

UNIVERSIDAD AUTÓNOMA METROPOLITANA IZTAPALAPA
DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA

***RECONSTRUCCIÓN Y SOMBREADO DE
SUPERFICIES TRIDIMENSIONALES
ANATÓMICAS A PARTIR DE
CORTES TOMOGRÁFICOS***

227468

TESIS PARA OBTENER EL GRADO DE
MAESTRÍA EN INGENIERÍA BIOMÉDICA

NORMA PILAR CASTELLANOS ABREGO

ASESORA: DRA. VERÓNICA MEDINA BAÑUELOS

México, D. F., agosto 1995

MT 20/2221

CONTENIDO

	Página
CAPITULO 1	
INTRODUCCION.	1
 CAPITULO 2	
OBJETIVOS.	5
 CAPITULO 3.	
CONCEPTOS TEORICOS.	6
3.1. Transformaciones en 3D.	
3.1.1. Traslación.	
3.1.2. Rotación.	
3.1.3. Reflexión.	
3.1.4. Cambio de sistema de coordenadas.	
3.2. Proyecciones.	9
3.2.1. Proyección central.	
3.2.2. Proyección Cavalier.	
3.2.3. Proyección en un punto arbitrario.	
3.3. Representación de curvas y superficies.	11
3.3.1. Continuidad geométrica y continuidad paramétrica.	
3.3.2. Funciones paramétricas.	
3.3.3. Superficies de Bézier.	
3.3.4. Método B-Spline.	
3.4. Eliminación de superficies ocultas.	17
3.4.1. Visibilidad de un objeto simple o convexo.	
3.4.2. Visibilidad de diversos objetos.	
3.4.3. Algoritmos Z-Buffer.	
3.5. Iluminación y sombreado.	20
3.5.1. Modelos de iluminación.	
3.5.2. Sombreado constante.	
3.5.3. Método de Gouraud.	
3.5.4. Método de Phong.	
3.5.5. Problemas con interpolación en sombreado.	
3.6. Sombreado en imágenes médicas.	25
3.6.1. Sombreado por gradiente por niveles de gris.	
3.6.2. Sombreado por gradiente z-buffer.	
3.6.3. Sombreado por gradiente en niveles de gris (GG).	
3.6.4. Sombreado por gradiente en niveles de gris adaptivo (AGG).	
3.7. Reconstrucción a partir de cortes en 2D.	29

CAPITULO 4	
METODO Y DESARROLLO	33
CAPITULO 5	
RESULTADOS Y DISCUSION.	40
CAPITULO 6	
CONCLUSIONES.	52
BIBLIOGRAFIA.	54

CAPITULO 1.

INTRODUCCION

Con el descubrimiento de los rayos X fué posible reconstruir el interior del cuerpo humano para su observación y estudio sin necesidad de cirugía dando inicio al campo de la radiología. En esos años, la reconstrucción estaba limitada a partes del cuerpo que fueran radiopacas (tejido óseo) o que pudieran ser visualizadas con material de contraste para rayos X. El reciente desarrollo de tomografía computarizada basada en rayos X (CT), ultrasonido (US), tomografía computarizada por emisión de fotón única (SPECT), tomografía por emisión positrón (PET), y resonancia magnética nuclear para imágenes médicas (MRI), proveen en conjunto, la capacidad de observar y estudiar el tejido blando y óseo, anatómica y funcionalmente en forma no invasiva. Esta capacidad es necesaria para la planeación de cirugías, diagnóstico y tratamiento de muchas enfermedades, y educación médica en general. La introducción de algoritmos para la reconstrucción en 3D de órganos de interés a partir de imágenes en 2D, complementa la utilidad de estas técnicas, además de continuar con el avance en la visualización no invasiva del cuerpo humano.

Modelar un objeto tridimensional (3D) en una computadora cuyo despliegue es en una pantalla gráfica (2D) presenta algunos problemas. ¿Cómo puede ser desplegada la profundidad o tercera dimensión en una pantalla gráfica?. ¿Cómo pueden las partes ocultas del objeto ser removidas e identificadas en la imagen?. ¿Cómo pueden contribuir la luminosidad, el sombreado, la textura y el color en la representación de la imagen?. ¿Cómo debe ser modelado el mundo tridimensional en una computadora?. Muchas técnicas han sido desarrolladas para contestar estas preguntas y algunas de las mayormente utilizadas se describen en los siguientes capítulos. Estas técnicas modelan y generan diferentes representaciones en imágenes mediante diversos algoritmos que están muy relacionados con la aplicación. Los métodos manejados en la representación tridimensional de imágenes médicas permiten al usuario explorar cualquier zona del cuerpo humano desde cualquier ángulo con una alta fidelidad.

