

Capítulo 5: Resultados y Conclusiones

5.1. Introducción

Durante en desarrollo e implementación de este proyecto, se encontraron varias limitantes que originalmente no fueron consideradas, haciendo que la estrategia tomada inicialmente para su implementación tomara un nuevo rumbo inesperado, obteniendo como resultado que la complejidad y ventajas originalmente planeadas para este modelo mermaran de forma importante.

A continuación se presentan algunos de los resultados obtenidos y limitantes, perfectibles, del modelo implementado y el posible trabajo a futuro con miras a mejorar los errores de diseño e implementación del modelo.

5.2. Resultados obtenidos

La elección del modelo BSP-OctTree como modelo básico de descripción geométrica fue, sin lugar a dudas, un gran acierto. El modelo presenta grandes ventajas en lo que a diseño se refiere, y la facilidad con la que se pueden lograr las operaciones booleanas entre objetos descritos de esta manera es un gran punto a su favor.

Sin embargo, cuando se intentó hacer uso de esta gran ventaja al momento de implementar el mecanismo de animación surgió una limitante no considerada en un principio, el cual puso al margen del proyecto esta opción. Esto es, que para la descripción de un hiperplano en 2D, para su posterior operación con el cuerpo descrito en el otro árbol, originalmente se planteó la simulación del plano mediante una “caja” infinitamente delgada sobre el plano de corte. Más, a la hora de tratar de describir esta caja infinitamente delgada mediante el modelo BSP-OctTree, resultó imposible de

lograr. Esto se debió a que, al ser una caja infinitamente delgada, los planos paralelos al hiperplano que deseamos representar están infinitesimalmente separados uno del otro. Y siendo que, el criterio básico del modelo BSP-OctTrees para crear un nodo terminal es que los polígonos que intersectan el octante en cuestión deben tener al menos un punto en común, incluso si este punto está fuera del octante dado.

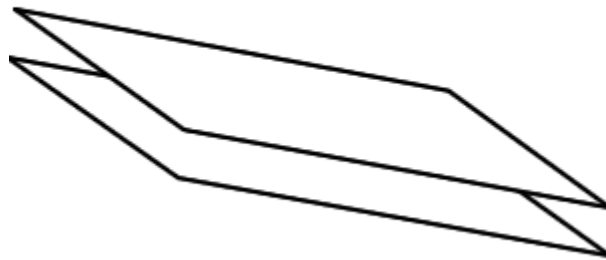


Figura 5.1: Parte intermedia de una caja infinitamente delgada dentro de un octante.

Por tal motivo, se pensó una forma alternativa para simular el plano, sin la ventaja de poder usar propiamente las operaciones booleanas de los BSP-OctTrees. La solución implementada sujeta a mejoramiento, consiste en la creación de un solo polígono que abarque el universo de la figura, y que sea paralelo al plano que deseamos usar como plano de avance. Éste único polígono sirve de referencia para seccionar los polígonos con los que se va intersectando. El proceso es básicamente el mismo usado por las demás operaciones booleanas, con la diferencia de que al momento de seleccionar, alguna de las secciones del objeto resultantes de la partición, tal y como se hace en las operaciones clásicas, se toma la arista con la cual el plano corta el polígono, y se almacena en una lista de aristas, las cuales posteriormente se usan para formar el polígono que representa el corte en ese momento. Sin embargo, debido a la partición octal de la figura, en ocasiones hacían falta aristas que conectaran la frontera

correctamente. Para tratar de subsanar esta anomalía, se hizo un método, el cual conectaba los lados buscando el punto más cercano al extremo de la arista no conectada, completando así el polígono a mostrar en la pantalla de animación.

Esta corrección trajo consigo una serie de defectos que limitó aún más el alcance de la implementación. Uno de ellos, es más, la principal limitación, impuso al proyecto una limitada cantidad de rutas a elegir para el recorrido de la animación; ya que se restringió únicamente a recorridos sobre ejes principales y sobre ejes de 45° que pasen sobre el origen. Este problema se atacó de raíz, sin embargo los resultados no fueron satisfactorios, dando incluso vistas erróneas en recorridos ya probados. Por lo que queda susceptible para corrección en futuros trabajos.

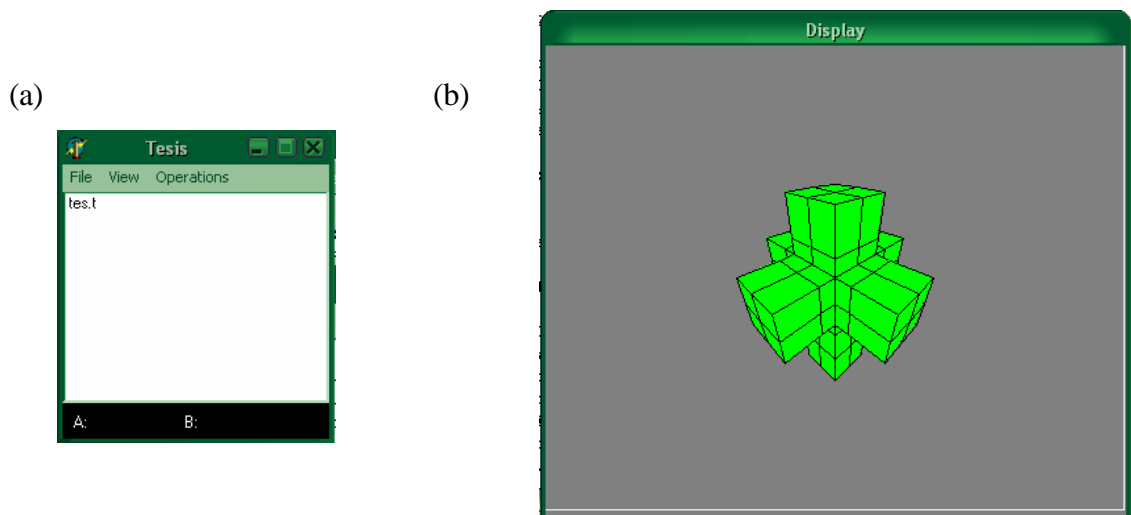


Figura 5.2: (a) Interfaz principal del sistema. (b) Pantalla que despliega el objeto mediante el modelo BSP-OctTree

