METABALLS PARA MODELADO 3D

Guillermo Díez

Una característica casi universal en los programas comerciales de animación 3D es el uso de polígonos triangulares para la representación de los objetos que intervienen en la animación.

Este tipo de descripción ayuda en la eficiencia de los algoritmos que tendrán en última instancia que crear las imágenes, y además hace que la representación sea uniforme (mientras todo en nuestro universo sean polígonos, todo lo que habremos de hacer es colgerlos todos y pasar cada uno de ellos por nuestra máquina de dibujar polígonos).

Sin embargo, cuando observamos una animación hecha por ordenador, podemos ver como ciertos objetos no parecen ser poligonales . Aparecen como superficies suaves, y no formados por pequeñas facetas planas, como cabría esperar. Esto no es sino producto de ingeniosos algoritmos que consiguen mostrar una versión suavizada de lo que internamente son triángulos planos.

Los polígonos son capaces de representar de manera exacta objetos que por su naturaleza son poligonales, como un cubo, una mesa sencilla, o las torres de KIO, pero tienen el inconveniente de que se necesitan muchos de ellos para representar superficies complicadas, con variaciones importantes de curvatura, y que por naturaleza no son poligonales. Podemos ver el efecto que este hecho tiene en la fase de modelado, común a toda producción de animación de síntesis.

GUILLERMO DÍEZ *es estudiante de 4°en la ETSIT de Madrid y es responsable de la creación de una nueva primitiva gráfica, basada en campos escalares y splines que generaliza los metaballs, incluida en la última versión del programa Metarreyes.*

Cuando modelamos, creamos la descripción geométrica de los objetos que van a aparecer en escena en la animación. En términos prácticos, esto se traduce en crear la 'malla' de polígonos que representará (de manera aproximada en la mayoría de los casos) el objeto que hemos modelado.

La forma en que esto se hace suele ser tomando un objeto real, y muestreando puntos sobre éste, mediante un dispositivo de posicionamiento 3D. Podemos intuir ya que, aunque existen más formas de proporcionar al ordenador un modelo poligonal de nuestro objeto, estas técnicas hacen que sea más sencillo por ejemplo, modelar un edificio -en el que quizás tan sólo habría que muestrear ocho puntos- que un bosque -con la complicación de sus quizás cientos de árboles, sus miles de hojas, y en cada una de ellas, todos sus precisos detalles.

Los polígonos son capaces de representar de manera exacta objetos que por su naturaleza son poligonales,pero tienen el inconveniente de que se necesitan muchos de ellos para representar superficies complicadas

Históricamente esto ha conllevado el que se prefiera realizar animaciones con objetos de naturaleza artificial, creados por el hombre (mucho más sencillos de modelar), a usar objetos de origen natural, cuya complejidad puede ser tan grande que su modelado sea inviable. En este contexto, los metaballs no pretenden resolver de manera drástica el problema del modelado de objetos naturales, pero constituyen una herramienta tremendamente útil e intuitiva que permite crear de manera sencilla objetos que de otra forma serían complejos de modelar. Para entenderlo comencemos por describir en que consisten.

1.Metaballs.

En el año 82, e independientemente, K.Omura en Japón y J.Blinn en Estados Unidos desarrollaron la idea de los metaballs, y los usaron para visualizar mapas de densidades de electrones en moléculas. Pronto se vio que presentaban propiedades muy interesantes para el modelado de objetos más generales.

De cara al usuario(al modelador), los metaballs son esferas que interactúan con el resto de metaballs, de forma que cuando uno se encuentra alejado del resto, su forma es esférica, pero a medida que se acerca a otros metaballs, va progresivamente fundiendose con aquellos que tiene más cerca.

