

3. La Multirresolución

Después de modelar los objetos en el espacio, es necesario un proceso de optimización para poder realizar diversas funciones gráficas, como el desplegado realístico. Existen técnicas de *Simplificación Poligonal* para producir *Niveles de Detalle* (Levels of Detail, LODs), ya que no todas las formas complejas que interesa modelar pueden describirse mediante las técnicas tradicionales de modelado geométrico, como por ejemplo los fractales.

En ciertas aplicaciones donde se requiere una respuesta rápida, como en simuladores, realidad virtual, donde es importante ver que tanto detalle es necesario en cierto momento. Es posible que un objeto visto de lejos cubra una parte tan pequeña de la imagen, que sea innecesario dibujar una versión con todo detalle, cuando una versión simplificada del objeto sea suficiente, ahorrando así tiempo. Una forma común de hacer esto es construir manualmente varias versiones del objeto con distintos niveles de detalle y cambiar de versión cuando sea necesario. Otra manera más novedosa es una técnica de modelado que cambie la complejidad del modelo automática y suavemente, se le conoce como técnica de multirresolución.

Un objeto puede ser manipulado a cualquier nivel de detalle. La idea básica es descomponer el modelo en dos partes: 1.- Parte de menor resolución; 2.- Parte de detalle. Un poliedro se descompone de una parte de menor resolución que será un conjunto de promedios ponderados de los vértices (como un filtro pasa bajos), todo detalle fino es eliminado y solamente se guardan algunos datos representativos.

Este proceso es altamente necesario para escenarios complejos de gran cantidad de información gráfica, donde los requerimientos de almacenamiento y la computadora son muy altos o exceden su capacidad, incluyen la necesidad de mayor velocidad. Estas escenas por su complejidad, requieren un gran número de polígonos, aunque se considere un pequeño ángulo de visión.



Figura 3.1

3.1 Los niveles de detalle

Los niveles de detalle son las simplificaciones, que se hacen a partir de un modelo complejo, en la búsqueda de una versión más simple del modelo, ignorando

algunas cualidades gráficas no distinguibles con cierta degradación visual. Estas simplificaciones pueden ser usadas cuando el objeto aparece pequeño, cuando se está moviendo y cuando está en una visión lejana del observador.

El objetivo de un LOD es la simplificación de la imagen sin crear una degradación visual, existen varios tipos de LOD, la idea principal es remover las primitivas de la malla principal, obteniendo una malla menos compleja pero con las mismas cualidades gráficas, encontrar un balance entre la belleza y nitidez de la imagen contra la velocidad de desplgado.

Los algoritmos deben tener una heurística adecuada para poder mantener este balance y poder elegir cuales primitivas simplificar, según el criterio que utilicen es el resultado del algoritmo.

3.1.1 Niveles continuos

Cuando los LODs generados solamente varía por uno o dos polígonos, los cambios son más suaves y se les llama continuos, debido a que la lista de polígonos a modificar no es muy notable, por lo tanto las variaciones son poco percibidas por el observador.¹

3.2 La preservación de la forma y topología

Cuando se hace la simplificación, una de las propiedades más importantes del objeto debe ser mantenido, la forma, ya que se deben mantener esas propiedades que hacen identificable al objeto. La formar contenida en el objeto consiste en varias propiedades que los algoritmos deben identificar y trabaja acorde a ellas; como son:

Basándose en el área planar o planos contenidos, se deben calcular las adyacencias de polígonos y mantener esos planos. Los bordes pronunciados y los bordes apuntados. Podemos desaparecer aquellos bordes adyacentes que son colineales.²

La topología del objeto es su estructura espacial, que puede ser simplificada dentro del proceso, sin embargo sus cambios son muy notorios, por lo tanto, no es muy útil, por ejemplo si se tiene un hoyo o espacio interior al simplificar puede generar partes no conectadas, lo cual es muy notorio.³

¹ Hoffman Christoph M., *Solid Modeling: an Introduction*, p. 93.

² Samet, Hanan, *Applications of Spatial Data Structures, Computer Graphics, Image Processing and GIS*, p. 189.