Los dos grandes campos en la representación tridimensional de imágenes médicas son la *representación en superficie (surface rendering)* y la *representación en volumen (volume rendering)*. En representación en superficie, la visualización y manejo de datos multidimensionales se basa únicamente en los límites del objeto y el despliegue de la superficie del objeto de interés está aislado del resto del volumen[6, 19]. En representación en volumen, se persigue la representación de múltiples estructuras con diferentes tipos de tejido. A cada voxel se le asocia un nivel de opacidad y se asume que el valor de cada uno está correlacionado con el tipo de material que lo integra. Para cada versión de despliegue se examina cada voxel, lo que hace a esta representación muy costosa computacionalmente [3,20,21,26, 37]. El tema de las ventajas y desventajas de uno u otro tipo de representación se ha convertido en controversia, pero se puede hacer una diferencia drástica entre los dos: la representación en volumen es muy útil cuando las estructuras en la escena son difusas, como puede ser un cancer dentro de un tejido; por otro lado la representación en superficie pierde la información del material del objeto que lo limita para fines de comparación con otros tejidos. La gran ventaja de tener únicamente información sobre los límites del objeto

es que permite una más rápida interacción y manipulación de los datos, además de obtener una aproximación mejor al detalle y textura de la superficie del objeto.

El interés de nuestro estudio es únicamente en imágenes que representan la anatomía del cuerpo humano (imágenes médicas anatómicas) facilitadas por sistemas CT, US y RMN. En el campo de la clínica médica la visualización en 3D puede ser de mucha utilidad en planeaciones de cirugía [11], ortopedia [9], traumas músculo-esqueléticos [22], estudios en hígado[32], análisis de tejidos [23,30], reconstrucción de vasos sanguíneos [7,8], trombosis venosa [34], y visualización neuronal [1,18,24], entre otros. Además es utilizada en diversas áreas además de la médica como es el Modelado molecular, Diseños de automóviles, barcos, aviones, partes mecánicas, etc., ayudados por computadora, Animación para programas educativos y Simulación de diferentes objetos .

Producir una imagen con una representación perfectamente real, es impráctico. Se necesitan técnicas que tomen en cuenta los diferentes tipos de realismo que necesita cada aplicación, la cantidad de procesamiento para generar una imagen, las capacidades del hardware de despliegue, la cantidad de datos que modelan la escena, y los efectos perceptuales de la imagen sobre el observador. El costo de la técnica implica número de operaciones, de manera que para reducir este costo y por consiguiente el tiempo de procesamiento se han creado dos soluciones. La primera busca simplificar el número de operaciones mediante algoritmos de representación de imágenes como es la eliminación de superficies ocultas (sección 3.4). La segunda estrategia hace uso de arquitecturas basadas en multiprocesadores para lograr una rápida velocidad de manejo de imágenes al repartir las tareas entre cada procesador (procesamiento en paralelo)[33,36].

Todos los algoritmos de las diferentes técnicas de reconstrucción persiguen el mismo objetivo: mostrar una imagen que se asemeje en lo posible a la realidad con el mínimo costo computacional. Si se desea que el usuario obtenga información veraz, la imagen generada debe mostrarse en forma clara, sin ambigüedades y con la menor cantidad de datos extrínsecos. La figura 1.1 presenta en forma de diagrama a bloques las diferentes etapas que intervienen en la reconstrucción tridimensional de un objeto u órgano.

La calidad de la imagen representada en forma tridimensional depende grandemente de tres de sus componentes principales:

- La precisión de la segmentación del objeto. La extracción de la estructura de interés dentro de una imagen 2D o de un volumen 3D para proporcionar información sobre los límites del objeto, es uno de los problemas más desafiantes en el campo. Existen segmentaciones desde una clasificación por umbral en niveles de gris, por identificación de patrones (features) en la imagen, etc.
- La precisión de las normales a la superficie. Determinadas por vectores de posición, promedios entre vecinos y por gradientes en los niveles de gris de la imagen en 2D.
- La calidad de sombreado. Se han generado diversos algoritmos para modelar la distribución de la luz reflejada por un objeto.

El objeto reconstruido y proyectado en la pantalla gráfica puede tener diversos manejos que permiten una interacción más amigable con el usuario, además de ayudar a la explotación de la información visual presente. Entre las manipulaciones adicionales a las contempladas en este trabajo, existen:

- **Retroproyección multiplanar:** Procedimiento para extraer una rebanada simple de un volumen a cualquier ángulo.
- **Color falso:** También llamado método de rango coloreado. Es una operación empleada para distinguir objetos o regiones dentro de un volumen.
- **Antialiasing:** Remueve bordes recortados dentro de la imagen y mantiene la fidelidad de la imagen reconstruida eliminando ruido (alta frecuencia) en la imagen final.
- **Ecuilibración del histograma:** Operación utilizada para aumentar el rango dinámico de la imagen. En este procedimiento se intenta obtener una función de densidad constante en el rango de niveles de grises.
- **Disección digital:** Es un corte hacia adentro del objeto tridimensional de manera que se observen las capas que lo componen.

