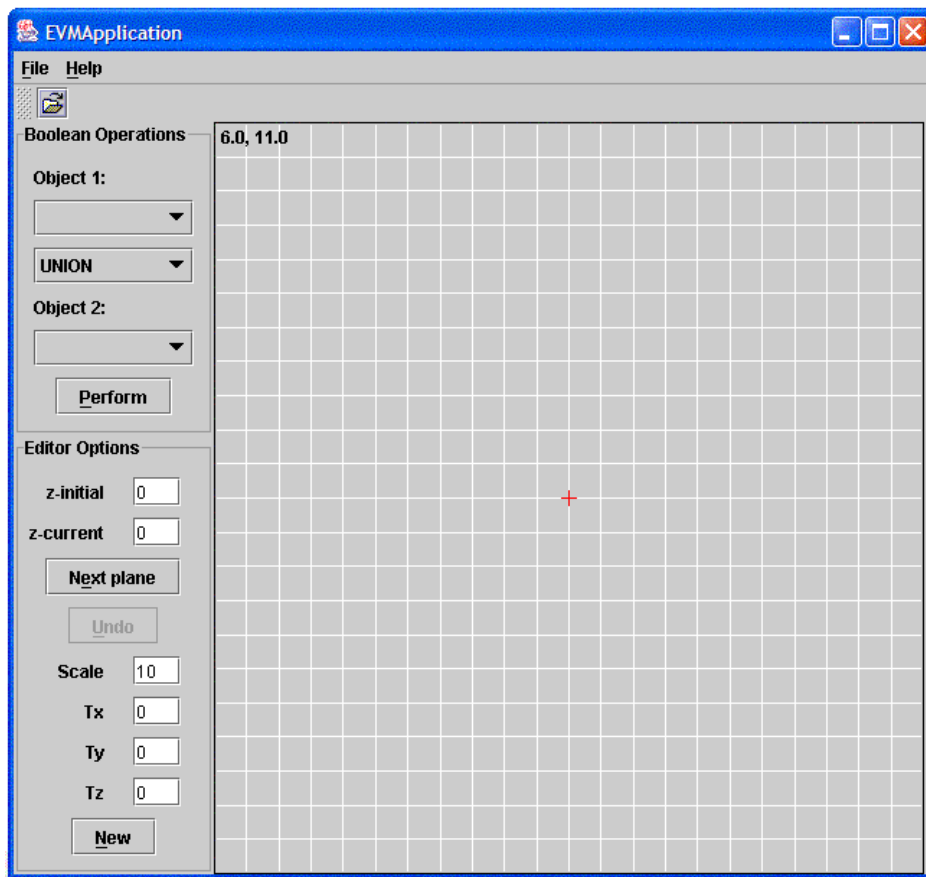


Figura 5-1 Interfaz principal de la aplicación del modelo de vértices extremos.

La interfaz principal de la aplicación desarrollada como parte de este proyecto se muestra en la Figura 5-1, en donde se observan las características principales de la misma. De manera general, esta interfaz está dividida en cinco partes: la barra de menú, la barra de herramientas, el panel de control para las operaciones booleanas, el panel de control para

las opciones del editor y un lienzo que sirve como área de dibujo para la definición de pseudo poliedros ortogonales. Cada una de estas partes será explicada por separado, las primeras tres en esta misma sección y las restantes en la sección relacionada con el editor.

En la Figura 5-2 se muestran tres de los elementos de interacción con los que cuenta la interfaz. En los incisos (a) y (b) de la misma se presenta la barra de menús principal, integrada por los elementos de menú relacionados con las funciones asociadas a los archivos y la ayuda. Dentro del menú archivo (parte (a) de la figura) se encuentran las opciones de abrir archivo y salir. Para abrir un archivo, se presentan opciones diferentes según sea el tipo de archivo que se desee abrir, la aplicación es capaz de abrir archivos que representen objetos EVM válidos en las representaciones de *EVM*, *BRep* y *Octree* tal y como se describió en el capítulo referente al diseño de la aplicación. Al ser seleccionada alguna de estas opciones, se presenta un cuadro de diálogo que permite buscar y seleccionar el archivo deseado. La opción de menú salir por su parte y como su nombre lo indica, termina con la aplicación.