³ Thalmann D., Magnenat-Thalmann N., *New Trends in Computer Graphics*, p. 167.

Al elegir que debemos preservar, el algoritmo debe ser cuidadoso y mantener las cualidades de identificación del objeto.

3.3 Simplificación poligonal de operadores

Podemos utilizar ciertos operadores matemáticos para procesar las simplificaciones necesarias, de acuerdo a ciertas características del sólido⁴:

- **Normalización:** remoción de caras degeneradas o bordes y cualquier primitiva definida múltiples veces.
- **Simplificación de Vértices:** combinación de todos los puntos incluidos dentro de un volumen (cualquiera una esfera o una celda de rejilla). Entonces, los puntos y las caras cercanas son combinadas.
- **Simplificación de Borde:** remoción de todos los bordes más pequeños que algún umbral.
- **Simplificación Basada en Angulo:** remoción de bordes los cuales forman un ángulo cerrado. Inversamente, los bordes que están alineados son combinados.
- **Simplificación del Tamaño de la Cara:** remoción de todas las caras las cuales tienen un área más pequeña que algún umbral. Los hoyos pudieran tener que ser llenados.

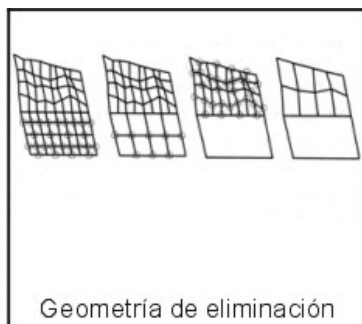


Figura 3.3

- **Simplificación de la Normal de la Cara:** combinación de todas las caras adyacentes con normales casi paralelas.

3.4 Clasificación de los algoritmos de simplificación poligonal

Los algoritmos de simplificación se encuentran clasificados en tres grupos, según⁵:

Algoritmos de geometría de eliminación

Algoritmos de subdivisión adaptativa

Algoritmos de muestreo

⁴ Astheimer, P. and Pöche, M.-L. *Level-of-detail generation and its applications in virtual reality*. 1994, p. 299-312.

⁵ Erickson, C. *Polygonal simplification: an overview*, p. 229-238.

3.4.1 Algoritmos de geometría de eliminación

Esta técnica es de las más populares, y mediante una análisis va haciendo una selección de primitivas que van a ser removidas, aquellas primitivas que se encuentran dentro de una política de heurística bien establecida, por ejemplo se puede usar un

criterio de planaridad, esto quiere decir, que se eliminan los vértices que se encuentran cerca de las regiones planares, como se ve en el ejemplo siguiente⁶.

3.4.2 Algoritmos de subdivisión adaptativa

Primeramente hace un acercamiento casi fidedigno al modelo cuando se le añade más detalles al modelo original, después es subdividido y la posición de cada nuevo vértice es cambiada para tener más cercanía a la superficie original, este proceso no es popular debido al primer paso.

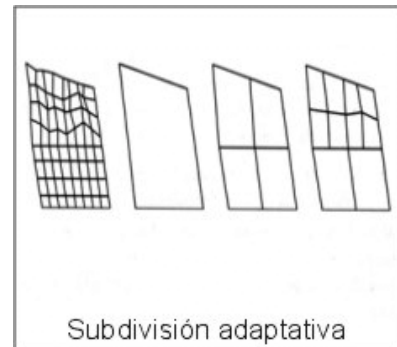


Figura 3.4

3.4.3 Algoritmos de muestreo

En este método se eligen las primitivas que se conservarán mediante un método heurístico; esta selección es pseudoaleatoria, otra manera es muestrear un grupo de primitivas y tomar la mas representativa. Esta técnica ya no es usada, debido a que es difícil la selección de muestra que ayuden a preservar al modelo.

