UNIDAD 4

4 Requerimientos de Memoria y Tiempo

Para hacer el análisis de Tiempo y memoria se utilizará la siguiente notación:

- A(Pi) Área de la superficie de una primitiva Pi.
- N(Pi) Número de Octantes requeridos para construir una primitiva Pi.
- A(OF) Área de la superficie del objeto final.
- N(OF) Número de Octantes requeridos para construir el Objeto final
- K Número de primitivas que comprenden la definición del Árbol CSG.

La memoria ocupada al construir un objeto mediante árboles octales vía la definición CSG se relaciona con la cantidad de octantes que se utilizan, ya sea para el objeto final, como la cantidad de octantes que se crearon durante el proceso de obtención de dicho objeto.

4.1 Requerimientos de memoria

4.1.1 Memoria requerida por el método clásico

Como ya se sabe, para crear un objeto el método clásico necesita construir los árboles octales de todas las primitivas que forman al objeto, así como los árboles octales que resulten de las operaciones Booleanas, entre los cuales se encuentra el objeto final, de estos árboles creados el único que sirve cuando ya se obtuvo el objeto Final es el que se encuentra en la raíz del árbol CSG porque es el que representa al objeto final, los otros árboles creados, entonces, ya no se necesitan. Además muchos de los octantes que se crearon no forman parte del objeto final, ya que son octantes que se encuentran dentro de alguna primitiva pero que se encuentra afuera del objeto final, estos octantes serán ignorados, por ejemplo si se tiene una unión y uno de los operandos es negro y el otro es gris, entonces todo el subárbol del nodo gris, no se va a ocupar, esto mismo es para la intersección cuando se tiene un operando blanco y uno gris, todo el subárbol del nodo gris es omitido. También muchas veces es posible que se creen más de una los octantes, haciendo uso de memoria

Un objeto está formado por k primitivas y cada una de estas necesita para crear su árbol octal N número de octantes, como cada Primitiva tiene un número de octantes diferente, se considerará que N(Pi) representa el número de octantes que contiene una primitiva Pi. Entonces el número de octantes que se utilizan para construir objeto final son:

$$\text{N\_Oct\_met\_clásico} = N(P_1) + N(P_2) + \ldots + N(P_k) \text{ octantes} \qquad \text{Ec. 4.1}$$

$$\text{N\_Oct\_met\_clásico} = \sum_{i=1}^{K} N(Pi) \text{ octantes} \qquad \text{Ec. 4.2}$$

4.1.2 Memoria requerida por el método Top Down

En el método Top Down solo se crea el árbol octal del objeto final, de esta forma no hay octantes innecesarios, El número de octantes que forma a un objeto final se va a representar por medio de N_Oct_met_TopDown, y es igual a N(OF). O representa al número de octantes que forman parte del árbol octal final.

$$N\_Oct\_met\_TopDown = N(OF) \qquad \text{Ec. 4.3}$$

4.1.3 Comparación de memoria

La ecuación 4.4 hace una comparación de la cantidad de memoria que ocupa cada método.

$$\frac{N\_Oct\_met\_TopDown}{N\_Oct\_met\_Clásico} = \frac{N(OF)}{\sum_{i=1}^{K} N(Pi)} \qquad \text{Ec. 4.4}$$

El número de octantes que se requieren para forman a un objeto o primitiva, es determinado por su superficie, por ejemplo si se tiene la unión de dos objetos disjuntos y la diferencia de estos mismo objetos el número de octantes es mayor en la unión que en la diferencia, por que la unión tendrá una superficie mayor, mientras que en la diferencia en lugar de crecer el número de octantes disminuyó. De hecho, el número de octantes es proporcional al área de la superficie de cada objeto, y entonces se dice que N(OF) es proporcional a A(OF) y N(P) es proporcional a A(P)

Si se considera a μ como la razón de mejoramiento de memoria entre el método Top Down y el método Clásico:

$$\mu = \frac{N\_Oct\_met\_TopDown}{N\_Oct\_met\_Clásico} = \frac{O(OF)}{\sum_{i=1}^{K} O(Pi) \text{ octantes}} = \frac{A(OF)}{\sum_{i=1}^{K} A(Pi) \text{ octantes}} < 1 \qquad \text{Ec. 4.5}$$

Por lo tanto entre más complejos sean los objetos y tengan más primitivas la razón μ disminuye debido a que en el método clásico se crean más árboles octales, con lo que se comprueba que se ocupa menos memoria en el método Top Down.

4.2 Análisis de Tiempo

Las siguientes definiciones serán utilizadas en las formulas del análisis de tiempo.

- Tiempo_examinacion: Tiempo promedio requerido para examinar la relación entre un octante y una primitiva.
- Tiempo_crear: Tiempo requerido para almacenar un octante .
- Tiempo_Operación_Booleana: Tiempo promedio requerido para ejecutar una operación Booleana de un octante.