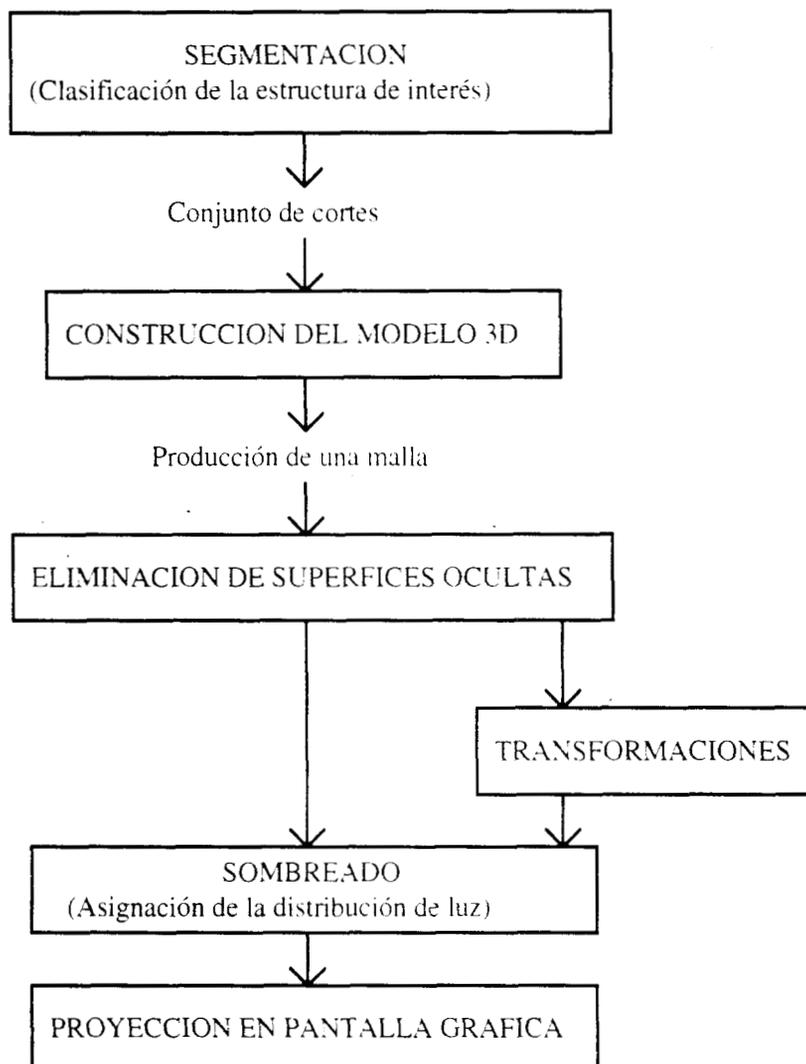


figura 1.1. Diagrama a bloques para la representación de imágenes médicas tridimensionales.

Este trabajo consiste por una parte en la investigación de técnicas para la representación gráfica en 3D y por otra parte en la implementación de las más adecuadas al problema de reconstrucción de imágenes médicas. El problema de la segmentación constituye por sí mismo una extensa línea de investigación, por lo que éste trabajo asume que la etapa de clasificación ya ha sido efectuada. Se inicia entonces de un conjunto de contornos para atacar a partir del segundo bloque de la figura 1.1.

En el siguiente capítulo se plantean formalmente los objetivos de la tesis, en el capítulo 3 se plantean los conceptos teóricos según lo establecido en Computación Gráfica, y lo reportado en representación tridimensional de imágenes médicas, que aunque no se aplicaron del todo, se consideraron dignos de comentarse por sus efectos en el resultado final de la representación. En el capítulo 4 se plantea el método y el desarrollo de la implementación de los algoritmos, y los problemas y restricciones de la implementación. Los resultados se reportan en el capítulo 5 y en el mismo se discuten. Por último, en el capítulo 6 se realiza un resumen y se concluye sobre la experiencia adquirida en el desarrollo de este trabajo.

CAPITULO 2.

OBJETIVOS

OBJETIVO GENERAL:

INVESTIGAR TECNICAS DE RECONSTRUCCION TRIDIMENSIONAL DE IMAGENES MEDICAS A PARTIR DE CORTES TOMOGRAFICOS.

