

Universitat Politècnica de Catalunya
Dep. Teoria del Senyal i Comunicacions
E.T.S. d'Enginyeria de Telecomunicació

**APLICACION DEL MODELADO GEOMETRICO A
PROBLEMAS DE DIFRACCION ELECTROMAGNETICA**

Mercè Vall-llossera Ferran

Director: Dr. Juan Manuel Rius Casals

UNIVERSITAT POLITÈCNICA DE CATALUNYA
ADMINISTRACIÓ D'ASSUMPTES ACADÈMICS

Aquesta Tesi ha estat onregistrada
a la pàgina 61 amb el número 566

Barcelona,

L'ENCARREGAT DEL REGISTRE,

Tesis doctoral presentada a la
Universitat Politècnica de Catalunya
para la obtención del título de
Doctor Ingeniero de Telecomunicación

Barcelona, Marzo 1994

CAPITULO 3

VISUALIZACION GRAFICA TRIDIMENSIONAL

En la actualidad se ha dedicado un esfuerzo muy importante a la creación de técnicas que posibiliten la generación de imágenes realistas. La principal razón de estos estudios se debe a que existen muchos campos en los que la visualización de imágenes realistas por ordenador facilitan enormemente sus objetivos; entre ellas cabe destacar la simulación, el diseño, la publicidad, la educación, etc.

La principal dificultad en la obtención de un realismo visual total es la complejidad del mundo real, solo hay que observar la riqueza de nuestro entorno para comprobarlo: Hay superficies de texturas muy variadas, gradaciones de color, sombras, reflexión y pequeñas irregularidades en los objetos que nos rodean. Los costes computacionales para simular todos estos efectos pueden ser muy elevados: la generación por software de estas imágenes puede requerir tiempos de minutos e incluso de horas en ordenadores potentes. Por ello, en las estaciones gráficas más potentes los algoritmos de generación de imágenes se implementan en hardware dedicado, denominado acelerador gráfico, el cual permite obtener imágenes muy complejas en tiempo real.

La visualización realista tridimensional presenta una gran importancia para el análisis electromagnético de modelos geométricos por los siguientes motivos:

- Por un lado, la representación de imágenes realistas permite la visualización de los modelos generados (el capítulo 2 se ha dedicado al modelado geométrico tridimensional) y con ello verificar si son correctos o no (interrogación), antes de realizar el estudio electromagnético mediante las técnicas numéricas que se presentarán en el capítulo 4.
- Por otro, la aplicación de análisis en alta frecuencia desarrollada en el departamento, *GRECO*, mediante el procesado gráfico de una imagen realista es

capaz de obtener, por ejemplo, la sección recta de blancos radar (ver capítulo 5) o el diagrama de radiación de reflectores conformados (capítulo 6). La gran velocidad de este método de análisis ha permitido desarrollar técnicas de diseño interactivo y optimización automática de formas de mínima sección recta radar (ver capítulo 7 de la presente tesis).

Por estos motivos ha sido necesario un estudio exhaustivo de las distintas técnicas de visualización de imágenes realistas, de las cuales las más útiles para los métodos de análisis electromagnético son las que se presentan en este capítulo.

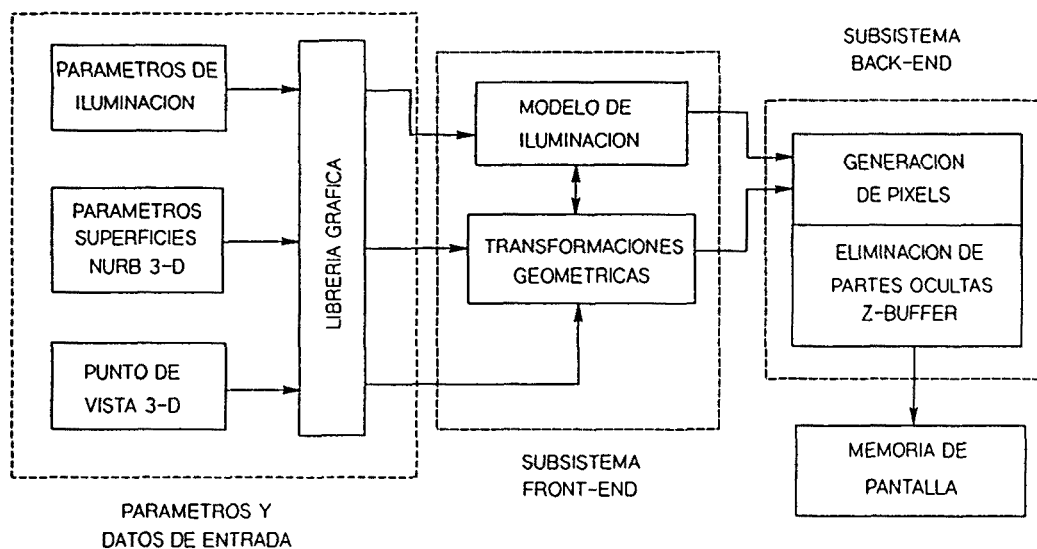


Figura 3-1: Visualización realista de una escena tridimensional

La figura 3-1 presenta un diagrama de bloques muy general en el que se muestran las distintas fases de la visualización realista tridimensional en las estaciones gráficas actuales. Primeramente se comentan los datos y parámetros que deben especificarse a través de la librería gráfica. En segundo lugar se estudia el subsistema *front-end*, constituido por los modelos de iluminación y las transformaciones geométricas necesarias para descomponer la escena en una colección de polígonos coloreados. Posteriormente se considera el subsistema *back-end*, que incluye la generación de píxeles y eliminación de partes ocultas, necesarias para representar en pantalla los polígonos anteriores. Finalmente, la última sección se dedica a las distintas soluciones hardware

desarrolladas para conferir a las estaciones gráficas actuales de una mayor potencia que permita la visualización de imágenes extremadamente complejas en tiempos razonables.

3.1 PARAMETROS Y DATOS DE ENTRADA

En el campo del diseño gráfico por ordenador se han desarrollado numerosas técnicas de visualización gráfica realista de objetos situados en un determinado entorno. Para ello es necesario determinar la posición y orientación de todos y cada uno de los objetos que intervienen en la escena, definir la iluminación y los coeficientes de reflectividad luminosa de sus superficies, eliminar las partes ocultas que quedan eclipsadas, etc. De esta forma se obtienen imágenes de gran realismo, en las que incluso se llega a simular la textura de los diversos materiales que componen los objetos.

a) Información básica

La generación gráfica realista requiere la especificación de una escena tridimensional completa (ver figura 3-1):

- **Observador:** Posición, dirección de observación y ángulo de visión. Utilizando el hardware y software adecuados, el punto de vista, puede variar con una botonera de mando, observándose en tiempo real como se actualiza la escena para cada punto de vista.
- **Modelo geométrico de los objetos:** Existen muchas técnicas de modelado geométrico tridimensional, al estudio de las mismas se ha dedicado el capítulo 2 de la presente tesis. Sin embargo, el método más común en los aceleradores gráficos hardware es el modelado por fronteras, en los que se define la superficie del objeto, bien aproximándola por facetas, o bien de forma precisa con superficies paramétricas racionales NURBS.
- **Reflectividad de las superficies:** Especificación de las propiedades físicas de las superficies de los objetos: coeficientes de reflexión para cada color de la luz, parámetros de dispersión de la energía luminosa incidente, etc.
- **Escena:** Situación de los objetos tridimensionales en la escena y orientación de los mismos relativa al observador.
- **Iluminación:** Parámetros de las fuentes luminosas de la escena: posición dirección de iluminación, diagrama y directividad del haz luminoso, etc.

El resultado es una imagen bidimensional correspondiente a la visión del observador de la escena tridimensional original, en la cual se han suprimido las superficies ocultas o eclipsadas y se muestran las superficies visibles con el coloreado y sombreado correspondientes a las condiciones de iluminación. Todo este proceso puede realizarse en tiempo real con ayuda de un acelerador gráfico hardware, capaz de representar instantáneamente y de forma realista superficies NURB de hasta grado 6 a partir de los parámetros que las definen.

b) Información opcional para mejora del realismo

Independientemente de la tecnología de visualización utilizada, la investigación dedicada a la generación de imágenes realistas tiene abiertas diversas líneas con el objetivo de conseguir modelos más precisos, tanto estáticos como dinámicos. Algunos investigadores han desarrollado modelos apropiados solamente a tipos especiales de estructuras tales como gases, ondas, montañas y árboles. Otros, en cambio, se han dedicado al desarrollo de algoritmos eficientes para posicionar de forma automática gran cantidad de objetos de una misma clase en escena, un ejemplo de dicha aplicación es la simulación por ordenador de un bosque, en la que se requiere situar un gran número de árboles. A continuación se comentan algunas de estas líneas de investigación.

- **Textura:** Se han desarrollado técnicas que son capaces de conferir textura a las superficies de los objetos que intervienen en una escena realista, para simular detalles sobre las superficies de los objetos. Existen muchas maneras de representar las texturas, las cuales van desde una variación de color sobre la superficie a, incluso, simular deformaciones de las geometrías de los objetos.

- **Dinamismo:** Por dinamismo se entiende cambios que se propagan a lo largo de una secuencia de imágenes, entendiéndose por cambio cualquier modificación de la posición u orientación de los objetos, del tamaño de los mismos, de las propiedades de los materiales o variaciones de iluminación y especificaciones de ángulo de vista. Por tanto, quedan incluidas modificaciones en cualquier parte de la escena o de las técnicas de visualización aplicadas a ella. La clase de dinamismo más popular es el movimiento pasando desde transformaciones sencillas controladas por el usuario a la animación [Foley'90].

- **Antialiasing:** Las primitivas representadas en una pantalla de ordenador tienen el problema de que son una representación digitalizada de la realidad y por tanto, como máximo pueden ofrecer una precisión de un pixel en la representación de los objetos. Ello produce que, por ejemplo, las líneas o aristas inclinadas sean siempre aproximadas en escalera. El efecto escalera, conocido como *aliasing*, es visible porque cada pixel tiene un color uniforme en todo el espacio que ocupa.

Se han desarrollado algunos métodos que combaten este efecto. A continuación, se va a comentar la filosofía de trabajo de una de estas técnicas de "antialiasing", quizá la más intuitiva de todas ellas, aplicada al problema particular de la representación de una línea inclinada en una imagen en blanco y negro.

Si se interpreta una recta como un rectángulo de un grosor determinado cubriendo una porción de malla de cuadros tal como se muestra en la figura 3-2, (en este caso cada pixel se corresponde a un cuadro) una técnica que no evite el aliasing asigna a los pixels totalmente dentro del rectángulo el color negro, a los que quedan totalmente fuera el color blanco, y a los situados en una posición intermedia (existentes en líneas inclinadas) según el porcentaje de área cubierta se les asigna uno u otro color, produciéndose el efecto escalera.

La técnica de *antialiasing*, en cambio, asigna a los pixels parcialmente dentro un color gris cuya intensidad será proporcional al porcentaje de área del pixel fuera del rectángulo. Esta aproximación suaviza la transición y, además, la línea vista desde una cierta distancia parece más real (ver figura 3-3), aunque se haya producido un ensanchamiento de la transición de una zona a otra.

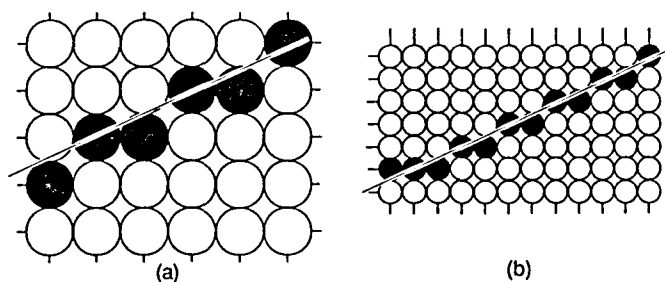


Figura 3-2: Efecto de aliasing en la representación de una línea inclinada. Aunque la figura b) tenga doble resolución respecto la a), el efecto de escalonamiento no se evita.

- **Stereopsis:** Todo lo que se ha comentado hasta aquí era referente a la obtención de imágenes realistas bidimensionales, pero en la actualidad también se han desarrollado técnicas para obtener imágenes tridimensionales que proporcionan la sensación de ocupar espacio y no una escena realista proyectada en un plano.

El hombre es capaz de ver los objetos en tres dimensiones gracias a que la imagen percibida en un ojo es ligeramente distinta de la percibida en el otro, ya que existe una separación de unos pocos centímetros entre ambos. El cerebro fusiona las dos imágenes en una, la cual es interpretada en tres dimensiones. Se

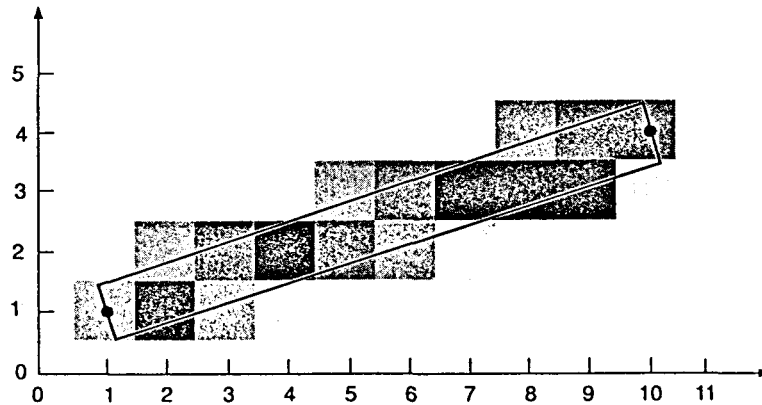


Figura 3-3: Ejemplo de aplicación de antialiasing sobre una línea inclinada: Intensidad proporcional al área cubierta.

han desarrollado varias técnicas que consiguen que los dos ojos capten imágenes ligeramente distintas a partir de una imagen en pantalla ya sea mediante el uso de gafas con filtros polarizadores distintos para cada ojo o bien mediante métodos de holografía [Foley'90].

c) Librerías gráficas

El continuo avance en la tecnología hardware ha hecho posible la evolución de las pantallas gráficas desde un tipo especial de dispositivo a llegar a convertirse en el interface humano estándar hacia el ordenador, posibilitando la visualización de imágenes realistas de gran calidad. Por otro lado, los paquetes de software para visualización y generación de imágenes realistas han evolucionado desde paquetes dependientes del equipo a bajo nivel y suministrados por las empresas para sus dispositivos de presentación particulares a paquetes de mayor nivel e independientes del dispositivo.

Estos paquetes software pueden funcionar en una gran variedad de dispositivos de presentación que van desde impresoras láser y plotters a registradores de película y visualizaciones en pantalla en tiempo real de altas prestaciones. Esto se ha conseguido con la creación de librerías gráficas que se utilizan como los lenguajes de programación de alto nivel lo cual las confiere de portabilidad. Esta portabilidad se proporciona de la misma forma que lo hacen los lenguajes de programación de alto nivel (como el Fortran, el C o el Pascal), que son independientes de la máquina, consiguiendo aislar al programador de la mayoría de características particulares de la misma.

La librería gráfica común a todas las estaciones de trabajo HEWLETT-PACKARD, utilizadas para la realización de esta tesis, es la librería gráfica *starbase*. Basada en los standards ANSI CGI y PHIGS, proporciona un lenguaje de alto nivel independiente del dispositivo de salida. Incluye primitivas avanzadas como NURBS, mallados, iluminación, imágenes realistas, visualización tridimensional, etc. Estas primitivas son independientes del dispositivo de salida: un controlador de dispositivo las traduce en instrucciones propias del dispositivo gráfico utilizado.

3.2 TRANSFORMACIONES GEOMETRICAS

Para representar una escena tridimensional en un espacio de dos dimensiones, como es la pantalla de la estación gráfica, deben realizarse una serie de operaciones y transformaciones geométricas con cada uno de los puntos, líneas y polígonos de los objetos que componen la escena. Se conoce como "*pipeline*" gráfico al modelo conceptual del proceso que sufren los datos en su conversión desde el modelo del usuario hasta la imagen en la pantalla, tal como se representa en la figura 3-4. Este *pipeline* está controlado por las librería gráficas, de forma que el usuario sólo debe proporcionar los datos necesarios.

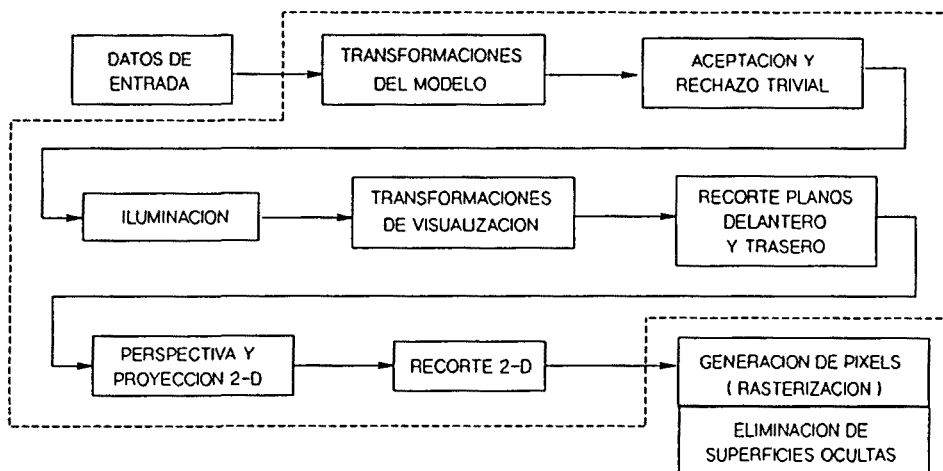


Figura 3-4: "*Pipeline*" para visualización realista tridimensional. Se ha enmarcado en línea de puntos el subsistema front-end, formado por las transformaciones geométricas y los modelos de iluminación.

El *pipeline* que se muestra en el diagrama de bloques de la figura 3-4 utiliza un modelo de iluminación local, con algoritmo de eliminación de superficies ocultas de *z-buffer* (ver sección 3.5.1) y suavizado de *Gouraud* (ver sección 3.3.3). Con otras técnicas de visualización pueden obtenerse *pipelines* algo distintos al que se presenta, pero con una estructura parecida. Se ha elegido este modelo porque es el utilizado por el programa

GRECO (ver capítulo 5) para visualizar en pantalla los objetos que posteriormente procesa.

Las distintas transformaciones geométricas que se realizan son:

- **Transformaciones del modelo:** En esta fase los objetos que forman la escena se transforman del sistema de coordenadas del modelo (MC Modelling Coordinate) al sistema de coordenadas del mundo (WC World Coordinate). Esto se realiza sobre los vértices de los polígonos y los puntos de control de los NURBS aplicando matrices de transformación (rotación, escalado y traslación) en coordenadas homogéneas.

Posteriormente se convierten las superficies paramétricas NURB en mallas de polígonos que las aproximan con la resolución deseada y se calculan las normales a todos los polígonos. Como los modelos de iluminación precisan del conocimiento de dichas normales, es necesario que cualquier transformación realizada sobre los objetos afecte también a todas las normales a las superficies de los mismos.

- **Aceptación y rechazo trivial:** En este paso se testean las primitivas (expresadas ya en coordenadas del mundo) para ver si los objetos quedan totalmente fuera o dentro de la ventana de visualización, eliminando aquellas que quedan totalmente fuera. También se eliminan en este paso aquellas superficies cuyas normales apuntan en una dirección contraria al observador (técnica *back-face culling*).

Eliminando todos estos objetos y superficies al principio de la cadena de visualización se consigue reducir considerablemente el procesado en pasos posteriores.

- **Iluminación:** En este paso se calcula el color y la luminosidad de cada punto de las superficies a visualizar. Para conseguir imágenes realistas se utilizan normalmente modelos de iluminación especular, difusa y ambiental. A este aspecto se dedica la sección 3.3.

Dado que las superficies curvadas han sido aproximadas con mallas de polígonos, es necesario aplicar algún algoritmo de suavizado de la imagen para evitar un aspecto facetizado de dichas superficies. Estas técnicas se comentan en la subsección 3.3.3.

- **Transformaciones de visualización:** En un sistema de coordenadas 3-D está definida la posición del punto de vista y la dirección de observación. Para

proyectar todos los puntos de la escena en un plano perpendicular a la dirección de observación, deben realizarse una serie de transformaciones geométricas (rotaciones y/o traslaciones de los objetos). Como interesa realizar todo tipo de transformaciones mediante el producto de matrices por los puntos a transformar, se utiliza un sistema de coordenadas homogéneo de cuatro dimensiones [Dewey'88].

El resultado de las transformaciones es una escena 3-D con el eje z en la dirección del punto de vista y los ejes x e y sobre el plano perpendicular a la dirección de observación. En este nuevo sistema de coordenadas de pantalla, el valor de z representa la distancia de cada punto al observador y x e y equivalen a las coordenadas 2-D de cada punto sobre la estación gráfica.

- **Recorte de planos delantero y trasero:** Se eliminan todos los puntos que quedan por detrás del observador, que no deben ser visibles, así como aquellos cuya distancia al punto de vista es mayor que un valor predeterminado.

- **Perspectiva y proyección 2-D:** Se sitúa cada punto sobre la pantalla en función de sus coordenadas x e y . En el caso de proyección con perspectiva, las coordenadas x e y se dividen por un valor proporcional a la coordenada z , con lo que aparecen más pequeños los objetos más alejados del observador.

- **Recorte 2-D (*clipping*):** Se eliminan las partes de los objetos que se encuentran fuera de los límites de la pantalla. Esta operación, que es trivial con los puntos, requiere algoritmos especiales con líneas o polígonos que se encuentran parcialmente fuera de la pantalla [Dewey'88].

Después de sufrir las transformaciones geométricas y de ser aplicado un modelo de iluminación, la salida del subsistema *front-end* entra en el subsistema *back-end* donde se realiza:

- **Generación de pixels (*rasterization*):** En esta última fase del *pipeline* se transforman las primitivas en colores de pixels de pantalla y se almacena esta información en la memoria de pantalla. El proceso de *rasterization* se compone de dos subfases: exploración de la información y determinación de las superficies ocultas, a las que se dedican respectivamente las secciones 3.4 y 3.5.

El proceso de *rasterization* requiere la determinación de los pixel de pantalla cubiertos por cada polígono. Los sistemas de visualización mediante software pueden realizarlo de dos formas distintas: haciendo un estudio primitiva a primitiva (siguiendo un orden a partir de los objetos que intervienen en la escena) o pixel a pixel (orden de la imagen). La mayoría de las técnicas

modernas realizan el proceso de *rasterization* objeto a objeto, usando el algoritmo de *z-buffer* para determinar las superficies ocultas, que no deben aparecer en la escena. Existen, sin embargo, otros muchos algoritmos de eliminación de superficies ocultas [Dewey'88] [Foley'90], como son la partición binaria del espacio y la cola ordenada de objetos. En la sección 3.5 se comentan con detalle cada uno de ellos.

3.3 MODELOS DE ILUMINACION

En una imagen fotorrealista es necesario generar sombreados sobre la superficie considerando la posición, orientación y las características tanto de las superficies de los objetos que intervienen en la escena como de las fuentes de luz que los iluminan. En esta sección se van a describir distintos modelos de iluminación que determinan el color de las superficies en todos los puntos de la imagen.

Los modelos de iluminación obtienen el color de cada superficie en función de la luz que incide sobre ella. Esta luz puede provenir directamente de las fuentes de iluminación o bien haberse reflejado o transmitido antes a través de otras superficies. Los modelos que calculan la luz reflejada y transmitida indirectamente se conocen como "modelos de iluminación global". En cambio, los que únicamente consideran la luz que llega directamente de las fuentes de iluminación se conocen como "modelos de iluminación local".

Sin embargo, la mayor parte de la luz en los entornos del mundo real no procede de fuentes de luz directas, por lo que los modelos de iluminación local necesitan incluir la contribución de la iluminación indirecta. Ello se realiza habitualmente mediante un término de iluminación ambiental que es constante en todas las superficies de los objetos. Este término no depende de la posición de los objetos, ni de la posición del observador, ni de la presencia o ausencia de objetos cercanos que pudieran bloquear la luz incidente. Recientemente se ha comprobado que algunos efectos propios de la iluminación global también pueden conseguirse mediante mapas de reflexión, transparencias o sombras [Foley'90]. Estas aproximaciones obtienen escenas bastante realistas con tiempos de cálculo muy bajos.

En la actualidad se han desarrollado dos tipos de algoritmos para generar imágenes que enfatizan la contribución de la iluminación global. En la sección 3.3.2 se estudian los modelos de iluminación global. Por un lado, el algoritmo de trazado de rayos, o "*ray tracing*", incluye la determinación de las superficies visibles y sombreado para representar las sombras, reflexión y refracción. Por otro lado, las técnicas de "*radiosity*" (o "radianza" en castellano) modelan el equilibrio energético entre las superficies de una

escena analizando la interacción entre cada par de superficies con independencia del ángulo de vista, antes de aplicar las transformaciones de visualización (ver sección 3.2).

Estos métodos de iluminación global consiguen imágenes mucho más realistas que aquéllas en las que sólo se considera la contribución de la iluminación directa. Sin embargo, esto repercute en un incremento considerable de los tiempos de cálculo. Esta es la razón por la que sólo se utilizan los métodos de iluminación global en casos en que los tiempos de cálculo no son muy restrictivos o en imágenes de poca complejidad.

3.3.1 Modelos de iluminación local

A partir de lo comentado en la introducción de esta sección se deduce que una de las técnicas más efectivas para dar realismo a las imágenes por ordenador es la inclusión de la iluminación en las escenas. Sin embargo, el modelado de la energía electromagnética y su interacción con el ojo humano no es un problema sencillo de tratar. La psicología y la fisiología de la percepción visual han sido estudiadas con mucho detalle, y se ha llegado a la conclusión de que hacer un modelo exacto sería demasiado complejo y representaría un coste computacional excesivo.

Por ello, los modelos matemáticos de iluminación utilizan ciertas simplificaciones que permiten obtener efectos suficientemente realistas en tiempo real utilizando técnicas de aceleración gráfica hardware. Uno de los modelos más populares es el de iluminación local de Phong [Phong'73].

En los modelos de iluminación local, la simplificación más importante que se introduce es considerar únicamente las reflexiones de primer orden de la luz en las superficies. La principal consecuencia es que se desprecian las reflexiones múltiples entre superficies, de forma que la luminosidad de cada una depende sólo de su orientación relativa respecto a la dirección de iluminación. Otra simplificación muy habitual es el uso de fuentes luminosas puntuales y direccionales, en lugar de fuentes superficiales como existen en la realidad física.

a) Fuentes de luz

Los modelos de iluminación local consideran dos tipos básicos de fuentes de luz:

- **Fuente direccional:** Es una fuente de luz puntual localizada en un punto infinitamente lejano, situado en una dirección conocida. Por tanto, los rayos que emite dicha fuente son paralelos, de forma que la dirección de la luz incidente es la misma para todos los objetos de la escena, independientemente de la posición. El frente de onda es plano, por lo que la luz no sufre atenuación en su trayecto de la fuente al objeto.

- **Fuente puntual:** Está situada relativamente cerca del volumen en el que se encuentra la escena, por lo que los rayos que emite no son paralelos y el ángulo con el que un rayo de luz llega a la superficie de un objeto depende de la posición del objeto respecto a la fuente de luz. El frente de onda es esférico, con lo que la luz sufre atenuación inversamente proporcional al cuadrado de la distancia recorrida por el rayo. Las fuentes puntuales de luz pueden definirse con un diagrama de radiación isotrópico, cónico, o bien de la forma $\cos^n \theta$.

La atenuación considera el hecho de que la energía emitida desde una fuente de luz puntual decrece con la inversa del cuadrado de la distancia. Sin embargo, en la práctica esto no da resultados del todo satisfactorios; ya que aunque este comportamiento sea el correcto para fuentes puntuales de luz, los objetos con los que nos encontramos en la vida real normalmente no están iluminados por fuentes de luz puntuales. Esta es la razón por la que es habitual utilizar un coeficiente de atenuación de la luz con la distancia de la forma:

$$\alpha = \frac{1}{C_1 + C_2 r + C_3 r^2} \quad (3.1)$$

el cual permite modelar gran cantidad de efectos luminosos. Las constantes C_1 , C_2 y C_3 están definidas por el usuario y asociadas a la fuente puntual de luz. La expresión impone como valor máximo de este factor la unidad para garantizar que siempre existe atenuación, y no amplificación, con la distancia.

b) Reflexión de la luz en las superficies

Cuando la luz incide sobre la superficie de un objeto, puede ser absorbida, transmitida o reflejada. Las técnicas de visualización gráfica por ordenador se interesan únicamente por la parte transmitida y la reflejada, ya que son las que hacen visible al objeto. La intensidad de la luz reflejada y de la transmitida por un objeto depende de la longitud de onda y de los ángulos de incidencia y de observación respecto a la normal a la superficie. Utilizando coeficientes de reflexión distintos para cada longitud de onda incidente, puede simularse con gran realismo la coloración y textura de distintos materiales que componen los objetos.

Los modelos de iluminación local consideran tres tipos básicos de reflexión en la superficie de los objetos: ambiental, debida a la luz que recibe el objeto proveniente del entorno que le rodea; difusa o independiente de la dirección de observación, que proporciona una apariencia mate a las superficies; y especular o direccional, que proporciona una apariencia brillante. La luminosidad total de la superficie es igual a la suma de los tres tipos de reflexión y se calcula por separado para cada color, consiguiéndose efectos de gran realismo.

Luz ambiental

La luz ambiental es la luz que recibe un objeto por dispersión de todo el entorno que le rodea, tanto proveniente de las fuentes de luz como de las demás superficies. En realidad, es una fuente de luz distribuida, y un modelo exacto sería extremadamente complejo y computacionalmente muy costoso. Por eso, la luz ambiental se suele modelar como una reflexión difusa y constante en todas las superficies de la escena, independientemente de su posición y orientación respecto a las fuentes de luz puntuales.

La energía reflejada de la luz ambiental L_α suele calcularse como el producto de la intensidad de luz presente en la escena L , debida a las fuentes de iluminación, por el coeficiente de reflexión difusa de cada superficie ρ_d :

$$L_\alpha = \rho_d L \quad (3.2)$$

- La intensidad incidente L se considera constante para todas las superficies de la escena
- El coeficiente de reflexión ρ_d modela la cantidad de luz reflejada desde la superficie de un objeto y su rango de valores está comprendido entre 0 y 1. Este coeficiente es una propiedad asociada al material del que está hecho el objeto y, en general, se define de forma empírica, sin correspondencia directa con ninguna propiedad física del material.

Para acentuar más el realismo de la imagen, la energía reflejada se calcula por separado para cada una de las componentes espectrales, o colores, de la luz ambiental.

Reflexión difusa

Aunque los objetos iluminados con luz ambiental quedan más o menos sombreados proporcionalmente a la intensidad de luz que reciben, esta iluminación es uniforme en toda su superficie. En la realidad física no es así: el brillo de la superficie no es uniforme, pues depende de la orientación de la misma y de la distancia a la fuente de luz.

Se dice que hay reflexión difusa cuando la luz es reflejada en todas direcciones con la misma intensidad, proporcionando una apariencia mate a las superficies. El fenómeno físico correspondiente a la reflexión difusa es la reemisión de la luz absorbida por la superficie del objeto.

La intensidad de la luz que llega al observador por reflexión difusa es independiente de la dirección de observación, y se calcula como el producto de la luminosidad incidente L , el coeficiente de reflexión difusa ρ_d y el coseno del ángulo de incidencia respecto a la normal a la superficie, θ_i , teniendo en cuenta el factor de atenuación por propagación α (ver figura 3-5):

$$L_d = \rho_d \alpha L \cos \theta_i \quad (3.3)$$

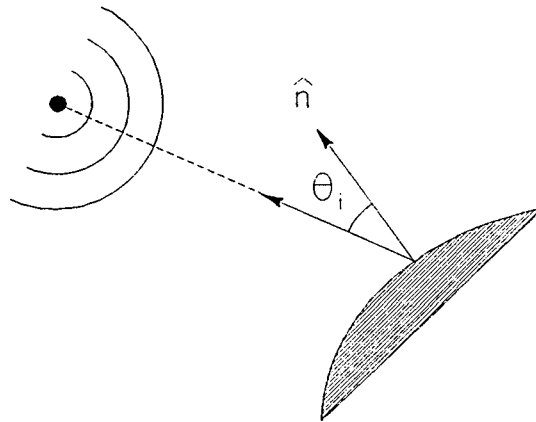


Figura 3-5: El ángulo de incidencia es el parámetro que determina la variación del brillo de las superficies.

Este tipo de dependencia con el ángulo de incidencia se debe a que la potencia incidente interceptada por la superficie, y reradiada por la misma es proporcional a la proyección del área de la superficie en un plano perpendicular a la dirección de incidencia y, por tanto, proporcional a $\cos \theta_i$.

También la reflexión difusa se calcula por separado para cada una de las componentes espectrales de la luz incidente.

Reflexión especular

La reflexión especular aparece en cualquier superficie pulida. Este efecto puede observarse al iluminar una manzana con luz blanca: la superficie de la misma presenta un punto muy brillante que es causa de la reflexión especular, mientras que la luz reflejada por el resto de superficie es el resultado de la reflexión difusa. Incluso puede notarse que este "punto brillante" no se ve de color rojo (color de la superficie de la manzana) sino blanco, que corresponde al color de luz incidente.