3.5 Algoritmos de simplificación

Ahora mostraremos algunos algoritmos de simplificación:

3.5.1 Análisis de Mallas Arbitrarias de Multiresolución

Este primer algoritmo, consiste en un preprocesador para otro algoritmo que produce una representación de **multirresolución** de una malla, el cual es un geomorfo compacto conteniendo una malla base simple y una serie de coeficientes "wavelet" que son usados para introducir detalles dentro de la malla. De esta representación, una nueva malla puede usar una *subdivisión recursiva* (es decir,

⁶ Erickson, C. 1996 *Polygonal Simplification: an overview*, p. 244-255.

donde cada triángulo es subdividido usando un operador de partición 4 a 1) hasta que la cantidad deseada de detalle es alcanzada. Semejante malla es encodificada dentro de una representación de multirresolución. El algoritmo que se presenta aquí puede ser usado para convertir cualquier malla a una en la cual se tiene la propiedad de subdivisión recursiva.

Este algoritmo es uno de subdivisión adaptativa el cual preserva la topología, pero identifica cualquier forma característica en la malla. El error de aproximación es medido usando la distancia a la malla original. Los **mapas armónicos** son usados en varios pasos para parametrizar una malla 3D dentro de una triangulación planar. El algoritmo tiene cuatro pasos principales:

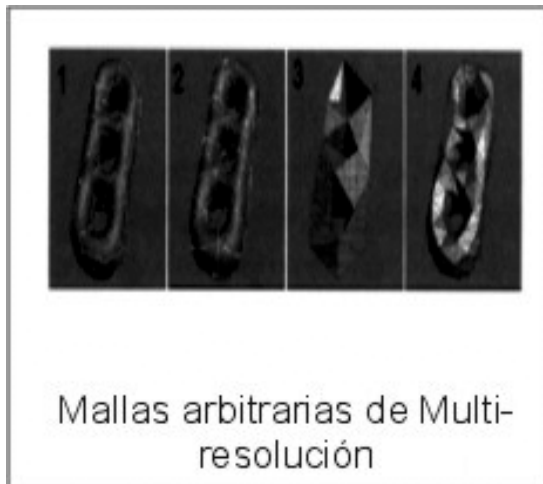


Figura 3.5

1. **Particionamiento:** Un diagrama parecido al de Voronoi es construido sobre la malla original usando un algoritmo de buscador de trayectoria multi-semilla en el gráfico dual de la malla (donde los nodos son las caras de la malla y los arcos representan adyacencias y son pesados usando la distancia entre los centros de caras adyacentes). Este diagrama es entonces triangulado usando un método parecido al de Delamay y los mapas armónico para hacer rectos los bordes.

2. **Parametrización:** el resultado es una malla base que es parametrizada usando un mapa armónico. La parametrización es forzada a ser continua a través de las caras para que el número de coeficientes "wavelet" sea mínimo.
3. **Remuestreo:** la malla base es ahora remuestreada usando un operador de división 4 a 1 hasta que la malla esté en una cierta distancia a la malla original. Cada paso es parametrizado como en el paso 3.
4. **Análisis de multirresolución:** la sucesión resultante de mallas es pasada al algoritmo de análisis de multirresolución para ser encodificada usando "wavelets".

Este algoritmo es atractivo en su formalismo matemático. Produce un amplio rango de simplificación y no obstante detalles que pueden ser añadidos en partes

específicas de la malla. Pero esto es computacionalmente caro también. Además, extrayendo una malla válida desde una representación basada en "wavelet" es caro.⁷

3.5.2 Algoritmo de Mallas Progresivas

El algoritmo de Mallas Progresivas⁸, produce geomorfos y es derivado de un algoritmo más antiguo. Este busca áreas planares y bordes característicos. La simplificación es hecha aplicando un operador **colapso de borde**, donde un colapso de borde produce un nuevo vértice removiendo las dos caras y un vértice. El resultado es una malla de base simplificada y series de **particiones de vértice** las cuales son un inverso de los colapsos de borde y son usadas para introducir detalles dentro de la base de la malla. Esta es llamada **Malla Progresiva** y un amplio número de simplificaciones pueden ser extraídas de este.