EN PARTICULAR:

- IMPLANTAR UNA TECNICA DE CONSTRUCCION DE UN MODELO DE SUPERFICIE A PARTIR DE CONTORNOS.
- IMPLANTAR Y COMPARAR LOS ALGORITMOS DE SOMBREADO DEL MODELO DE SUPERFICIE.

CAPITULO 3.

CONCEPTOS TEORICOS

3.1. TRANSFORMACIONES EN 3D

Las transformaciones geométricas son muy importantes para generar imágenes en un espacio 3D. Se utilizan para lograr el efecto de observar el objeto en diferentes direcciones y posiciones.

3.1.1. Traslación.

Es la transformación que traslada un punto (x,y,z) a un nuevo punto (x',y',z') a través de (L,M,N) :

$$[x', y', z', 1] = [x, y, z, 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ L & M & N & 1 \end{bmatrix}$$

donde L, M, y N, son las distancias a ser movidas desde x, y, y z, respectivamente .

3.1.2. Rotación.

En tres dimensiones es útil realizar transformaciones de rotación a través de cualquier eje.

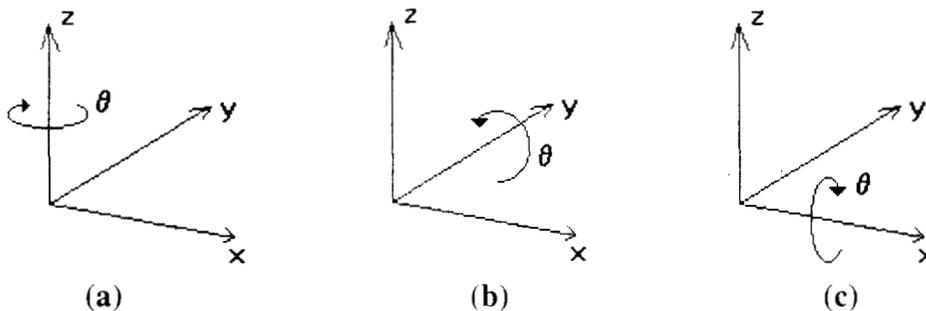


figura 3.1. Rotaciones en tres dimensiones. (a)En el eje z. (b)En el eje y. (c)En el eje x.

Si la rotación es en el plano z, la transformación solo afecta los valores de las coordenadas x y y. La rotación alrededor del eje z, con un ángulo θ , se lleva a cabo mediante la siguiente transformación:

$$[x', y', z', 1] = [x, y, z, 1] \begin{bmatrix} \cos \theta & -\text{sen} \theta & 0 & 0 \\ \text{sen} \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La rotación en el eje coordenado y:

$$[x', y', z', 1] = [x, y, z, 1] \begin{bmatrix} \cos \theta & 0 & \text{sen} \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\text{sen} \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La rotación en el eje coordenado x:

$$[x', y', z', 1] = [x, y, z, 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\text{sen} \theta & 0 \\ 0 & \text{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.1.3. Reflexión.

Un objeto es reflejado a través de un plano mediante la manipulación de los elementos diagonales en la matriz de transformación en 3D. El objeto es reflejado a través de un plano generando una imagen espejo de todos los puntos sobre el lado opuesto del plano. La matriz para una transformación en los planos

$$XY: \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad YZ: \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{y } XZ: \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

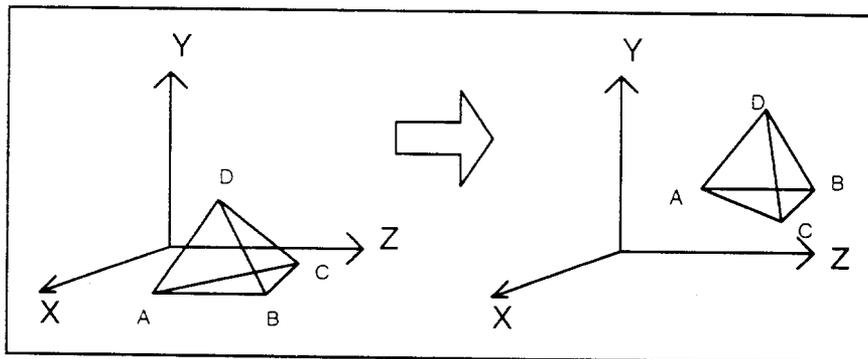


figura 3.2. Reflexión sobre el plano YZ.