Al contrario de la reflexión difusa, la reflexión especular está caracterizada por una fuerte dependencia con la dirección de observación, por lo que produce puntos brillantes

sobre las superficies pulimentadas. La reflexión especular se produce en aquellas superficies en las que no penetra la energía electromagnética. Sobre una superficie plana infinita, según la ley de Snell, toda la energía se refleja en un ángulo de reflexión igual al de incidencia. Sobre superficies planas finitas aparece un cierto diagrama de reflexión, con el máximo en la dirección de reflexión especular dada por la ley de Snell.

Phong desarrolló el modelo de reflexión especular más popular para reflectores no perfectos. Dicho modelo asume que el máximo de reflexión especular se produce en la dirección dada por la ley de Snell. Para las demás direcciones, la intensidad de la luz reflejada se aproxima por $\cos^n \theta$, donde θ es el ángulo que forma la dirección de observación con la especular, ver figura 3-6, y n es un coeficiente cuyo valor varía según el material.

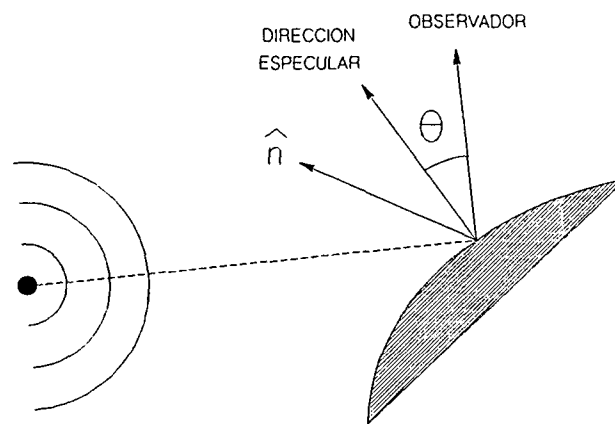


Figura 3-6: El parámetro fundamental del modelo de Phong es el ángulo que forma la dirección de observación con la de reflexión especular.

Los valores del coeficiente n pueden variar desde 1 hasta varios centenares, según el material a simular: $n = 1$ proporciona un decaimiento de la energía lento y, por tanto, la zona en la que se produce reflexión especular es grande. Valores grandes de n simulan puntos brillantes más focalizados, como se muestra en la figura 3-7.

El cálculo de la luz obtenida por reflexión especular se obtiene de

$$L_e = \rho_e \alpha L \cos^n \theta \quad (3.4)$$

Como en el caso de reflexión difusa, L es la luminosidad incidente y α el factor de atenuación de la luz en el medio. En el caso de la luz especular, se definen distintos coeficientes de reflexión ρ_e para cada componente espectral, y, como en los casos anteriores, se calcula la luz reflejada por separado para cada una de ellas.

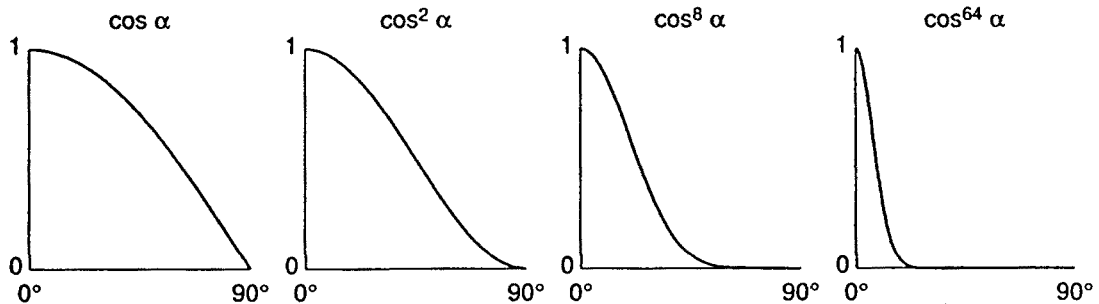


Figura 3-7: El valor del coeficiente "n" en el modelo de iluminación de Phong determina el tamaño de los puntos brillantes.

Suma de reflexiones

La luz reflejada por una superficie para cada componente espectral, y por tanto su coloración, es la suma de las reflexión de luz ambiental y de las reflexiones difusa y especular debidas a cada una de las N fuentes de luz presentes en la escena:

$$L_{TOTAL} = L_a + \sum_{i=1}^N L_{d_i} + \sum_{i=1}^N L_{e_i} \quad (3.5)$$

3.3.2 Modelos de iluminación global

Los métodos de iluminación local descritos en el apartado anterior sólo calculan la luz que se propaga desde las fuentes que están iluminando la escena hasta cada uno de los objetos, se refleja en cada superficie por separado, y llega finalmente al observador. Para ello tiene en cuenta la posición de las fuentes de luz, la atenuación de la luz en el medio, la orientación y los coeficientes de reflexión de las superficies, y la posición del observador. Las fuentes de luz en estos modelos son siempre puntuales o direccionales (puntuales en el infinito), y no permiten la representación de zonas de sombra.

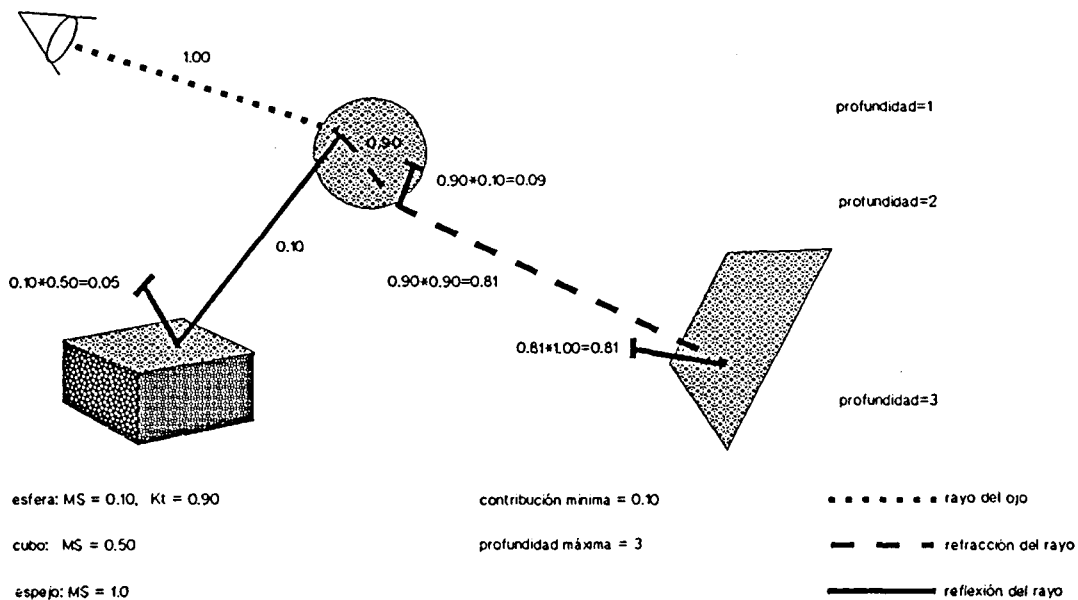
Los modelos de iluminación global, que se van a estudiar en esta sección, añaden el efecto de las reflexiones múltiples de la luz entre las distintas superficies que forman la escena. Asimismo, también pueden definir superficies semitransparentes con una transmitancia, y fuentes de luz superficiales. Estos modelos no modelan la reflexión de la luz en cada superficie por separado, como los modelos de iluminación local, procesan todas las superficies de la escena de forma unitaria, calculando la interacción múltiple entre ellas de forma iterativa y obteniendo de esta manera las zonas de sombra. Los algoritmos de iluminación global más populares son los de trazado de rayos ("ray

tracing") y los métodos basados en la conservación de la energía ("*radiosity*"). Estos métodos son muy distintos entre sí, tanto en sus fundamentos teóricos como en los resultados que obtienen.

Estos modelos obtienen imágenes de mayor realismo que los modelos anteriores, pero son técnicas muy complejas que requieren de un coste computacional muy elevado. Los métodos de trazado de rayos no pueden utilizar en ningún momento el hardware de los aceleradores gráficos, por lo que el procesado debe realizarse por software. Por ello, ya no puede pensarse en obtener imágenes en tiempo real, perdiéndose la interactividad con el usuario que ofrecían las imágenes con iluminación local, pues el tiempo de procesado puede llegar a ser de horas, en imágenes bastantes complejas. Esta es la principal razón por la que aún en la actualidad muy a menudo se utilicen los modelos de iluminación local en la obtención de imágenes fotorrealistas.

a) Ray tracing

El trazado de rayos es ya una técnica clásica de generación de imágenes fotorrealistas [Glassner'89]. Se basa en el seguimiento del camino seguido por los rayos desde el punto de vista (cada punto de la imagen) hasta el entorno exterior de la misma. Cada vez que un rayo intersecta una superficie se divide en dos: un rayo transmitido y otro rayo reflejado. Al repetir el proceso recursivamente, se genera un árbol en el que cada nodo es un punto de intersección entre rayo y superficie (ver figura 3-8).



La imagen se genera considerando cada pixel de pantalla como la raíz de un árbol. La luminosidad de cada pixel se obtiene siguiendo el árbol desde las hojas hacia la raíz. En cada nodo se calcula la intensidad de la contribución de la superficie intersectada a través de un modelo de iluminación local, de forma que la luminosidad final de cada pixel es la suma de las contribuciones de todos los modelos de árbol asociado.

Los fundamentos físicos del trazado de rayos se basan en la aproximación de óptica geométrica, según la cual sólo se refleja energía en la dirección de reflexión especular, por lo que no consideran efectos de difracción. De esta forma se obtienen representaciones muy realistas de objetos reflejados en superficies especulares, así como de la visión a través de superficies semitransparentes. También puede modelarse correctamente el fenómeno de la refracción.

Como principal consecuencia de la aproximación de óptica geométrica, debe mencionarse la generación de sombras con bordes excesivamente abruptos [Greenberg'86], sin zonas de penumbra. Por ello, los algoritmos de *Ray tracing* más avanzados incorporan técnicas de suavizado de bordes. Por el mismo motivo, el modelado de reflexiones difusas y mezclas de colores no puede realizarse de forma suficientemente realista.

Otra característica de las técnicas de trazado de rayos es que generan imágenes estáticas, es decir que deben recalcularse completamente cada vez que cambia el punto de vista del observador. Además, sólo permite definir fuentes puntuales de luz, pero no fuentes superficiales o distribuidas. Esta última limitación ha sido subsanada recientemente con la creación de algoritmos de trazado de rayos distribuidos y trazados de rayos cónicos.

Sin embargo, la principal limitación del trazado de rayos es el tremendo coste computacional que supone el generar un árbol de reflexiones para cada pixel de pantalla, con el agravante de que no permite utilizar el hardware del acelerador gráfico en ningún momento del procesado, ni tan solo para las operaciones más elementales, como son el cálculo de las contribuciones de cada reflexión en las superficies y la representación en pantalla de la escena tridimensional.

b) Radiosity

Aunque la iluminación global por trazado de rayos obtiene un excelente modelo de la reflexión especular y la transparencia refractiva sin dispersión, todavía necesita de la definición de un término de luz ambiental que considere las demás contribuciones globales de la escena. Existen otras aproximaciones basadas en modelos de ingeniería térmica para la emisión y reflexión de radiación los cuales eliminan la necesidad del término de luz ambiental mediante un tratamiento más preciso de las reflexiones entre objetos. Estos algoritmos, conocidos como métodos de "*radiosity*" (radianza en

castellano), fueron introducidos por primera vez por Goral, Torrance, Greenberg y Battaile [Goral'84] y por Nishita y Nakamae [Nish'85] y asumen la conservación de la energía luminosa en un entorno cerrado.

El algoritmo de "radiosity" modela correctamente la interacción entre superficies si la reflexión en ellas es perfectamente difusa. Se basa en el equilibrio energético global de las superficies que forman la escena. Para ello, se define una superficie ficticia que encierra completamente la escena que se desea modelar. Esta superficie está formada por fuentes de luz superficiales, paredes reflectoras y paredes transparentes ("ventanas"). La luz emitida por las superficies, o reflejada en ellas, se considera siempre difusa. También se pueden definir fuentes puntuales y direccionales.

Equilibrio energético global

Supongamos una escena dividida en N elementos de superficie, incluyendo la envoltura exterior. El método de *radiosity* asegura el equilibrio de la energía radiada dentro de la superficie que encierra la escena.

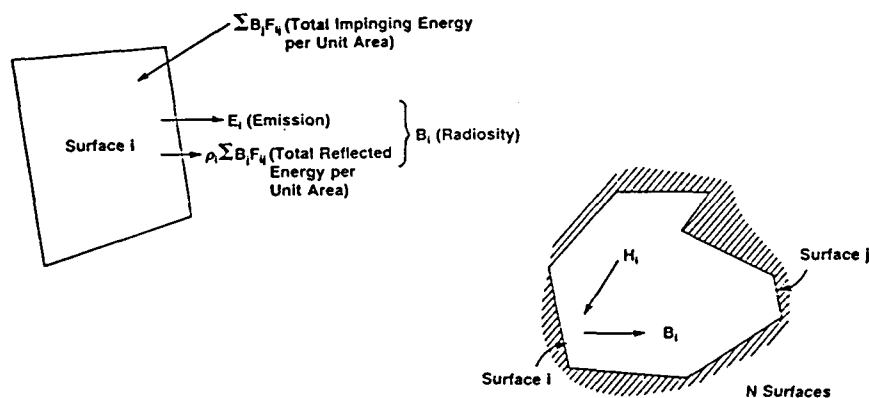


Figura 3-9: Radiación por un elemento de superficie

La densidad de potencia por unidad de superficie B_i (*radiosity*) radiada por la superficie i -ésima es (figura 3-9):

$$B_i = E_i + \rho_i H_i \quad (3.6)$$

donde E_i es la densidad de potencia emitida por la superficie, ρ_i el coeficiente de reflexión y H_i la densidad de potencia incidente por unidad de superficie. Esta última se calcula a partir de la proveniente de las otras $N - 1$ superficies, y la debida a la propia superficie i , si ésta es cóncava:

$$H_i A_i = \sum_{j=1}^N B_j A_j F_{ji} \quad (3.7)$$

con A_i y A_j iguales al área de las superficies i y j , respectivamente, y F_{ij} la porción de la energía radiada por la superficie j que incide sobre la superficie i . Este último parámetro se denomina "factor de forma".

Si consideramos la naturaleza recíproca de los factores de forma:

$$A_i F_{ij} = A_j F_{ji} \quad (3.8)$$

resulta que la combinación de las ecuaciones (3-6) y (3-7) puede escribirse como:

$$B_i - \rho_i \sum_{j=1}^N B_j F_{ij} = E_i \quad (3.9)$$

y para las N superficies:

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1N} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \cdots & -\rho_2 F_{2N} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \rho_N F_{N1} & -\rho_N F_{N2} & \cdots & 1 - \rho_N F_{NN} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \cdot \\ \cdot \\ E_N \end{bmatrix} \quad (3.10)$$

Este sistema de N ecuaciones lineales con N incógnitas B_i tiene como parámetros E_i , ρ_i y F_{ij} . La densidad de potencia emitida por las superficies, E_i , es diferente de cero para las fuentes de iluminación superficiales y para las primeras reflexiones de una fuente puntual o direccional en una superficie, y cero para el resto de las superficies. Los coeficientes de reflexión ρ_i son datos conocidos, y los factores de forma F_{ij} deben calcularse previamente.

Generación de la imagen

Como el sistema de ecuaciones lineales (3-10) es de diagonal dominante, puede resolverse por el método iterativo de Gauss-Seidel: En cada iteración se suma a la potencia radiada por una superficie la contribución proveniente de todas las demás. De esta forma se consigue el refinamiento progresivo de la imagen. Normalmente se consigue la convergencia con un número pequeño de iteraciones del orden de 6 ó 8.

Para obtener una buena resolución en la discretización del modelo de iluminación global, las superficies sobre las que se aplica el algoritmo de *radiosity* son una subdivisión lo

suficientemente fina (elementos diferenciales) de las superficies que forman los objetos de la escena. Una vez se ha obtenido la densidad de potencia radiada por cada elemento de superficie, se realiza una interpolación bilineal sobre las superficies que forman los objetos, evitando de esta forma que el sombreado tenga una apariencia "escalonada".

El proceso de generación de una imagen utilizando el modelo de *radiosity* para iluminación global puede resumirse de la siguiente forma:

- Dividir las superficies de los objetos en parches elementales.
- Calcular los factores de forma entre cada par de elementos de superficie [Rius'91]
- Resolver el sistema de ecuaciones lineales (3-10), obteniendo las densidades de potencia radiadas B_i por cada parche, para cada uno de los tres colores primarios.
- Realizar una interpolación bilineal de los valores obtenidos en cada parche, para conseguir un sombreado suave de las superficies.

Los factores de forma dependen únicamente de la geometría de los objetos que forman la escena, por lo que si se producen modificaciones en la iluminación de la escena E_i , o en los coeficientes de reflexión de las superficies, basta con volver a resolver el sistema de ecuaciones sin recalculer los factores de forma. Además, las imágenes generadas son dinámicas: puede moverse el punto de vista en tiempo real utilizando el acelerador gráfico.

c) Conclusiones

El método de iluminación global por *radiosity* soluciona muchas de las deficiencias del trazado de rayos: produce penumbras y mezclas de colores de gran realismo además de permitir el modelado de fuentes superficiales. Es un modelo "correcto" en el sentido en que se basa en leyes físicas (equilibrio energético).

La principal ventaja con respecto al trazado de rayos es la velocidad: se generan imágenes de escenas muy complejas en minutos, en lugar de horas.

Una ventaja adicional es el desacoplo entre los distintos pasos en la generación de la imagen: cálculo de los factores de forma, resolución del sistema de ecuaciones por refinamiento progresivo y representación bidimensional en la pantalla de la estación de trabajo. Ello permite modificar el punto de vista en tiempo real, o variar parámetros de

iluminación o reflexión sin repetir todo el proceso completo, como sucede con el trazado de rayos. De esta forma se consigue un grado de interactividad con el usuario.

Lógicamente, el método también tiene sus inconvenientes: no puede modelar la reflexión especular, la transparencia y la refracción.

Ambos métodos son complementarios y ninguno modela por sí sólo perfectamente la realidad. Por ello, en la actualidad la investigación se dirige hacia la utilización de métodos híbridos: *radiosity* con cálculos de factores de forma por trazado de rayos [Wallance'89].

3.3.3 Suavizado de la imagen

Para aplicar los modelos de iluminación no es práctico colorear cada pixel de pantalla calculando la normal a la superficie y aplicando el modelo de iluminación y reflexión deseado, dado el elevado costo computacional que supondría por el gran número de pixels presente en la imagen. Por ello se aplican modelos de visualización más eficientes basados en el suavizado de superficies definidas por polígonos o mallas de polígonos.

a) Sombreado constante

El modelo de sombreado o coloración más sencillo consiste en aplicar un valor constante de color para todos los puntos de cada polígono. Este color se calcula a partir de la orientación del polígono en relación con las fuentes de luz y el observador, según el modelo de iluminación utilizado. Ello equivale a muestrear el valor de la ecuación de iluminación una vez para cada polígono, manteniéndose este valor para toda el área del mismo. Este método tan sencillo obtiene soluciones muy burdas en las que no se puede disimular la apariencia facetizada de las superficies curvadas, ni siquiera discretizándolas con un número muy elevado de polígonos.

b) Sombreado con interpolación

Como una alternativa a la evaluación de la ecuación de iluminación en cualquier punto del polígono se han desarrollado técnicas de interpolación de sombreado, en las que la información en el interior de los polígonos se interpola linealmente a partir de los valores en sus vértices.

Para evitar que las superficies curvadas presenten una apariencia facetizada, todos los polígonos que comparten un vértice deben tomar la misma dirección de la normal a la superficie en este vértice, la cual se calcula habitualmente a partir del modelo de NURBS que define la superficie. De esta forma las técnicas de interpolación lineal obtienen la misma coloración a lo largo de una arista para los dos polígonos que la

comparten, disimulando muy bien la facetización de la superficie y dándole una apariencia curvada muy efectiva.

Existen dos técnicas de sombreado por interpolación que se encuentran implementadas en el acelerador gráfico de la mayoría de las estaciones de trabajo potentes. Estas técnicas se conocen como suavizado de Gouraud y suavizado de Phong [Foley'90].

- **Suavizado de Gouraud:** El método de Gouraud realiza una interpolación bilineal de las componentes de color que corresponden a cada uno de los vértices del polígono. Las ecuaciones de interpolación, que son triviales para el caso de triángulos y cuadrados, fueron desarrolladas por Gouraud para el caso de polígonos no regulares con número arbitrario de lados.

- **Suavizado de Phong:** Interpola la normal a la superficie en el interior del polígono a partir de los valores en los vértices. La coloración en todos los puntos del polígono se obtiene a partir de los valores interpolados de la normal.

El suavizado de Phong obtiene generalmente mejores resultados que el de Gouraud, y esta mejora se hace especialmente notable cuando se trabaja con modelos de reflexión especular. En la figura 3-10 se comparan estas dos técnicas de suavizado:

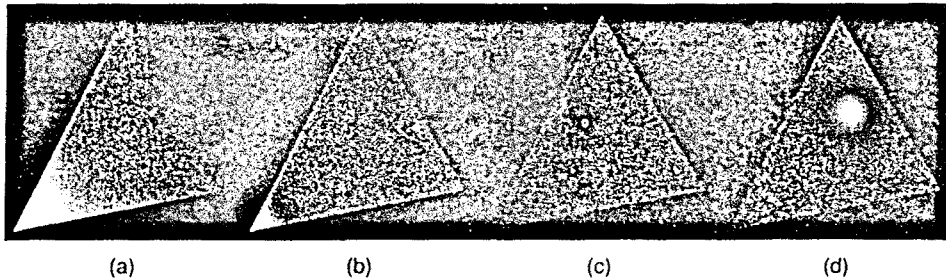


Figura 3-10: Comparación de las técnicas de suavizado de Gouraud y de Phong para una superficie con reflexión especular.

En las imágenes 3-10 a) y b) el punto de reflexión especular se encuentra en un vértice del polígono. La primera, suavizado de Gouraud, tiende a difuminar el brillo especular por el resto del polígono, mientras que la segunda, suavizado de Phong, interpola correctamente el brillo.

Las imágenes 3-10 c) y d) tienen el punto brillante en el interior del polígono. Como el brillo de los vértices es muy bajo, el suavizado de Gouraud no es capaz de

representar un punto brillante en el interior, 3-10 c), mientras que el método de Phong obtiene correctamente la normal a la superficie y el brillo en todos los puntos del polígono, figura 3-10 d).

3.3.4 Aplicación de los métodos de iluminación al análisis electromagnético

Recientemente se han desarrollado técnicas de procesamiento gráfico para el análisis electromagnético en alta frecuencia [Rius'91], [Rius'93]. Estas técnicas, a las que se conoce con el nombre genérico de *GRECO* (GRaphical Electromagnetic COmputing), se basan en procesar una imagen del objeto para extraer sus características geométricas de interés. Para ello es necesario iluminar la escena con el modelo de iluminación en cada caso:

- El procesamiento de una imagen del objeto permite aprovechar la **eliminación de superficies ocultas** para analizar únicamente las superficies iluminadas por el campo incidente. Para ello basta situar al observador de la escena en la misma posición espacial que la fuente electromagnética.
- Puede obtenerse la sección recta radar de un objeto según la aproximación de **óptica geométrica** si se utiliza el modelo de reflexión especular. En este caso, la sección recta radar de una superficie curvada es proporcional al área del punto brillante que aparece en la pantalla.
- Para obtener la sección recta radar según la aproximación de **óptica física** se utiliza el modelo de reflexión difusa: la contribución de cada punto de la imagen al campo dispersado es proporcional al brillo del pixel correspondiente.
- En general, es posible obtener las **normales a la superficie** en todos los puntos visibles de la misma. Para ello se ilumina con tres fuentes de luz de colores rojo, verde y azul respectivamente, situadas sobre los ejes x , y , z . Si la reflexión es difusa, las componentes roja, verde y azul del color de un pixel son iguales a las componentes x , y , z de la normal unitaria en este punto.
- Conociendo las normales para todos los puntos de una superficie es trivial detectar la presencia de **aristas** como discontinuidades de la normal. El ángulo interno y la orientación de una arista pueden determinarse a partir de las normales a ambas caras de la misma. Ello permite aplicar las técnicas de alta frecuencia para análisis de difracción en aristas, como el método de las corrientes equivalentes.
- Pueden analizarse las **dobles reflexiones** entre superficies, según las aproximaciones de óptica geométrica para la primera reflexión y óptica física

para la segunda, mediante una técnica de trazado de rayos si se conoce la normal a todos los puntos de la superficie.

El método de procesamiento gráfico se ha aplicado para el cálculo de sección recta radar y análisis de reflectores en alta frecuencia, lo que se discute en detalle en los capítulos 5 y 6 de esta Tesis.

3.4 GENERACION DE PIXELS

En esta sección y la siguiente se estudiará el subsistema *back-end* del acelerador gráfico. Tal como se indica en la figura 3-1, este subsistema determina los pixels cubiertos por cada polígono y escribe la información correspondiente en la memoria de pantalla, eliminando simultáneamente los pixels ocultos mediante la técnica del *z-buffer*.

3.4.1 Memoria de pantalla y tabla de colores

La memoria de pantalla, o *frame-buffer*, es una porción de memoria RAM que constituye la salida del acelerador gráfico. Para cada pixel de la pantalla gráfica, el *frame-buffer* contiene la información de color. Se denomina número de planos (o profundidad) del *frame-buffer* al número de bits utilizados para codificar el color de cada pixel. Tal como se muestra en la figura 3-11, el número que indica el color de cada pixel es un índice a una "tabla de colores" que contiene las proporciones de rojo, verde y azul (R,G,B) de este color.

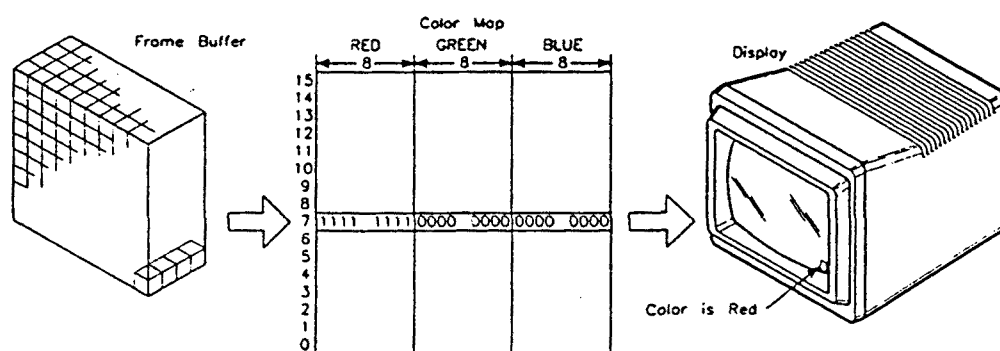


Figura 3-11: Memoria de pantalla y tabla de colores, para una profundidad de 4 planos de color. En una imagen pueden indexarse simultáneamente 16 colores de una paleta de 2^{24} colores posibles.

Las profundidades más habituales del *frame-buffer* son 4, 6 ó 8 planos, que permiten representar simultáneamente 16, 64 ó 256 colores. Para aumentar la profundidad de la pantalla a 24 planos sin incrementar excesivamente el tamaño físico de la memoria, las estaciones gráficas más potentes tienen 3 *frame-buffer* de 8 bits cada uno. Cada uno de los *frame-buffer* contiene las proporciones de rojo, verde y azul del color de cada pixel. La mezcla de las tres componentes del color se realiza mediante la técnica de *image-blending*: se muestra el contenido de los tres *frame-buffer* alternativamente, cada uno de ellos con el color rojo, verde o azul correspondiente, de forma que el ojo humano realiza la mezcla de colores integrando en el tiempo lo que ve en la pantalla.

3.4.2 Generación de pixels

La información de salida del subsistema *front-end* es típicamente un conjunto de polígonos, líneas o puntos especificado mediante las coordenadas de pantalla y las componentes (R,G,B) del color de sus vértices o extremos. El subsistema de generación de pixels (*back-end*) crea la imagen en la memoria de pantalla correspondiente a cada una de estas primitivas, determinando qué pixels están contenidos en su interior y cuál es el color cada uno de estos pixels.

La implementación en hardware más eficiente se consigue mediante arquitecturas de tareas distribuidas (*pipeline*) en las que varios procesadores realizan tareas en cadena. Existen dos posibles modelos de *pipeline* según el orden en que se generan los pixels:

- **Siguiendo el orden de los objetos:** Se procesan las primitivas en el mismo orden en que aparecen en la salida del subsistema *front-end*. Para cada una de ellas se generan los pixels contenidos en su interior. Para representar estos pixels en la pantalla, como aparecen desordenadamente (según el orden de las primitivas) es necesario almacenarlos en una memoria intermedia: el *frame-buffer* o memoria de pantalla. Esta técnica puede utilizarse con los métodos de eliminación de superficies ocultas siguiendo el orden de los objetos (*z-buffer*, orden según profundidad y algoritmos de árbol BSP), que se explican en la sección 3.5.
- **Siguiendo el orden de la pantalla:** Se generan los pixels en el mismo orden en que aparecen en la pantalla, lo cual evita la necesidad de almacenarlos en el *frame-buffer*: pueden enviarse directamente al conversor digital/analógico para la formación de la señal de video. Para ello es necesario asociar a cada línea de pantalla la lista de primitivas cortadas por esta línea, ordenadas de izquierda a derecha.

En la actualidad las estaciones gráficas más potentes utilizan la primera de estas dos técnicas, por lo que es la única que estudiaremos en esta sección.

Arquitecturas distribuidas siguiendo el orden de los objetos

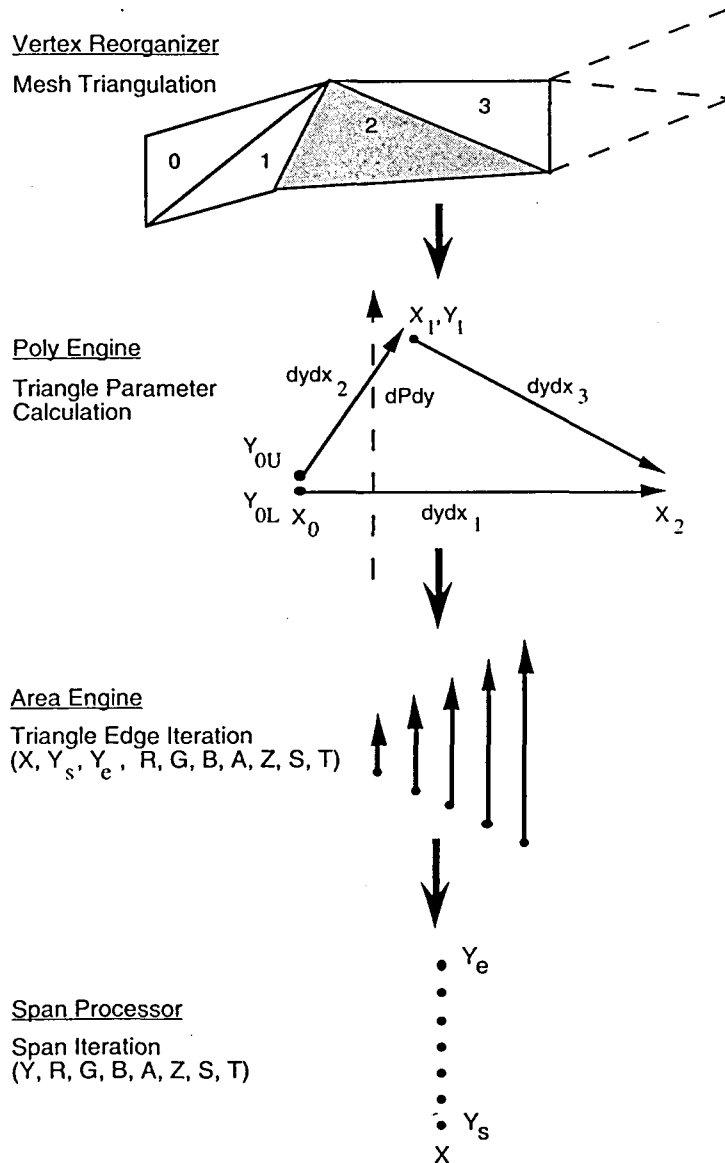


Figura 3-12: Proceso de generación de pixels para una malla de triángulos

Como se ha indicado anteriormente, esta familia de algoritmos obtiene la posición (x,y,z) y el color (R,G,B) de los pixels procesando ordenadamente las primitivas que salen del *front-end*. En la figura 3-12 se muestran las cuatro etapas del algoritmo de generación de pixels más utilizado para representar polígonos convexos. Cada una de estas etapas

suele tener un procesador dedicado, formando así un sistema de arquitectura distribuida con estructura *pipeline*, como se muestra en la figura 3-13.