Más, finalmente se logró una versión capaz de hacer la animación completa sin fallos en los principales recorridos. De igual forma, y únicamente como ayuda visual, se creó un recorrido sobre la figura en sí, donde es posible ver cómo es que el plano de recorrido va reduciendo el tamaño de la figura en sí. Sin embargo, este recorrido no

logró corregirse por completo para figuras más complejas, mas con un poco más de revisión a detalle de estos procedimientos se podrá lograr sin mayor problema.

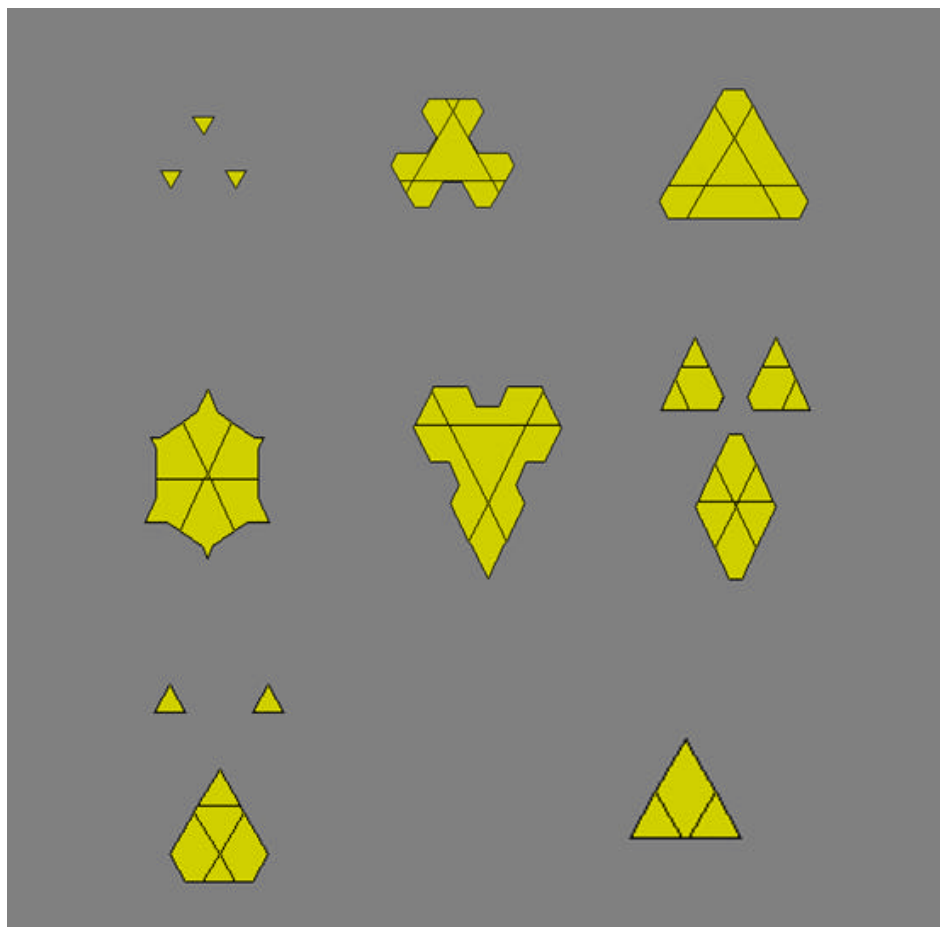


Figura 5.2: Secuencia de cuadros del recorrido del tesseracto

5.3.Trabajo a futuro

Es claro que debido a que los objetivos originales de este trabajo no fueron alcanzados, el trabajo a futuro es mucho y muy variado. Primero que nada, estaría la revisión detallada del diseño original del sistema para el rastreo de posibles mejoras al modelo propuesto para lograr la animación, y su posible generalización para lograr la multidimensionalidad anhelada desde un principio. Dentro de la línea trazada por la propuesta implementada en este trabajo, queda abierta la posibilidad de encontrar un método mejor y más eficiente para corregir o encontrar las líneas faltantes en la creación de la vista del recorrido. Y de igual forma, tratar de hacerlo extensible para la n-

dimensionalidad. Y para el caso particular de la animación en 2D y el recorrido auxiliar en 3D, también es posible buscar una forma más idónea para ver el recorrido actual.

También queda pendiente la extensión del modelo BSP-OctTree para lograr un soporte multidimensional. Esto es, desarrollar los mecanismos propicios para que la estructura soporte más dimensiones y hacer extensibles los algoritmos de operaciones booleanas a este nuevo modelo. Por ejemplo, para el caso de una descripción de un objeto en 4D, será necesario, ya no usar un Oct-Tree, si no un Hexa-Tree (16 nodos por nivel) con lo cual lograríamos replicar la descripción espacial que logran los Oct-Trees en el espacio 3D, en cuatro dimensiones. Esta extensión debe llevar un estudio minucioso sobre las modificaciones que se tendrán que hacer a los algoritmos de creación, operación y visualización. Ya que es muy probable, que estos tengan que ser modificados; principalmente la visualización, ya que se tendría que hacer una proyección a 3D y luego sobre ésta hacer nuevamente una proyección 2D para poder ser visualizada en la pantalla.