El elemento de menú ayuda (Figura 5-2 (b)) tiene una opción para presentar en pantalla una pequeña guía de uso del editor de pseudo poliedros y de los elementos presentados en ésta interfaz principal; también presenta una opción que hace referencia a la versión de la aplicación.

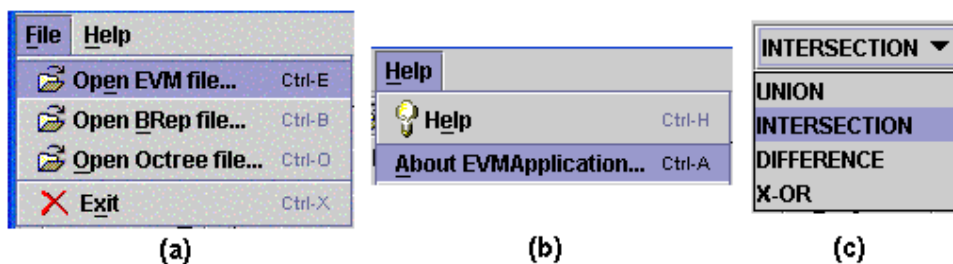


Figura 5-2 Elementos de interacción de la interfaz principal. En (a) se muestra el menú archivo y en (b) el menú de ayuda, cada uno de ellos con sus opciones respectivas. La figura (c) muestra una lista de opciones para las operaciones booleanas.

Es importante mencionar que cada uno de los elementos de menú que son o que podrían ser de frecuente uso, tienen asociado su método abreviado de teclado, esto como guía de usabilidad dentro de las técnicas de interacción que se integraron a la interfaz.

La Figura 5-1 ilustra también que el panel de control de las operaciones booleanas cuenta con listas de selección como las de la Figura 5-2 (c), la cual es la lista de selección correspondiente al operador para dichas operaciones. Las otras dos listas de selección correspondientes a objeto1 y objeto2, contendrán los nombres de los objetos EVM que hayan sido cargados desde un archivo o creados por medio de algún otro mecanismo de los descritos más adelante, como lo son el efectuar una operación booleana sobre dos objetos EVM, a través del uso de editor, o al abrir un archivo definido en otra representación por ejemplo. La barra de herramientas sólo posee un botón que permite abrir archivos definidos dentro del modelo EVM, pero su finalidad es que pueda crecer al agregar más características a la aplicación.

En la siguiente sección se describe un componente importante de esta interfaz: el editor de pseudo poliedros ortogonales.

5.2 Editor de pseudo poliedros ortogonales.

El editor de pseudo poliedros ortogonales es una herramienta integrada dentro de la aplicación general del modelo de vértices extremos, se construyó con la finalidad de poder definir y crear pseudo poliedros ortogonales de mediana complejidad que, en combinación de las operaciones de escalamiento, traslación y las operaciones booleanas permitirán la creación de modelos más complejos de pseudo poliedros ortogonales a partir de otros más sencillos.

El editor cuenta con un área de dibujo cuadrículada para facilitar la definición de los objetos (véase la Figura 5-1). Esta malla representa una discretización del espacio euclidiano bidimensional en un rango de $[-10, 10]$ tanto para el eje y como para el eje x . La definición de los objetos se realiza con la ayuda del ratón. El botón izquierdo del ratón sirve para marcar los puntos extremos de un segmento de línea dado. Estas marcas son realizadas al

efectuar *clicks* (presionar y soltar) con dicho botón, no es necesario realizar la operación de arrastre del ratón. Con el botón derecho, se puede ir eliminando líneas previamente definidas en una forma LIFO¹. Debe mencionarse también que dentro de las características con las que cuenta el editor está el panel de control del mismo (Figura 5-3), dicho panel está compuesto de lo siguientes campos:

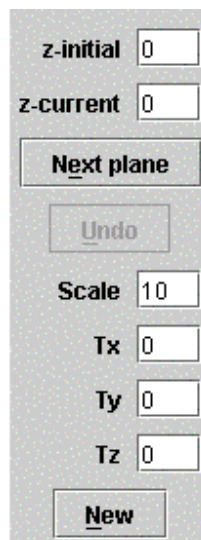


Figura 5-3 Panel de control del Editor.