3.1.4. Cambio de sistema de coordenadas.

Cualquier tipo de transformación se puede realizar con un cambio de sistema de coordenadas. El cambio de sistema de coordenadas es de gran utilidad cuando se desea que diversos objetos, con un propio sistema de coordenadas, se desplieguen juntos en un sistema de coordenadas global. La transformación desde el sistema coordenado XYZ al $X'Y'Z'$ se obtiene trasladando del punto origen O a O' .

$$P' = P.T = [X, Y, Z, 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ O & P & Q & 1 \end{bmatrix} = [X', Y', Z', 1].$$

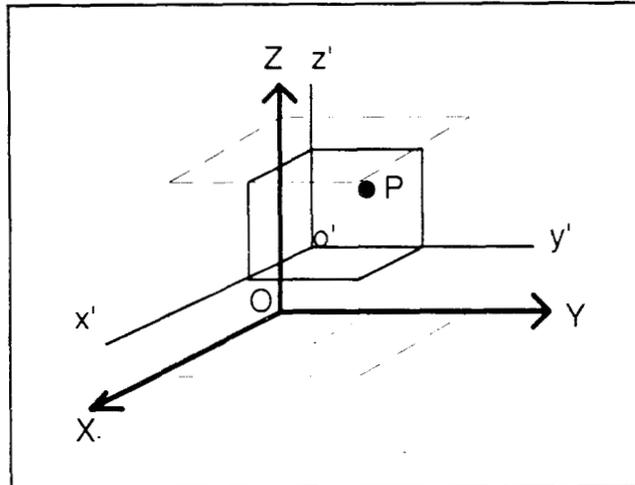


figura 3.3. Punto P en dos diferentes ejes coordenados.

3.2. PROYECCIONES

En general, las proyecciones transforman puntos dentro de un sistema de coordenadas de dimensión n a puntos en un sistema de coordenadas de dimensión menor que n . Para nuestro propósito solo trataremos la segunda y tercera dimensión, ya que se discutirá cómo desplegar objetos tridimensionales en una pantalla gráfica. Debido a que la pantalla es un espacio bidimensional se necesita esta herramienta matemática para obtener una representación en dos dimensiones de un objeto real tridimensional.

La proyección de un objeto tridimensional está definida por rayos de proyección derechos (*proyectores*) surgiendo desde un *centro de proyección*, y que pasan a través de cada punto del objeto e intersectan el *plano de proyección*.

El tipo de proyecciones que se describirán son conocidas como *proyecciones geométricas planares* a causa de que la proyección es sobre un plano y los proyectores son rectilíneos. Estas proyecciones son divididas en dos clases básicas: *perspectiva* y *paralela*. Si la distancia entre el centro de proyección y el plano de proyección es finita, entonces es perspectiva; si la distancia es infinita, la proyección es paralela. Esta última llamada así porque los proyectores son paralelos debido a que el centro de proyección se localiza en un punto infinito.

La proyección paralela a su vez se subdivide en *ortográfica* y *oblicua*, las cuales difieren en la relación entre la dirección de proyección y la normal al plano de proyección. En proyección ortográfica estas direcciones son las mismas (proyector normal al plano de proyección) y en oblicua no lo son.

3.2.1. Proyección central.

La proyección central es paralela y ortográfica, el centro de proyección está localizado en uno de los ejes coordenados del objeto real (espacio mundo), donde los ejes coordenados del ojo están establecidos en el centro de proyección (fig. 3.4).

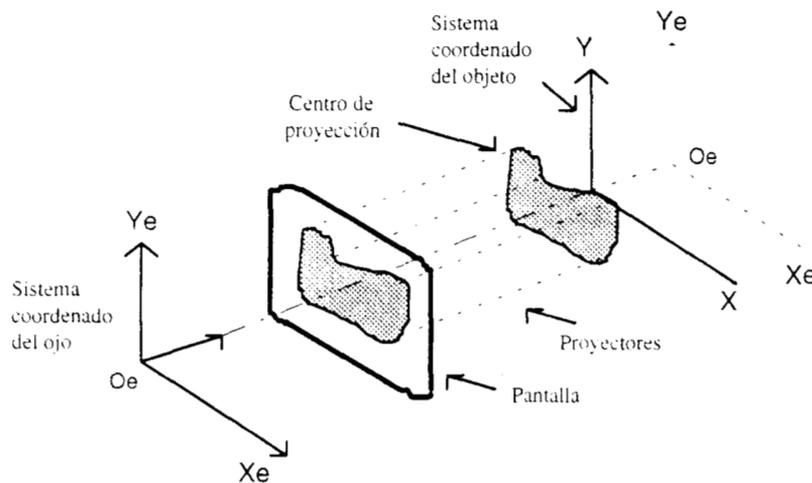


figura 3.4. Proyección central.