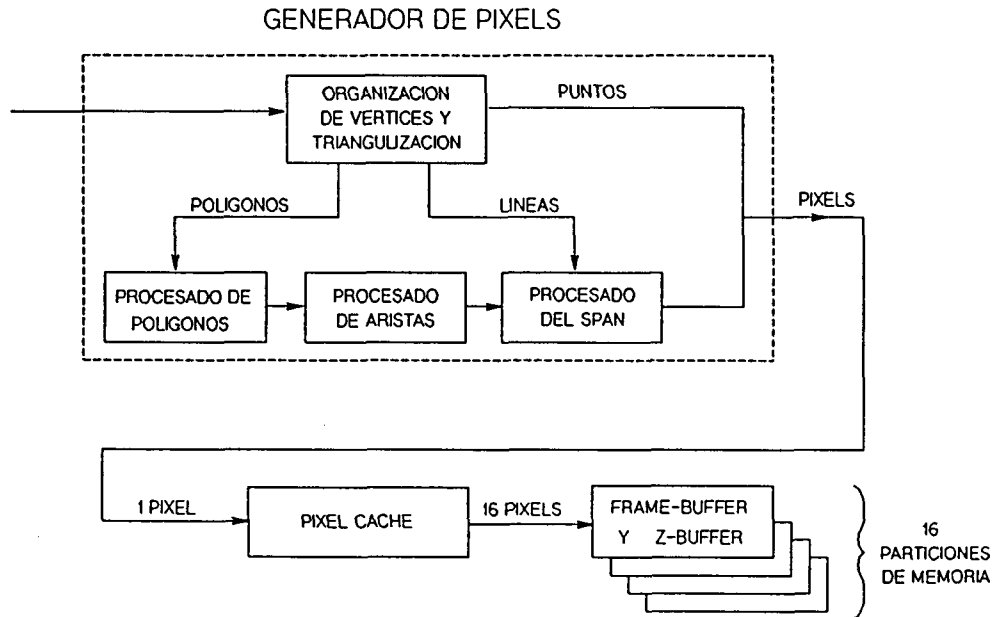


Figura 3-13: Estructura pipeline del generador de píxeles. El pixel cache sincroniza un enlace serie de alta velocidad proveniente del generador de píxeles con un enlace paralelo de baja velocidad hacia la memoria de pantalla.

- **Organización de vértices y triangulación:** La primera etapa separa la información del *front-end* en polígonos, vértices o puntos y los envía a una de las siguientes etapas. En el caso de polígonos de más de tres lados, los divide en un conjunto de triángulos adyacentes, como se muestra en la figura 3-12 (a).
- **Procesado de polígonos (*Poly engine*):** Los cálculos que se realizan una única vez por polígono se agrupan en esta etapa. El contorno del polígono se divide en aristas y se calcula la variación de cada una de las coordenadas y , z , R , G y B a lo largo de cada arista para cada incremento de un pixel en la dirección x (figura 3-12 a). A estos valores se les denomina muy a menudo "pendientes" (*slopes*).
- **Procesado de aristas (*Area engine*):** Esta etapa incluye los cálculos que se realizan sólo una vez para cada línea de pantalla contenida dentro del polígono (denominada *span*). Como se muestra en la figura 3-12 (c), para cada x (línea vertical) se obtienen las coordenadas y , z , R , G y B de los puntos inicial y final. Para ello se utilizan las pendientes calculadas en la etapa anterior. A su vez, se calculan las pendientes de variación de estos parámetros a lo largo de cada *span*, para incrementos de un pixel en la dirección vertical.

- **Procesado del span:** Las operaciones realizadas sobre cada pixel del *span* se agrupan en este procesado. Para cada pixel se calculan los valores *z*, *R*, *G* y *B* añadiendo las pendientes a los valores calculados para el pixel inmediatamente anterior. El valor *z* se compara con el valor previo almacenado en esta posición de memoria y sólo en el caso de que sea menor el nuevo valor de pixel reemplaza al anterior (algoritmo del *z-buffer*, sección 3.5).

Los sistemas distribuidos que asignan un procesador a cada uno de los cuatro pasos anteriores generan las imágenes mucho más rápidamente de lo que puede aceptar una memoria *frame-buffer* estándar. Por ejemplo, el acelerador gráfico VGX de las estaciones POWERVISION de Silicon Graphics es capaz de generar hasta 200 millones de pixels por segundo, logrando una velocidad muchas veces más rápida que la escritura en un sistema de memoria VRAM típico.

Para evitar el cuello de botella supone el acceso a la memoria del *frame-buffer*, se realizan escrituras o lecturas en paralelo a un conjunto de pixels, típicamente 16. Para ello debe utilizarse una memoria intermedia (*pixel cache*) que acepte los pixels generados a alta velocidad y de uno en uno por el generador de pixels y los escriba de 16 en 16 en el *frame-buffer*, a una velocidad más baja (figura 3-13).

El diseño de las memorias *cache* se hace, en general, considerando que los nuevos accesos a memoria son más probables sobre direcciones próximas a las del acceso anterior. Por tanto, si los accesos siguen secuencias muy erráticas se degradan las prestaciones del *cache*. En la visualización de polígonos puede preverse con mucha precisión a que región del *frame-buffer* se va a acceder en cada momento, ya que se conocen la situación del polígono en la pantalla y el orden de generación de los pixels. De esta forma, el controlador de memoria *cache* puede trabajar muy eficientemente empezando a leer el siguiente bloque de pixels del *frame buffer* mientras se procesa el bloque previo.

3.5 ELIMINACION DE PARTES OCULTAS

Una fase muy importante en la generación de imágenes realistas es la eliminación de las partes ocultas o eclipsadas de los objetos. La eliminación de las caras ocultas de los objetos -que se encuentran en la parte trasera, y por tanto, no visible- se realiza dibujando únicamente aquellos polígonos tales que el producto escalar de la normal exterior al polígono por la dirección de observación sea positivo. Esto se realiza al principio del *pipeline* de transformaciones geométricas en la etapa de "aceptación y rechazo trivial" (ver figura 3-4).

La eliminación de las superficies eclipsadas por otros objetos es más complicada, y requiere algoritmos especializados. Estos se dividen en dos clases, según sean independientes o dependientes del dispositivo de salida. Los primeros eliminan las partes ocultas en el espacio del objeto, antes de generar la imagen, mientras que los segundos trabajan sobre la propia imagen [Brunet'90]. Estos últimos al ser dependientes del dispositivo de salida, optimizan los recursos disponibles y son más eficientes que los primeros.

Los algoritmos de eliminación de superficies ocultas también pueden clasificarse en si requieren una ordenación previa de los polígonos o no:

- **Algoritmos de lista de prioridades:** La mayoría de los algoritmos se basan en una ordenación de todos los polígonos según su distancia al observador: la imagen se genera dibujando en primer lugar los polígonos más alejados, y sobre ellos los más cercanos ("algoritmo del pintor"). Para ello es necesario un preprocesado de la escena, anterior a la visualización de la imagen. Ejemplos típicos de estos algoritmos son el árbol de partición binaria de espacio y el algoritmo de ordenación según la profundidad.

- **Algoritmos sin ordenación previa:** El algoritmo de *z-buffer*, muy utilizado en la actualidad, no requiere de una ordenación previa de los objetos a visualizar: procesa los algoritmos en cualquier orden y realiza la eliminación de superficies ocultas pixel a pixel, directamente sobre el *frame-buffer*.

a) Algoritmo del z-buffer

El algoritmo del z-buffer desarrollado por Catmull es una de las técnicas de visualización de superficies más sencillas de implementar, tanto en software como en hardware. Este algoritmo requiere que en el ordenador esté disponible el *z-buffer*, una porción de memoria de pantalla que contiene la coordenada z, o distancia al observador, de cada pixel.

El proceso de eliminación de superficies ocultas es el siguiente: Al almacenar los pixel de salida del generador de pixels en el *frame-buffer* se compara el valor de la coordenada z del nuevo pixel con la del pixel ya existente en esta posición de pantalla, almacenada previamente en el *z-buffer*. Si la nueva z es menor que la antigua, el nuevo pixel está más cerca del observador, y se visualiza actualizando los valores del *frame-buffer* (color) y *z-buffer* (profundidad). Si la nueva z es mayor que la antigua, el nuevo pixel está más lejos del observador y se descarta (no se visualiza).

Dado que el algoritmo del *z-buffer* trabaja en el espacio de la imagen es un método fuertemente dependiente del dispositivo de representación gráfica. La principal ventaja

de este algoritmo frente a otros independientes del dispositivo de salida es que no precisa una ordenación previa de los polígonos y superficies de la escena, pudiéndose dibujar éstos en cualquier orden. Por ello, el coste computacional de dicho algoritmo es mucho menor, especialmente cuando el número de facetas en el modelado de los objetos es muy elevado, ya que el tiempo de cálculo adicional para eliminar las superficies ocultas tiende a ser independiente del número de polígonos.

Por otra parte, este algoritmo padece un serio inconveniente: la gran cantidad de memoria que requiere. Por ejemplo, un *z-buffer* con 24 bits para almacenar la información de profundidad de todos los pixels de una pantalla de 1024 x 1024 puntos ocupa 3 Megabytes de memoria de pantalla.

La gran simplicidad de este método y el no requerir de estructuras de datos especiales para la representación en escena junto con el decrecimiento de los costes de la memoria han repercutido en el desarrollo de numerosas implementaciones, tanto hardware como firmware, de algoritmos de *z-buffer*.

Además, la información contenida en el *z-buffer* es útil incluso después de la visualización de la imagen. El programa *GRECO* de cálculo de RCS en tiempo real utiliza la coordenada *z* de cada punto del objeto para obtener la distancia de este punto al observador, y así sumar coherentemente la contribución de todos los puntos de la superficie con la fase adecuada (ver capítulo 5).

b) Algoritmos de listas de prioridades

Los algoritmos de listas de prioridades determinan un orden de visualización para los objetos asegurando la obtención de una imagen correcta siempre que los objetos se visualicen en el orden preestablecido. Si los objetos no se intersectan entre sí y, por tanto, unos quedan totalmente por detrás de los otros es sencillo determinar un orden de visualización. Sin embargo, cuando en la imagen existen objetos que se intersectan entre sí y, por tanto, hay zonas de la imagen en las que unos quedan por delante de los otros y otras zonas en las que se produce el caso contrario, no es posible una ordenación válida de los objetos a visualizar. En estos casos la solución es subdividir estos objetos para crear un orden lineal de visualización.

Existen distintos algoritmos de visualización por lista de prioridades, los cuales se diferencian unos de otros en cómo determinan el orden, en cómo determinan los objetos a subdividir y en qué instante se produce la subdivisión. A continuación, se comentan los más utilizados.

Algoritmo de ordenación según la profundidad

La idea básica de este algoritmo desarrollado por Newell y Sacha es visualizar los objetos en el *frame-buffer* en orden decreciente de la distancia respecto al punto de vista. Son necesarios los tres pasos siguientes:

- Ordenar todos los polígonos en orden decreciente de la coordenada z.
- Solucionar las posibles ambigüedades que puedan aparecer entre polígonos que se intersectan, subdividiéndolos cuando sea necesario.
- Visualizar los objetos siguiendo el orden preestablecido por la lista.

Este método se conoce como el *algoritmo del pintor* ya que, al igual que los pintores, los objetos más cercanos se visualizan sobre los más alejados.

Método de partición binaria del espacio

El algoritmo de árbol de partición binaria del espacio (BSP), desarrollado por Fuchs, Kedem y Naylor, es un método extremadamente eficiente para visualizar un conjunto estático de polígonos en tres dimensiones desde un punto de vista arbitrario. Este método consigue un algoritmo de visualización lineal para cualquier punto de vista de la escena, gracias a una fase de preprocesado anterior a la visualización. Ello hace que sea un algoritmo apropiado para aplicaciones en las que el ángulo de vista cambie, pero los objetos no. Un ejemplo típico son los simuladores de vuelo.

Este método requiere que la escena tridimensional haya sido modelada con polígonos. Para determinar el orden correcto de visualización se utiliza la técnica de partición binaria del espacio (BSP), presentada en la sección 2.2.4 d). La raíz del árbol binario es un polígono elegido de entre los que hay que visualizar. El algoritmo trabaja bien sea cual sea el polígono elegido como raíz, pero es más eficiente si se elige un polígono situado en el centro de la escena.

Este polígono raíz divide la escena en dos subespacios: uno de ellos contiene todos los polígonos que están por delante del polígono raíz, según la dirección de la normal, mientras que el otro contiene los que quedan detrás de él. Cualquier polígono intersectado por el plano del polígono raíz se divide por la línea de intersección y los dos polígonos resultantes se asignan al subespacio apropiado. Los polígonos hijos de la raíz son dos polígonos cualesquiera, situados en cada uno de los dos subespacios generados anteriormente. Igual como se ha realizado con el nodo raíz, estos nuevos nodos subdividen el subespacio asociado a cada nodo en las partes anterior y posterior al polígono del nodo. El algoritmo termina cuando el subespacio asociado a cada nodo

contiene un único polígono. La figura 3-14 muestra la generación de un árbol binario a partir de una escena formada por polígonos.

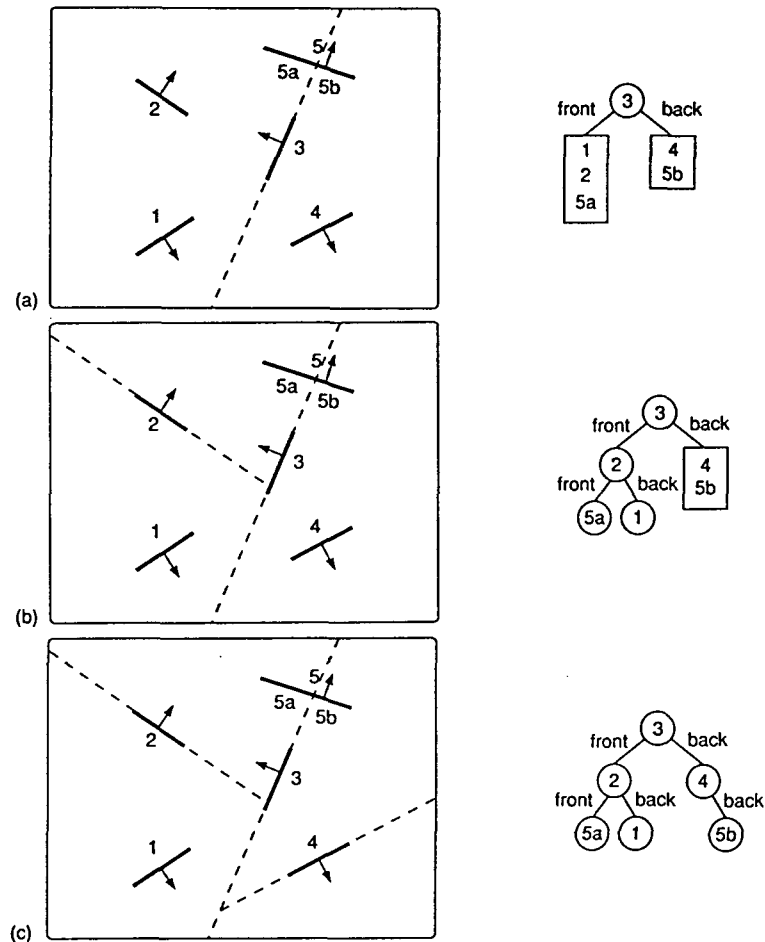


Figura 3-14: Arbol de partici3n binaria del espacio (BSP): a) Primera subdivisi3n con el pol3gono 3 como ra3z. b) Subdivisi3n de la rama izquierda. c) Subdivisi3n de la rama derecha (3rbol completo).

Para obtener la lista de pol3gonos con el orden adecuado a cada punto de vista, se recorre el 3rbol de la siguiente forma: En cada uno de los nodos del 3rbol se visualiza primero la rama correspondiente al subespacio m3s alejado del observador, despu3s el pol3gono del nodo, y despu3s la rama correspondiente al subespacio en el que est3 situado el observador. Este criterio permite recorrer todo el 3rbol visualizando los pol3gonos en el orden adecuado.

3.6 HARDWARE DEL ACELERADOR GRAFICO

El diagrama de bloques para visualización de imágenes que se ha presentado en las secciones anteriores es un modelo lógico que no coincide necesariamente con un modelo físico, ya que las etapas del diagrama de bloques o *pipeline* pueden implementarse tanto en software como en hardware.

En la figura 3-1 se ha mostrado el diagrama de bloques del conjunto de funciones necesarias para la representación de imágenes realistas por ordenador. A continuación, se van a comentar las soluciones hardware que se han desarrollado para implementar algunas de estas etapas y con ello acelerar el proceso de generación de imágenes realistas. Estos sistemas hardware se conocen con el nombre genérico de "acelerador gráfico".

La visualización de imágenes muy complejas, formadas por un número muy elevado de polígonos y superficies paramétricas, requiere un coste computacional excesivo para ser llevada a cabo por un sólo procesador. La única forma de obtener la imagen en tiempo real es mediante procesamiento concurrente, en el que múltiples operaciones son realizadas simultáneamente por distintos procesadores. Este procesamiento concurrente puede implementarse en dos formas básicas: el *pipelining* o distribución y el paralelismo, como se muestra en la figura 3-15.

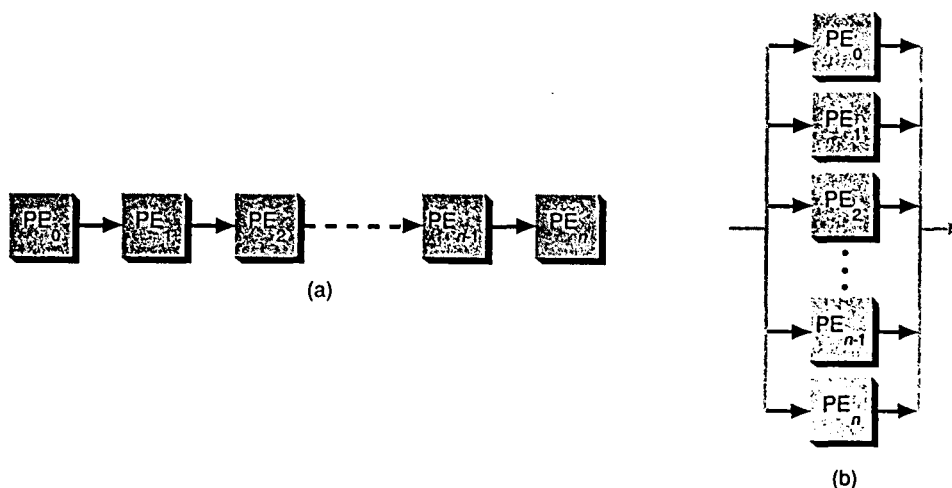


Figura 3-15: Formas básicas de procesamiento concurrente: a) *pipelining* o distribución, b) paralelismo.

- Un procesador distribuido (*pipeline processor*) contiene un número de elementos de procesamiento (*PE*) dispuestos de forma que la salida de uno de ellos es la entrada del siguiente, figura 3-15 (a). El proceso se distribuye en varias

etapas de forma que los mismos datos son procesadas secuencialmente por elementos de cálculo independientes (*PE*).

- En una estructura distribuida en paralelo todos los procesadores (*PE*) realizan simultáneamente las mismas operaciones sobre porciones distintas de los datos, figura 3-15 (b). Para ello es necesario repartir los datos en conjuntos iguales a la entrada de los procesadores y recombinar la salida de los mismos en un único flujo de datos.

Los procesadores paralelo y *pipeline* son las estructuras más utilizadas en el hardware de los aceleradores gráficos de altas prestaciones. Ambas técnicas pueden utilizarse para acelerar tanto el subsistema *front-end* como el *back-end* de un sistema gráfico (ver figura 3-1).

3.6.1 Subsistema *front-end*

El bloque *front-end* de un sistema de visualización gráfica tiene dos funciones principales: lectura del modelo a visualizar y transformación de las primitivas de alto nivel, especificadas en coordenadas del modelo, en un conjunto de primitivas de bajo nivel (polígonos, rectas y puntos) en coordenadas de pantalla.

Para conseguir las velocidades de visualización requeridas por las aplicaciones gráficas en tiempo real es necesario utilizar concurrencia para acelerar estos cálculos. Tanto la distribución de tareas como el paralelismo han sido usados durante décadas para diseñar el hardware del subsistema *front-end* de los sistemas gráficos de altas prestaciones:

- Como el procesado lógico del *front-end* es intrínsecamente de tipo distribuido (ver figura 3-4), cada etapa puede asignarse a una unidad hardware separada. Sin embargo, debe tenerse en cuenta que la heterogeneidad en las tareas que llevan a cabo las distintas etapas puede dar lugar a la aparición de "cuellos de botella" en las más lentas. Esta situación se evita paralelizando aquellas etapas que requieran un coste computacional más elevado.

- Por contra, la estructura paralela para el subsistema *front-end* presenta dificultades a la entrada y a la salida de los procesadores, pues la lectura y escritura serie de los datos y de los resultados del proceso no puede realizarse a la velocidad requerida por los procesadores, debido al limitado ancho de banda de las memorias. Aunque se han propuesto algunas soluciones basadas en la lectura paralelo de bases de datos compartidas y en el procesado paralelo de los datos de salida por el subsistema *back-end*, la complicación que esto supone desaconseja el uso del paralelismo.

Por estos motivos los sistemas gráficos actuales prefieren la estructura distribuida sobre la paralela. A continuación se va a comentar algunas de las soluciones hardware para la implementación de las distintas etapas del subsistema *front-end*:

- **Lectura de primitivas en las bases de datos:** Cuando un único procesador no es suficiente para leer toda la base de datos con la velocidad requerida, suele dividirse la base de datos y leerla en paralelo.

- **Transformaciones del modelo y de visualización:** Esta etapa incluye normalmente un número muy elevado de multiplicaciones de matrices y vectores para realizar cambios de coordenadas, traslaciones y giros sobre todas las primitivas del modelo, lo cual requiere una potencia de cálculo elevada.

Afortunadamente, las multiplicaciones de matrices y vectores son cálculos simples que pueden ser implementados fácilmente en hardware. Uno de los primeros procesadores de este tipo fue el multiplicador de matrices, el cual podía realizar el producto de un vector de 4 elementos por una matriz homogénea (4x4 en 3D) en 20 microsegundos. Sin embargo, algunos de los procesadores más recientes han explotado la potencia y programabilidad de los chips en coma flotante comerciales, como por ejemplo el acelerador gráfico Hewlett-Packard Turbo VRX que utiliza el procesador 860i de INTEL.

La implementación más común de esta etapa es la de un único procesador que transforma secuencialmente una serie de vértices, aunque recientemente las estaciones gráficas más potentes utilizan un número reducido de procesadores en paralelo. Algunas de las operaciones que se pueden paralelizar son: procesado paralelo de las cuatro componentes x, y, z, w de cada vértice, procesado paralelo de los tres vértices de cada triángulo y procesado de distintas primitivas en paralelo.

- **Aceptación y rechazo trivial:** La operación de aceptación o rechazo se puede implementar de forma muy directa, ya que en el peor de los casos se requiere un producto escalar (para eliminar polígonos que no están orientados hacia el observador) y en el mejor caso una comparación (para determinar en que lado de los planos de *clipping* se encuentra cada vértice). Como estas operaciones no son muy costosas se suelen realizar con el mismo procesador que realiza las transformaciones geométricas.

- **Iluminación:** Un procesador especializado puede calcular los colores de los vértices en función de la normal a la superficie y los parámetros del modelo de iluminación. Sin embargo, en las estaciones más modernas la tendencia es utilizar un procesador en coma flotante de propósito general y programable. En

sistemas de menores prestaciones los cálculos de iluminación los realiza el mismo procesador que calcula las transformaciones geométricas. Cuando se realiza suavizado de Phong los cálculos de iluminación se aplazan hasta la etapa de generación de los pixels (*rasterization*).

- **Recorte:** El proceso de recorte (*clipping*) debe eliminar los vértices de los polígonos que se encuentran fuera del volumen de visualización. La subdivisión de polígonos que se encuentran parcialmente fuera de este volumen siempre se había considerado un proceso muy engorroso, ya que el número de vértices a analizar puede cambiar durante el proceso de cálculo y los polígonos cóncavos pueden subdividirse en varios polígonos.

Sutherland y Hodgman [Suth'74] demostraron que cualquier polígono cóncavo o convexo arbitrario podía someterse a un proceso de clipping en un volumen convexo pasando los vértices a través de una única unidad de proceso múltiples veces. En cada paso por la unidad se estudiaba la posible intersección con un plano distinto.

En 1980, Clark propone descomponer este lazo de procesado en una secuencia simple de procesadores idénticos, cada uno de los cuales se compone de único chip especializado, denominando a esta distribución "*geometry engine*" [Clark'82]. Este *geometry engine* es un procesador suficientemente general para ser capaz también de transformar primitivas y obtener perspectivas.

Recientemente algunos procesadores de coma flotante de propósito general han empezado a reemplazar las unidades de transformación y procesadores especializados. Por ejemplo, Silicon Graphics, empresa que siempre había empleado procesadores "*custom*" en el subsistema *front-end*, en 1989 aplicó el chip Weitek 3332 tanto para las transformaciones como para recorte en sus sistemas POWER IRIS.

El proceso de *clipping* puede ser realizado bien por un único procesador que recorta cada polígono por los planos que limitan en volumen de visualización, o bien mediante un conjunto de procesadores en paralelo, uno para cada plano.

- **Perspectiva y proyección en la pantalla:** Igual como sucede con las etapas de iluminación y de transformaciones geométricas, los cálculos en esta etapa son sencillos pero requieren bastantes recursos por la gran cantidad de datos a procesar. Pueden implementarse mediante unidades especializadas o mediante un procesador en coma flotante de propósito general. En sistemas de muy altas prestaciones, estos cálculos los realiza un sistema distribuido de procesado concurrente.

3.6.2 Subsistema *back-end*

El subsistema *back-end* realiza básicamente dos funciones:

- Conversión de las primitivas de bajo nivel que constituyen la salida del *front-end* en una serie de pixels y procesamiento de los mismos.
- Almacenar los pixels en la memoria de pantalla.

A continuación se va a comentar muy brevemente el hardware que realiza estas funciones en las estaciones de trabajo más potentes.

a) Generación y procesamiento de pixels (*scan conversion*)

Los algoritmos de generación de pixels se han presentado en el apartado 3.4.2. Estos algoritmos han sido diseñados para permitir una fácil implementación en hardware, por lo que el diagrama de bloques de la arquitectura de procesamiento distribuido sigue fielmente al diagrama de flujo del algoritmo (figuras 3-12 y 3-13).

El proceso de generación y procesamiento de los pixels suele dividirse en cuatro unidades hardware:

- **Buffer de datos:** Cuando el subsistema *front-end* utiliza varios procesadores en paralelo, es necesario un buffer de datos a la entrada de *back-end* para uniformar el flujo de salida del *front-end* y eliminar tiempos de espera en el generador de pixels. Este buffer utiliza una cola de prioridad que garantiza el orden correcto en la visualización de las primitivas.
- **Generación de pixels:** Las cuatro etapas de generación de pixels (ver apartado 3.4.2) suelen implementarse en un circuito integrado VLSI de tipo *custom*, con lo que se consiguen prestaciones del orden de 16 millones de pixels generados por segundo. Para conseguir un rendimiento todavía mayor, las estaciones POWERVISION de Silicon Graphics utilizan varios procesadores en paralelo de tipo MIMD (*Multiple-Instruction Multiple-Data*), con lo que consiguen generar hasta 200 millones de pixels por segundo.
- **Procesado de pixels:** La salida del generador de pixels es post-procesada por otro circuito integrado *custom* que añade prestaciones adicionales, como por ejemplo: *dithering* (tramas de color simulado), transparencia (eliminando tramas de pixels en superficies semitransparentes), corrección gamma (para adaptar la

respuesta del CRT al ojo humano), sistema de ventanas (solapamiento, recorte, etc.) y *antialiasing* (ver sección 3.1.b).

- **Buffer de pixels:** Como los pixels son generados y procesados a ráfagas, mientras que la velocidad de escritura de pixels en la memoria de pantalla es constante, para evitar tiempos de espera a la entrada del *pixel cache* (ver figura 3-13) es necesario colocar un *buffer* en forma de cola FIFO que uniformice el flujo de pixels.

b) Memoria de pantalla (*Raster Engine*)

Como se ha indicado en la sección 3.4.2, la velocidad de generación de pixels es mucho mayor que la velocidad de escritura en memoria de pantalla. Ello obliga a realizar la escritura de varios pixels en paralelo, por lo que es necesario convertir el flujo de pixels en serie a uno paralelo. Esta operación la realiza el *pixel cache*, que habitualmente se implementa en un circuito integrado *custom*.

Para escribir los pixels en paralelo en la memoria de pantalla es necesario dividir ésta en bloques pequeños, usualmente de 16 pixels. Además, la manipulación de esta memoria se realiza por un conjunto de procesadores en paralelo que tienen asignada una región de la pantalla. Estas particiones pueden realizarse mediante bloques contiguos (partición continua) o bloques entrelazados (partición entrelazada).

Ejemplos típicos de partición son bloques contiguos de 16x1 ó 4x4 pixels a la salida del *pixel cache* y 20 ó 40 procesadores en paralelo que controlan bloques entrelazados de 64 kilopixels.

La información contenida en cada pixel de la memoria de pantalla está almacenada en una serie de bits asignados a este pixel. El conjunto de los bits en todos los pixels de la pantalla que almacenan un mismo tipo de información constituye un "plano". Los distintos planos que existen en la memoria de pantalla de las estaciones más potentes son:

- **Planos de imagen:** Almacenan 24 bits con la información del color asociado a cada pixel, 8 bits para cada componente R,G,B. Si se utiliza la técnica *antialiasing*, también existen 8 bits con la componente α de cada pixel. Para poder utilizar la técnica del *doble-buffer*, algunas estaciones tienen dos bancos de 24 ó 32 planos de imagen, totalizando 48 ó 64 planos.

- **Planos de profundidad (*z-buffer*):** Contienen la coordenada z o profundidad de cada pixel, necesaria para la eliminación de superficies ocultas por la técnica del *z-buffer*. Habitualmente se utilizan 16 ó 24 planos de profundidad.

- **Planos de textura:** Contienen la información necesaria para representar texturas (ver sección 3.1.c). Suelen haber entre 8 planos de textura, para modelar únicamente la intensidad de la textura, y 32 planos para modelar las 4 componentes R,G,B y α de la textura con 8 bits cada una.

- **Planos de *overlay*:** Se utilizan para representar primitivas ajenas a la imagen y superpuestas a la misma, como por ejemplo marcos de ventanas, menús desplegables, el cursor del ratón, etc. Son habituales entre 4 y 8 planos de este tipo.

- **Planos de ventana:** En algunas estaciones este conjunto de planos contiene la información que refleja el estado de cada pixel en relación con el sistema de ventanas (a qué ventana pertenece y el tipo de la misma). Las estaciones Silicon Graphics POWERVISION emplean 2 conjuntos de 4 planos de ventana.

Las configuraciones más sofisticadas poseen planos de bits adicionales para realizar funciones avanzadas de textura, deformaciones, transparencias, convolución, etc., así como paralelismo para acelerar la representación de la imagen. El número de planos de las estaciones gráficas más potentes puede llegar hasta los 268 bits por pixel del acelerador VGXT de Silicon Graphics.

Como ejemplo hardware gráfico la figura 3-16 muestra la arquitectura del acelerador VGX de Silicon Graphics. En la figura pueden apreciarse los distintos bloques que se han estudiado en esta sección:

- ***Geometry subsystem:*** Corresponde al subsistema *front-end*. Obsérvese la estructura distribuida de procesado concurrente. Cada una de las etapas contiene cuatro procesadores de 32 Mflops con memoria local que procesan cuatro vértices en paralelo.

- ***Scan conversion:*** Constituye la primera parte del subsistema *back-end*, dedicada a la generación de pixels. Presenta una estructura distribuida en las fases de procesado de polígonos y aristas, y paralela en la fase de procesado del *span*.

- ***Raster subsystem:*** Es la segunda parte del *back-end*. El procesado de los pixel de pantalla se realiza en paralelo por 20 procesadores IE (*Image Engines*), sobre una arquitectura del *frame-buffer* con partición de memoria entrelazada. Este procesado de los pixel incluye la técnica del *z-buffer* para eliminación de partes ocultas y la representación de texturas.

- ***Display subsystem:*** Es el subsistema de visualización. Está formado por las tablas de color, los conversores digital a analógico y el monitor.