Quizás lo más ilustrativo que existe a la hora de entender el comportamiento de los metaballs es ver-

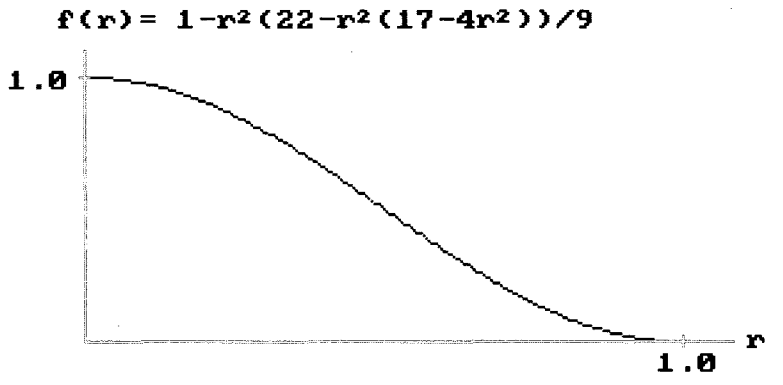

Figura 1.

los en una animación en que aparezcan,como en los efectos especiales de la pelicula Terminator II, o en varias animaciones japonesas que aparecieron en Siggraph (en Estados Unidos) o Imagina (en Monte Carlo).

De cara a la implementación, la forma en que se consigue el efecto de fusion es conceptualmente sencillo, aunque conlleva alguna dificultad a la hora de su representación en pantalla.

Imaginemos por ejemplo dos cargas eléctricas de igual signo, puntuales y muy alejadas una de otra. Imaginemos tambien el potencial eléctrico que crean (que será solo dependiente de la distancia a la carga y decrecerá con ésta), éste será un campo escalar, que toma un solo valor en cada punto del espacio, y que será la suma de los potenciales producidos por cada carga.

Por estar alejadas las cargas, si consideramos una superficie equipotencial, ésta consistirá en dos esferas, una centrada en cada carga. Esto es conveniente porque queremos que cuando dos metaballs estén alejados no interactúen, y tengan forma esférica.

En cambio cuando aproximamos lo suficiente las dos cargas, si volvemos a imaginar el potencial eléctrico que originan, dado que éste se suma en cada punto del espacio, el efecto que obtenemos cuando consideramos una superficie equipotencial es que las dos esferas se funden.

Luego lo que necesitamos conceptualmente para idear metaballs (abstrayendo los detalles irrelevantes en graficos por ordenador) es signar a cada metaball un campo escalar, función del punto del espacio en que nos encontremos, de tipo aditivo (tal que el campo total en cada punto será la suma de los campos producidos por cada uno de los metaballs que consideremos), que además cumpla la propiedad de que el valor del campo de nuestro metaball ha de depender sólo de la distancia al centro de éste, y tal que su valor sea estrictamente decreciente con ésta y se haga nulo a partir de un punto (aunque ese punto sea el infinito).

Generalmente se suele usar como función de la distancia al centro del metaball, la que aparece en la figura 1, o una proporcional, y como superficie equipotencial (equiescalar en este caso) la que surge de igualar el campo total a 0.5.

La razón por la que se escoge casi siempre esta función y no otra es que se han apreciado dos ventajas muy útiles en la práctica.

La primera es que dado que el campo se anula a una distancia finita del centro del metaball, en el resto del espacio no tendremos que preocuparnos de calcular la aportación de ese metaball al campo total y eso hace que los algoritmos que usemos para la visualización sean más eficientes y que su complejidad sea menor.

La segunda de las ventajas es que cumple la condición de que su derivada es nula en el punto en que el campo se hace cero, y esto lo que implica es que cuando nuestros metaballs se fusionen, lo harán de manera suave (en el sentido matemático de la palabra).

Quizás la ventaja más importante que aporta el modelado con metaballs es la intuitividad.

2.Poligonalización de metaballs.

Antes de entrar en evaluar la aportación que los metaballs hacen en el campo del modelado 3D, tan sólo mencionaré de pasada algunas de las técnicas que se usan en la práctica para su visualización.
La idea fundamental en la quese apoyan todos los algoritmos que pretenden dibujar metaballs es que si evaluamos el campo que produce una distribución de metaballs en un punto del espacio, y éste resulta ser mayor que el umbral que hemos establecido (0.5 en el ejemplo anterior), el punto estará en el interior del volumen que limitan los metaballs. De la misma manera, si el campo es menor que el umbral, estaremos en algún punto exterior a ese volumen.

Generalmente, lo que resulta más conveniente, es crear una poligonalización de los metaballs. Esto consiste, a grandes rasgos, en crear una malla de polígonos, que aproximen la superficie de los metaballs.

El interés de este modo de proceder, es que, una vez que tenemos la malla de polígonos creada, cualquier programa de 'rendering' (creación de imágenes 3D) orientado a polígonos podrá ocuparse del resto.