- **Campo z-initial:** en este campo se indica el plano inicial en z del objeto a dibujar. Una vez que se presiona el botón *Next plane* este campo es deshabilitado.
- **Campo z-current:** indica el plano actual en z para el pseudo poliedro. La extrusión tiene como planos en z la coordenada del plano actual, definida por este campo, y la coordenada del plano anterior del objeto. Para el caso inicial, la coordenada anterior es precisamente *z-initial*.
- **Botón Next plane:** permite obtener una representación gráfica del objeto según se vaya creando, así como alimentar al sistema para indicarle que una nueva

¹ La última línea definida, es la primera en ser eliminada.

parte del objeto en edición ha sido generada, y debe ser agregada al mismo. La representación inicial durante el proceso de edición es en el modelo de alambres, pero puede ser cambiada a una visualización como objeto sólido. Este y otros detalles se describen con mayor amplitud en este mismo capítulo.

- **Botón Undo:** permite restaurar el estado anterior del objeto hasta como estaba antes de haber presionado por última vez el botón *Next Plane*. Para el caso inicial. Este botón aparece deshabilitado y sólo puede deshacerse la última edición introducida.
 - **Campo Scale:** es el factor de escala para el poliedro. Dado que el objeto generado se encuentra limitado al espacio bidimensional de $[-10, 10]$ tanto como para x como para y , este factor de escala determinará las proporciones reales del objeto final. El escalamiento efectuado sobre el objeto es un escalamiento uniforme y se realiza con respecto al origen. Para obtener transformaciones uniformes, esta opción es deshabilitada una vez que se ha comenzado la edición.
 - **Campo Tx:** indica el valor de traslación del objeto en x . Para obtener transformaciones uniformes, esta opción es deshabilitada una vez que se ha comenzado la edición.
 - **Campo Ty:** indica el valor de traslación del objeto en y . Para obtener transformaciones uniformes, esta opción es deshabilitada una vez que se ha comenzado la edición.
 - **Campo Tz:** indica el valor de traslación del objeto en z . Para obtener transformaciones uniformes, esta opción es deshabilitada una vez que se ha comenzado la edición.
 - **Botón New:** es utilizado para indicarle a la aplicación que se desea empezar con la generación de un nuevo pseudo poliedro, por lo que entre otras cosas, realiza una limpieza de las estructuras de datos internas para permitir la definición de un nuevo objeto.
-

En la esquina superior izquierda del área de dibujo (Figura 5-1) aparece constantemente las coordenadas lógicas de la posición actual del ratón para facilitar la ubicación de las líneas generadas. Vale la pena hacer mención aquí de que no importa cuál sea la dimensión de la ventana, esto es, cuáles sean las coordenadas físicas del área de trabajo, las coordenadas lógicas siempre permanecen constantes.

El editor permitirá únicamente y por obvias razones la creación de segmentos de líneas ortogonales, no se pueden crear segmentos de líneas de cualquier tipo. En cada definición de segmento de línea, el sistema define de manera automática, al punto final del segmento de línea recién creado, como el punto inicial del nuevo segmento de línea en proceso.