4.2.1 Análisis de Tiempo usado en el método Clásico

Si se supone que un objeto está formado por k primitivas y, por lo tanto al menos k-1 operaciones Booleanas (al menos, por que alguna primitiva puede tener operación de complemento) el tiempo que se tardará en crear un objeto final es el tiempo en que se tarda en hacer k examinaciones para determinar la relación entre un octante y las primitiva, y por ultimo el tiempo que se tarda en realizar hasta k creaciones del mismo octante, entonces el tiempo que se tarda el método Clásico en obtener el árbol octal del objeto final es:

$$Tmet\_clásico = N(P)*[k * Tiempo\_Prueba + (k - 1) * Tiempo\_Operación\_Boleana + k*Tiempo\_crear] \quad \text{Ec. 4-6}$$

El tiempo que se tarda en hacerse las operación Booleana es menor al que se tarda en hacer las pruebas y la creación de octantes, entonces se puede trabajar únicamente k en lugar de k-1 por facilidad. De este modo al cambiar k-1 por k la ecuación 4.6 queda de la siguiente forma:

$$Tmet\_clásico = N(P)*[(k) * Tiempo\_Prueba + (k) * Tiempo\_Operación\_Boleana + (k)*Tiempo\_crear] \quad \text{Ec. 4.7}$$

Factorizando k, la ecuación queda como:

$$Tmet\_clásico = N(P)*k*[Tiempo\_prueba + Tiempo\_Octante\_en\_Operación\_Booleana + Tiempo\_crear] \quad \text{Ec. 4.8}$$

Como se dijo en la demostración de la memoria que el número de octantes es proporcional al área del objeto, entonces N(P) es proporcional a A(P) y la ecuación 4.9 queda:

$$Tmet\_clásico \; \alpha \; A(P)*k*[Tiempo\_prueba + Tiempo\_Octante\_en\_Operación\_Booleana] + Tiempo\_crear] \quad \text{Ec. 4.9}$$

Donde α significa proporcional

4.2.2 Análisis de Tiempo usado en el método Top Down

A diferencia del método clásico, en el método Top Down solo se va a crear cada octante una vez, pero, el tiempo que se tarda en examinar si un octante esta dentro del objeto y el que se tarda en realizar las operaciones Booleanas entre los octantes de las primitivas es el mismo. Entonces el tiempo que se tarda en crear un objeto final el método Top Down esta dado en la siguiente ecuación.

Tmet_TopDown = N(OF) * [k * (Tiempo_prueba + (k-1) * Tiempo_Operacion_Booleana) + Tiempo_crear] Ec. 4.10

Si se hace una aproximación de la Ecuación 4.10 al tomar en cuenta k * Tiempo_Octante_en_Operación_Booleana en lugar de (k-1) * Tiempo_Octante_en_Operación_Booleana. La ecuación se transforma en:

Tmet_TopDown = N(OF) * [k * ((Tiempo_prueba) + Tiempo_Operacion_Booleana) + Tiempo_crear] Ec. 4.11

Dado que N(P) es proporcional a A(P) y la ecuación 4.11 queda:

Tmet_TopDown α A(OF) * [k * ((Tiempo_prueba) + Tiempo_Operacion_Booleana) + Tiempo_crear] Ec 4.12

Donde α significa proporcional

4.2.3 Comparación de tiempo

La razón entre los dos métodos con respecto a l tiempo esta dado por:

$$\tau = \frac{Tmet\_TopDown}{Tmet\_clásico} = \frac{A(OF)}{A(P)} * \left[\frac{k*(TiempoPrueba + Tiempo\_Operación\_Boleana\_Top\_Down) + Tiempo\_crear}{k*(Tiempo\_prueba + Tiempo\_Operación\_Booleana + Tiempo\_crear)} \right]$$

Ec. 4.13

ó, dividiendo el enumerador y el denominador por k

$$\tau = \frac{Tmet\_TopDown}{Tmet\_clásico} = \frac{A(OF)}{A(P)} * \left[\frac{Tiempo\_prueba + Tiempo\_Operación\_Boleana\_TopDown + Tiempo\_crear/k]}{Tiempo\_prueba + Tiempo\_Octante\_en\_ Operación\_Booleana + Tiempo\_crear]} \right]$$

Ec. 4.14

Se puede observar en la ecuación 4.14 que la expresión entre corchetes es siempre menor que 1. Mientras que como A(OF) representa al área del objeto final no necesariamente es más pequeña que el área de la primitiva A(P). Sin embargo, la razón α esta siempre limitada por $A(OF)/A(P) < K$.

4 Conclusiones

El método ofrece construir un solo Árbol Octal, desde la definición CSG, por lo que no es necesario crear los Árboles Octales de las primitivas que conforman al objeto, de esta forma proporciona un ahorro de memoria, una forma de no tener ahorro de memoria es cuando se construya una primitiva únicamente, pero esto rara vez ocurre y aún cuando esto suceda, el número de octantes creados será el mismo que en el método clásico y nunca va a ser superior.

Entonces siempre se asegura que el número de octantes en el método Top Down es menor, ó en el peor de los casos igual que en el método Clásico.

Aunque el método Top Down no crea los árboles octales innecesarios, no siempre asegura ser más rápido, esto depende del Área de la superficie del objeto final, por ejemplo, si se trabaja con primitivas disjuntas y uniones entre éstas, el tiempo puede ser más grande, esto se debe a que se crean más octantes de los que cada primitiva tienen, no obstante, la diferencia de tiempo en este caso va a estar limitada por el número de primitivas.

Generalmente cuando se construye un objeto es poco probable que contenga únicamente primitivas disjuntas y uniones, hay mucho más posibilidades que se construya un objeto con un conjunto de primitivas no disjuntas y que se apliquen a ésta las diferentes operaciones Booleanas: Intersección, Unión, Diferencia y Complemento.