En cuanto a los algoritmos que se usan para la poligonalización de metaballs, sólo nombrar los más importantes: 'marching cubes' (algoritmo que se usa principalmente en visualización científica y medica [Foley95] pág. 1035), 'compact cubes' [Moore92], y un tercero, que aparece descrito en [Foley95] pág. 1048 (de forma no muy clara, pero que es sin duda tan importante como los otros dos). Estas son las referencias indispensables si se prentende hacer una implementación de un poligonalizador de metaballs.

3.Conclusión.

Quizás la ventaja más importante que aporta el modelado con metaballs es la intuitividad.

La manera en que construimos objetos con ésta técnica es constructiva. Comenzamos por un conjunto de primitivas metaball y vamos añadiendo más según queramos que continúe nuestro objeto. Si el resultado no nos convence en alguna región, podemos quitar, poner o mover los metaballs que no encajan bien.

Esta no es la única técnica que posee la virtud de ser intuitiva y fácil de usar por usuarios no experimentados, pero es quizás la más representativa.

Los tipos de objetos que podemos representar usando metaballs son muchos, en teoría. Incluso existen métodos de aproximar cualquier objeto con una distribución de metaballs [Shig91]. Sin embargo, aquellos para los que parecen pensados son los fluidos y los modelos orgánicos, los cuerpos humanos, animales etc...

En estos casos, la propiedad que se puede aprovechar es que con un número no muy grande de primitivas (como mucho del orden de cientos) no podemos representar más que un subconjunto muy reducido de objetos, pero tenemos gran control sobre aquello que modelamos.

Este es un claro ejemplo de una de las casi constantes situaciones en gráficos por ordenador, que es el conflicto entre complejidad y control. En este contexto, el modelado usando metaballs se encuentra en una posición realmente buena, porque nos permite crear objetos de muy razonable complejidad, y nos ofrece un gran control.

Aunque con un número reducido de primitivas no podemos representar más que un subconjunto muy reducido de objetos, la ventaja radica en que tenemos gran control sobre aquello que modelamos.

4.Direcciones futuras.

Parece que la característica que puede convenir mantener en futuras técnicas, frente a los metaballs, es la intuitividad de su forma de modelado.

En esta línea existe más de una generalización de éstos. Existen metaballs en los que su forma original es un elipsoide (esto es fácil de conseguir sin más que hacer que la distancia al centro se defina de manera que sea constante en un elipsoide). Tambien hay metaballs que en su forma original son una supercuádrica [Barr92], o los llamados supermetaballs, metaballs elipsoidales en los que el campo no sólo depende de la distancia sino de la dirección.

Una de estas generalizaciones, probablemente la más útil, fue concebida por G.Wyvill, uno de los investigadores de mayor prestigio en el modelado con campos escalares. Esta consiste en considerar, no un punto que crea un campo, sino una curva en el espacio (un spline) que crea un campo escalar de forma que, cuando dicha curva no esté en interacción con ninguna otra, su forma sea aquella que se obtiene de deformar un cilindro hasta que su eje es la curva que hemos creado. Asimismo, el radio de dicho 'cilindro' se podría variar según avanzamos en la curva, consiguiendo así formas más complejas.

Wyvill sin embargo no consiguió resolver el problema del cálculo del campo para dicha primitiva y por ello no pudo avanzar.

Posteriormente se ha encontrado una solución a este problema de la que el autor es responsable, así como de una implementación de la misma. Si el futuro a partir de aquí se parece en algo al pasado más reciente, se puede decir con seguridad que han de existir más puntos de vista interesantes (útiles) que pueden tomar como inspiración inicial los métodos de modelado basado en campos escalares que aquí se han descrito.

BIBLIOGRAFIA:

[Foley95] FOLEY, J.D.; VANDAM, A; FEINER, S.; HUGUES, J.F.; *Computer Graphics.Principles and Practice.* Addison-Wesley.

[Moore92] MOORE, D., WARREN, J., *Compact Isocontours from Sampled Data*, Graphics Gems III, págs 23-28, Academic Press.

[Barr92] BARR, A.H., *Rigid Physically Based Superquadrics*, Graphics Gems III, págs 137-159, Academic Press.

[Shig91] SHIREGU, M., *Volumetric Shape Description of Range Data using «Blobby Model»*, SIGGRAPH' 91 Conference Proceedings, 227-235,ACM Press.