La definición del objeto se va haciendo en planos perpendiculares a z en un esquema de dibujo de atrás hacia delante. La idea es pensar en el objeto como una sucesión de objetos EVM e irlos acomodando sobre el eje z , una vez que se ha terminado de definir alguno de los objetos EVM, se realiza una validación para verificar que el objeto EVM recién definido representa un EVM válido. El sistema genera extrusiones de cada una de las cajas definidas y las une con las cajas previamente definidas para constituir un objeto EVM unificado. Durante el proceso de fusión, el sistema realiza entre otros procesos internos, la eliminación de vértices repetidos, con la finalidad de que, a partir de objetos EVM válidos previamente verificados, se genere un nuevo objeto EVM que también sea válido. Este proceso de validación de las cajas que van definiendo al objeto, involucra la aplicación del teorema de validación para objetos EVM, dicho teorema ha sido descrito en el capítulo 3.

Es importante recordar que el editor posee características de corrección de puntos de los objetos EVM definidos sobre un plano determinado, este proceso es descrito en el capítulo referente al diseño e implementación del sistema.

Por último, la Figura 5-4 muestra un objeto EVM generado con el editor.

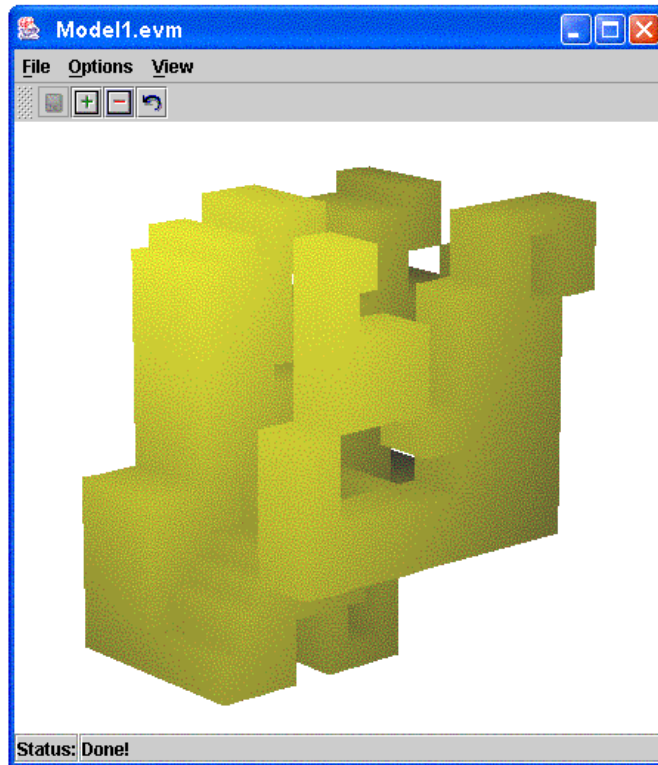


Figura 5-4 Objeto generado con el editor de pseudo poliedros ortogonales.

5.3 Ventana de visualización.

Cada vez que se abre un archivo, se genera un nuevo objeto con el editor, o se realiza alguna operación que involucre la creación de objetos EVM nuevos, se crea una nueva ventana de visualización como la de la Figura 5-5 para cada uno de ellos. Cada una de estas ventanas, aunadas a la de la aplicación principal constituyen los elementos principales de interacción con el modelo EVM.

La ventana de visualización en íntima coordinación con la aplicación principal mantiene y administra la lista de objetos EVM activos. Cada una de estas ventanas de visualización tienen como título el nombre del objeto que presentan, si dicho objeto es nuevo debido a que ha sido creado a partir de alguna de las operaciones del modelo o del editor, el nombre por omisión que el sistema asigna a los objetos nuevos es *Model#.evm*, en

donde # representa un número entero secuencial administrado por la aplicación, dicho nombre es también actualizado en las listas de selección de objetos dentro del panel de las operaciones booleanas y permanece así hasta que el usuario de manera explícita guarde el objeto EVM realizándose los cambios y actualizaciones correspondientes.