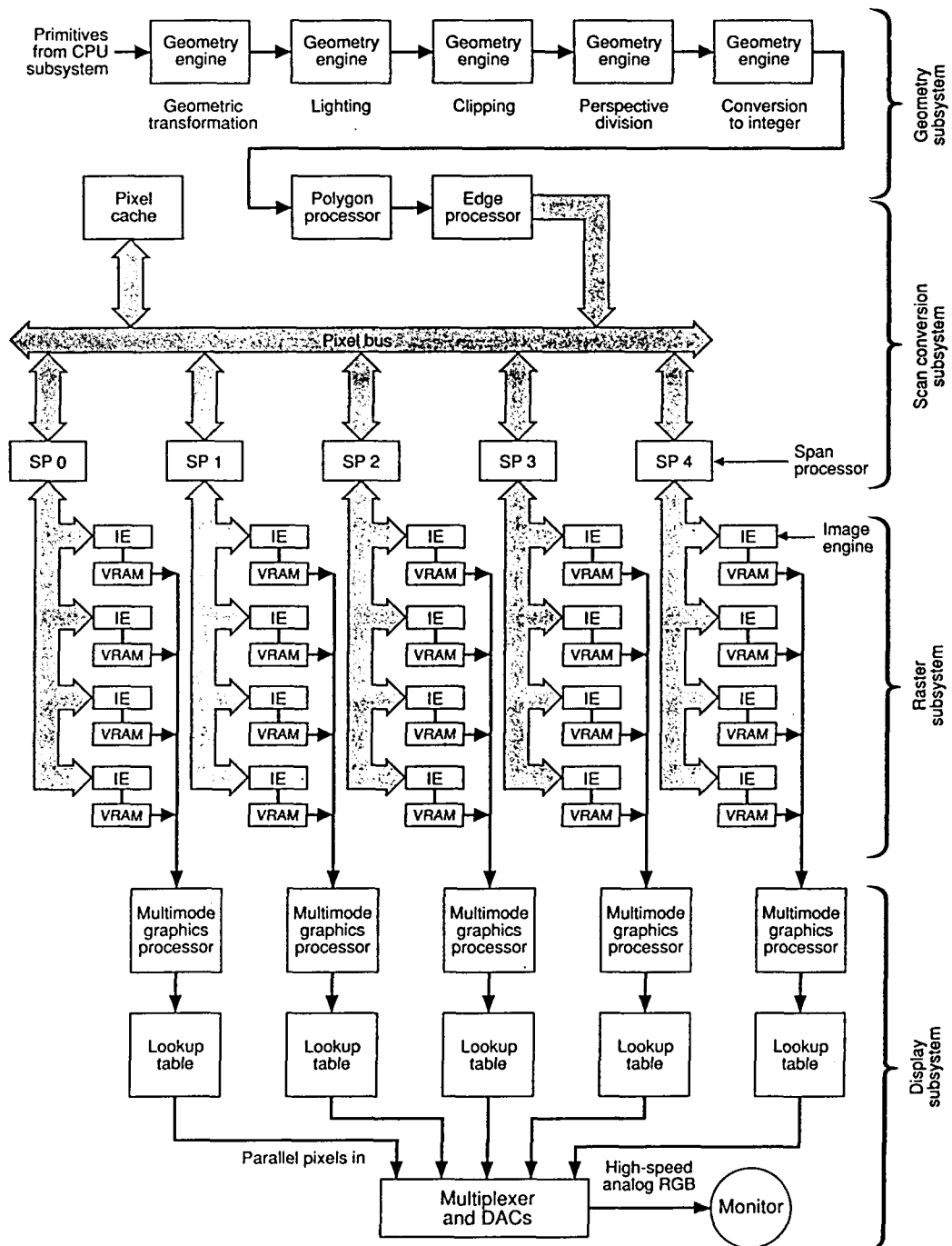


Figura 3-16: Arquitectura del acelerador gráfico VGX de Silicon Graphics

CAPITULO 4

MODELADO DE OBJETOS PARA ANALISIS NUMERICO

La solución analítica del problema electromagnético solamente es posible para algunas geometrías muy particulares, llamadas canónicas, en las que las condiciones de contorno se aplican a superficies con alguna coordenada constante en un sistema de coordenadas en el que la ecuación de onda sea separable. Además, el medio comprendido entre estas superficies debe ser homogéneo. Ejemplos típicos son las soluciones modales para guías de onda rectangulares o cilíndricas y la difracción de una onda incidente por objetos cilíndricos o esféricos infinitos.

Como los objetos arbitrarios, representados por un modelo geométrico a los que se dedica esta tesis, en general no responden a este tipo de geometría, no es posible una solución analítica exacta y es necesario recurrir a la solución numérica del problema. Ella suele realizarse a través de los siguientes pasos:

- 1) **Formulación del problema electromagnético:** En general, se parte de las ecuaciones de Maxwell, de la ecuación de onda o de algunas ecuaciones integro-diferenciales derivadas de ellas.
- 2) **Discretización de la formulación matemática:** Como el cálculo numérico mediante ordenador sólo puede procesar números y no funciones, es necesario discretizar en vectores numéricos todas las funciones y los resultados de aplicar los operadores integro-diferenciales sobre las funciones. Así se transforma un sistema de ecuaciones funcionales en un sistema de ecuaciones algebraicas de dimensión finita, que puede resolverse mediante un ordenador.
- 3) **Discretización de las condiciones de contorno:** Sólo es posible imponer exactamente las condiciones de contorno cuando la geometría del problema tiene formas canónicas. En general, es necesario discretizar la superficie, procesando

el modelo geométrico del objeto, y además imponer de forma aproximada las condiciones de contorno sobre el modelo.

- 4) Finalmente, en los problemas abiertos de análisis de antenas (se busca la solución en el exterior de la superficie y no en el interior) es necesario asegurar que los campos obtenidos en la solución cumplan la "condición de radiación de Sommerfeld". Esta condición impone a los campos que se propaguen desde la fuente hacia el exterior, y no a la inversa, y que se atenúen con la distancia de la forma adecuada, conservando la potencia total del frente de onda.

En este capítulo se estudia la aplicación del modelado geométrico a los métodos numéricos para análisis electromagnético. Se consideran 3 métodos numéricos, que precisan modelos geométricos muy distintos:

- Método de los elementos de contorno (BEM), que utiliza modelado por fronteras y descomposición mediante celdas de la frontera.
- Método de los elementos finitos (FEM), basado en la representación mediante partición del espacio y descomposición en celdas volumétricas.
- Método del gradiente conjugado con transformada rápida de Fourier (CG+FFT), que utiliza enumeración de ocupación del espacio. Los modelos generados para este método de análisis también son útiles para el método de diferencias finitas en el dominio del tiempo (FDTD).

A continuación se comentan estas tres aplicaciones con una breve introducción a la teoría de cada método.

4.1 METODO DE LOS ELEMENTOS DE CONTORNO

El método más utilizado para el análisis de difracción en obstáculos y de radiación de antenas es el método de los momentos. Este método se basa en la resolución de una ecuación integral obtenida a través del principio de equivalencia superficial. Como las integrales de la ecuación a resolver se extienden únicamente sobre la superficie de los objetos y no sobre el volumen interior de los mismos, solamente es necesario discretizar esta superficie, lo cual constituye una ventaja sobre los métodos que precisan discretizar todo el volumen.

El método de los elementos de contorno se basa en la formulación integral del método de los momentos, pero utiliza una discretización superficial en subdominios utilizando funciones base polinómicas, iguales a las que utiliza el método de los elementos finitos (ver sección 4.2). Por este motivo, algunos autores consideran al método de los

elementos de contorno como un caso particular del método de los momentos, con funciones base polinómicas, mientras que para otros es un caso particular del método de los elementos finitos, con formulación integral. Nosotros, para no entrar en esta controversia, hemos preferido presentarlo como un método independiente.

4.1.1 Formulación

a) Equivalencia superficial

El teorema de equivalencia permite formular un problema equivalente al problema original, en el que unas corrientes equivalentes (no reales) radian en el espacio libre el mismo campo que existe en el problema real. Ello permite utilizar en las integrales de radiación funciones de Green del espacio libre, con lo que se simplifican enormemente los cálculos. Cuando existen condiciones de contorno canónicas, en lugar de la función de Green en espacio libre puede utilizarse la radiación de una fuente puntual en presencia de estas condiciones de contorno. De esta forma se asegura que los campos obtenidos cumplirán las condiciones de contorno canónicas sin necesidad de imponerlas explícitamente.

Existen dos tipos de equivalencia: "la equivalencia volumétrica" y la "equivalencia superficial" [Cardama'93]. El método de los elementos de contorno resulta de aplicar la equivalencia superficial. Dicha equivalencia solamente puede aplicarse cuando el objeto está formado por un medio homogéneo a trozos. En dicho caso se utiliza una formulación del teorema de equivalencia en la que las corrientes equivalentes se definen únicamente sobre las superficies de separación entre las regiones homogéneas del medio.

En el problema equivalente superficial existen dos conjuntos de fuentes, las cuales permiten separar los campos \vec{E}, \vec{H} en sus componentes "incidente" \vec{E}^i, \vec{H}^i y "dispersada" \vec{E}^s, \vec{H}^s debidas, respectivamente, a las fuentes originales y a las fuentes equivalentes, radiando ambas en espacio libre en lugar de en presencia del objeto. Los campos dispersados $\vec{E}^s = \vec{E} - \vec{E}^i$ y $\vec{H}^s = \vec{H} - \vec{H}^i$ pueden calcularse, en el exterior de la superficie S, a través de la radiación de las fuentes equivalentes \vec{J}_{eqS} y \vec{M}_{eqS} en el espacio libre

$$\vec{E}^s = L_E(\vec{J}_{eqS}, \vec{M}_{eqS}) \quad \vec{H}^s = L_H(\vec{J}_{eqS}, \vec{M}_{eqS}) \quad (4.1)$$

donde L_E y L_H son operadores lineales basados en la convolución de las corrientes con la función de Green [Cardama'93].

Como en el problema equivalente el campo total es nulo en el interior de la superficie S, las condiciones de contorno sobre dicha superficie son [Cardama'93]

$$\begin{aligned}\vec{J}_{eqS} &= \hat{n} \times \vec{H} = \hat{n} \times (\vec{H}^s + \vec{H}^i) = \hat{n} \times L_H(\vec{J}_{eqS}, \vec{M}_{eqS}) + \hat{n} \times \vec{H}^i \\ \vec{M}_{eqS} &= -\hat{n} \times \vec{E} = -\hat{n} \times (\vec{E}^s + \vec{E}^i) = -\hat{n} \times L_E(\vec{J}_{eqS}, \vec{M}_{eqS}) - \hat{n} \times \vec{E}^i\end{aligned}\quad (4.2)$$

La resolución de este sistema de ecuaciones funcionales permite obtener el valor de las incógnitas \vec{J}_{eqS} , \vec{M}_{eqS} . En el caso particular de superficies perfectamente conductoras (PEC), $\vec{M}_{eqS} = -\hat{n} \times \vec{E} = 0$ y basta resolver una de las dos ecuaciones con incógnita \vec{J}_{eqS} .

b) Análisis de cuerpos con regiones homogéneas

El teorema de equivalencia superficial tiene muchas aplicaciones prácticas como las que se van a considerar a continuación. Sin embargo, cualquier objeto que se analice por este método debe poder ser aproximado por un medio homogéneo a trozos. En este caso se aplica el principio de equivalencia superficial a las superficies conductoras y a las interfaces de separación entre regiones dieléctricas homogéneas. Solamente en el caso de existir regiones dieléctricas heterogéneas es imprescindible una formulación mixta de equivalencia volumétrica y equivalencia superficial.

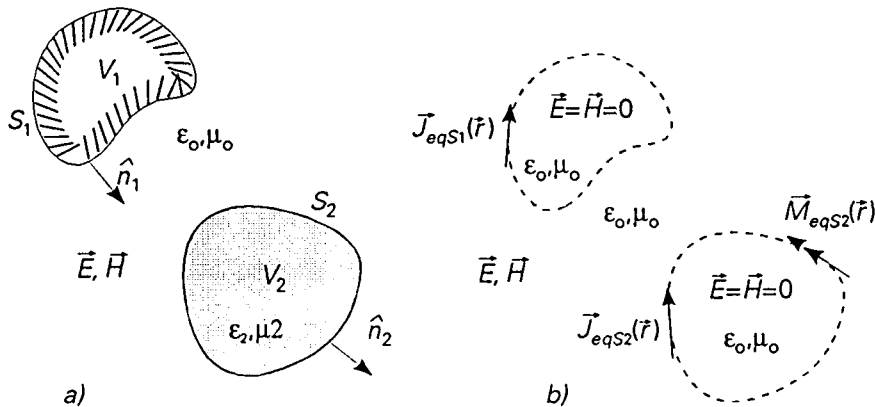


Figura 4-1: Teorema de equivalencia con regiones conductoras y dieléctricas homogéneas a trozos: a) problema original, b) problema equivalente superficial-superficial

En el caso de tener regiones dieléctricas homogéneas y zonas de conductor perfecto se tendrá que resolver una de las dos ecuaciones definidas sobre la superficie conductora S_1

$$\begin{aligned}\vec{M}_{eqS_1} = 0 &= -\hat{n} \times [\vec{E}^i + L_E(\vec{J}_{eqS_1}) + L_E(\vec{J}_{eqS_2}, \vec{M}_{eqS_2})] \\ \vec{J}_{eqS_1} &= \hat{n} \times [\vec{H}^i + L_H(\vec{J}_{eqS_1}) + L_H(\vec{J}_{eqS_2}, \vec{M}_{eqS_2})]\end{aligned}\quad (4.3)$$

Y las correspondientes ecuaciones superficiales en las superficies de separación dieléctrica (S_2 en la figura)

$$\begin{aligned}\vec{M}_{eqS_2} &= -\hat{n} \times [\vec{E}^i + L_E(\vec{J}_{eqS_1}) + L_E(\vec{J}_{eqS_2}, \vec{M}_{eqS_2})] \\ \vec{J}_{eqS_2} &= \hat{n} \times [\vec{H}^i + L_H(\vec{J}_{eqS_1}) + L_H(\vec{J}_{eqS_2}, \vec{M}_{eqS_2})]\end{aligned}\quad (4.4)$$

de forma que tenemos tres ecuaciones con tres incógnitas ($\vec{J}_{eqS_1}, \vec{J}_{eqS_2}, \vec{M}_{eqS_2}$).

Esta formulación permite obtener el campo total en el exterior de V_1 y de V_2 . El campo obtenido con las ecuaciones (4.3) y (4.4) en el interior de V_1 y de V_2 es nulo, por hipótesis del teorema de equivalencia. En el problema original, los campos en el interior de V_1 son realmente nulos, suponiendo que se trata de un conductor perfecto. Sin embargo, en el interior de V_2 los campos no son nulos, y pueden calcularse a través de la formulación del teorema de equivalencia superficial con las componentes tangenciales de los campos en la cara interior de la superficie

$$\vec{J}_{eqS_2^-} = -\hat{n} \times \vec{H}|_{S_2^-} = -\vec{J}_{eqS_2^+}, \quad \vec{M}_{eqS_2^-} = \hat{n} \times \vec{E}|_{S_2^-} = -\vec{M}_{eqS_2^+}\quad (4.5)$$

donde los superíndices + y - se refieren respectivamente a las caras exterior e interior de S_2 . Las corrientes equivalentes del problema interno son iguales a las del externo cambiadas de signo, debido a la continuidad de las componentes tangenciales de campo en la interfase. El cambio de signo se debe a que el vector normal a la superficie \hat{n} apunta hacia el exterior, en sentido contrario a la región donde se calcula el campo.

El campo radiado en el interior de V_2 por estas corrientes equivalentes es igual al campo total del problema original, suponiendo que no existen fuentes reales \vec{J}^i, \vec{M}^i en el interior de V_2 . Para resolver el problema interior a V_2 , debe replantearse la ecuación (4.4) con las nuevas corrientes equivalentes

$$\begin{aligned}\vec{M}_{eqS_2^-} &= \hat{n} \times L_E(\vec{J}_{eqS_2^-}, \vec{M}_{eqS_2^-}) \\ \vec{J}_{eqS_2^-} &= -\hat{n} \times L_H(\vec{J}_{eqS_2^-}, \vec{M}_{eqS_2^-})\end{aligned}\quad (4.6)$$

Ahora tenemos 4 ecuaciones definidas sobre S_2 : el par (4.4) en la cara exterior de la superficie y el par (4.6) en la cara interior. Los operadores L_E y L_H de las ecuaciones (4.4) corresponden a la radiación de las corrientes en el medio externo, definido por

ϵ_0, μ_0 , mientras que los de las ecuaciones (4.6) corresponden al medio interno, ϵ_2, μ_2 .

Como solamente hay dos incógnitas, $\vec{J}_{eqS_2^-}$ y $\vec{M}_{eqS_2^-}$, es suficiente con resolver un sistema formado por dos de las ecuaciones anteriores. Las elecciones más habituales son: bien las dos EFIE (ecuaciones integrales de campo eléctrico), una en cada cara de la superficie, o bien las dos MFIE (ecuaciones integrales de campo magnético), una en cada cara de la superficie, o bien un sistema formado por la suma de las dos EFIE y la suma de las dos MFIE. Nótese que debe incluirse información relativa a las dos caras de la superficie: un sistema formado por una EFIE y la MFIE correspondiente a la misma cara no contiene información del dieléctrico situado en la otra cara y, por tanto, no permite resolver el problema.

Cuando existen varias superficies de separación entre regiones dieléctricas homogéneas y/o conductoras, debe plantearse un sistema formado por un par de ecuaciones para cada interfase entre dieléctricos, según se ha descrito en el párrafo anterior, y la EFIE o la MFIE (4.3) correspondientes a cada interfase dieléctrico-conductor.

c) Formulación de las ecuaciones integrales

Según el teorema de equivalencia superficial, los campos en una región V del espacio libre, cerrada por una superficie, son los originados por las fuentes en el interior de dicho volumen más los radiados por las corrientes equivalentes en la superficie que envuelve el volumen. Si consideramos un problema exterior o abierto, la superficie del teorema de equivalencia es la superficie S más la superficie del infinito S_∞ . Según la condición de radiación de Sommerfeld los campos en el infinito son nulos, por lo que también lo son las corrientes equivalentes en s_∞ y basta tener en cuenta únicamente la contribución de la superficie S más las fuentes en el volumen V . Teniendo en cuenta estas consideraciones el campo en el volumen V es:

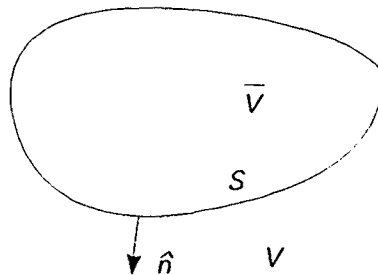


Figura 4-2: Región exterior V limitada por la superficie S y por el infinito

$$\begin{aligned}
 & \int_V [-j\omega\mu \vec{J}G - \vec{M} \times \nabla'G + \frac{\rho}{\epsilon} \nabla'G] dv' \\
 & + \int_S [-j\omega\mu (\hat{n} \times \vec{H})G + (\hat{n} \times \vec{E}) \times \nabla'G + (\hat{n} \cdot \vec{E}) \nabla'G] ds' = \\
 & = \begin{cases} \vec{E} & \text{en } \vec{r} \in V \\ 0 & \text{en } \vec{r} \in \bar{V} \end{cases} \quad (4.8)
 \end{aligned}$$

$$\begin{aligned}
 & \int_V [-j\omega\epsilon \vec{M}G + \vec{J} \times \nabla'G + \frac{\tau}{\mu} \nabla'G] dv' \\
 & + \int_S [j\omega\epsilon (\hat{n} \times \vec{E})G + (\hat{n} \times \vec{H}) \times \nabla'G + (\hat{n} \cdot \vec{H}) \nabla'G] ds' = \\
 & = \begin{cases} \vec{H} & \text{en } \vec{r} \in V \\ 0 & \text{en } \vec{r} \in \bar{V} \end{cases} \quad (4.9)
 \end{aligned}$$

Estas expresiones reciben el nombre de "ecuaciones de Stratton-Chu" [Stratton'41]. Al aplicar el teorema de equivalencia el "campo incidente" es, en general, el radiado por las fuentes en el interior del volumen V

$$\begin{aligned}
 \int_V (-j\omega\mu \vec{J}G - \vec{M} \times \nabla'G + \frac{\rho}{\epsilon} \nabla'G) dv' &= \vec{E}^i \\
 \int_V (-j\omega\epsilon \vec{M}G + \vec{J} \times \nabla'G + \frac{\tau}{\mu} \nabla'G) dv' &= \vec{H}^i
 \end{aligned} \quad (4.10)$$

Los operadores L_E y L_H pueden obtenerse al hacer la sustitución $\vec{J}_{eq} = \hat{n} \times \vec{H}$ y $\vec{M}_{eq} = -\hat{n} \times \vec{E}$ en las ecuaciones de Stratton-Chu. Teniendo en cuenta la definición del campo dispersado $\vec{E}^s = \vec{E} - \vec{E}^i$ y $\vec{H}^s = \vec{H} \times \vec{H}^i$, para $\vec{r} \in V$ las expresiones de estos operadores son:

$$\begin{aligned}
 \vec{E}_S &= L_E(\vec{J}_{eq}, \vec{M}_{eq}) = \int_S [-j\omega\mu \vec{J}_{eq}G - \vec{M}_{eq} \times \nabla'G - \frac{\nabla' \cdot \vec{J}_{eq}}{j\omega\epsilon} \nabla'G] ds' \\
 \vec{H}_S &= L_H(\vec{J}_{eq}, \vec{M}_{eq}) = \int_S [-j\omega\epsilon \vec{M}_{eq}G + \vec{J}_{eq} \times \nabla'G - \frac{\nabla' \cdot \vec{M}_{eq}}{j\omega\mu} \nabla'G] ds'
 \end{aligned} \quad (4.11)$$

4.1.2 Discretización

En la sección anterior se ha presentado la formulación de las ecuaciones integrales para problemas de radiación y difracción. Estas ecuaciones funcionales pueden representarse en la forma general

$$LX = Y \quad (4.12)$$

donde L es uno de los operadores lineales L_E, L_H , la función X es la incógnita (corrientes equivalentes) y la función Y el término independiente (campo incidente).

Para resolver (4.12) mediante ordenador es necesario discretizar las funciones y operadores y convertir la ecuación funcional en una ecuación matricial. Esta idea básica fue desarrollada a principios de siglo (1915) por el ingeniero ruso Galerkin. Las aplicaciones electromagnéticas aparecieron con el desarrollo de las comunicaciones en bandas de microondas: en la década de los cuarenta los métodos variacionales y en la de los cincuenta el método de Rayleigh-Ritz permitían resolver el problema con un gran esfuerzo analítico y un pequeño esfuerzo numérico. No fue hasta la década de los sesenta cuando el desarrollo de la tecnología informática permitió plantear métodos que requieran un menor esfuerzo analítico. La aparición en 1968 de la obra "*Field Computation by Moment Methods*" del Prof. R.F. Harrington supuso la formulación sistemática de los métodos numéricos entonces existentes en un concepto muy general que se denominó el "Método de los momentos" y que constituye la herramienta de análisis numérico más utilizada actualmente en electromagnetismo.

a) Método de los momentos

El primer paso consiste en aproximar la función incógnita X por una combinación lineal de "funciones base" x_j

$$X_N = \sum_j^N a_j x_j \approx X \quad (4.13)$$

donde los N coeficientes a_j son las muestras de la discretización de X y constituyen las incógnitas del problema numérico a resolver. Para que X_N sea una buena aproximación de X , es necesario que las funciones base x_j pertenezcan al dominio del operador L , es decir, que cumplan las condiciones de diferenciabilidad y de contorno del operador.

Al substituir (4.13) en la ecuación funcional (4.12) se obtiene:

$$\sum_j^N a_j L x_j = Y_N \approx Y \quad (4.14)$$

que es una ecuación funcional con N incógnitas a_j . Para que exista una buena solución es necesario que una combinación lineal de las funciones Lx_j pueda aproximar adecuadamente a la función Y . Para ello, las funciones base x_j deben ser tales que las funciones Lx_j pertenezcan al rango del operador L y que, cuando el orden de la discretización N tienda a infinito, el conjunto $\{Lx_j\}$ sea una base completa del rango de L . Esto equivale a decir que el campo debe poder representarse en la base de funciones $\{Lx_j\}$.

El error en las condiciones de contorno o "residuo" es

$$R = Y - Y_N = Y - \sum_j^N a_j L x_j \quad (4.15)$$

Para convertir la ecuación funcional (4.14) en un sistema de M ecuaciones con N incógnitas, se hace nulo el residuo ponderado con M funciones peso W_i

$$\langle W_i, R \rangle = 0 \quad i = 1 \dots M \quad (4.16)$$

donde el producto escalar está definido como el producto interno de Hilbert

$$\langle f, g \rangle = \int_{D(\omega)} f^*(\vec{r}) \cdot g(\vec{r}) d\vec{r} \quad (4.17)$$

El sistema de M ecuaciones lineales con N incógnitas que resulta es

$$\langle W_i, Y \rangle = \sum_j^N a_j \langle W_i, L x_j \rangle \quad i = 1 \dots M \quad (4.18)$$

el cual puede escribirse en forma matricial como

$$A a = b \quad (4.19)$$

donde A es la matriz $M \times N$ $A_{ij} = \langle W_i, L x_j \rangle$, a es el vector columna formado por los elementos a_j y b es el vector columna $b_i = \langle W_i, Y \rangle$. Para resolver el sistema (4.19), si $M=N$ la matriz A es cuadrada y

$$a = A^{-1} b \quad (4.20)$$

Si $M > N$, el sistema (4.19) es sobredeterminado y, en general, no tiene una solución exacta. La solución de mínimos cuadrados, que proporciona error o residuo R mínimo, puede obtenerse a través de la "ecuación normal"

$$A^* A a = A^* b \quad a = (A^* A)^{-1} A^* b \quad (4.21)$$

donde A^* es la matriz adjunta de A , es decir, la traspuesta conjugada de A , $A^* = (A^T)^*$. La matriz $A^* A$ es cuadrada y, por tanto, se puede invertir si no es singular.

Cuando la matriz A es singular, existen soluciones homogéneas del sistema de ecuaciones (para $b = 0$ ó $Y = 0$) lo que significa que existen campos que verifican las condiciones de contorno sin excitación y, por tanto, nos encontramos en condiciones de resonancia.

Conocidos los coeficientes a_j mediante (4.20) o (4.21), la solución numérica del problema es X_N de (4.13). En general, es una solución aproximada porque $\langle W_i, R \rangle = 0$ no implica que el residuo sea cero y porque el espacio funcional en el que está definida la función X es de infinitas dimensiones y sólo podemos representarla con X_N mediante un conjunto finito de N coeficientes a_j .

Método de colocación ("Point-Matching")

Si se eligen funciones peso iguales a funciones delta de Dirac

$$W_i(\vec{r}) = \delta(\vec{r} - \vec{r}_i) \quad i = 1 \dots M \quad (4.22)$$

con \vec{r}_i un punto de la superficie S sobre la que se aplican las condiciones de contorno, resulta que $\langle W_i, Y \rangle = \langle W_i, Y_N \rangle$ implica $Y(\vec{r}_i) = Y_N(\vec{r}_i)$, es decir, se impone la condición de contorno en un número finito de M puntos \vec{r}_i .

En este caso el sistema de ecuaciones algebraicas (4.18) es

$$Y(\vec{r}_i) = [L X_N]_{(\vec{r}_i)} \quad i = 1 \dots M \quad (4.23)$$

Esta versión del método de los momentos se utiliza con frecuencia en la práctica, debido a su sencillez ya que no son necesarios los productos escalares con las funciones peso (integraciones numéricas), lo cual se traduce en una mayor velocidad de cálculo.

Método de Galerkin

El método de Galerkin se caracteriza por la elección de funciones peso iguales a las funciones base, $W_i = x_i$. En este caso, el sistema de ecuaciones a resolver es:

$$\langle x_i, Y \rangle = \sum_j^N a_j \langle x_i, L x_j \rangle \quad i = 1 \dots M \quad (4.24)$$

Esta elección debe cumplir el criterio general de que cuando M tiende a infinito, las funciones peso sean una base completa del rango del operador L . Puede demostrarse que para ello es condición suficiente que sean una base completa del dominio del operador adjunto L^a . Como las funciones peso son iguales a las funciones base, que son una base completa del dominio del operador L , resulta que si el operador L es autoadjunto, $L = L^a$, las funciones peso son siempre una base completa del rango de L y el método de los momentos es aplicable en este caso.

b) Método de los elementos de contorno

En el método de los elementos de contorno los subdominios de la discretización se denominan "elementos". Con objeto de conseguir que las funciones base aproximen simultáneamente la geometría de la superficie y la corriente que fluye por ella, se utilizan polinomios que interpolan las coordenadas de la superficie y el valor de la corriente en unos puntos fijos llamados "nodos". Para facilitar la construcción de los polinomios de interpolación, estos se definen en función de coordenadas locales (ξ, η) sobre un plano bidimensional y con formas sencillas, como por ejemplo triángulos rectángulos o cuadrados. De esta forma, las operaciones de integración y diferenciación de las funciones base también se simplifican, llevándose a cabo mediante un cambio de variable del espacio real (x, y, z) a las coordenadas locales (ξ, η) .

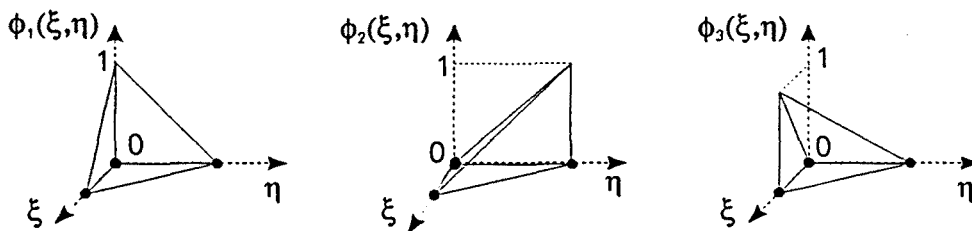


Figura 4-3: Funciones de interpolación para elementos finitos triangulares lineales. En general, $\phi_i(\xi, \eta) = 1$ en el nodo i -ésimo y 0 en los demás.

La interpolación de la geometría de un elemento y la corriente que fluye por él se realiza mediante las ecuaciones

$$\vec{r} = \sum_{i=1}^{N_g} \phi_i(\xi, \eta) \vec{r}_i \qquad \vec{J} = \sum_{i=1}^{N_f} \phi_i(\xi, \eta) \vec{J}_i \qquad (4.25)$$

donde ϕ_i son funciones de interpolación definidas sobre un elemento que valen 1 en el nodo i -ésimo y 0 en los demás (figura 4-3), y se construyen mediante polinomios de interpolación de Lagrange en (ξ, η) . \vec{r}_i son las coordenadas del nodo i -ésimo y \vec{J}_i el valor de la corriente en ese nodo. N_g y N_f son, respectivamente, los números de nodos del elemento utilizado para modelar la geometría de la superficie y del elemento que representa la función.

La figura 4-4 muestra una esfera cuya superficie ha sido discretizada con elementos de contorno cuadráticos.

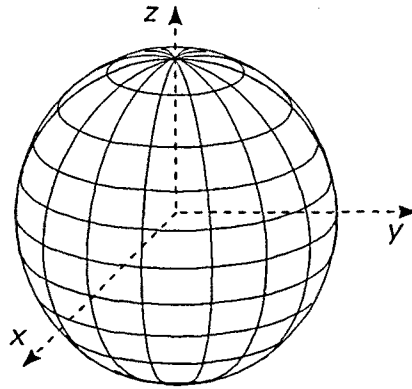


Figura 4-4: Esfera modelada con elementos finitos triangulares y cuadrangulares de orden 3.

Se denomina orden del elemento al orden de los polinomios de interpolación ϕ_i . El número de nodos existentes en cada elemento aumenta con el orden del mismo. Los elementos de contorno se denominan isoparamétricos cuando se utilizan las mismas funciones de interpolación para representar la geometría del elemento y la función incógnita (mismo orden de los polinomios y $N_g = N_f$), superparamétricos cuando el orden en la aproximación de la superficie es mayor que el de la función ($N_g > N_f$) y subparamétricos cuando es menor.

Las integraciones y diferenciaciones de los operadores L_E y L_H se realizan sobre el elemento en coordenadas locales (ξ, η) , utilizando el jacobiano de la transformación donde sea necesario. El operador se discretiza con las funciones peso habituales, normalmente con el método de colocación muestreando el campo en la posición de los nodos. Así, se obtiene el campo en el nodo i -ésimo como la suma de las contribuciones de las corrientes en todos los elementos

$$\vec{E}_i^s = \sum_{e=1}^{N_e} L_E(\vec{J}_e) \quad i = 1 \dots N_f \quad (4.26)$$

donde N_e es el número de elementos. Sustituyendo el valor de la corriente en el interior de cada elemento, \vec{J}_e , por la interpolación (4.25) de los valores de la corriente en los nodos, tenemos

$$\vec{E}_i^s = \sum_{e=1}^{N_e} \sum_{j=1}^{N_f} \vec{J}_{ej} L_E[\phi_j(\xi, \eta)] \quad i = 1 \dots N_f \quad (4.27)$$

y sumando para todos los elementos que comparten el mismo nodo (operación de "ensamblado" de matrices locales en una matriz global) resulta, tras cambiar la numeración local de los nodos en cada elemento (j) por una numeración global (n),

$$\vec{E}_i^s = \sum_{n=1}^{N_t} \vec{J}_n L_E[\phi_n(\xi, \eta)] \quad i = 1 \dots N_t \quad (4.28)$$

donde \vec{J}_n representa la suma de las corrientes de los distintos elementos que fluyen por el nodo n-ésimo y N_t es el número total de nodos en la superficie. Para transformar la ecuación (4.28) en un sistema de ecuaciones escalares es necesario utilizar una incógnita para cada una de las tres componentes de la corriente en cada nodo. El resultado de aplicar la EFIE o MFIE a una ecuación del tipo (4.28) es un sistema de ecuaciones $Aa = b$ donde el vector columna incógnita a contiene los valores de cada componente de la corriente en cada nodo.