3.2.2 Proyección Cavalier.

La proyección Cavalier es muy frecuentemente utilizada y se clasifica en paralela y oblicua. La dirección de proyección (proyector) forma un ángulo de 45° con el plano de proyección. La figura 3.5. muestra la proyección Cavalier de un cubo unitario dentro del plano (x,y) , las líneas inclinadas son las proyecciones de las orillas de cubo que son perpendiculares al plano (x,y) (pero no al plano de proyección) y forman un ángulo α con el proyector.

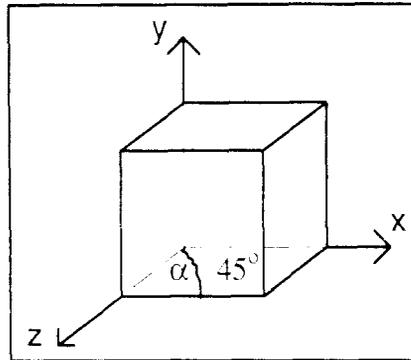


figura 3.5. Proyección Cavalier.

3.2.3. Proyección en un punto arbitrario.

La pantalla de despliegue es colocada perpendicularmente a la línea que conecta el ojo del usuario con el origen en el espacio mundo. El sistema coordenado del ojo orienta los ejes para que la línea de visión sea dirigida hacia el origen del espacio mundo (eje Z_e), los puntos del eje X_e a la derecha y los puntos del eje Y_e hacia arriba (fig. 3.6).

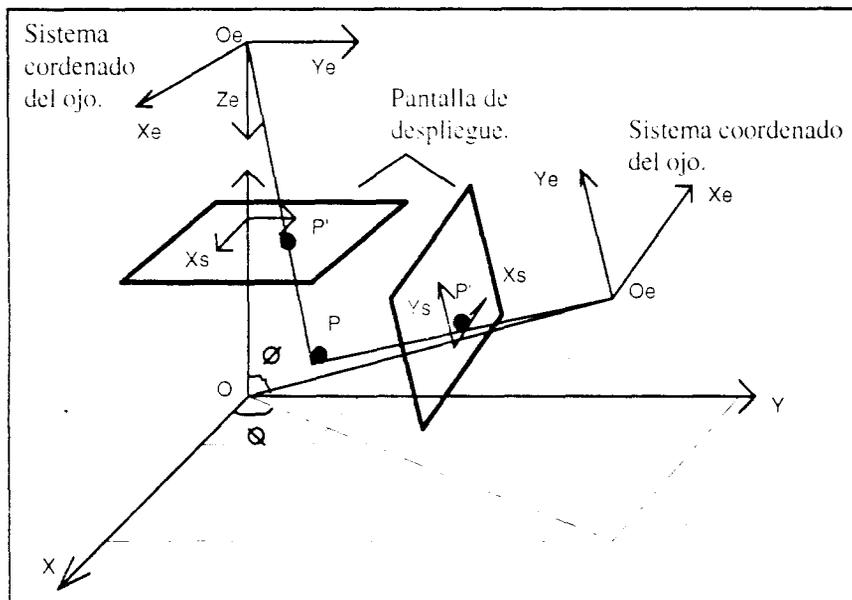


figura 3.6. Proyección en un punto arbitrario.

3.3. REPRESENTACION DE CURVAS Y SUPERFICIES.

La necesidad de modelar curvas y superficies se crea por dos motivos: el poder crear modelos que existen en el mundo real y modelar objetos físicos inexistentes. Para imágenes médicas la interpolación es empleada para convertir los cortes o rebanadas bidimensionales de alguna estructura anatómica digitalizada a partir de CT, MRI, PET, SPECT, o US, en un

objeto continuo tridimensional. Por lo tanto su objetivo es llenar los espacios entre cortes adyacentes por medio de aplicar una función de promediación sobre los voxels.

A causa de que nos encontramos a menudo con objetos en 3D con estructuras complejas, se han desarrollado diversas técnicas especiales para modelar y generar imágenes más realistas. Aunque estos objetos pueden ser aproximados con una precisión relativamente fina, estas representaciones pueden llegar a ser muy burdas e intratables¹. Nosotros necesitamos una representación más práctica, tratable tanto por la computadora como por el usuario que intenta modelar curvas o superficies.

3.3.1. Continuidad geométrica y continuidad paramétrica.

La continuidad geométrica (G) y la continuidad paramétrica (C) entre dos segmentos de curva que se unen en un punto, es un parámetro muy importante que debe tomarse en cuenta al modelar un objeto.