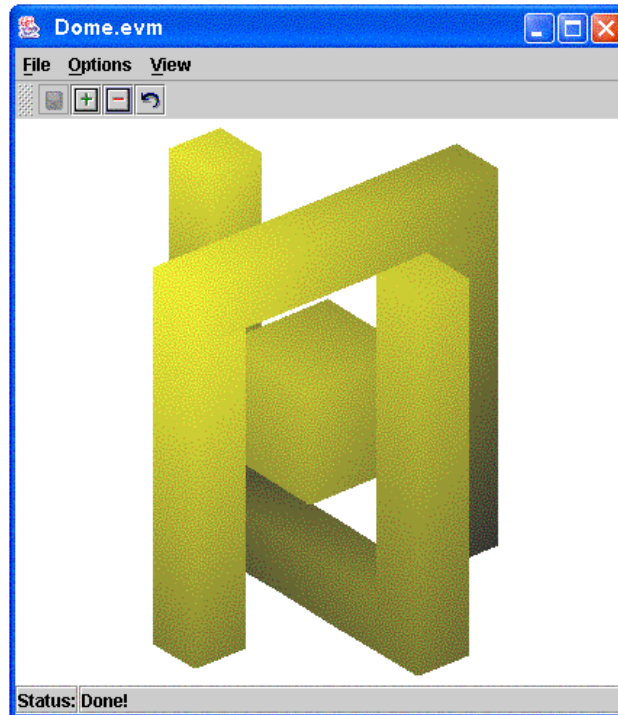


Figura 5-5 Ventana de visualización de objetos EVM.

Cada objeto contenido dentro de una ventana de visualización puede ser manipulado de manera independiente, se puede rotar por ejemplo, utilizando el ratón a través del principio de manipulación directa, esto es, mover el objeto directamente con el ratón para producir la rotación deseada sobre el mismo. También es posible escalar el objeto utilizando los botones destinados para este fin, estos botones se encuentran en la barra de herramientas de cada ventana de visualización, misma que se ilustran también en la Figura 5-5. Los botones de la barra de herramientas, así como los elementos de mayor utilidad han sido dotados con la característica de “*tip text*”, la cual es una pequeña guía acerca de la

función del elemento apuntado por el ratón. Esto último constituye también una característica de usabilidad y de guía de diseño de interfaces.

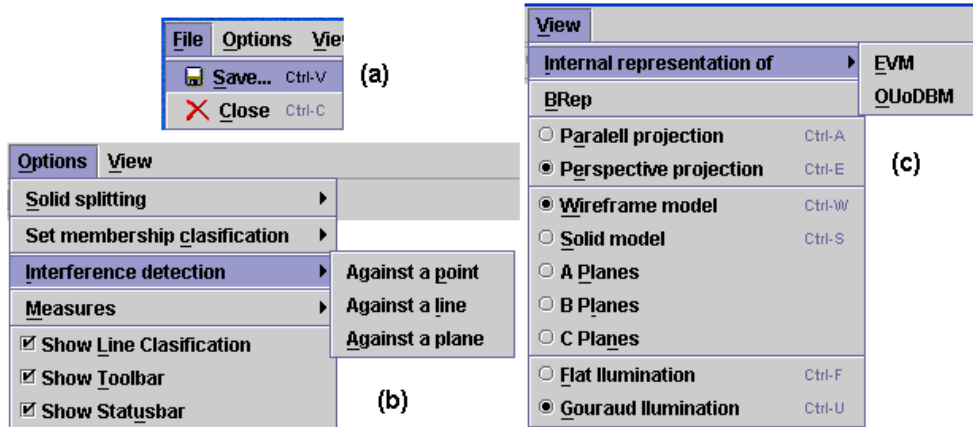


Figura 5-6 Menús de la ventana de visualización.

La Figura 5-6 ilustra los menús integrados dentro de la ventana de visualización. En la figura (a) se observa las opciones del menú archivo, las cuales son guardar y cerrar la ventana, la opción de guardar está activa sólo para los elementos que no tengan todavía un nombre asignado por el usuario.

En la Figura 5-6 (b) se observa el menú de opciones. Este menú contiene las operaciones que pueden ser realizadas sobre el objeto en cuestión, estas operaciones están agrupadas y son: la partición de sólidos, la clasificación de elementos geométricos, la detección de interferencia y algunas medidas del pseudo poliedro ortogonal. Este menú muestra también opciones para mostrar u ocultar la barra de estado, la barra de herramientas y la visualización de la línea como parte de la opción de la clasificación de elementos geométricos. Este último aspecto se discute un poco más adelante.