4.1.3 Modelado

En estrecha colaboración con los trabajos de esta tesis se han desarrollado en nuestro grupo implementaciones del método de los elementos de contorno [Duffo'93b] [Duffo'90]. Para poder analizar los objetos con cualquier método numérico es necesario disponer de un modelo geométrico de los mismos. Por ello, se hace imprescindible una fase de creación y una fase de discretización del modelo, las cuales forman el preprocesado del análisis electromagnético.

En la actualidad existen diversas herramientas CAD que permiten el diseño de objetos tridimensionales y su posterior discretización. En el departamento se ha adquirido el software comercial I-DEAS (Integrated Design Engineering Analysis Software) de SDRC (Structural Dynamics Research Corporation). Dicho software es un completo paquete informático de CAE (ingeniería asistida por ordenador) que permite integrar las funciones de diseño, análisis, *drafting*, test y fabricación de moldes por control numérico en cada fase del desarrollo de un producto.

I-DEAS se divide en una serie de familias o módulos con funciones bien diferenciadas [I-DEAS]. En el diseño de objetos para el análisis por el método de los elementos de contorno ha sido necesario utilizar solamente dos módulos: *Solid Modeling* y *Finite Element Modeling & Analysis*.

El módulo *Solid Modeling* ofrece herramientas de software para desarrollar los conceptos geométricos y de diseño de un sistema mecánico. Permite crear objetos sólidos usando formas geométricas 3-D, 2-D o conjuntos de secciones transversales.

Además, se puede manipular la geometría diseñada orientándola, combinándola con otras, cortándola, deformándola, etc.

El módulo *Finite Element Modeling & Analysis*, construye un modelo de elementos finitos en el que se incluye la localización de los nodos y de los elementos, las propiedades físicas del material, cargas y condiciones de contorno. Dicho modelo se puede construir a partir de una estructura diseñada en el módulo *Solid Modeling*

La creación del objeto con la información necesaria, de forma que se pueda analizar con el software desarrollado en el departamento [Duffo'93a][Duffo'93b] se divide en dos procesos encadenados. En primer lugar se crea la figura deseada y una vez obtenida se malla su superficie (se discretiza), generando elementos y nodos sobre la misma, los cuales constituyen los parámetros de entrada del software de análisis numérico. El primer proceso se realiza en el *Solid Modeling* y el segundo en el *Finite Element Modeling & Analysis*.

La familia *Solid Modeling* ofrece al usuario la posibilidad de elegir entre varias técnicas de modelado, puesto que, como se ha demostrado en el capítulo 2, no hay una técnica que sea la más apropiada para cualquier diseño sino que para cada tipo de geometría existe una técnica que es más adecuada que las demás. Entre otras técnicas ofrece la posibilidad del modelado por medio de la combinación de estructuras canónicas, por revolución de un perfil 2D, mediante facetas y mediante superficies B-spline. Independientemente de la técnica de modelado empleada para definir el objeto, todos los objetos son almacenados como modelos de fronteras, las cuales están definidas por superficies paramétricas B-Spline racionales no uniformes (NURBS).

Una vez diseñado el objeto se procede a la discretización de la superficie. Los pasos seguidos para dicha discretización se resumen en la figura 4-5 y se detallan a continuación.

- 1) Primero se recupera el objeto diseñado en *Solid Modeling* o se crea uno en el módulo *Finite Elements Modeling & Analysis*, que incorpora algunas funciones de modelado tridimensional, aunque mucho más sencillas y limitadas que las del otro módulo.

- 2) Una vez recuperado el sólido se procede a realizar el mallado que consiste en crear nodos y elementos basados en los puntos, curvas y superficies que definen la geometría de la estructura. Para ello, se generan las áreas de mallado (*mesh areas*). Un área de mallado es la entidad topológica que se discretiza mediante elementos finitos y se define a través del conjunto de curvas que la limitan.

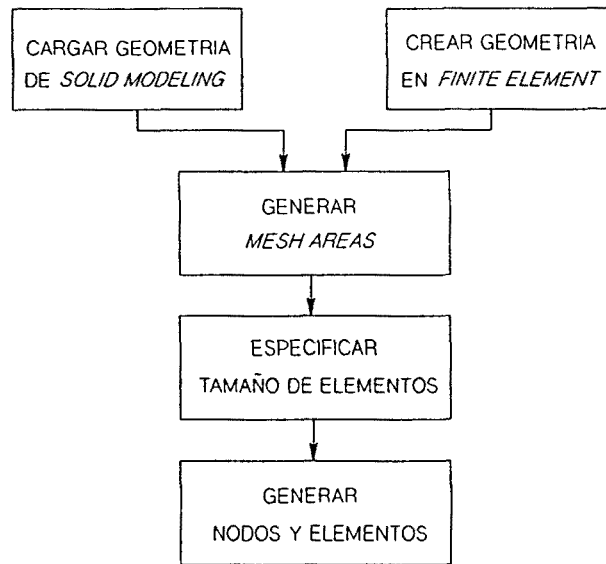


Figura 4-5: Diagrama de bloques del proceso de discretización de un objeto

La definición del área de mallado es imprescindible para poder especificar el tamaño y forma de los elementos, el número de nodos por elemento, el tipo de mallado, etc, ya que puntos, curvas y superficies no contienen toda la información necesaria para controlar la generación del mallado.

La creación de las áreas de mallado y su posterior discretización se puede realizar de manera manual o automática. Tanto en un caso como en otro se puede realizar un mallado regular conformado (*mapped meshing*) o un mallado libre (*free meshing*). El mallado regular está limitado a elementos de 3 ó 4 lados dispuestos en filas y columnas, cuya forma se adapta a los límites de la superficie mallada. El mallado libre puede describir regiones más complicadas porque los elementos pueden ser de formas y tamaños muy distintos y pueden situarse libremente. Esta técnica permite realizar el mallado de áreas con agujeros y con cualquier número de aristas.

3) Una vez formadas las áreas y decidido el tipo de mallado, se activa la formación de elementos y nodos mediante el comando *Generate mesh*.

En la figura 4-6 se presenta un ejemplo de un objeto discretizado siguiendo el algoritmo presentado arriba mediante el programa I-DEAS.

Una vez realizada la discretización de un objeto este módulo permite modificar algunas características de la malla obtenida. Se pueden modificar los nodos y elementos creados: es posible volverlos a numerar, cambiar sus propiedades físicas y color, moverlos, borrarlos e incluso cambiar el sistema de coordenadas de referencia.

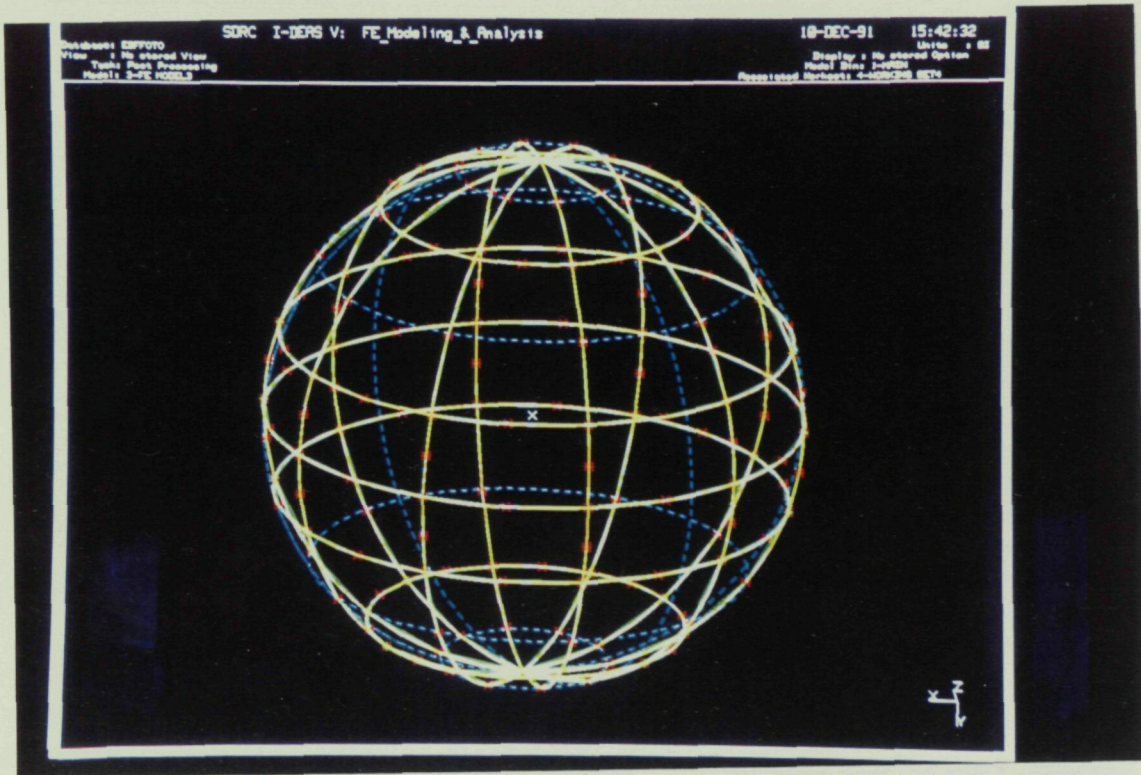


Figura 4-6: Esfera discretizada en I-DEAS por el método de los elementos de contorno

Este módulo permite hacer discretizaciones superficiales con elementos triangulares y con elementos cuadrangulares, los cuales pueden ser lineales, parabólicos y cúbicos. Los elementos parabólicos triangulares y cuadrangulares indistintamente han sido los que más se han utilizado en los diseños realizados para el posterior análisis electromagnético. Dichos elementos presentan la ventaja sobre los lineales de adaptarse mejor a las superficies curvadas y, además, tienen una definición más simple que la de los elementos cúbicos, puesto que un elemento cúbico triangular está formado por 9 nodos y el cuadrangular por 12 nodos mientras que los elementos parabólicos triangulares quedan definidos con 6 nodos y los cuadrangulares con 8 nodos (ver figura 4-7).

En ocasiones también se ha optado por el uso de elementos lineales en la discretización, ya que aunque adolezcan del inconveniente de no ajustarse bien a las superficies curvadas presentan la ventaja de reducir el número de nodos, con lo que se reduce la cantidad de memoria requerida para su almacenamiento y, además, se simplifica notablemente su procesado mediante el método de los elementos de contorno. Ello se traduce en la posibilidad de poder analizar objetos eléctricamente mas grandes, aunque a costa de una peor precisión en el modelo geométrico.

Hasta aquí se ha expuesto el procedimiento para el modelado y discretización de objetos tridimensionales. En el caso 2-D se sigue un procedimiento totalmente paralelo al

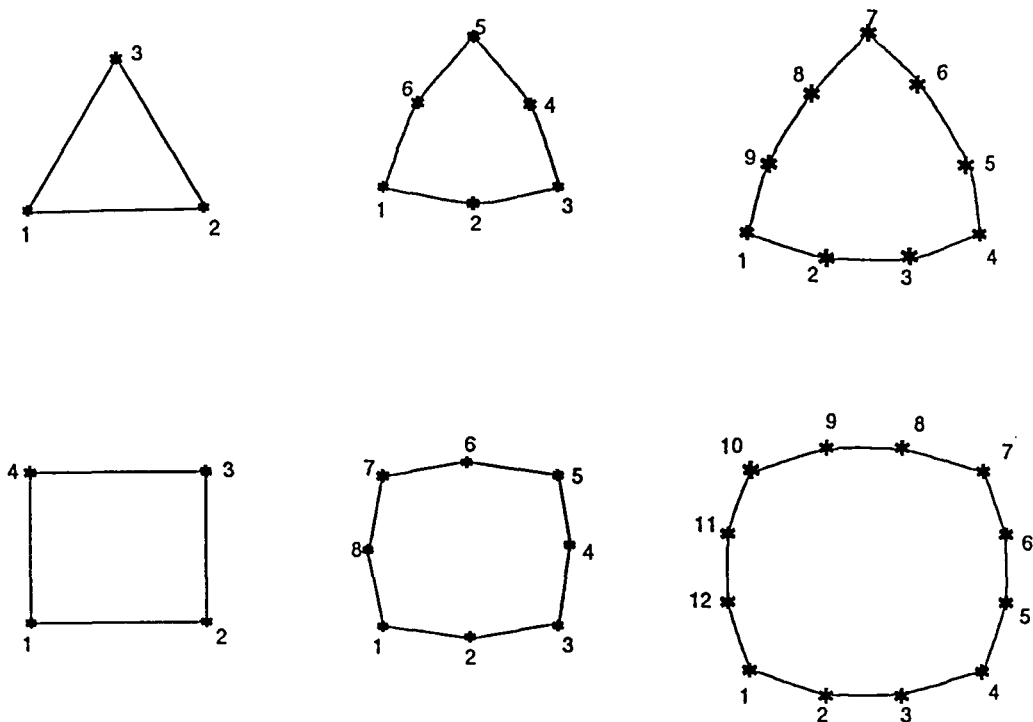


Figura 4-7: Tipos de elementos bidimensionales que permite utilizar I-DEAS en el mallado superficial

presentado, sólo que en este caso el objeto es una superficie plana, la cual se puede modelar fácilmente mediante las herramientas de modelado que proporciona la familia *Finite Element Modeling & Analysis*. Sin embargo, ahora lo que hay que discretizar es el contorno de la superficie que se va a estudiar. Por tanto, los elementos utilizados son unidimensionales y también en este caso pueden ser lineales (2 nodos), parabólicos (3 nodos) y cúbicos (4 nodos). Los más utilizados son los parabólicos porque se ajustan bien a las curvas, aunque también pueden emplearse los lineales para conseguir un ahorro de memoria de almacenamiento.

El proceso de generación de elementos de contorno para problemas bi y tridimensionales se ha automatizado mediante el desarrollo de programas en el lenguaje "IDEAL", propio de I-DEAS, que invocan los comandos necesarios teniendo en cuenta todas las condiciones que pueden influir en el proceso.

Como conclusión podemos señalar que las nuevas herramientas de diseño asistido por ordenador como el programa I-DEAS simplifican mucho el problema de modelado y discretización de objetos arbitrarios implícito en los problemas de análisis electromagnético. Hasta ahora se diseñaban los objetos de forma manual, con lo que

sólo era posible el diseño y discretización de formas muy sencillas. Gracias a las herramientas CAD es posible el diseño y discretización de estructuras totalmente arbitrarias. La única gran limitación que existe en la actualidad en el análisis electromagnético con métodos numéricos está en la cantidad de información necesaria para definir los objetos eléctricamente grandes.

4.2 METODO DE LOS ELEMENTOS FINITOS

En el apartado anterior, dedicado al método de los elementos de contorno (BEM), se ha demostrado como es posible discretizar una ecuación funcional del tipo $Lu=v$ mediante el método de los momentos. Sin embargo, esta no es la única solución posible. El método de los elementos finitos (FEM) es el método más general para resolver ecuaciones funcionales mediante una discretización en subdominios, de forma que casi todos los demás métodos, incluido el método de los momentos, pueden considerarse como un caso particular del FEM.

La característica principal del FEM es la discretización simultánea de la función y de la geometría en subdominios o elementos utilizando funciones base polinómicas. Estos polinomios interpolan el valor de la función a partir de muestras en unos puntos fijos llamados nodos. Los valores en los nodos constituyen el vector incógnita en la resolución matricial del problema.

En cuanto a la formulación de las ecuaciones funcionales, puede aplicarse el FEM con las ecuaciones integrales derivadas del teorema de equivalencia, con ecuaciones diferenciales o con ecuaciones variacionales. En el primer caso el FEM se reduce al método de los elementos de contorno presentado en el apartado 4.1. El segundo y tercer casos, ecuaciones diferenciales o variacionales, constituyen lo que se entiende habitualmente como método de los elementos finitos en un sentido más restringido.

Con este tipo de formulación diferencial o variacional la discretización es volumétrica y no superficial, como sucede en el BEM. A cambio, la matriz a invertir es dispersa y no llena, lo cual compensa en parte el mayor número de incógnitas que aparecen con la discretización volumétrica.

4.2.1 Formulación

a) Formulación diferencial

Sea una ecuación diferencial

$$v = L_d u \tag{4.29}$$

donde u es la función incógnita y v una función conocida. El operador diferencial L_d es habitualmente la ecuación de onda inhomogénea. Discretizando la incógnita en funciones base polinómicas

$$u(\vec{r}) = \sum_{j=0}^{N-1} u_j A_j(\vec{r}) \tag{4.30}$$

y aplicando el método de Galerkin se obtiene un sistema matricial de la forma

$$\langle A_i, v \rangle = \sum_{j=0}^{N-1} u_j \langle A_i, L_d A_j \rangle \quad i = 0 \dots M-1 \tag{4.31}$$

Este sistema de ecuaciones se diferencia del obtenido para el método de los elementos de contorno, ecuación (4.24), en que la matriz a invertir $\langle A_i, L_d A_i \rangle$ es dispersa y que el número de incógnitas es mucho mayor que en el BEM por discretizarse todo el volumen del objeto en lugar de su superficie.

Puede demostrarse que esta formulación conduce a una solución de la ecuación diferencial con mínimo error cuadrático medio cuando el operador L_d es definido positivo, lo cual sucede para la ecuación de onda con las condiciones de contorno habituales.

b) Formulación variacional

Se dice que un funcional escalar $J(u)$ es variacional cuando presenta un mínimo o un máximo para $u(\vec{r})$ igual a la solución correcta del problema que queremos resolver. Ello implica que el escalar $J(u)$ es menor (o mayor) con la solución correcta que con cualquier otra función u imaginable. Habitualmente el significado físico de estos operadores variacionales está relacionado con la energía electromagnética almacenada en el sistema bajo estudio: la solución u debe ser una función tal que la energía total sea mínima.

Un ejemplo típico es el siguiente: si u es la solución de la ecuación de onda homogénea,

$$\nabla^2 u + k^2 u = 0 \quad (4.32)$$

entonces el funcional

$$J(u) = \frac{\int_s (\nabla u)^2 ds}{\int_s u^2 ds} \quad (4.33)$$

es mínimo.

Para discretizar el variacional se descompone la incógnita en funciones base polinómicas, ecuación (4.30), y se buscan los coeficientes u_j que hacen mínimo el funcional:

$$\frac{\partial J(u)}{\partial u_j} = 0 \quad \forall j=0 \dots N-1 \quad (4.34)$$

con lo que se obtiene un sistema de N ecuaciones con N incógnitas. Este procedimiento de discretización de una expresión variacional se conoce con el nombre de "método de Rayleigh-Ritz".

4.2.2 Discretización

El procedimiento de discretización que se realiza en el método de los elementos finitos es, como se ha indicado en la introducción de este apartado, lo más general posible: realiza una buena aproximación de las funciones y de la geometría de los objetos que discretiza, adaptándose muy bien a la complejidad de estas funciones u objetos.

Para aproximar una función $u(\vec{r})$ se divide su dominio D en pequeños subdominios o elementos. Dentro de cada elemento puede aproximarse la función $u(\vec{r})$ a partir de los valores $u_i = u(\vec{r}_i)$ en los nodos \vec{r}_i ,

$$u(\vec{r}) = \sum_{i=1}^n u_i A_i(\vec{r}) = [A_1 \ A_2 \ A_3 \ \dots \ A_n] \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad (4.35)$$

donde las funciones base $A_i(\vec{r})$ son polinomios que interpolan la función u a partir de sus valores nodales u_i . El número de nodos n definidos en cada elemento está relacionado con el orden de los polinomios que forman las funciones base. Por ejemplo, para realizar una aproximación lineal con polinomios de primer grado sobre un elemento triangular o rectangular los nodos son los vértices del triángulo o rectángulo (figura 4-8

a), mientras que para una aproximación cuadrática con polinomios de segundo grado los nodos son los vértices y los puntos medios de los lados (figura 4-8 b).

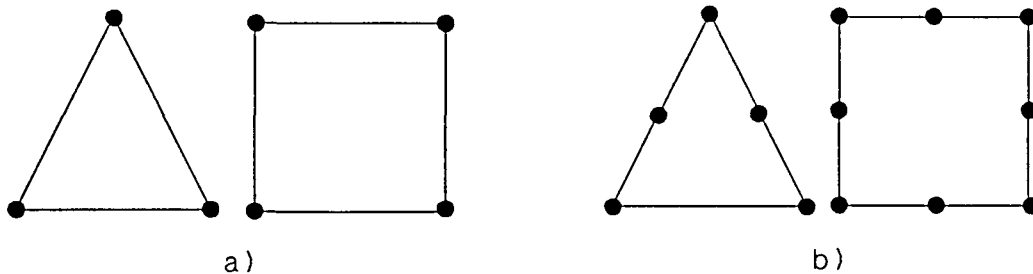


Figura 4-8: Elementos finitos triangulares y rectangulares: a) Interpolación lineal (de primer grado), b) interpolación cuadrática (de segundo grado)

Los polinomios de interpolación $A_i(\vec{r})$ son tales que

$$A_i(\vec{r}_j) = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad (4.36)$$

de forma que se asegura que las incógnitas u_i sean los valores nodales de la función, $u(\vec{r}_i) = u_i$.

Las expresiones de los polinomios $A_i(\vec{r})$ para cualquier orden de aproximación pueden obtenerse muy fácilmente definiendo las coordenadas del área local para problemas 2-D, o de volumen local para problemas 3-D, y utilizando el método de interpolación de Lagrange [Itoh'89] con la condición (4.36).

Las coordenadas de área local L_1, L_2, L_3 de un punto P en el interior de un triángulo se definen como la relación (ver figura 4-9)

$$L_1 = \frac{\text{área del triángulo } P \ v_2 \ v_3}{\text{área del triángulo } v_1 \ v_2 \ v_3} \quad (4.37)$$

siendo las coordenadas L_2 y L_3 análogas.

Evidentemente, se cumple la relación $L_1 + L_2 + L_3 = 1$ para todo punto P en el interior del triángulo, por lo que una de las tres coordenadas de área local es redundante. Para un triángulo de tres nodos situados en los vértices, las coordenadas de área local de los nodos cumplen:

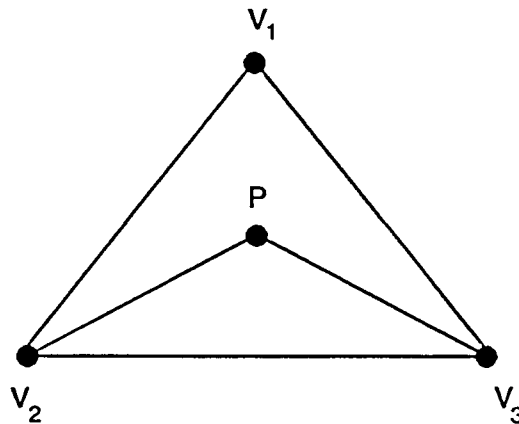


Figura 4-9: Cálculo de las coordenadas de área local.

$$L_i(\mathbf{V}_j) = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad (4.38)$$

de forma que los polinomios de interpolación de primer orden $A_i(\vec{r})$ son iguales a las coordenadas de área local:

$$A_i(\vec{L}) = L_i \quad (4.39)$$

Para polinomios de orden superior a uno se utiliza el método de interpolación de Lagrange en las coordenadas de área local L_i . Estos polinomios $A_i(\vec{L})$ se encuentran tabulados en la literatura para distintas formas de los elementos, tanto bidimensionales como tridimensionales, y distintos órdenes de los polinomios.

En el caso 3-D, para un tetraedro tenemos cuatro coordenadas de volumen local L_1, L_2, L_3, L_4 definidas análogamente a la ecuación (4.37) con volumen en lugar de área.

Con este procedimiento de discretización, si para elementos adyacentes se toman los mismos nodos en el lado o cara común, puede demostrarse fácilmente que la función $u(\vec{r})$ será continua en la transición de un elemento a otro.

El paso de coordenadas de área local L_1, L_2, L_3 (y L_4 en 3-D) a coordenadas globales \vec{r} se realiza a partir de las posiciones de los nodos \vec{r}_i

$$\bar{r} = \sum_j^n \bar{r}_j A_j(\bar{L}) \tag{4.40}$$

Esta expresión es también la ecuación de interpolación de la geometría para el caso de elementos isoparamétricos, en los que los polinomios de interpolación de la función y de la geometría son del mismo orden.

La función $u(\bar{r})$ se aproxima en todo su dominio D a partir de sus valores nodales u_i como:

$$u(\bar{r}) = \sum_{i=1}^N u_i B_i(\bar{r}) \tag{4.41}$$

donde esta ecuación se diferencia de la aproximación local (4.35) en que ahora el sumatorio se extiende a los N nodos existentes en el dominio mientras que en (4.35) el sumatorio se extiende sólo sobre los n nodos de cada elemento.

Las funciones base $B_i(\bar{r})$ valen 1 en el nodo i -ésimo y 0 en todos los demás nodos y están definidas sobre todos los elementos que comparten el nodo i -ésimo. Una función base $B_i(\bar{r})$ es igual a la función local $A_i(\bar{L})$ correspondiente al elemento en el interior del cual se encuentra \bar{r} . En el contorno de un elemento, $B_i(\bar{r})$ es igual a la suma de los $A_i(\bar{r})$ de todos los elementos que comparten este contorno.

Por ejemplo, un funcional lineal $L(u)$ se discretiza como:

$$L(u) = L\left[\sum_i^N u_i B_i(\bar{r})\right] = \sum_i^N u_i L[B_i(\bar{r})] \tag{4.42}$$

y un variacional lineal $J(u^2)$:

$$\begin{aligned} J(u^2) &= J\left[\sum_i u_i B_i(\bar{r}) \sum_j u_j B_j(\bar{r})\right] \\ &= \sum_i \sum_j u_i u_j J[B_i(\bar{r}) B_j(\bar{r})] \end{aligned} \tag{4.43}$$

que se escriben más cómodamente en forma matricial:

$$L(u) = [u_1 \dots u_N] \begin{bmatrix} L(B_1) \\ \vdots \\ L(B_N) \end{bmatrix} \tag{4.44}$$

y

$$J(u^2) = [u_1 \dots u_N] \begin{bmatrix} J(B_1 B_1) & \dots & J(B_1 B_N) \\ \vdots & & \vdots \\ J(B_N B_1) & \dots & J(B_N B_N) \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \quad (4.45)$$

Las matrices de las ecuaciones (4.44) y (4.45) se conocen como "matrices globales" pues interpolan todos los puntos \vec{r} del dominio D a partir de todos los nodos.

Es habitual calcular los operadores L o J sobre los polinomios de interpolación en coordenadas locales, pues en este caso la formulación se simplifica y, además, los elementos de las matrices (4.44) o (4.45) se encuentran tabulados en la literatura para los operadores más frecuentes. Para ello es necesario tener en cuenta el cambio de coordenadas (4.40) a través del jacobiano correspondiente.

Las matrices $[L(A_i)]$ y $[J(A_i A_j)]$ de cada elemento se conocen como "matrices del elemento". Estas matrices se ensamblan para formar las matrices globales $[L(B_i)]$ y $[J(B_i B_j)]$. La operación de ensamblado consiste en sustituir en las ecuaciones (4.44) o (4.45) B_i por el A_i del elemento en el interior del cual está el nodo, o la suma de varios A_i cuando el nodo i -ésimo se encuentra en un contorno común a varios elementos. Esta operación de ensamblado puede realizarse de forma mecánica mediante una reenumeración de nodos y suma de submatrices.

4.2.3 Modelado

El método de los elementos finitos (FEM), al igual que el método de los elementos de contorno (BEM), precisa de un modelo geométrico del objeto que se va analizar. Y, por tanto, también requiere una fase de diseño y otra de discretización del modelo antes de proceder al análisis electromagnético.

Tal como se ha comentado en la sección 4.1.3, se ha adquirido el software comercial I-DEAS que simplifica mucho tanto el diseño como la discretización de los objetos. El proceso que hay que seguir para la creación de un modelo que contenga toda la información necesaria para el análisis electromagnético por el método de los elementos finitos es muy parecido al que se ha descrito para el BEM.

Sin embargo, mientras que para resolver el problema tridimensional por el BEM se requería un modelo en el se discretizaba la superficie del objeto mediante elementos bidimensionales, ahora es necesaria una discretización de todo el volumen que ocupa el objeto mediante elementos tridimensionales. Ello repercute en un aumento considerable de la cantidad de información a almacenar y en que el proceso de mallado precise de algunos pasos adicionales respecto al que se ha descrito anteriormente.

En la figura 4-10 se muestra un objeto discretizado por elementos finitos tridimensionales.

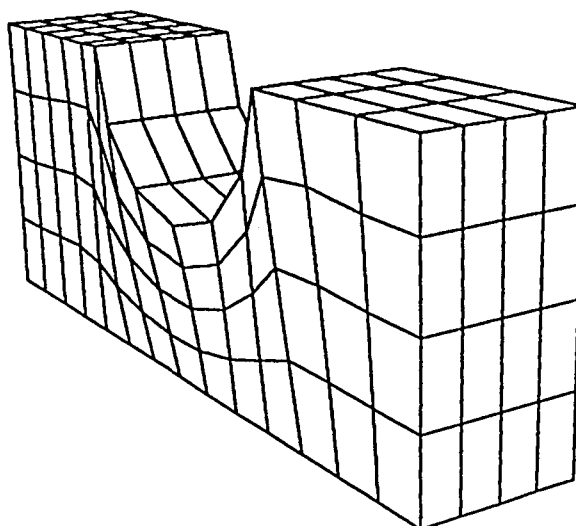


Figura 4-10: Ejemplo de un objeto discretizado por elementos finitos 3-D lineales.

A continuación se indican los pasos a seguir en el modelado y discretización de objetos para el análisis por el FEM, destacando las características del que lo diferencian del BEM. En la sección 4.1.3 se describen con mayor detalle los pasos comunes.

- 1) Diseño del objeto en el módulo *Solid Modeling de I-DEAS*.
- 2) En el familia *Finite Element Modeling & Analysis* se recupera el objeto diseñado en *Solid modeling* o se modela uno objeto con las herramientas que proporciona este módulo.
- 3) El proceso de mallado se inicia con la definición de las áreas de mallado (*mesh areas*) que recubren el objeto. Dichas áreas se definen a partir de las curvas que las limitan.
- 4) En segundo lugar se define el volumen de mallado (*mesh volume*) a partir de la superficie que lo encierra, la cual se ha definido anteriormente mediante las *mesh areas*. La definición del volumen de mallado es imprescindible para especificar el tamaño y forma de los elementos que se van a utilizar, el número de nodos por elemento y el tipo de mallado (regular conformado o libre), ya que

los puntos, curvas y superficies no son suficientes para proceder a la discretización.

6) Una vez formados los volúmenes y decidido el tipo de mallado, se activa la formación de nodos y elementos mediante el comando *Generate mesh*.

El programa I-DEAS permite al usuario elegir entre varios tipos de elementos tridimensionales, los cuales se presentan en la figura 4-11. En la columna de la izquierda se muestran los elementos lineales, con un nodo en cada vértice, en la columna central los elementos parabólicos, con un nodo en cada vértice y en la mitad de cada arista, y en la columna de la derecha los cúbicos, con un nodo en cada vértice y dos en cada arista.

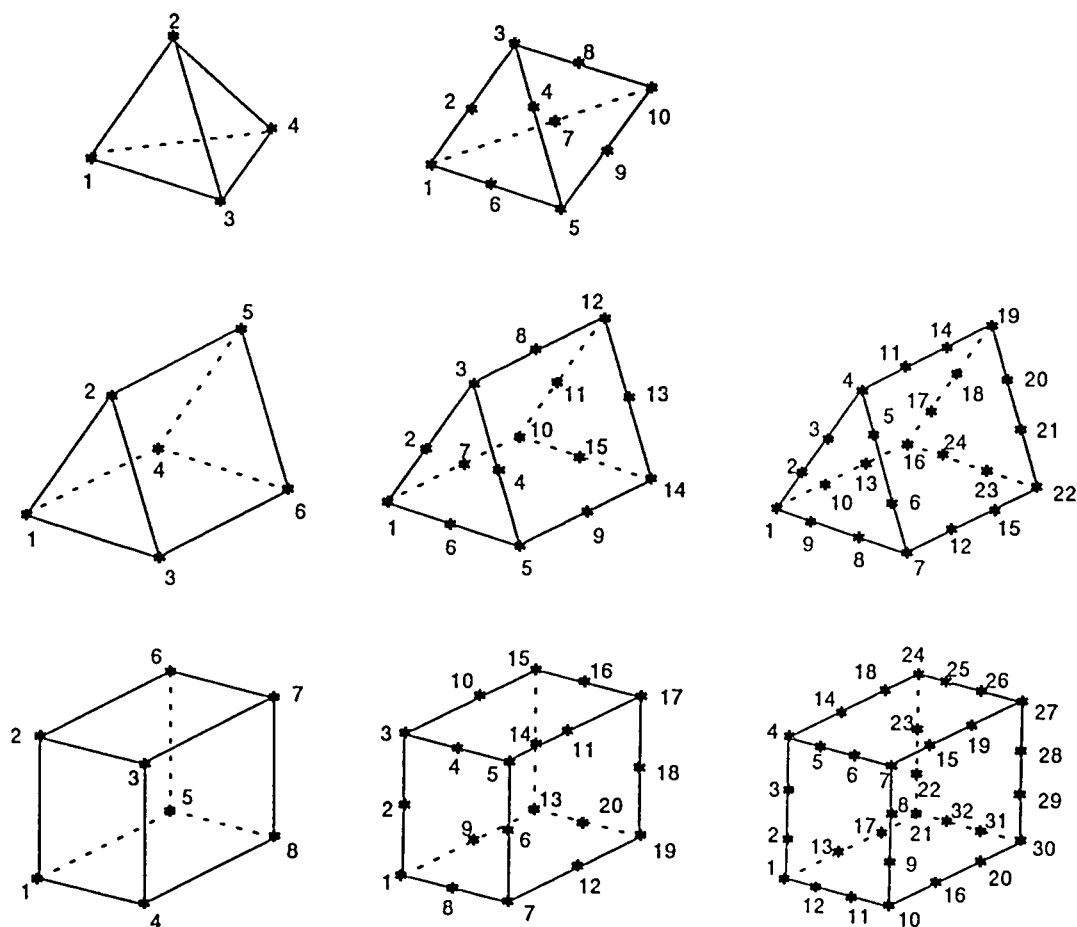


Figura 4-11: Tipos de elementos tridimensionales que permite utilizar I-DEAS para el mallado volumétrico.