Si las direcciones (no necesariamente las magnitudes) de los vectores tangenciales de los dos segmentos de curva que se unen en un punto común son iguales, entonces la curva tiene una continuidad geométrica G¹. Continuidad G¹ significa que la 'suavidad' o caída de ambas curvas en el punto de unión son iguales. Para que dos vectores tangenciales VT₁ y VT₂, tengan la misma dirección, es necesario que uno sea el múltiplo escalar del otro (VT₁=k *VT₂, ∀ k>0).

Si los vectores tangenciales de ambas terminaciones de segmentos son iguales (dirección y magnitud), entonces la aproximación tiene continuidad paramétrica de primer orden (C¹). Si la dirección y la magnitud de las n derivadas $\frac{d^n}{dt^n}[Q(t)]$ son iguales en el punto de unión, la curva presenta continuidad Cⁿ.

3.3.2. Funciones paramétricas.

La forma dominante para modelar curvas y superficies es la función paramétrica. Esta aproximación se ha reportado en diversos trabajos sobre reconstrucción de imágenes médicas. La función paramétrica define los puntos en el espacio 3D utilizando tres polinomios univariados (interpolación de una curva) o bivariados (interpolación a través de una superficie), uno por cada eje. Los coeficientes del polinomio son seleccionados de tal forma que la curva aproximada represente lo deseado. Aunque existen aproximaciones de n-grado, únicamente se estudiará el grado cúbico debido a que ofrece mayor flexibilidad que los polinomios de ordenes mayores. Los polinomios de orden alto pueden producir ondulaciones no deseables, además de que el costo computacional se incrementa considerablemente.

Un punto en una curva se representa por el vector $P(t_0) = [x(t_0) \quad y(t_0) \quad z(t_0)]$, donde cada polinomio cúbico se define como:

¹ El modelo del objeto está formado por poliedros que se unen unos con otros por sus vértices. Estos poliedros son conocidos como facetas. Siempre que se tenga una mayor cantidad de facetas se tendrá una más exacta aproximación a la superficie.

$$\begin{aligned}
 x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x, \\
 y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y, \quad \text{para } 0 \leq t \leq 1. \\
 z(t) &= a_z t^3 + b_z t^2 + c_z t + d_z.
 \end{aligned}$$

Un segmento de curva $Q(t)$ se define por ciertas restricciones en los puntos terminales, en los vectores tangenciales y en la continuidad entre segmentos. Reescribiendo en forma matricial la curva $Q(t) = [x(t) \ y(t) \ z(t)] = T \cdot C$,

$$\text{donde } T = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \text{ y } C = \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix} = M \cdot G = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix}.$$

La matriz M se conoce como la *matriz base* y G involucra restricciones geométricas (puntos terminales o vectores tangentes) por lo que se le conoce como el *vector geométrico*. Los pesos son cada polinomio cúbico de t , y son conocidos como las *funciones de mezcla* (blending functions), $B = T \cdot M$.

Las curvas o superficies de *Bézier* están definidas por dos puntos terminales y otros dos puntos que controlan los vectores tangenciales de los puntos terminales. Por otro lado las curvas o superficies *Splines* están definidas por cuatro puntos de control.

3.3.3. Superficies de Bézier.

La curva de Bézier indirectamente especifica el vector tangente a los puntos terminales por medio de dos puntos intermedios que no están sobre la curva (fig. 3.7). Los vectores tangentes en los puntos iniciales (VT_i) y terminales (VT_f) se calculan como $VT_i = Q'(0) = 3(P_2 - P_1)$ y $VT_f = Q'(1) = 3(P_4 - P_3)$.

Bézier define la superficie $Q(u,v)$ en tres dimensiones generando el producto cartesiano de dos curvas. Se utilizan dos *funciones de mezcla* (B) similares para cada parámetro (u, v).

$$Q(u,v) = \sum_{i=0}^n \sum_{j=0}^m p_{i,j} B_{i,n}(u) B_{j,m}(v)$$

donde $B_{i,n}(u) = C(n,i)u^i(1-u)^{n-i}$ y $C(n,i) = \frac{n!}{i!(n-i)!}$ (coeficiente binomial).

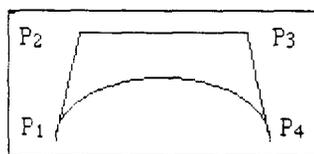


figura 3.7. Curva de Bézier y sus cuatro puntos de control.