La característica mas importante de este algoritmo es que toma en cuenta información tal como color, textura, y discontinuidades normales sobre la superficie de cada malla. Las formas importantes del modelo las cuales son representadas por esta clase de información (y no por simple geometría) son también preservadas. La minimización de una función de energía es usada para guiar la simplificación. Esta función tiene cuatro términos. El primero asegura que la malla simplificada permanece cercana a la original. La segunda favorece triángulos con mejores proporciones. El tercer término desalienta la simplificación de discontinuidades de color y textura. Finalmente, el último término desalienta la simplificación de las discontinuidades topológicas y normales. Los pasos básicos del algoritmo son estos:

1. Ordenar los bordes usando el mínimo costo de simplificación. Este costo es medido usando una variación de la función de energía.
2. Aplicar el operador de colapso de bordes para el borde en la cabeza de la lista y registrar la correspondiente partición del vértice en la estructura de malla progresiva (incluyendo color, textura e información normal)
3. La posición del nuevo vértice es seleccionado entre los dos vértices iniciales y el centro del borde, dependiendo sobre el cual uno es el más cercano a la malla original.
4. Recalcular el costo para los bordes que han sido afectados por el operador y reordenar la lista.

⁷ Hearn Donald, Baker M. Pauline, *Computer Graphics*, p. 67.

⁸ Hoppe, H. 1996. *Progressive meshes*, p.321.

5. Si la lista está vacía o el costo de la siguiente simplificación excede un cierto límite, el algoritmo termina y regresa la malla final progresiva. De otra manera, regresa al paso 2. Figura 3.6⁹

Este algoritmo es relativamente rápido y toma en cuenta el color y la textura, los resultados son usualmente muy buenos. Este es el único que usa información tal como color y textura. Pero como los modelos se mejoran, esta clase de datos serán más y más vitales al usuario. Ellos hacen el objeto más identificable y entonces debería ser preservado. Esta simplificación debería ser hecha usando la característica de la percepción visual humana.

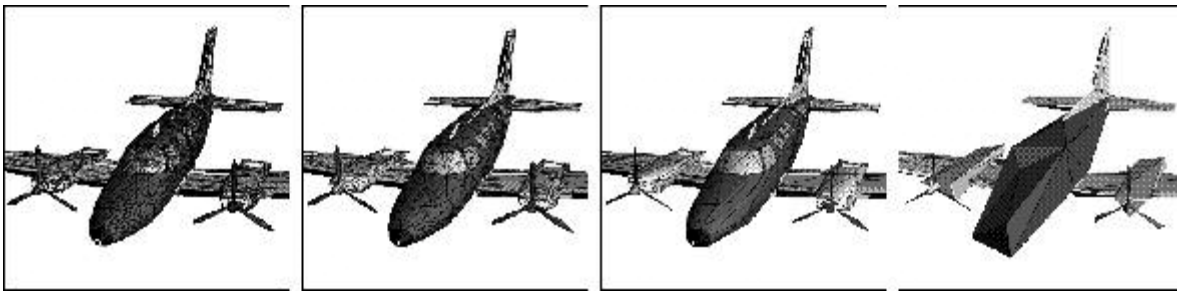


Fig. 3.6 Mallas Progresivas.

3.6 Aproximación del error

Para controlar la simplificación, el error de aproximación debería de ser medido localmente (en cada primitiva). Pero para que el usuario sea capaz de especificar la simplificación, una cota global debería ser puesta para el error. Algunos algoritmos usan una cota para el error local, como el segundo, o una distancia a la malla original como el primero. Otros usan una construcción geométrica para asegurar que la simplificación no exceda un cierto límite. Un uso para la medida del error es determinar si una simplificación puede ser usada ya que sus diferencias con el modelo original pueden no ser notables.

La mayoría de los algoritmos permite al usuario especificar el límite superior para la aproximación del error local. Esto no es muy intuitivo y requiere alguna práctica. Alternativamente, algunos algoritmos permiten al usuario especificar cuantos polígonos deberían de ser dejados en la simplificación.

⁹ Hoppe, H. *Progressive meshes*, p.63.