Igual que en el mallado superficial, cuanto mayor es el grado de los elementos éstos se adaptan mejor a las superficies curvadas, pero requieren más nodos en su definición y, por tanto, más memoria.

Finalmente, es preciso comentar que la resolución del problema electromagnético 2-D por el FEM requiere un modelo de la sección transversal del objeto cilíndrico bajo estudio. Esta superficie plana se discretiza con elementos bidimensionales de forma idéntica a como se ha descrito en la sección 4.1.3 para el mallado del BEM tridimensional.

4.3 METODO DEL GRADIENTE CONJUGADO + FFT

En la sección 4.1 se ha visto que problemas tridimensionales homogéneos a trozos podían resolverse con una formulación integral y discretización superficial gracias a la aplicación del teorema de equivalencia superficial. Sin embargo, para analizar objetos tridimensionales inhomogéneos es imprescindible una discretización volumétrica.

Debido al elevado número de incógnitas que aparece con la discretización volumétrica, solamente es práctico solucionar el problema utilizando técnicas que reduzcan los requerimientos de memoria y tiempo de proceso. Los ejemplos más utilizados de estas técnicas son:

- El método de los elementos finitos, en el que la reducción de memoria y tiempo de proceso se consigue gracias a la presencia de matrices dispersas. El tipo de modelado que requiere es el mallado en elementos finitos o descomposición del espacio en celdas. Este método se ha implementado en nuestro grupo y ha sido estudiado en la sección 4.2.
- El método del gradiente conjugado con transformada rápida de Fourier (CG+FFT), en el que la reducción de coste computacional se consigue realizando las convoluciones del gradiente conjugado en el dominio transformado de Fourier. Esta técnica necesita un mallado regular en celdas cuadradas (2-D) o rectangulares (3-D), que corresponde a un modelo de "enumeración de ocupación del espacio", que se ha presentado en la sección 2.2.4 b). Esta técnica controla la presencia o ausencia del objeto, o el tipo de material del mismo, en cada posición de la malla regular que divide el espacio. Este método también ha sido implementado en nuestro grupo y por ello se estudia en esta sección.
- El método de diferencias finitas en el dominio del tiempo (FDTD), que evita la inversión de grandes matrices mediante la solución iterativa del problema en el dominio temporal. Este método utiliza el mismo tipo de modelado por

enumeración de ocupación del espacio que el CG+FFT. Como los trabajos de modelado de esta tesis se desarrollaron originalmente para el método CG+FFT, no se va a estudiar el método FDTD, a pesar de que ha sido implementado muy recientemente en nuestro grupo utilizando los modelos del CG+FFT.

4.3.1 Formulación

El método del gradiente conjugado utiliza una formulación integral basada en el teorema de equivalencia superficial. La discretización del operador integral se realiza mediante descomposición en funciones base y peso, de forma análoga al método de los momentos (ver 4.1.2 a). Muchos autores consideran al método del gradiente conjugado como una solución del método de los momentos en la que se utiliza el gradiente conjugado para invertir la matriz de impedancias. Otros, lo consideran como un método independiente que parte de una formulación muy diferente y que también realiza una discretización en funciones base y peso.

a) Equivalencia volumétrica

La versión volumétrica del teorema de equivalencia substituye el objeto del problema original por un conjunto de fuentes equivalentes definidas en el interior del mismo. La figura 4-12 muestra el problema original (a) y el problema equivalente volumétrico (b).

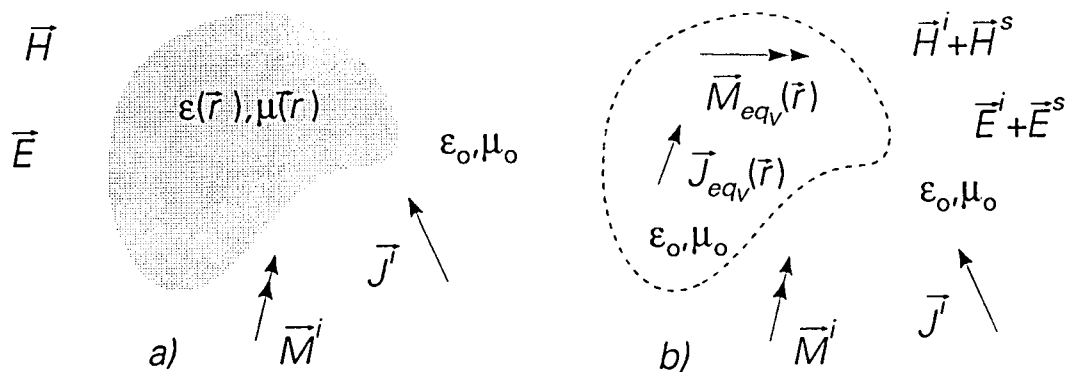


Figura 4-12: Teorema de equivalencia: a) problema original, b) problema equivalente volumétrico

La formulación del problema equivalente volumétrico se obtiene substituyendo en las ecuaciones de Maxwell el medio original por el espacio libre y añadiendo unas corrientes equivalentes que compensen el cambio de medio,

$$\begin{aligned}\nabla \times \vec{E} &= -\vec{M}^i - j\omega \mu(\vec{r}) \vec{H} = -\vec{M}^i - j\omega \mu_0 \vec{H} - \vec{M}_{eqV} \\ \nabla \times \vec{H} &= \vec{J}^i + j\omega \epsilon(\vec{r}) \vec{E} = \vec{J}^i + j\omega \epsilon_0 \vec{E} + \vec{J}_{eqV}\end{aligned}\quad (4.46)$$

donde, para satisfacer la igualdad, las corrientes equivalentes volumétricas son:

$$\vec{M}_{eqV} = j\omega (\mu(\vec{r}) - \mu_0) \vec{H} \quad \vec{J}_{eqV} = j\omega (\epsilon(\vec{r}) - \epsilon_0) \vec{E} \quad (4.47)$$

Las corrientes equivalentes constituyen la incógnita del problema y deben cumplir la ecuación (4.47)

$$\vec{E} = \frac{\vec{J}_{eqV}}{j\omega (\epsilon(\vec{r}) - \epsilon_0)} \quad \vec{H} = \frac{\vec{M}_{eqV}}{j\omega (\mu(\vec{r}) - \mu_0)} \quad (4.48)$$

Calculando los campos \vec{E}^s y \vec{H}^s a partir de la radiación de las fuentes equivalentes e imponiendo la condición (4.48) queda el sistema de ecuaciones funcionales, definidas en todo el volumen,

$$\begin{aligned}\frac{\vec{J}_{eqV}}{j\omega (\epsilon(\vec{r}) - \epsilon_0)} - \vec{E}^i &= L_E(\vec{J}_{eqV}, \vec{M}_{eqV}) \\ \frac{\vec{M}_{eqV}}{j\omega (\mu(\vec{r}) - \mu_0)} - \vec{H}^i &= L_H(\vec{J}_{eqV}, \vec{M}_{eqV})\end{aligned}\quad (4.49)$$

La resolución de este sistema permite obtener el valor de las incógnitas \vec{J}_{eqV} y \vec{M}_{eqV} . Habitualmente $\mu(\vec{r}) = \mu_0$ y $\vec{M}_{eqV} = 0$, con lo que sólo es necesario resolver la ecuación de \vec{E}^i con incógnita \vec{J}_{eqV} .

b) Formulación de las ecuaciones integrales

Si se hace un desarrollo paralelo al expuesto en el apartado equivalente para el método de los elementos de contorno se llega a una ecuación similar, sólo que en este caso al trabajar con equivalencia volumétrica se tienen corrientes equivalentes volumétricas, con lo que se llega a la siguiente expresión para calcular los campos dispersados:

$$\begin{aligned}\vec{E}_S &= L_E(\vec{J}_{eqV}, \vec{M}_{eqV}) = \int_V [-j\omega \mu \vec{J}_{eqV} G - \vec{M}_{eqV} \times \nabla' G - \frac{\nabla' \vec{J}_{eqV}}{j\omega \epsilon} \nabla' G] ds' \\ \vec{H}_S &= L_H(\vec{J}_{eqV}, \vec{M}_{eqV}) = \int_V [-j\omega \epsilon \vec{M}_{eqV} G + \vec{J}_{eqV} \times \nabla' G - \frac{\nabla' \vec{M}_{eqV}}{j\omega \mu} \nabla' G] ds'\end{aligned}\quad (4.50)$$

4.3.2 Discretización

a) Gradiente conjugado

En la sección 4.1.2 se ha mostrado la solución numérica de una ecuación funcional del tipo $LX=Y$. Otra forma de resolver esta ecuación puede obtenerse minimizando un funcional proporcional a la norma del residuo

$$f(X) = \frac{1}{2} \|R\|^2 = \frac{1}{2} \|Y - LX\|^2 \quad (4.51)$$

La solución se obtiene cuando el gradiente de $f(X)$ es nulo

$$\nabla f(X) = L^a LX - L^a Y = -L^a R = 0 \quad (4.52)$$

lo cual equivale a resolver la ecuación $L^a LX = L^a Y$ que conduce a la solución de mínimos cuadrados ($\|R\|$ mínima) del problema original $LX = Y$.

Para minimizar el funcional $f(X)$ se actualiza de forma iterativa una aproximación inicial $X^{(0)}$

$$X^{(i+1)} = X^{(i)} + \lambda^{(i)} P^{(i)} \quad (4.53)$$

donde $P^{(i)}$ es una dirección en la cual $f(X)$ disminuye. $\lambda^{(i)}$ es un escalar que minimiza $f(X)$ a lo largo de la dirección $P^{(i)}$ y, por tanto, debe ser tal que el gradiente del funcional en el mínimo $X^{(i+1)}$ sea normal a la dirección de avance $P^{(i)}$:

$$\langle \nabla f(X^{(i+1)}), P^{(i)} \rangle = 0 \quad (4.54)$$

La elección de la dirección $P^{(i)}$ conduce a distintos métodos de minimización. Por ejemplo, en el método del gradiente descendiente se toma $P^{(i)} = -\nabla f(X^{(i)})$. En este método el número de iteraciones es grande porque al minimizar el funcional en cada iteración se "estropea" la minimización en iteraciones anteriores. Esto no sucedería si en cada iteración el nuevo gradiente $\nabla f(X^{(i+1)})$, además de ser ortogonal a $P^{(i)}$, fuera ortogonal a todos los $P^{(j)}$ anteriores

$$\langle \nabla f(X^{(i+1)}), P^{(j)} \rangle = 0 \quad \forall j \leq i \quad (4.55)$$

Esta condición implica que el gradiente es ortogonal al subespacio generado por todas las direcciones $P^{(j)}$ en las que se ha minimizado en iteraciones anteriores. Las direcciones $P^{(j)}$ que cumplen esta propiedad se llaman "direcciones conjugadas" (ver capítulo 7) respecto al operador $L^a L$. Como sólo es posible minimizar $f(X)$ en un

subespacio de N dimensiones, debido a la aproximación (4.13) $X \sim X_N = \sum a_j x_j$, bastan N iteraciones para que la proyección del gradiente en este subespacio sea nula:

$$\langle \nabla f(X^{(N)}), P^{(j)} \rangle = 0 \quad \forall j \leq N-1 \Rightarrow \nabla f(X^{(N)}) = 0 \quad (4.56)$$

En estas condiciones, X_N es la solución de $LX = Y$ con residuo mínimo, dentro del subespacio generado por las funciones base.

Para la implementación del algoritmo del gradiente conjugado (sección 7.3), es necesario discretizar numéricamente las funciones y los operadores. Para ello se expande la función incógnita X en funciones base (4.13) y el resultado de aplicar los operadores se expande en funciones peso (4.16)-(4.18). De esta forma, el operador L discretizado se representa mediante el producto del vector correspondiente a la función discretizada por una matriz A tal que $A_{ij} = \langle W_i, Lx_j \rangle$, mientras que el operador adjunto L^a se representa mediante el producto por la matriz adjunta o Hermítica A^* .

Cuando el operador L es de tipo convolución, por ejemplo la EFIE y la MFIE, y se discretiza mediante funciones base definidas en subdominios que forman una malla regular, la matriz A de $Aa = b$ es de Toeplitz o de Toeplitz a bloques. En la figura 4-13 se muestran ejemplos de discretizaciones en una, dos o tres dimensiones que producen este tipo de matrices.

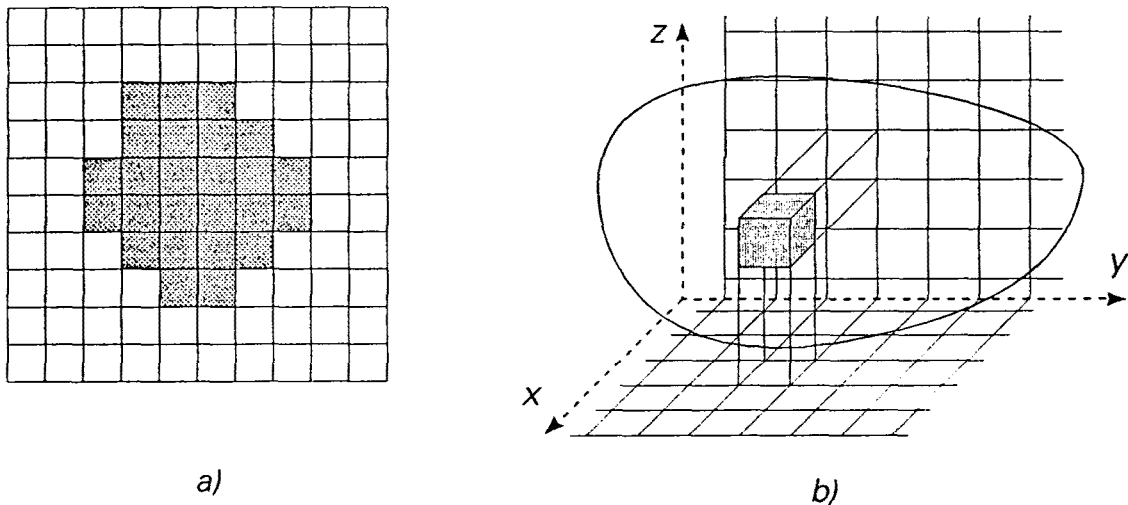


Figura 4-13: Discretización con malla regular para obtener una matriz de Toeplitz o de Toeplitz a bloques: a) Problemas 2-D, b) Problemas 3-D.

En este caso, las operaciones $AP^{(k)}$ y $A^*R^{(k)}$ del método del gradiente conjugado discretizado tienen la forma de una convolución discreta en una, dos o tres dimensiones y pueden implementarse eficientemente mediante la transformada rápida de Fourier (FFT). Ahora el número de operaciones en cada iteración es del orden de

$N \log N \sim N$ en lugar de N^2 y, además, no es necesario almacenar la matriz A completa (tamaño N^2) sino sólo los elementos no redundantes (tamaño N).

Sin embargo, en un problema tridimensional arbitrario, para obtener una malla regular es necesaria una discretización volumétrica (figura 4-13). Por ello, si el tamaño eléctrico de los elementos de la malla regular se mantiene constante, el número de incógnitas aumenta con la frecuencia en la forma $N \sim f^3$ en lugar de $N \sim f^2$ como sucede con discretización superficial.

La técnica del gradiente conjugado tiene otras muchas aplicaciones. De entre ellas se puede destacar la optimización de funciones no lineales sin restricciones. En esta tesis se ha planteado el problema de optimización de la sección recta radar de objetos arbitrarios (ver capítulo 7). Uno de los métodos que se han aplicado ha sido el método del gradiente conjugado. En las secciones 7.3 y 7.4 se hace un estudio más exhaustivo de este método presentando tanto su algoritmo como las propiedades del mismo.

b) Funciones base "tejado" (*roof top*)

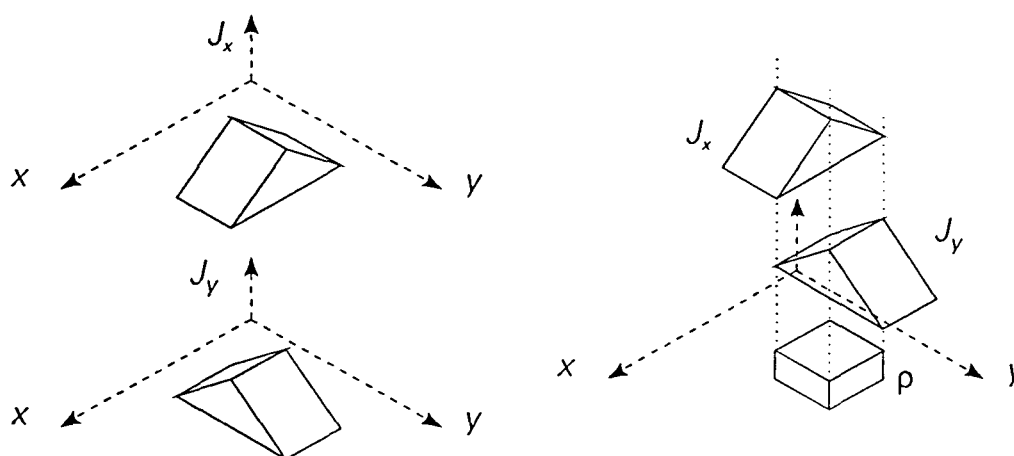


Figura 4-14: Funciones base "tejado" definidas sobre subdominios rectangulares del plano

Con objeto de evitar las singularidades en la distribución de carga que aparecen cuando el modelo discreto de \vec{J} es discontinuo, las funciones base que representan \vec{J} se eligen de forma que $J_x(x,y)$ sea continua en x y $J_y(x,y)$ continua en y . En este caso la densidad volumétrica de carga

$$-j\omega\rho = \nabla \cdot \vec{J} = \frac{\partial}{\partial x} J_x + \frac{\partial}{\partial y} J_y \tag{4.57}$$

no presenta singularidades.

En la figura 4-14 se muestran las funciones "tejado" ("roof top"), iguales a la combinación de una función pulso en x y una función triángulo en y , o viceversa. Juntamente con estas funciones base suelen emplearse funciones peso del tipo "hoja de navaja" ("razor blade"), que se representan en la figura 4-15.

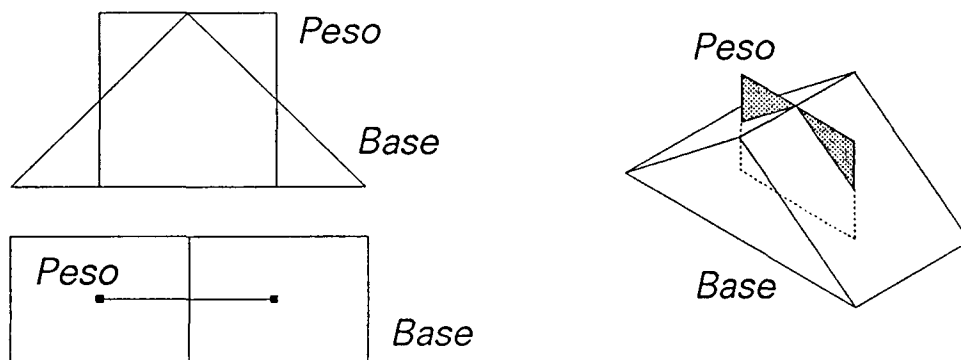


Figura 4-15: Funciones peso "hoja de navaja" en combinación con funciones base "tejado".

En el caso de discretización volumétrica, pueden emplearse funciones base y peso tridimensionales análogas a las funciones "tejado" y "hojas de navaja" de dos dimensiones.

4.3.3 Modelado

En el departamento se ha utilizado la técnica de modelado por enumeración de ocupación del espacio en varios campos del análisis electromagnético, como tomografía de microondas [Broquetas'91], [Rius'92a] y análisis de la penetración electromagnética en tejidos biológicos [Vaquer'93]. Antes de la realización de esta tesis era habitual realizar los modelos de forma totalmente manual, marcando con el ratón del ordenador las celdas ocupadas por el objeto e indicando el tipo de material de las mismas. Este procedimiento manual, que era tedioso en el caso de objetos bidimensionales, se hacía totalmente inviable en el caso tridimensional.

La adquisición del software I-DEAS, que permite generar mallados de elementos finitos, no solucionó el problema, pues el cuadrado y el cubo perfectos no forman parte de las

librerías de elementos finitos 2-D y 3-D disponibles. Por ello, no es posible generar mallados regulares con I-DEAS.

Por este motivo, en esta tesis se ha desarrollado un algoritmo para generar automáticamente mallados regulares de cuadrados 2-D o cubos 3-D a partir de mallados en elementos finitos.

Este algoritmo necesita un mallado en elementos finitos triangulares lineales en 2-D o tetraédricos lineales en 3-D. Dada un malla regular, para todos los puntos centrales de las celdas cuadradas (2-D) o cúbicas (3-D) comprueba si están contenidos en alguno de los triángulos o tetraedros del mallado en elementos finitos. En caso afirmativo, asigna el material de este elemento finito a la celda cuadrada o cúbica de la malla regular.

Evidentemente, la doble iteración para todas las celdas del mallado regular y para todos los elementos finitos se realiza de forma optimizada: para cada elemento finito solamente se comprueban las celdas del mallado regular que están contenidas en el menor rectángulo (2-D) o paralelepípedo (3-D) que circunscribe el elemento finito. Los principales pasos del algoritmo son los siguientes:

- Para cada elemento finito (EF):
 - Calcular las esquinas del menor rectángulo (2-D) o paralelepípedo (3-D) que circunscribe el EF.
 - Para cada celda del mallado regular contenida en este rectángulo o paralelepípedo:
 - Comprobar si el centro de la celda está contenido en el EF. Si lo está, asignar el material del EF al material de la celda.

Para implementar este algoritmo es necesario poder determinar si un punto cualquiera del plano está contenido dentro de un triángulo o si un punto del espacio está contenido en un tetraedro. El método desarrollado para estas tareas se muestra en las figuras 4-16 y 4-17 respectivamente.

En el caso de un triángulo en el plano (x,y) , un punto P estará en su interior si para todos los vértices se cumple que el vector que va del vértice al punto P se encuentra a la derecha de la arista que parte de este vértice y deja el triángulo a la derecha. Es decir, la rotación que va de la arista al vector debe realizarse en el sentido de las agujas del reloj o positivo. Matemáticamente, el producto vectorial de dicha arista por el vector que va del vértice al punto P debe tener el sentido de las z negativas. Por ejemplo, en el triángulo de la figura 4-16, el vector AP se encuentra a la derecha del vector AB ,

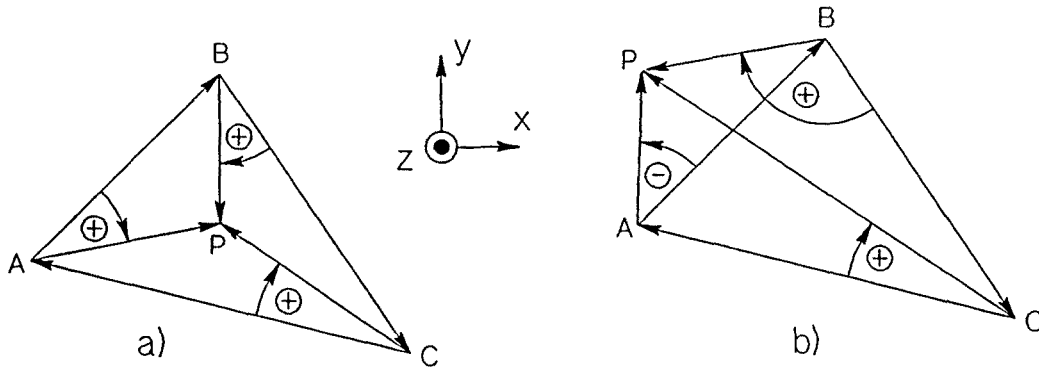


Figura 4-16: Determinación de si un punto P del plano (x,y) está contenido dentro de un triángulo.

mirando desde el vértice A hacia el centro del triángulo: la rotación desde AB hacia AP tiene sentido positivo. Si esta condición se cumple para los tres vértices, el punto P está en el interior del triángulo. En la figura 4-16 se compara el caso del punto P dentro del triángulo (a) con el punto P fuera del mismo (b).

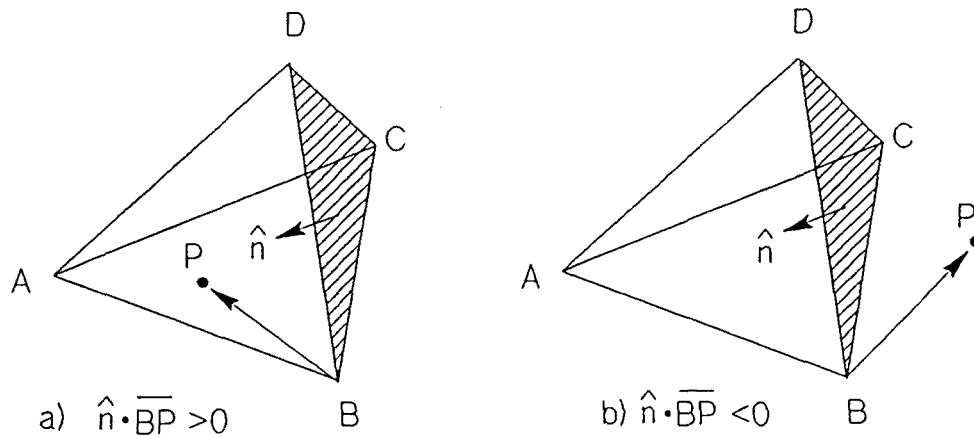


Figura 4-17: Determinación de si un punto P del espacio está contenido en el interior de un tetraedro.

En el caso de un tetraedro se calcula el vector normal a cada una de las caras que apunta hacia el interior del tetraedro. Si para todas las caras el producto escalar de este vector por el vector que va de un vértice de esta cara al punto P es positivo entonces el punto P está en el interior de tetraedro.

En las figuras 4-18 y 4-19 se muestran ejemplos de objetos 2-D y 3-D modelados por elementos finitos con I-DEAS y el resultado de la conversión a mallado regular de enumeración de ocupación del espacio.

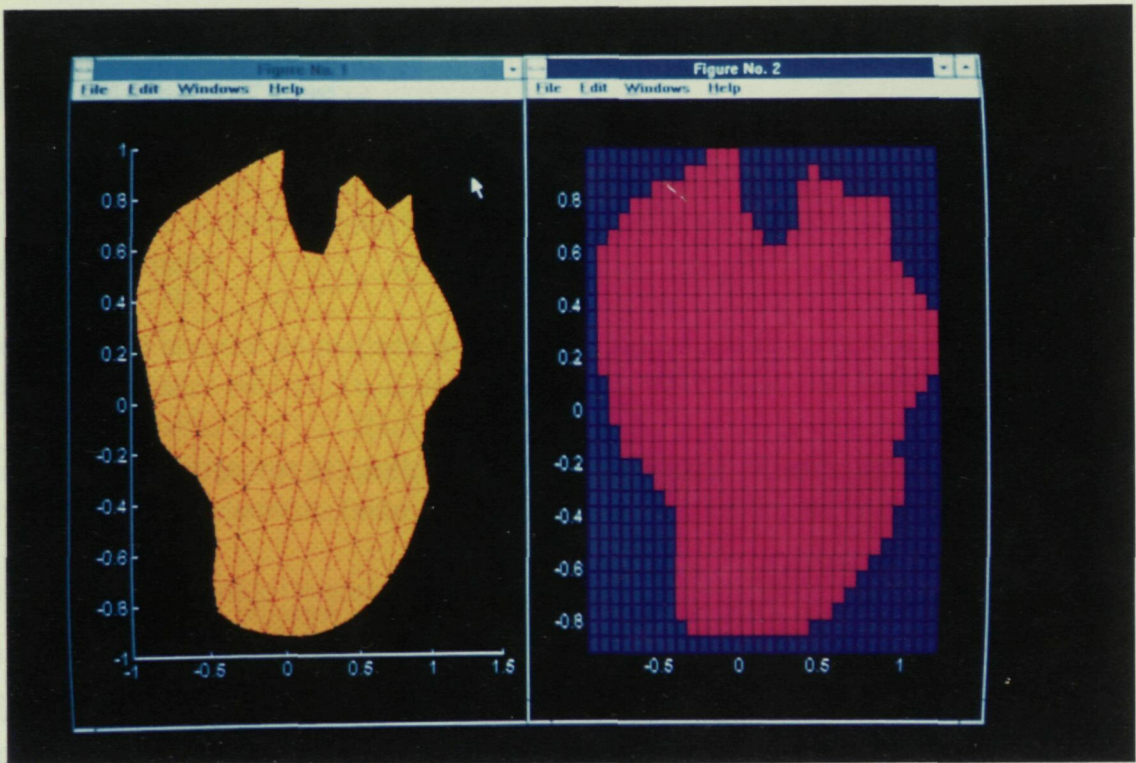


Figura 4-18: Generación de un mallado regular en dos dimensiones a partir de un mallado en elementos finitos triangulares lineales.

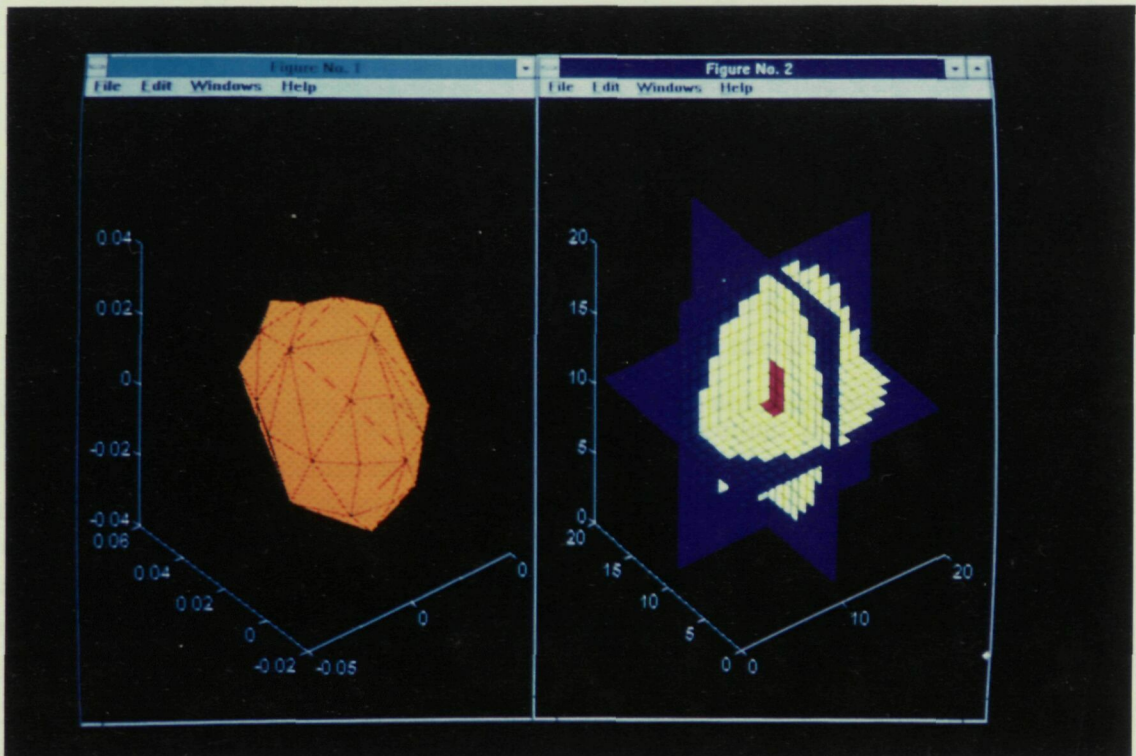


Figura 4-19: Generación de un mallado regular tridimensional a partir de un mallado en elementos finitos tetraédricos lineales.

CAPITULO 5

PREDICCIÓN DE RCS

El principal objetivo de los algoritmos presentados en este capítulo es el cálculo de la sección recta radar en tiempo real de cualquier objeto de geometría arbitraria usando una estación de trabajo gráfica. Este método permite integrar fácilmente la predicción de la RCS con un paquete software de modelado geométrico consiguiéndose una herramienta eficiente de modelado, diseño y análisis interactivos de aviones con unas especificaciones de RCS. En el capítulo 7 se estudiará en detalle la aplicación de los algoritmos de procesado gráfico a la predicción de RCS.