Finalmente, la Figura 5-6 (c) muestra el menú de vista. En este menú se encuentran opciones para que se pueda visualizar una representación interna del objeto como EVM, como OUoDBM o como BRep en su estructura jerárquica. La representación en BRep es una representación en texto de una estructura de datos interna utilizada para la

representación jerárquica del modelo de fronteras. El objeto en curso puede ser presentado en proyección perspectiva o paralela, en modelo de alambres, como un objeto sólido o en alguna de sus representaciones de planos de vértices. También es posible cambiar aquí el modelo de iluminación.

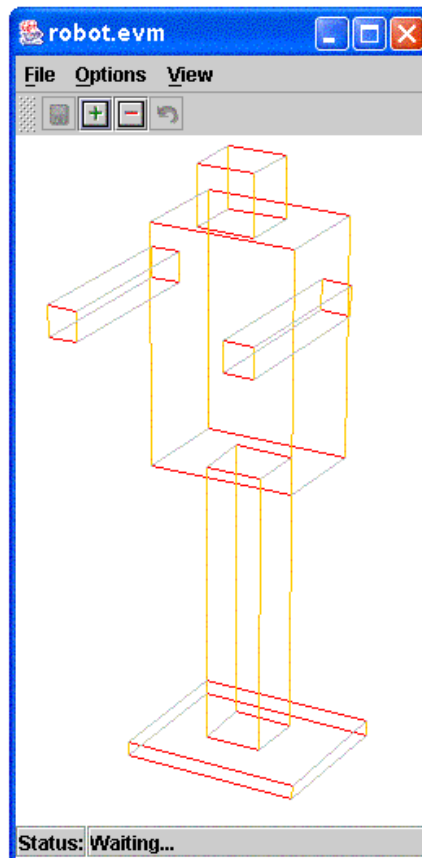


Figura 5-7 Robot ortogonal en modelo de alambres.

5.4 Operaciones booleanas regularizadas.

La aplicación principal cuenta con un panel de control para las operaciones booleanas entre objetos evm (véase Figura 5-1). Dicho panel consiste básicamente de cuatro elementos, tres de los cuales son listas de selección y el último de ellos un botón. Las listas de selección etiquetas con *Object 1* y *Object 2* mantienen un registro de los

objetos actualmente activos en el sistema, esto es, mantienen una lista de los nombres de los objetos evm actualmente en uso. La lista de selección *Boolean operator* permite seleccionar el operador booleano a ser aplicado a los objetos seleccionados en las listas referentes a *Object 1* y *Object 2* respectivamente, mientras que el botón *Perform* lleva a cabo la operación booleana especificada entre los dos objetos previamente seleccionados.

La Figura 5-7 muestra un sencillo robot en su modelo de alambres compuesto por elementos ortogonales. Dichos elementos fueron construidos utilizando el editor de pseudo poliedros ortogonales creando extrusiones simples y combinándolos con operaciones booleanas para obtener la definición del robot.

Es importante mencionar que así como para cada objeto nuevo abierto en el sistema se crea una nueva ventana de visualización que lo administra, el efectuar una operación booleana entre un par de objetos tiene el mismo efecto, esto es, se crea una nueva ventana como resultado de la operación.

El resultado de la operación booleana dependerá de los objetos seleccionados. El modelo de vértices extremos soporta las operaciones booleanas regularizadas de **unión**, **intersección**, **diferencia** y **o exclusivo**.

5.5 División de sólidos.

Cada uno de los objetos evm contenidos dentro de la ventana de visualización puede ser dividido por un plano perpendicular a A, B o C².

Dado que la partición con planos generales derivará en un objeto no ortogonal, sólo es posible realizar particiones con planos ortogonales por lo que el sistema interrogará al usuario sobre la coordenada del plano de partición, acto seguido, la operación es llevada a cabo tal y como se presenta en la Figura 5-8.

