

UTILIZACION COMPARADA DE DIVERSOS ALGORITMOS DE VISUALIZACION DE FACETAS EN C.A.D.

O.Santana, J.B.Albertini, B.Aubry,
G.Meunier, J.L.Coulomb

Laboratoire d'Electrotechnique de Grenoble (UA CNRS 355)
ENSIEG, BP.45
38402 St. Martin d'Hères
FRANCIA

En el *Laboratoire d'Electrotechnique de Grenoble* (LEG), trabajamos en la elaboración de un software de cálculo de campos electromagnéticos tridimensionales bajo el método de los elementos finitos. La finalidad de este software es la concepción y la modelización de máquinas eléctricas. La descripción geométrica y el mallaje (*mesh*) de los objetos tratados con este software, necesitan una visualización frecuente bajo diversas posiciones de los objetos. Por esta razón, la rapidez de la visualización es imprescindible.

Para la visualización, cada faceta es considerada como plana y poligonal, de ahí la necesidad de una descomposición de las facetas no planas en sub facetas (por ejemplo las cilíndricas o esféricas).

En el LEG disponemos de pantallas de visualización de tecnologías diferentes : con barrido aleatorio *random-scan display* (Tektronix 4014) y con barrido *TVraster-scan display*. Las pantallas más recientes realizan el llenado de polígonos (Apollo, Tektronix 4105,4107,4205...) ; las menos recientes coloreaban pixel por pixel (Ramtek 6211, Secapa 550,...)

Hemos elaborado algoritmos de eliminación de partes escondidas (*hidden parts*) específicos a cada tipo de pantalla de manera tal de explotar las posibilidades de cada pantalla al máximo. Estos algoritmos se utilizan en bibliotecas de subrutinas fortran-77. Cada software que necesita estos algoritmos, debe inicialmente seleccionar el tipo de pantalla gráfica; esta tarea se hace de forma interactiva gracias a una biblioteca de diálogo.

Los algoritmos realizados son los siguientes:

El algoritmo de *Z-Buffer* que consiste en un barrido de la pantalla pixel por pixel de arriba hacia abajo y de izquierda a derecha. En cada pixel una línea perpendicular a la pantalla atraviesa el objeto, de esta manera un cierto número de facetas serán intersectadas. La faceta de profundidad máxima es almacenada (el color o la intensidad por ejemplo). El punto de intersección de profundidad máxima (el eje Z apunta hacia el usuario) corresponde a la faceta vista. Este método permite el coloreado de objetos y exige procesadores rápidos.

El algoritmo *Scan-Line* barre la pantalla en una serie de planos (XZ) pasando por cada pixel del eje Y. Cada intersección de estos planos con el objeto es descompuesto en una serie de segmentos. Para cada capa de pixels en Y, dibujamos los segmentos después de haberlos clasificado. En relación al algoritmo precedente, éste gana una dimensión.

El algoritmo del *Pintor* consiste en clasificar las facetas para el dibujo según el orden decreciente de la coordenada Z del centro de gravedad. Este algoritmo es simple y rápido, y da buenos resultados si se trabaja con facetas pequeñas, obtenidas después de un mallaje en elementos finitos, por ejemplo. Este algoritmo funciona solamente en pantallas a borrado selectivo y es fácil de poner en operación.

El algoritmo de *Newell, Newell y Sancha* funciona igualmente en pantallas a borrado selectivo y llenado de polígonos. El objetivo de este algoritmo es de dibujar sucesivamente las facetas que no

son escondidas o tapadas por las facetas ya dibujadas precedentemente. El problema del grupo de facetas que se cubren mutuamente (por ejemplo A cubre B, B cubre C et C cubre A) es resuelto con el recorte de una o varias facetas del grupo. Gracias a este recorte eventual, el algoritmo resuelve cualquier tipo de ambigüedad.

El algoritmo de *eliminación de líneas escondidas* calcula en forma exacta los segmentos realmente vistos. Este algoritmo trata solamente facetas triangulares; por lo tanto, es necesario descomponer las facetas iniciales en triángulos (después de un mallaje tetraédrico por ejemplo). Para cada triángulo hay que determinar los pedazos de segmentos que no son tapados por los otros triángulos.

Los algoritmos [1] y [2] no han sido introducidos en el software de CAD *flux3d* elaborado en el equipo de Modelización del LEG dada su lentitud. Los algoritmos [3] y [4] se adaptan mejor a la visualización de facetas planas. Con nuestras pantallas a borrado selectivo utilizamos el algoritmo [4] para visualizar los objetos y el algoritmo [3] para representar el mallaje. Con nuestras antiguas pantallas a borrado no selectivo y con las mesas trazadoras, utilizamos el algoritmo [5] que es mucho más lento.