En este capítulo, primeramente se va a hacer un estudio de las técnicas clásicas de cálculo de RCS en alta frecuencia, planteando los problemas que presentan. A continuación se va a hacer una descripción del nuevo método de cálculo de RCS resaltando las innovaciones que presenta para resolver las dificultades de las técnicas clásicas, como son aprovechamiento de las herramientas gráficas de las estaciones de trabajo y el modelado por superficies paramétricas. En las dos secciones siguientes se desarrollan brevemente los dos etapas principales del proceso: la sección 5.3 se dedica a la solución adoptada por el programa *GRECO* en la realización del análisis geométrico, mientras que la sección 5.4 expone la solución adoptada en el análisis electromagnético para obtener las distintas aproximaciones de alta frecuencia a partir del procesado de la imagen visualizada en la pantalla de la estación gráfica.

5.1 TECNICAS CLASICAS DE PREDICCIÓN DE RCS

El cálculo de la RCS de objetos grandes y complejos incluye diferentes fenómenos de difracción, tales como reflexión especular, difracción en las aristas, reflexiones múltiples, efectos de sombra, difracción de las ondas de superficie en las discontinuidades, creeping waves, etc. Las técnicas numéricas presentadas en el capítulo 4 tienen en cuenta todos estos efectos, obteniendo resultados muy precisos, pero, por otro lado, el costo computacional resulta prohibitivo para blancos eléctricamente muy grandes.

Afortunadamente, como es bien conocido, la difracción en alta frecuencia es un fenómeno local: la principal contribución se debe a los puntos de reflexión especular en las superficies y aristas. Por ello, la contribución de los efectos de segundo orden normalmente no es importante y la RCS de blancos complejos se puede predecir con un grado de precisión razonable, por aproximaciones asintóticas de alta frecuencia. La reflexión superficial se analiza por óptica física (PO) y la contribución de la difracción en aristas se analiza por el método de las corrientes equivalentes (MEC). Con estos métodos el campo lejano se obtiene a través de una integral de superficie o de línea, de unas corrientes equivalentes.

Las principales dificultades de la predicción de la RCS de objetos complejos mediante técnicas de alta frecuencia se encuentran en el cálculo de las integrales de línea y superficie sobre una forma arbitraria, la cual se define con un software de modelado geométrico CAD. En ambos casos, es necesario conocer la normal a la superficie en cada punto del modelo geométrico, para poder realizar los cálculos electromagnéticos. Un problema adicional es que, de acuerdo con la teoría de alta frecuencia, estas corrientes equivalentes se consideran nulas sobre las superficies o las aristas no iluminadas por la onda incidente (por ejemplo zonas que quedan eclipsadas por otras partes del avión), con lo que las integrales de superficie o línea se extienden solamente sobre la región del blanco iluminada por el campo incidente y, por tanto, hay que diferenciar unas regiones de las otras.

Las técnicas clásicas de predicción de sección recta radar en alta frecuencia se basan en el modelado mediante facetas y aristas. Las aproximaciones de alta frecuencia se pueden aplicar separadamente a cada faceta, con lo que se obtiene una fórmula cerrada para óptica física y para el método de las corrientes equivalentes. La RCS del blanco completo se calcula sumando coherentemente la contribución separada de todas las facetas y aristas iluminadas. Las técnicas clásicas también son capaces de considerar las interacciones entre facetas y/o aristas para mejorar la aproximación. Se requiere, además, un procesado adicional para suprimir la contribución de las partes del modelo no iluminadas. La identificación de las regiones no iluminadas en el modelo geométrico del blanco es un problema difícil que incrementa mucho los tiempos de cálculo totales. Por tanto, los algoritmos clásicos de alta frecuencia deben dedicar un gran esfuerzo al análisis geométrico del modelo antes de poder realizar los cálculos propios del análisis electromagnético.

Además, hay que tener en cuenta que el coste computacional necesario para la identificación de las zonas ocultas y los cálculos de reflexiones múltiples se incrementan rápidamente con el aumento del número de facetas en los modelos complejos. Esta es la razón por la que las técnicas clásicas se implementan normalmente sobre ordenadores muy potentes, distintos de las estaciones gráficas utilizadas para modelar blancos. Por contra, la técnica de procesado gráfico es capaz de obtener la RCS en tiempo real en

la misma estación gráfica en la que se diseñan los modelos de análisis. En la sección siguiente se exponen las herramientas utilizadas para conseguirlo.

5.2 METODO DE PROCESADO GRAFICO

Muchas de las dificultades descritas en la sección anterior se resuelven con la nueva técnica de procesamiento gráfico. Esta técnica se basa en procesar una imagen tridimensional del blanco usando una estación gráfica (por ejemplo, la imagen generada por el software de modelado geométrico CAD). Si el punto de vista del blanco está situado en la posición del radar monoestático, entonces la imagen en la pantalla contiene solamente las superficies y aristas iluminadas por el radar: las que quedan ocultas al radar no son visibles desde el punto de vista del observador, porque han sido eliminadas por el hardware de visualización tridimensional.

Además, si la imagen se obtiene usando iluminación y representación tridimensional, se pueden definir los parámetros de las fuentes de iluminación de forma que las componentes roja, verde y azul (R, G, B) de cada pixel de la imagen correspondan respectivamente a las componentes (n_x , n_y , n_z) del vector unitario normal a la superficie en aquel punto. La memoria de pantalla de la estación de trabajo contiene, entonces, la siguiente información en cada pixel: las coordenadas x , y , z y las componentes de color R , G , B , las cuales son iguales a las componentes de la normal (n_x , n_y , n_z). Por tanto, la memoria de pantalla tiene disponible toda la información necesaria para obtener las aproximaciones de alta frecuencia. Esta información se proporciona en tiempo real por el acelerador de hardware gráfico de la estación de trabajo.

De acuerdo con lo comentado en los párrafos anteriores se llega a que el problema de predicción de la RCS se reduce solamente a los cálculos electromagnéticos, los cuales incluyen las integrales de línea y superficie de alta frecuencia. Estas se obtienen procesando la información x , y , z , n_x , n_y , n_z de las superficies iluminadas del blanco en la pantalla de la estación de trabajo.

Además, se pueden obtener los coeficientes de reflexión de Fresnel para cada punto de la superficie, porque la normal a cada pixel es conocida. Por tanto, la integral de superficie de óptica física se calcula mediante la suma coherente de la contribución de cada pixel. Para calcular la integral de línea del MEC se identifican las aristas del modelo mediante la detección de discontinuidades del vector unitario normal a la superficie. Se puede conocer el ángulo interior a la cuña y la orientación de la arista relativa al observador, porque se conocen las normales unitarias a ambos lados de la arista y la dirección de observación. A partir de estos parámetros se pueden calcular los coeficientes de difracción de longitud incremental (ILDC) para cada pixel de la imagen que quede sobre una arista y se puede calcular la integral de línea del MEC sumando coherentemente las contribuciones de los ILDC.

Si la estación de trabajo posee acelerador de hardware gráfico (ver capítulo 3) la imagen del blanco y la información x , y , z , n_x , n_y , n_z para cada pixel puede ser obtenida en tiempo real, de forma que el tiempo de CPU sólo incluye los cálculos de la parte electromagnética, dejando la parte geométrica al hardware gráfico. Por tanto, se obtiene la predicción de la RCS en tiempo real de un objeto arbitrario sea cual sea su complejidad.

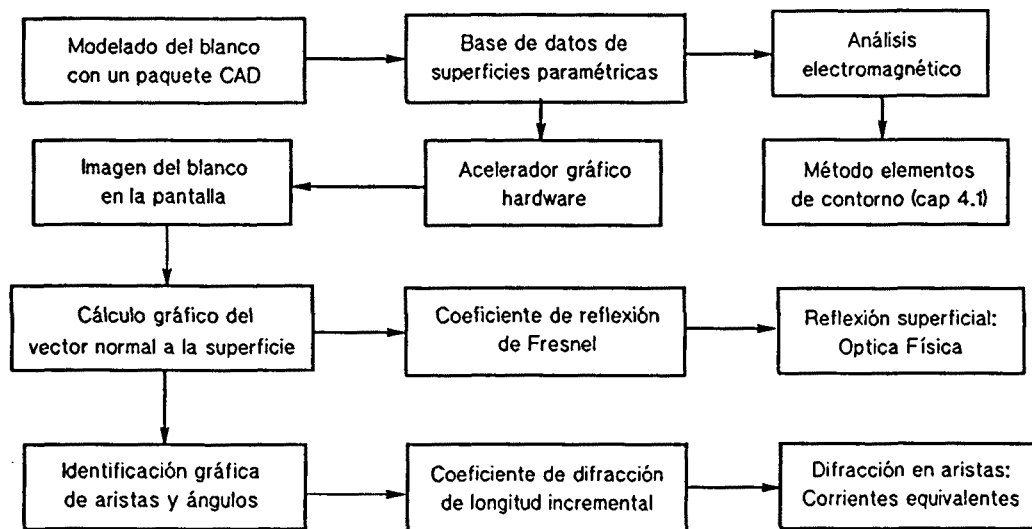


Figura 5-1: Diagrama de bloques con los diferentes pasos del método de procesamiento gráfico para la predicción de RCS.

Tal como se muestra en la figura 5-1 las distintas etapas del cálculo de RCS por procesamiento gráfico en el programa *GRECO* son:

- 1 - Modelado geométrico del blanco con un paquete CAD.
- 2 - Visualización del blanco en la pantalla por el acelerador gráfico hardware en tiempo real.
- 3 - Obtención de las coordenadas x , y , z , n_x , n_y , n_z de cada punto de las superficies iluminadas mediante procesamiento gráfico de la imagen.
- 4 - Se llevan a cabo las siguientes aproximaciones de alta frecuencia:
 - a) Reflexión en las superficies conductoras mediante la aproximación de óptica física.

b) Reflexión en superficies recubiertas por materiales RAM mediante óptica física y condición de contorno de impedancia.

c) Difracción en las aristas por el método de las corrientes equivalentes, usando coeficientes de difracción de longitud incremental.

Los tres primeros pasos incluyen la parte geométrica y gráfica de cálculos, a la que se dedica la sección 5.3 de este capítulo, mientras que el cuarto paso, desarrollado brevemente en el apartado 5.4, realiza el cálculo electromagnético.

5.3 ANALISIS GEOMETRICO

El primer paso del programa *GRECO* es el modelado del blanco. Para ello, se utiliza un paquete software de modelado geométrico de sólidos, como se ha comentado en el capítulo 2. El programa *GRECO* es capaz de procesar tanto modelos definidos por facetas y aristas como los definidos por superficies paramétricas B-spline racionales no uniformes (NURB), estudiados en el anexo 2A.

La representación mediante NURBS, como se ha comentado en el capítulo 2, presenta sobre el modelado por facetas clásico las siguientes ventajas.

- La aproximación de un objeto complejo requiere un número muy elevado de facetas mientras que el número NURBS que lo definen es mucho menor, consiguiendo una aproximación mucho mejor de las superficies curvadas arbitrarias. Ello repercute en una definición más compacta del modelo con una cantidad de información mucho menor, con lo cual disminuye muchísimo la cantidad de memoria ocupada y el tiempo de proceso. Otra ventaja para aplicaciones de algoritmos de optimización es que el número de grados de libertad es también mucho menor.

- En segundo lugar tenemos que los diseños mediante facetas introducen aristas y vértices artificiales entre las facetas, mientras que la superficie diseñada por NURBS presenta una apariencia suave y más aproximada a la superficie real. Por este motivo los cálculos de RCS realizados con modelos de facetas contienen el llamado "ruido de facetas", que no aparece en los modelos de superficies paramétricas.

Por todo ello, los modelos de superficies paramétricas se han utilizado en el programa *GRECO* y en otras implementaciones de métodos de alta frecuencia desarrollados en la Universidad de Cantabria [Pérez'92] y en la Universidad de Vigo [Hermida'93].

El segundo paso es la representación del modelo tridimensional de superficies paramétricas en tiempo real. Las partes ocultas de la escena son eliminadas de la imagen por el hardware gráfico, con lo que la imagen en la pantalla de la estación de trabajo contiene solamente las superficies visibles desde el punto de vista del observador. Todo lo relacionado con las técnicas de visualización realista de imágenes tridimensionales se encuentra en el capítulo 3.

El procesado gráfico de la imagen, tercer paso del algoritmo, comprende la identificación de las superficies no visibles por el observador, la obtención de las tres componentes de la normal a la superficie en cada pixel, la detección de aristas y el cálculo de los radios de curvatura. En las secciones que se incluyen a continuación se comenta la solución adoptada para cada uno de estos cálculos geométricos. En la descripción de todos ellos las coordenadas x , y se corresponden a las coordenadas 2D de la pantalla, mientras que la coordenada z define la dirección normal al plano de la pantalla. Si la posición del observador es la misma que la del punto de vista (en el caso de radar monoestático) el resultado es que la coordenada z de cada pixel coincide con la distancia del observador a cada elemento de superficie (figura 5-2). Desde el punto de vista electromagnético, esta información de la coordenada z es de gran importancia para obtener la suma coherente de la contribución de alta frecuencia de cada elemento de superficie.

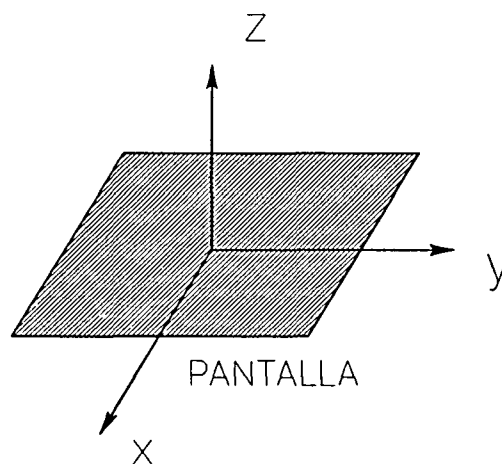


Figura 5-2: Ejes de coordenadas para el procesado gráfico.

5.3.1 Identificación de superficies ocultas

Como se ha comentado al principio de este capítulo, una de las mayores dificultades que presenta el cálculo de la integral de superficie de óptica física por las técnicas clásicas (aproximación por facetas) es la detección de las regiones de sombra. Sin embargo, *GRECO* no necesita enfrentarse a este problema porque las superficies ocultas de la imagen han sido eliminadas por el acelerador gráfico hardware en el proceso de visualización.

Una forma sencilla de validar la eliminación de las superficies ocultas y la obtención correcta de la coordenada z es el análisis de un sistema formado por dos esferas, tal como el que se muestra en la figura 5-3. La distancia entre las esferas es λ y el radio 0.1λ . La gráfica que se presenta a la derecha de la figura es la RCS calculada por *GRECO* normalizada respecto a la RCS de una única esfera. Puede comprobarse que cuando una esfera está ocultando a la otra, para ángulos de aspecto próximos a los 90° , la RCS del sistema de 2 esferas es igual a la RCS de la única que queda visible.

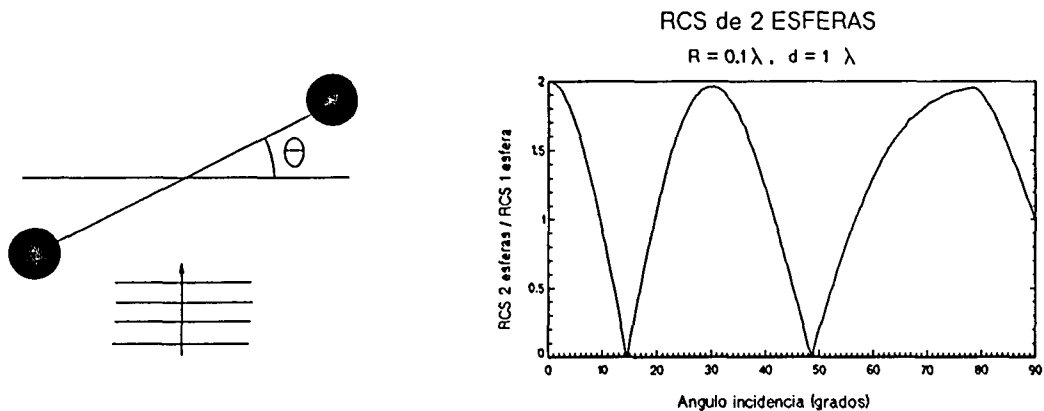


Figura 5-3: RCS de un sistema de dos esferas mediante el cual se valida la eliminación de las partes ocultas.

5.3.2 Cálculo de las normales a la superficie

Si la escena se visualiza utilizando el modelo de iluminación local de Phong (ver sección 3.3.1) el color de cada pixel depende únicamente de la normal a la superficie en este punto y de la situación del observador y de las fuentes de luz. Como las posiciones de las fuentes de luz y del observador son ambas conocidas, es posible

obtener la normal a la superficie de cada pixel de la imagen a partir de la información del color.

De acuerdo con el modelo de iluminación de Phong, ver apartado 3.3.1, cuando la reflexión en la superficie es difusa, no especular, la luminosidad de cada pixel se calcula separadamente para cada color (R , G , B) utilizando la ecuación:

$$L_d = L_s \alpha_L \rho_d \hat{n} \cdot \hat{r}_i \quad (5.1)$$

donde L_d es la luminosidad difusa de un color, L_s la intensidad de la fuente de luz, de este color, α_L la atenuación de la luz por propagación en el medio, ρ_d el coeficiente de reflexión difusa, \hat{n} el vector unitario normal a la superficie en cada pixel y \hat{r}_i el vector unitario en la dirección de incidencia (ver figura 5-4).

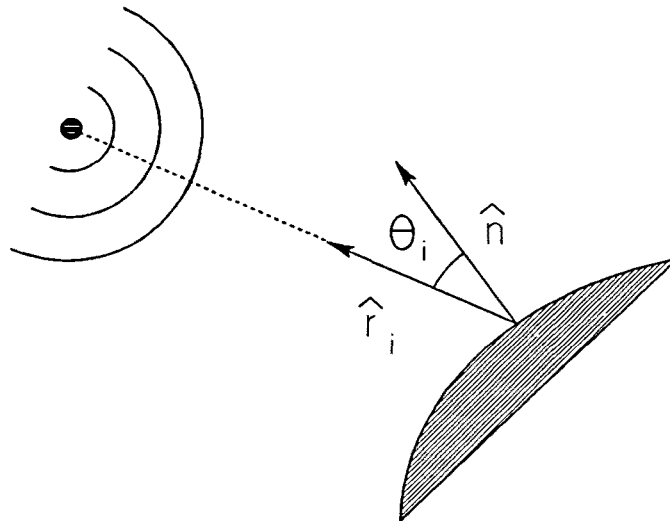


Figura 5-4: Geometría de la reflexión difusa según el modelo de iluminación de Phong.

Si en la ecuación (5.1) los coeficientes L_s , α_L , ρ_d son iguales a la unidad, se puede calcular la luminosidad del pixel como la proyección $\hat{n} \cdot \hat{r}_i$ del vector unitario normal a la superficie en la dirección de iluminación (ver figura 5-4). Por tanto, si se ilumina la imagen con tres fuentes de luz de colores puros verde, rojo y azul localizadas cada una de ellas en uno de los tres ejes de coordenadas, las tres componentes de color para cada pixel son iguales a las componentes (n_x , n_y , n_z) del vector unitario normal a la misma:

La figura 5-5 muestra una imagen del avión stealth F-117 iluminado según indica la ecuación (5.2). El color rojo puro significa que el vector unitario normal a la superficie es horizontal (dirección eje x), el color azul puro que es vertical (dirección eje y) y el color verde que es perpendicular a la pantalla (dirección eje z). Cuando la dirección de la normal no es paralela a ninguna de las tres direcciones de los ejes de coordenadas,

$$\begin{aligned}
 \text{ROJO: } \hat{r}_i = \hat{x} &\Rightarrow R = \hat{n} \cdot \hat{x} = n_x \\
 \text{AZUL: } \hat{r}_i = \hat{y} &\Rightarrow B = \hat{n} \cdot \hat{y} = n_y \\
 \text{VERDE: } \hat{r}_i = \hat{z} &\Rightarrow G = \hat{n} \cdot \hat{z} = n_z \\
 \hat{n} &= (n_x, n_y, n_z)
 \end{aligned}
 \tag{5.2}$$

la combinación de las tres componentes de color (R , G , B) es igual a las componentes del vector unitario normal (n_x , n_y , n_z).

Como las tres componentes de color son siempre cantidades positivas, hay una ambigüedad en el signo (sentido) de las componentes del vector. Para obtener los valores positivos, el acelerador gráfico visualiza solamente las superficies iluminadas, con $\cos\theta_i > 0$, y elimina aquellas que se encuentran en la cara posterior del objeto, con $\cos\theta_i < 0$. Para obtener los valores positivos y negativos de (n_x , n_y , n_z) es necesario iluminar al blanco tanto desde la dirección positiva como desde la negativa, utilizando diferentes fuentes de luz. Como necesitamos un total de 6 fuentes de luz para obtener los seis valores positivos y negativos de (n_x , n_y , n_z) y como sólo hay 3 componentes de color independientes (R , G , B), es necesario visualizar dos imágenes separadas. La figura 5-6 muestra las dos imágenes de un misil genérico, definido por N. Youssef en [Youssef'89], visualizadas con las 6 fuentes de luz. La combinación de los tres colores en cada pixel de las dos imágenes corresponden a las componentes (n_x , n_y , n_z) positiva y negativa del vector unitario normal.

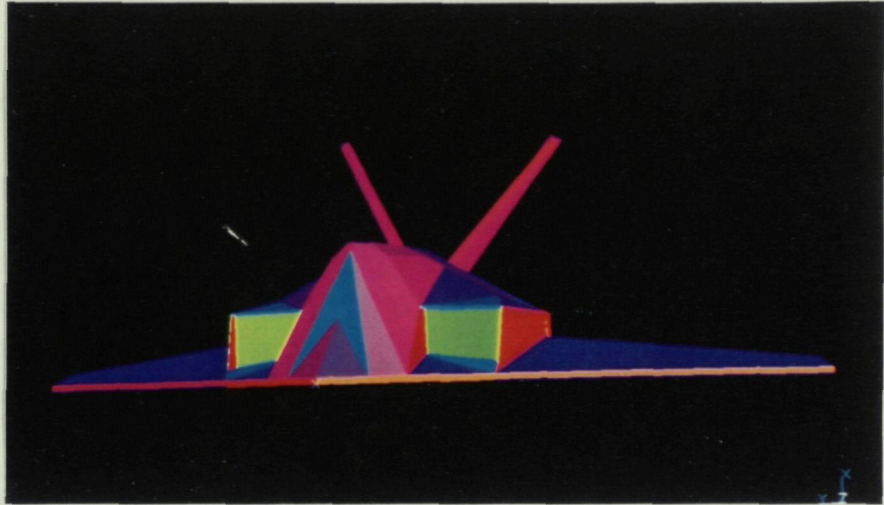


Figura 5-5: Visualización del avión F-117 en una estación gráfica iluminado por tres fuentes de luz de colores rojo, azul y verde situadas en los ejes x, y, z.

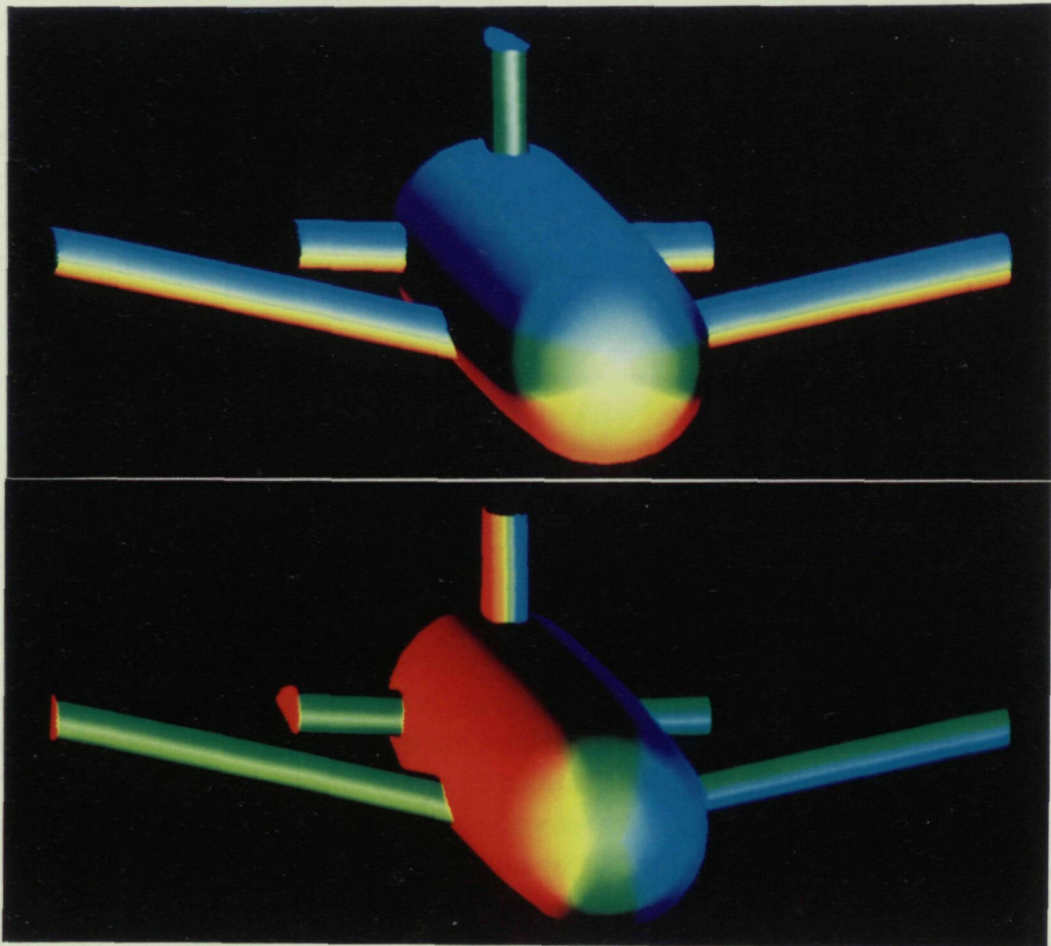


Figura 5-6: Visualización de las dos imágenes de un misil necesarias para obtener las componentes de la normal a la superficie en cada pixel.

5.3.3 Detección de aristas

Las aristas se detectan sobre la superficie del blanco como discontinuidades de la normal. Para ello, se realiza un barrido de la pantalla pixel a pixel. Cuando entre dos pixels consecutivos aparece una discontinuidad en alguna de las componentes de la normal, significa que hay una arista o un eclipse (ver figura 5-7).

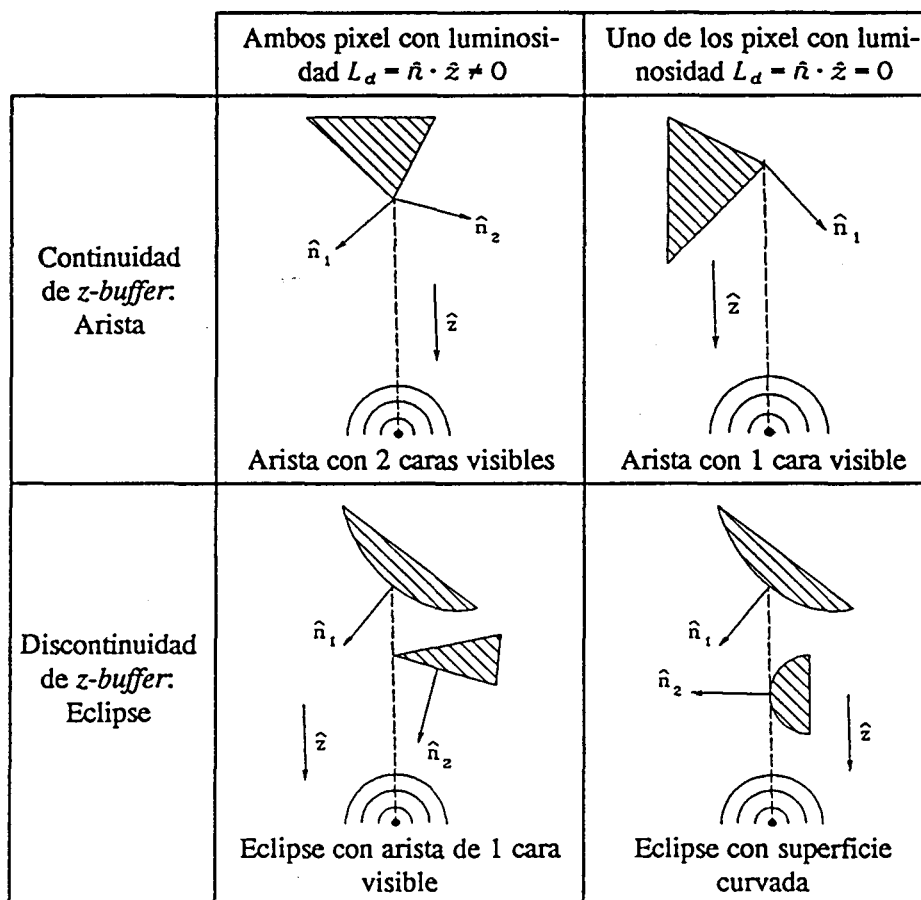


Figura 5-7: Cuatro casos posibles de discontinuidad de la normal a la superficie entre dos pixels consecutivos.

En la figura 5-7 puede observarse, a su vez, que las discontinuidades de la normal a la superficie debidas a aristas se discriminan de las debidas a eclipses por medio del z-buffer: si la coordenada z es continua hay una arista y si es discontinua se ha producido un eclipse.

Con objeto de evitar la detección de las cuñas que aparecen en la malla poligonal de la superficie NURB discretizada por el acelerador gráfico (ver sección 3.6), debe utilizarse

el sombreado de Gouraud (ver sección 3.3.3 b)), que produce una variación suave y continua de la luminosidad sobre una superficie curvada. De esta forma, cuando la discontinuidad en el valor de las componentes de la normal es mayor que un cierto umbral de decisión, se detectan correctamente las aristas reales del objeto y no las debidas a la malla poligonal generada por el hardware gráfico.

5.3.4 Cálculo de los radios de curvatura

Conociendo la normal unitaria a la superficie en todos los puntos pueden calcularse muy fácilmente los radios de curvatura de la superficie en los planos xz y zy . Estos radios de curvatura no coinciden en general con los radios principales de curvatura de la superficie.

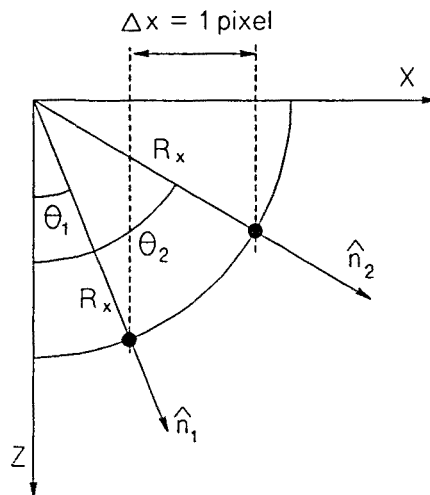


Figura 5-8: Método utilizado para el cálculo del radio de curvatura de una superficie en el plano "zx".

En la figura 5-8 se muestra el procedimiento para el cálculo del radio de curvatura R_x en el plano xz :

$$1 \text{ pixel} = \Delta x = R_x \sin\theta_2 - R_x \sin\theta_1 = R_x (n_{2x} - n_{1x}) \quad (5.3)$$

El procedimiento es análogo para el radio de curvatura R_y en el plano zy . Las expresiones para los radios de curvatura son:

$$\frac{1}{R_x} = \frac{\Delta n_x}{\Delta x} \quad \frac{1}{R_y} = \frac{\Delta n_y}{\Delta y} \quad (5.4)$$

que pueden utilizarse, por ejemplo, para detectar aristas cuando el inverso del radio de curvatura entre dos pixels consecutivos sea muy grande.

5.4 ANALISIS ELECTROMAGNÉTICO

El análisis electromagnético, segunda etapa de *GRECO*, utiliza como datos de entrada las coordenadas x, y, z y el vector unitario normal n_x, n_y, n_z en cada pixel iluminado del blanco. A partir del conocimiento de esta información, se pueden desarrollar algunas aproximaciones de alta frecuencia, de forma que los cálculos son independientes de la geometría del modelo. A continuación se va a dedicar una sección a cada una de las distintas aproximaciones implementadas.

5.4.1 Aproximación de óptica física

De acuerdo con la técnica de óptica física (PO), la RCS monoestática de una superficie perfectamente conductora puede aproximarse a altas frecuencias, por la expresión:

$$\sigma = \frac{4\pi}{\lambda^2} \left| \int_s \cos\theta e^{2jkz} ds \right|^2 \quad (5.5)$$

donde θ es el ángulo entre la normal a la superficie y la dirección de incidencia y z es la distancia desde el diferencial de superficie, ds , al observador, proyectada en la dirección de incidencia. La integral de superficie se extiende solamente sobre la región iluminada por la onda incidente.

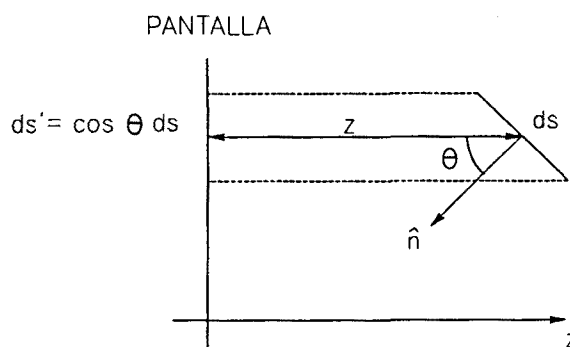


Figura 5-9: Elemento de superficie y su proyección en la pantalla gráfica.