² Recuérdese que el modelo EVM trabaja con un tipo de estructura *ABC-sorted* la cual puede tener seis tipos de ordenación XYZ, XZY, YXZ, YZX, ZXY, ZYX, aquí se ha conservado el plano de división perpendicular a A, B o C para mantener la coherencia con *ABC-sorted*.

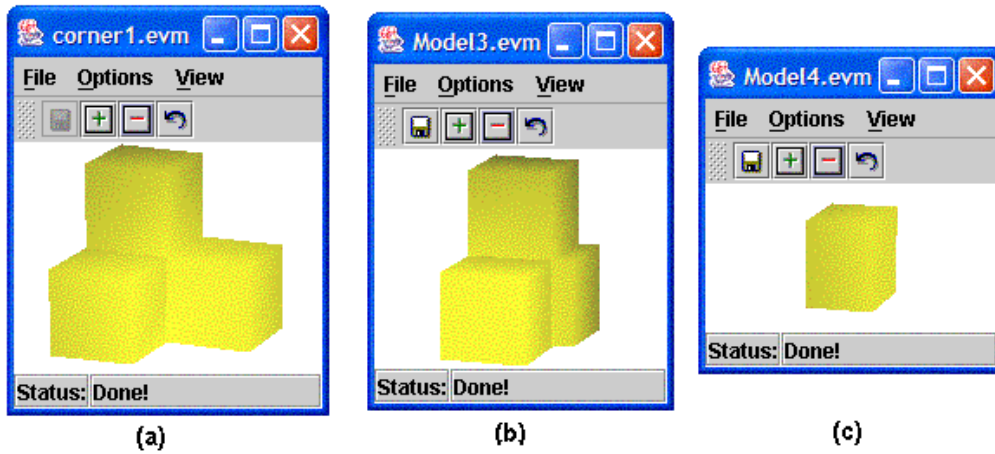


Figura 5-8 Ejemplo de partición de sólidos.

Obsérvese que una operación de este tipo derivará en dos objetos (y por lo tanto en dos ventanas de visualización), siendo cada uno de ellos el resultado respectivo de la partición seleccionada.

5.6 Clasificación de elementos geométricos.

La clasificación de elementos geométricos se refiere a la capacidad de determinar si un elemento geométrico determinado como un punto o un segmento de línea está dentro, fuera o sobre el objeto evm contra el cual se está comparando.

La Figura 5-9 muestra la clasificación de los segmentos de línea que componen al segmento de línea introducido por el usuario. El segmento de línea introducido es en base a los puntos extremos del segmento de recta. Con estos datos, el algoritmo determina los segmentos de línea que están fuera, dentro y sobre el pseudo poliedro ortogonal en cuestión y actúa en correspondencia. En la misma figura se puede observar una representación visual en diferentes colores de los segmentos de línea que están fuera, dentro, y sobre el objeto.

Para la clasificación de puntos el esquema es el mismo, sólo que aquí el sistema responde con una indicación acerca de si el punto dado está dentro, sobre o fuera del objeto EVM en cuestión.

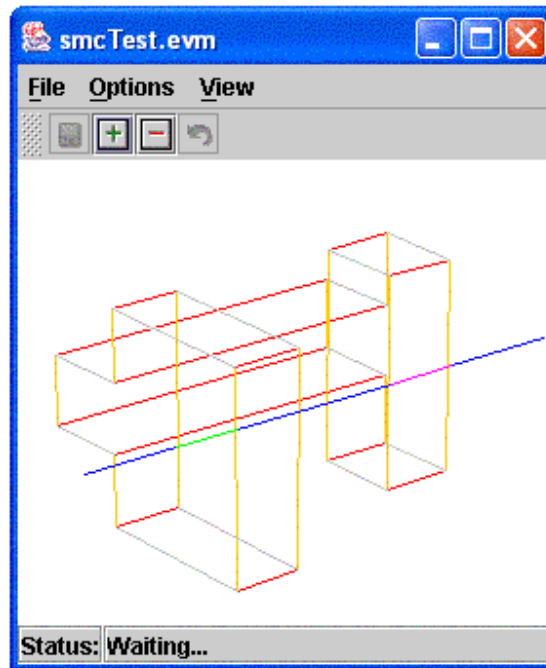


Figura 5-9 Ejemplo de clasificación de un segmento de línea.