Sin embargo, la imagen del blanco procesada por *GRECO* es una proyección sobre la pantalla de la estación de trabajo, de la superficie real tridimensional (ver figura 5-9). Puesto que el diferencial de superficie en la pantalla, ds' , equivalente a un pixel es igual a $ds' = \cos\theta ds$, la integral de superficie PO puede escribirse como:

$$\sigma = \frac{4\pi}{\lambda^2} \left| \int_s \cos\theta e^{2jkz} ds \right|^2 = \frac{4\pi}{\lambda^2} \left| \int_{\text{pantalla}} e^{2jkz} ds' \right|^2 \quad (5.6)$$

El cálculo discreto de la integral de superficie (5.6) conduce a una expresión de la forma:

$$\sigma = \frac{4\pi}{\lambda^2} \left| \sum_{\text{pixels}} e^{2jKz} \right|^2 \quad (5.7)$$

que es igual a la suma coherente de la contribución de fase de todos los pixels en la imagen del blanco. Esta contribución de fase se debe a la distancia z desde cada pixel al observador.

Debe observarse que la ecuación (5.7) es correcta solamente si un pixel radia como una apertura elemental, es decir, es equivalente a la proyección en la pantalla de un diferencial de superficie, ds , mucho menor que la longitud de onda. Si el número de pixels en la pantalla es suficientemente grande, esta condición se suele cumplir. Sin embargo, cuando la dirección de incidencia es rasante sobre la superficie, $\theta \rightarrow 90^\circ$, la proyección ds' en la pantalla (un pixel) es muy pequeña, pero la superficie ds puede ser muy grande.

Por tanto, en general no podemos asumir que cada pixel radia como una apertura elemental, pero sí como una que es eléctricamente grande. Asumiendo que un pixel es equivalente a una apertura rectangular con iluminación uniforme, su contribución al campo lejano puede ser aproximada por una función "sinc" del ángulo θ , y la integral de superficie se convierte, en el dominio discreto, en la siguiente expresión:

$$\sigma = \frac{4\pi}{\lambda^2} \left| \sum_{\text{pixels}} \text{sinc} \left(K \frac{l}{\cos\theta} \sin\theta \right) e^{2jKz} \right|^2 \quad (5.8)$$

donde l es el tamaño de un pixel ds' en la pantalla, y $(l/\cos\theta)$ es el tamaño del ds proyectado en este pixel.

Un problema que aparece muy a menudo cuando se calcula la RCS de superficies curvadas utilizando la aproximación de óptica física es la presencia de oscilaciones espúreas al variar la frecuencia. Estas oscilaciones se deben a la contribución de la transición abrupta de las corrientes equivalentes en frontera entre las regiones iluminada y en sombra.

Según la teoría de alta frecuencia, la principal contribución a la RCS monoestática de una superficie curvada está en el punto de reflexión especular, en el que la normal a la superficie es casi paralela a la dirección de incidencia, $\theta \rightarrow 0^\circ$ en la figura 5-4. La contribución de la región de variación rápida de fase, cerca de la frontera entre las regiones iluminada y en sombra, en teoría debe cancelarse, pero no lo hace debido al error en la discretización de la superficie de discretización. Como esta región de transición tiene $\theta \rightarrow 90^\circ$, esta contribución espúrea se puede eliminar sin modificar la contribución especular multiplicando la contribución de cada pixel por una función

$\cos^n\theta$ con un máximo en el punto de reflexión especular, de forma que el valor de la RCS se aproxima por la ecuación:

$$\sigma = \frac{4\pi}{\lambda^2} \left| \sum_{\text{pixels}} \cos^n\theta \operatorname{sinc}(Kl \tan\theta) e^{2jkz} \right|^2 \quad (5.9)$$

donde "n" es un parámetro que controla el efecto de la aproximación de fase estacionaria. Cuando $n = 0$ se calcula rigurosamente la integral de superficie de óptica física sobre la superficie discretizada y cuando $n > 1$ se disminuye el efecto negativo de la región de transición pero, por otro lado, se pueden introducir errores significativos en los niveles de los lóbulos secundarios que aparecen en las placas planas.

Las ecuaciones (5.8) y (5.9) pueden implementarse eficientemente en *GRECO* porque tanto la función $\cos\theta$ como la función "sinc" dependen solamente del ángulo θ . Utilizando las fuentes de luz descritas en la sección 5.2.2 (ver imagen del F-117 en la figura 5-5), la componente de color verde de cada pixel es

$$\text{VERDE} = n_z = \cos\theta \quad (5.10)$$

Como el color verde se codifica con 8 bits, la función $\cos^n\theta \operatorname{sinc}(Kl \tan\theta)$ puede ser tabulada en una tabla de 256 entradas, indexada por la componente de color verde de cada pixel. Si la fase exponencial e^{-2jkz} está también tabulada en una tabla de 2^{16} entradas, indexadas por la coordenada z, se puede añadir la contribución desde cada pixel a la integral de PO con solamente dos multiplicaciones y dos sumas de números reales en coma flotante.

5.4.2 Condición de contorno de impedancia

Los recubrimientos absorbentes radar se consideran mediante las aproximaciones de la condición de impedancia superficial (IBC) y de óptica física (PO), las cuales conducen a una formulación muy sencilla de la integral de superficie de PO. Hay que tener en cuenta que la aproximación PO+IBC es válida solamente para reflexión superficial cuando existe un punto de reflexión especular. Las aristas y esquinas del objeto deben ser tratadas por una aproximación de orden superior de la condición de contorno.

Según la aproximación de IBC la contribución de cada pixel en la ecuación (5.7) debe ser multiplicada por los coeficientes de reflexión de Fresnel para las polarizaciones paralela (Γ_{\parallel}) y perpendicular (Γ_{\perp}) al plano de incidencia:

$$\sigma = \frac{4\pi}{\lambda^2} \left| \sum_{pixels} [\Gamma_{\parallel} E_{\parallel}^i \hat{e}_{\parallel}^i + \Gamma_{\perp} E_{\perp}^i \hat{e}_{\perp}^i] \cos\theta \operatorname{sinc}(Kl \tan\theta) e^{2jKz} \right|^2 \quad (5.11)$$

donde E_{\parallel}^i y E_{\perp}^i son, respectivamente, las componentes del campo incidente en las direcciones paralela y perpendicular al plano de incidencia, \hat{e}_{\parallel}^i y \hat{e}_{\perp}^i . La descomposición del campo incidente en las componentes paralela y perpendicular conduce a las siguientes expresiones para las polarizaciones copolar y contrapolar:

Polarización copolar vertical:

$$\sigma_{yy} = \frac{4\pi}{\lambda^2} \left| \int_s \cos\theta \operatorname{sinc}(Kl \tan\theta) \left[\frac{\Gamma_{\parallel} n_y^2 + \Gamma_{\perp} n_x^2}{n_x^2 + n_y^2} \right] e^{2jKz} ds' \right|^2 \quad (5.12)$$

Polarización copolar horizontal:

$$\sigma_{xx} = \frac{4\pi}{\lambda^2} \left| \int_s \cos\theta \operatorname{sinc}(Kl \tan\theta) \left[\frac{\Gamma_{\parallel} n_x^2 + \Gamma_{\perp} n_y^2}{n_x^2 + n_y^2} \right] e^{2jKz} ds' \right|^2 \quad (5.13)$$

Polarización contrapolar:

$$\sigma_{xy} = \frac{4\pi}{\lambda^2} \left| \int_s \cos\theta \operatorname{sinc}(Kl \tan\theta) \left[\frac{(\Gamma_{\parallel} - \Gamma_{\perp}) n_y n_x}{n_x^2 + n_y^2} \right] e^{2jKz} ds' \right|^2 \quad (5.14)$$

donde n_x , n_y , n_z son respectivamente las componentes del vector unitario normal a la superficie en las direcciones horizontal, vertical y de incidencia. Utilizando el esquema de iluminación presentado en la sección 5.3.2 n_x , n_y , n_z en cada ds' son iguales a las componentes de color R, G, B del pixel correspondiente a ese ds' (ver figuras 5-5 y 5-6).

La formulación de los coeficientes de reflexión de Fresnel como funciones de la impedancia superficial equivalente es bien conocida y puede encontrarse en [Knott'85], [Maffet'89], [Klement'88]. La impedancia superficial de un recubrimiento dieléctrico sobre una superficie perfectamente conductora se obtiene a través del circuito equivalente en línea de transmisión.

Como los coeficientes de reflexión dependen solamente del ángulo θ , el cual, de acuerdo con la ecuación (5.10), se relaciona con la componente de color verde de cada pixel en la imagen (ver figura 5-5), los coeficientes de reflexión paralela y perpendicular pueden tabularse en una tabla de 256 entradas indexada por la

componente de color verde, evitando así los cálculos de los coeficientes de reflexión para cada pixel.

5.4.3 Método de las corrientes equivalentes

Según la teoría de alta frecuencia, el campo lejano dispersado por una arista puede aproximarse por el radiado por una línea de corriente equivalente situada en la arista. Esta corriente equivalente depende tanto de la dirección de incidencia como de la de observación relativas a la orientación de la arista, por lo que su valor no es constante a lo largo de aristas curvadas. Las corrientes equivalentes se expresan habitualmente como una función de los coeficientes de difracción de longitud incremental (ILDC). El campo lejano monoestático dispersado desde la arista, resultado de la radiación de las corrientes equivalentes, es:

$$\vec{E}^r = E_0 \frac{e^{-jKr}}{2\pi r} \int_{EDGE} \left[-D_{\parallel} \sin\gamma \hat{e}_{\parallel}^i - D_x \cos\gamma \hat{e}_x^i - D_{\perp} \cos\gamma \hat{e}_{\perp}^i \right] e^{2jKz} dl' \quad (5.15)$$

donde la integral de línea se extiende a lo largo de las aristas iluminadas por la onda incidente, \hat{e}_{\parallel}^i y \hat{e}_{\perp}^i son respectivamente los vectores unitarios paralelo y perpendicular al plano de incidencia, el cual viene definido por las direcciones de incidencia \hat{z} y la dirección de la arista \hat{t} , y γ es el ángulo entre el campo eléctrico incidente y \hat{e}_{\perp}^i .

En la ecuación (5.15) D_{\parallel} , D_{\perp} , D_x representan los coeficientes ILDC monoestáticos, los cuales expresan la dependencia del campo dispersado con los ángulos α , ϕ , β_r definidos en la figura 5-10. La formulación de los ILDC y las referencias a los artículos originales de Keller (GTD), Ufimtsev (PTD), Michaeli (MEC) y Mitzner (ILDC) se pueden encontrar en [Knott'85], [Rius'91] y [Maffet'89].

La separación del campo incidente de la ecuación (5.15) en las componentes perpendicular y paralela conduce a las expresiones siguientes para las polarizaciones copolar y contrapolar [Rius'91]:

Copolar horizontal:

$$E_{xx}^s = E_0 \frac{e^{-jKr}}{2\pi r} \int_{EDGE} \left[-D_{\parallel} \frac{t_x^2}{t_x^2 + t_y^2} + D_x \frac{t_x t_y}{t_x^2 + t_y^2} - D_{\perp} \frac{t_y^2}{t_x^2 + t_y^2} \right] e^{2jKz} dl' \quad (5.16)$$

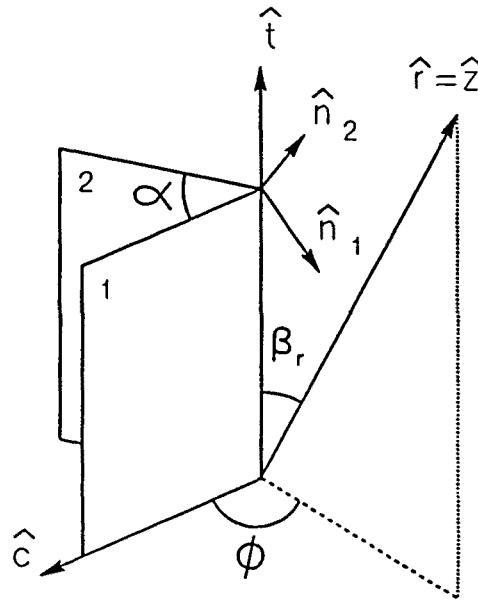


Figura 5-10: Geometría de una cuña: la dirección de incidencia y observación es la del vector z .

Copolar vertical:

$$E_{yy}^s = E_0 \frac{e^{-jKr}}{2\pi r} \int_{EDGE} \left[-D_1 \frac{t_y^2}{t_x^2 + t_y^2} - D_x \frac{t_x t_y}{t_x^2 + t_y^2} - D_\perp \frac{t_x^2}{t_x^2 + t_y^2} \right] e^{2jKz} dl' \quad (5.17)$$

Contrapolar con transmisión horizontal y recepción vertical:

$$E_{yx}^s = E_0 \frac{e^{-jKr}}{2\pi r} \int_{EDGE} \left[+D_1 \frac{t_x t_y}{t_x^2 + t_y^2} - D_x \frac{t_y^2}{t_x^2 + t_y^2} - D_\perp \frac{t_x t_y}{t_x^2 + t_y^2} \right] e^{2jKz} dl' \quad (5.18)$$

Contrapolar con transmisión vertical y recepción horizontal:

$$E_{yx}^s = E_0 \frac{e^{-jKr}}{2\pi r} \int_{EDGE} \left[-D_1 \frac{t_x t_y}{t_x^2 + t_y^2} - D_x \frac{t_x^2}{t_x^2 + t_y^2} + D_\perp \frac{t_x t_y}{t_x^2 + t_y^2} \right] e^{2jKz} dl' \quad (5.19)$$

donde $\hat{t} = (t_x, t_y, t_z)$ es el vector unitario paralelo a $d\vec{l}'$, y los ejes x, y, z se han definido como las direcciones horizontal, vertical y de incidencia, respectivamente.

Para realizar las integrales de línea del MEC, ecuaciones desde la (5.16) a la (5.19), el programa *GRECO* debe obtener los ILDC para cada pixel de la imagen situado sobre una arista, y sumar coherentemente las contribuciones de todos los pixels. Como los ILDC dependen de los ángulos α , ϕ , β_r , definidos en la figura 5-10, *GRECO* debe calcular primero estos ángulos a partir del conocimiento del vector unitario normal a ambas caras de la arista [Rius'91]:

$$\alpha = \cos^{-1}(-\hat{n}_1 \cdot \hat{n}_2) \quad (5.20)$$

$$\sin\beta_r = | \hat{r} \times \hat{t} | = \sqrt{t_x^2 + t_y^2} \quad (5.21)$$

$$\cos\theta = \frac{n_{1x}t_y - n_{1y}t_x}{\sqrt{t_x^2 + t_y^2}} \quad (5.22)$$

donde \hat{t} es el vector unitario en la dirección del eje

$$\hat{t} = \frac{\hat{n}_1 \times \hat{n}_2}{|\hat{n}_1 \times \hat{n}_2|} \quad (5.23)$$

Como la reflexión en las caras planas o curvadas de la arista ya se ha calculado utilizando óptica física, ahora sólo hay que calcular la contribución en campo lejano debida a la difracción en las aristas. Esto puede conseguirse utilizando los coeficientes ILDC de Ufimtsev o Mitzner [Knott'85], [Rius'91], [Maffet'89]. Los primeros son válidos solamente cuando la dirección de observación se encuentra en el cono de Keller [Knott'85], [Rius'91], [Maffet'89], mientras que los últimos son válidos para una observación biestática arbitraria.

Sin embargo, para problemas de RCS monoestática, se puede demostrar fácilmente [Rius'91] que los coeficientes de Mitzner y PTD para las polarizaciones paralela o perpendicular al plano de incidencia, D_{\parallel} y D_{\perp} respectivamente, son iguales. La diferencia reside en que la PTD asume que el coeficiente D_x es nulo. Esta última hipótesis implica que no haya efecto de polarización cruzada cuando el campo incidente es perpendicular al plano de incidencia.

Aunque los coeficientes más adecuados son los ILDC de Mitzner, en el programa *GRECO* se han implementado los coeficientes de la teoría física de la difracción (PTD). Esta es una buena aproximación cuando la dirección de incidencia está cercana al plano perpendicular a la arista [Rius'91], y es bien conocido que la principal contribución de

la arista a la RCS se produce cuando la incidencia es normal, mientras que la contribución para incidencia oblicua es normalmente despreciable.

Las figuras 5-11 y 5-12 muestran respectivamente, los coeficientes PTD monoestáticos [Rius'91] para la polarización paralela ($f = D_{\parallel}$) y perpendicular ($g = D_{\perp}$) al plano de incidencia, respecto al ángulo ϕ entre la dirección de incidencia y la superficie de la cuña, como se ha definido en la figura 5-10. Las expresiones para estos coeficientes de PTD pueden encontrarse en [Knott'85], [Rius'91], [Maffet'89], etc. Estas expresiones incluyen varias funciones trigonométricas, por tanto debe evitarse el cálculo exacto de los coeficientes de PTD para cada uno de los pixels situados en una arista. Por esta razón, se ha desarrollado la siguiente aproximación lineal de los coeficientes para la PTD monoestática:

1 cara visible $0 \leq \phi < \pi - \alpha$	2 caras visibles $\pi - \alpha \leq \phi \leq \pi - \alpha/2$
$f \approx -\frac{\phi}{\pi}$	$f \approx -(n - 1) \quad (5.24)$
$g \approx -\left(\frac{n}{2} - \frac{\phi}{\pi}\right)$	$g \approx \tan\frac{\alpha}{2} - \frac{2}{n}\cot\frac{\pi}{n} \quad (5.25)$

donde $n = (2\pi - \alpha)/\pi$ es el ángulo exterior de la cuña normalizado por π .

En resumen, el programa *GRECO* calcula la difracción de las aristas en alta frecuencia por el método de las corrientes equivalentes (MEC) en los pasos siguientes:

- 1- Se visualiza una imagen del blanco en la pantalla de la estación de trabajo. Las aristas ocultas son eliminadas por el acelerador hardware gráfico, con lo que solamente se visualizan las que quedan visibles al radar.
- 2- La normal unitaria a la superficie en cada pixel de la imagen se calcula por procesado gráfico de la imagen (ver sección 5.3.2).

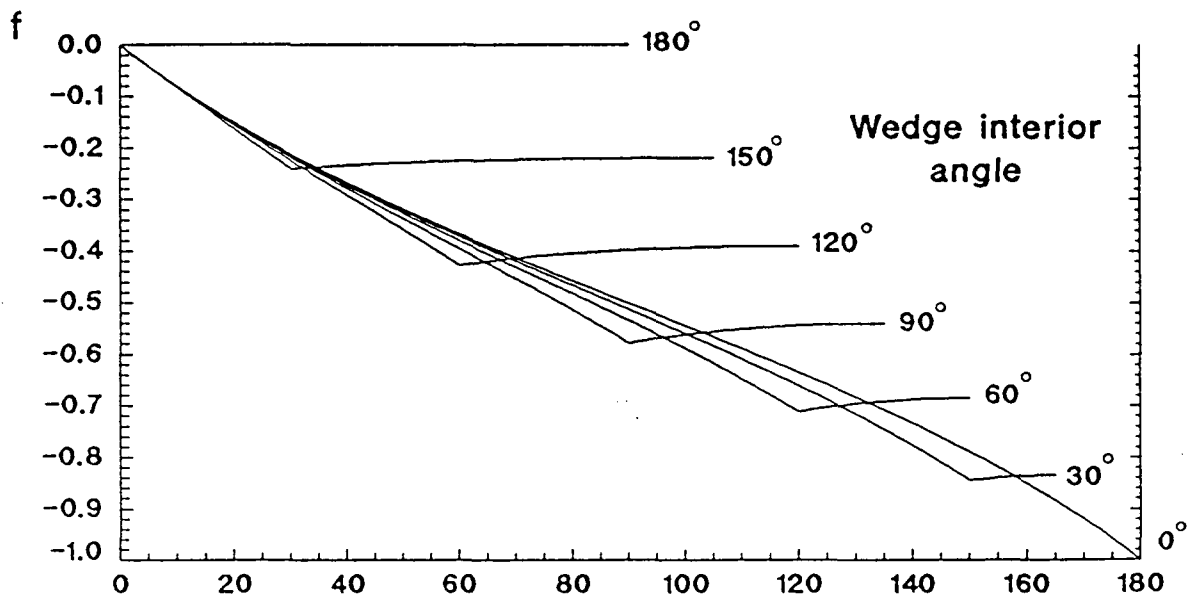


Figura 5-11: Coeficiente PTD monoestático $f=D_{\parallel}$ para la polarización paralela al plano de incidencia, en función del ángulo ϕ entre la dirección de incidencia y una cara de la cuña.

3- Se detectan las aristas en la imagen del blanco como discontinuidades de la normal unitaria a la superficie cuando la coordenada z se mantiene continua. Nótese que las superficies eclipsadas podrían producir también discontinuidades en la normal, ver figura 5-2, pero en ese caso la coordenada z sería discontinua. También pueden detectarse las aristas a partir de una discontinuidad del radio de curvatura (ver sección 5.3.4)

4- Para cada pixel de todas las aristas detectadas, se calculan $\alpha, \phi, \beta,$, a partir del vector normal a cada cara de la cuña utilizando las ecuaciones (5.20) a (5.23). Los coeficientes de difracción de la PTD monoestática se obtienen entonces utilizando una aproximación lineal muy simple, ecuaciones (5.24) y (5.25), las cuales optimizan el tiempo de ejecución evitando el cálculo de las funciones trigonométricas.

5- Las integrales de línea (5.16) a (5.19) del MEC se evalúan sumando coherentemente los coeficientes de PTD en cada pixel.

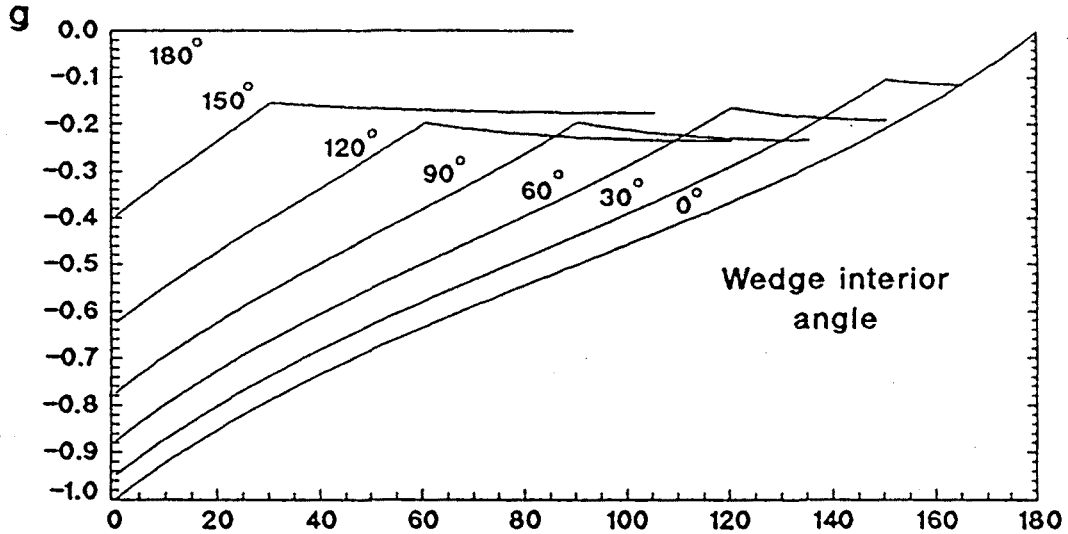


Figura 5-12: Coeficiente PTD monoestático $g=D_{\perp}$ para la polarización perpendicular al plano de incidencia, en función del ángulo ϕ entre la dirección de incidencia y una cara de la cuña.

5.4.4 Reflexiones múltiples por trazado de rayos

En el cálculo del efecto de las dobles reflexiones entre superficies perfectamente conductoras (figura 5-13), el campo magnético reflejado según óptica física por una superficie (la 1 en la figura) e incidente sobre otra superficie (la 2 en la figura) se obtiene a partir de la expresión:

$$\vec{H}_1^s(\vec{r}'') = 2 \iint_{S_1'} \hat{s}' \times [\hat{n}_1 \times \vec{H}_0^i] \left(jK + \frac{1}{s'} \right) \frac{e^{-jKs'}}{4\pi s'} e^{-jK\hat{i} \cdot \vec{r}'} dS_1' \quad (5.26)$$

donde $\vec{H}_0^i e^{-jK\hat{i} \cdot \vec{r}'}$ es el campo magnético incidente, \vec{r}' es un punto en la superficie 1, \vec{r}'' es un punto en la superficie 2 y $\vec{s}' = \vec{r}'' - \vec{r}' = s' \hat{s}'$.

El campo lejano reflejado según óptica física por la superficie 2 es

$$\vec{H}_{12}^s = 2jK \frac{e^{-jKr}}{4\pi r} \iint_{S_2''} \hat{r} \times [\hat{n}_2 \times \vec{H}_1^s] e^{jK\hat{r} \cdot \vec{r}''} dS_2'' \quad (5.27)$$

El campo magnético \vec{H}_1^s puede calcularse mediante óptica geométrica utilizando la aproximación de fase estacionaria en la integral sobre la superficie S_1' , si en cada

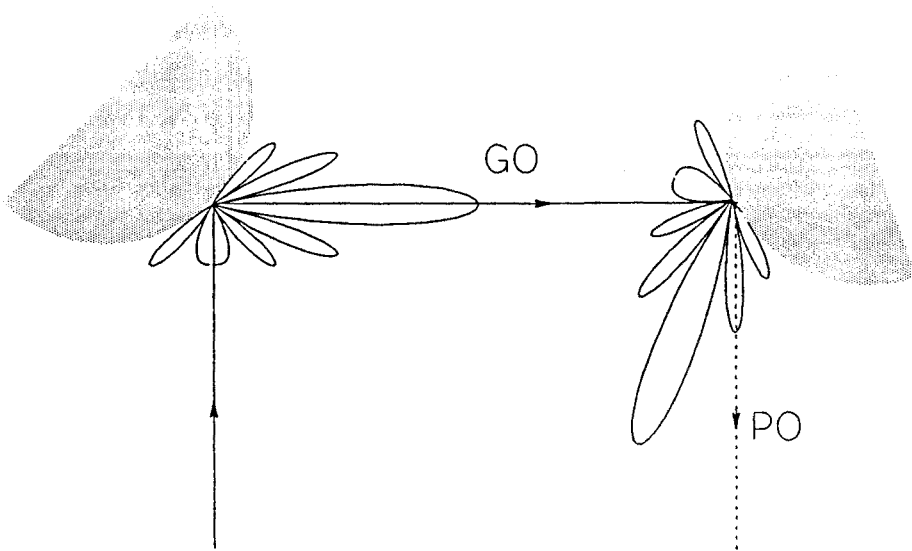


Figura 5-13: Dobles reflexiones.

punto se aproxima localmente la curva por un plano infinito tangente a la superficie en aquel punto. En este caso la reflexión de la onda plana incidente es especular y, para un punto de observación dado \vec{r}'' , el campo difractado es cero para cualquier \vec{r}' excepto para el punto de fase estacionaria, \vec{r}'_s . Aplicando el principio de fase estacionaria [Felsen'72] para calcular la integral en S_1 se llega a la expresión:

$$\vec{H}_{12}^s = 2jK \frac{e^{-jKr}}{4\pi r} H_0 \iint_{S_2''} \hat{r} \times [\hat{n}_2 \times (\hat{h}^i - 2\hat{n}_1 (\hat{n}_1 \cdot \hat{h}^i))] \left(1 + \frac{1}{jKs'}\right) e^{jK(\hat{r} \cdot \vec{r}'' - s' - \hat{i} \cdot \vec{r}'_s)} dS_2'' \quad (5.28)$$

donde \hat{r} es la dirección de observación e \hat{i} la de incidencia, \vec{r}'_s es el punto de reflexión especular en S_1 y \vec{r}'' el punto de integración en S_2 . Esta es la aproximación de óptica geométrica-óptica física, pues la primera integral de superficie (sobre S_1) se ha realizado asintóticamente, mientras que la segunda (sobre S_2) se realiza numéricamente.

Cuando se utiliza el procesado gráfico, la integral sobre S_2 se convierte en la suma de los pixels alcanzados por los rayos reflejados. El número de *impactos* en un elemento dS_2'' (iluminado por la primera reflexión) es igual al número de rayos o pixels de la zona dS_1' iluminada. Así se obtiene un cálculo muy aproximado del factor de divergencia cuando S_1' no es plana, sino curvada, como se representa en la figura 5-14. Ello es debido a que el número de impactos es inversamente proporcional al factor de divergencia, con lo que no hay que incluirlo explícitamente en la formulación.

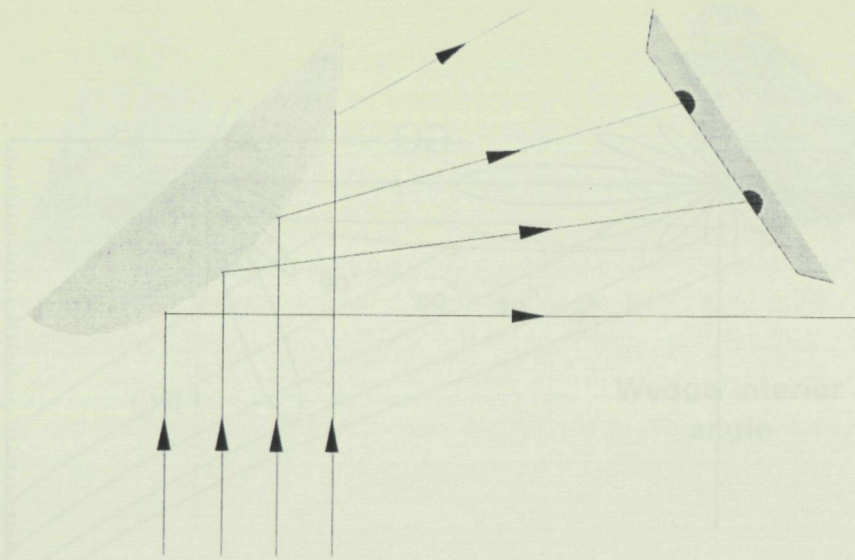


Figura 5-14: La dispersión de rayos tiene en cuenta implícitamente el factor de divergencia.

En la figura 5.15 se muestra el resultado del trazado de rayos sobre un avión Boeing 727. Los pixel blancos representan los puntos sobre los que incide un rayo reflejado especularmente en otra superficie, es decir, son los puntos de la segunda reflexión.

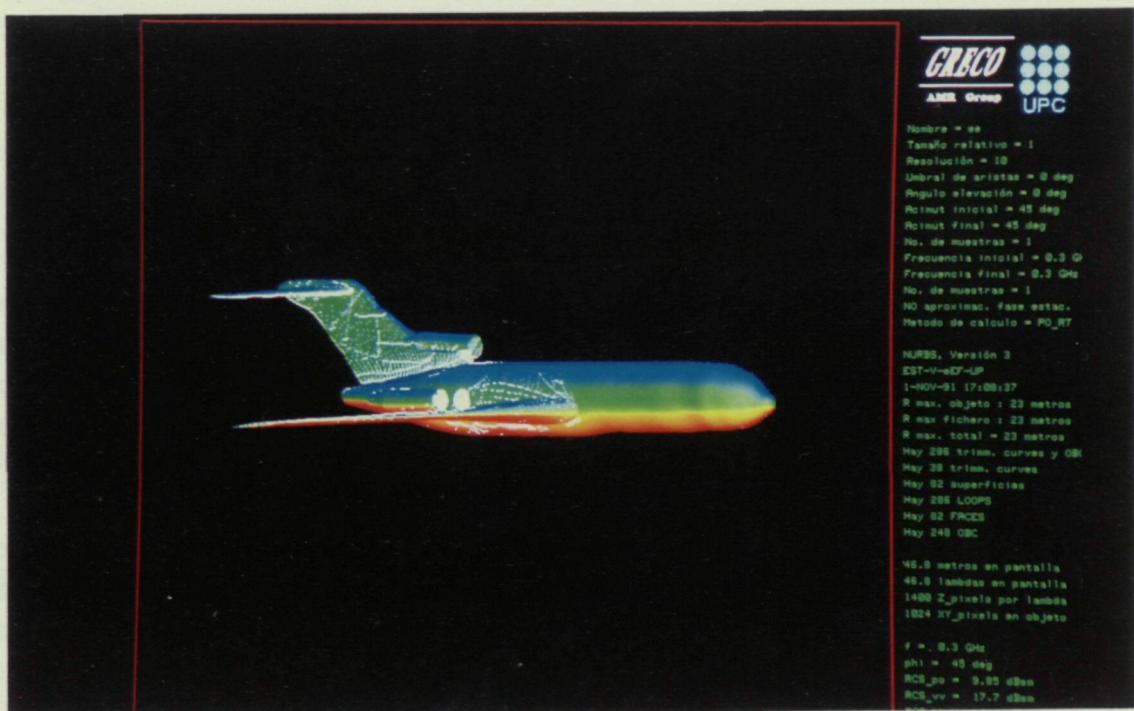


Figura 5-15: Trazado de rayos en un avión Boeing 727.