5.7 Detección de interferencia.

La detección de interferencias está íntimamente relacionada con la clasificación de elementos geométricos, ya que mientras en esta última interesa saber por ejemplo la situación de un punto respecto a un objeto, en la detección de interferencia sólo interesa saber si un elemento geométrico determinado interfiere o no con el objeto EVM contra el cual se compara.

Si P es un objeto representado en EVM, es posible determinar si un punto dado interfiere o no con P , lo mismo sucede para el caso de un segmento de recta y un plano. El segmento de recta estará también determinado por dos puntos diferentes, mientras que el plano lo estará por tres puntos en el espacio que no sean colineales.

La respuesta del sistema hacia estos aspectos es de carácter indicativo, ya que los algoritmos son de naturaleza booleana, por lo que la respuesta sólo será afirmativa o

negativa dependiendo de si el elemento de interés interfiere o no con el objeto EVM en cuestión.

5.8 Medidas.

Dependiendo del objeto evm que esté siendo procesado, es posible determinar algunas de sus características físicas relacionadas con las medidas del mismo. Estas características son el perímetro, el área, la superficie y el volumen.

En función del objeto contenido dentro de la ventana de visualización, es posible calcular algunas de las medidas antes mencionadas. El resultado es presentado en un cuadro de diálogo con la información que fue requerida, ya que los algoritmos son de carácter numérico.

5.9 Transformación entre modelos.

Dentro de los diferentes esquemas de representación de sólidos, es imposible decir que uno es completamente mejor que otro, cada uno de ellos tiene sus ventajas y sus desventajas, de ahí la importancia de integrar mecanismos de conversión entre los esquemas de representación más usuales.

Dentro del sistema implementado, se encuentra con la conversión del modelo EVM a su representación de fronteras (*B-Rep*), esta implementación construye una estructura jerárquica de contornos, en donde los contornos que representan caras del poliedro estarán determinados por una secuencia de vértices en el sentido contrario a las manecillas del reloj, mientras que los hoyos estarán determinados por una secuencia de vértices en el orden del sentido de las manecillas del reloj. La información presentada se muestra con sangría, con la finalidad de visualizar de manera más fácil cuales son los contornos dentro de otros contornos.

La representación interna de esta estructura, es la de un árbol, en donde un nodo determinado representa un contorno del objeto. Este nodo tiene apuntadores hacia sus posibles hermanos y hacia sus posibles hijos. Un nodo hermano, representa un contorno al

mismo nivel que el de su hermano, mientras que un hijo, representa un contorno perteneciente a un hoyo. Debe tomarse en cuenta que un hoyo dentro de otro hoyo constituye en realidad una cara. La estructura de árbol implementada mantiene esta jerarquía y su recorrido puede hacerse según convenga.

5.10 Notas del capítulo.

Este capítulo ha presentado de manera general algunos de los resultados más importantes. La mayoría de ellos han sido particularmente seleccionados para ilustrar las características más sobresalientes en cada una de las secciones en las que se presentaron. Así mismo, el capítulo estuvo ampliamente ilustrado con material que sirviera al lector para la mejor comprensión de lo que se detallaba.

Cabe hacer mención de que no se intentó hacer un análisis exhaustivo ni una ilustración completa de todas las características del software implementado, simplemente se presentó un esbozo de algunos de los resultados generados por el sistema. Sin embargo, las características mostradas cubren los aspectos planteados en los objetivos del presente trabajo de tesis, ya que se ha obtenido un software que implementa las características del modelo de vértices extremos, que integra aspectos importantes en términos de interacción, y que es resultado de un modelado orientado a objetos.