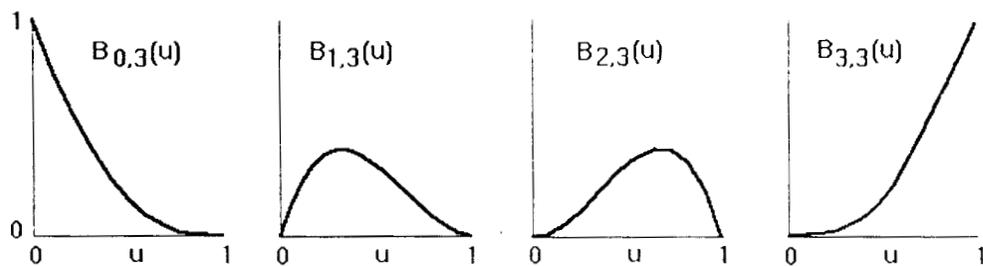


figura 3.8. Las cuatro *funciones de mezcla* de Bézier para $n=3$

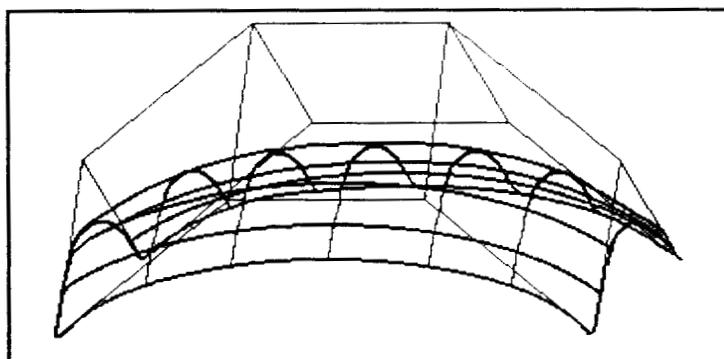


figura 3.9. Superficie de Bézier, $n=2$, $m=3$. Se muestra una superficie de Bézier con $(n+1) \times (m+1)$ puntos de control arreglados en una malla.

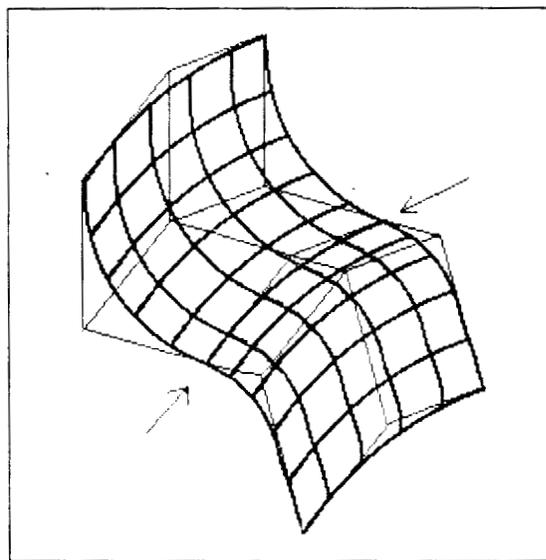


figura 3.10. Unión de dos pedazos de superficie de Bézier en el límite indicado por las flechas. Los puntos de control se escogen para mantener una continuidad de primer orden en el límite de ambos pedazos.

Propiedades de las curvas y superficies de Bézier:

1. Los puntos de control satisfacen dos propiedades importantes: la curva atraviesa los dos puntos terminales y es tangente en los mismos.

2. Permite representar formas multi-valuadas. Por lo tanto, si el primer y último punto de control coinciden, la curva es cerrada.
3. Al mover cualquier punto de control cambiará la forma de cada parte de la curva. En las *funciones de mezcla* de la figura 3.8 se observa que todas las funciones son diferentes de cero (excepto en $u=0$ y $u=1$) y consecuentemente la localización de cada punto de control intermedio influencia la localización de la curva.
4. La variación de la curva es en forma de disminución y por lo tanto garantiza su *no* oscilación cerca de los puntos de control definidos.

3.3.4. Método B-Spline.

Aunque las curvas y superficies de Bézier son muy utilizadas en muchos problemas para modelar diversas formas, se requiere de construcciones geométricas más complejas para garantizar continuidad geométrica cuando se unen segmentos de curvas o superficies. La curva Spline de orden cúbico tiene continuidad de C^0 , C^1 y C^2 (un orden mayor que la curva de Bézier). Los coeficientes polinomiales para las curvas *splines cúbicas naturales* dependen de los n puntos de control y la curva aproximada atraviesa cada uno de estos puntos .

Los *splines naturales* tienen dos desventajas: los coeficientes dependen de todos los n puntos de control y su cálculo requiere de invertir una matriz de $n+1$ por $n+1$, lo que afecta la velocidad de formación de la curva completa.

La más común de las curvas *splines* es la formulación B-Spline que utiliza un conjunto de funciones con soporte local o control local (depende únicamente de unos pocos puntos de control vecinos) como se muestra en la figura 3.11. Los B-Splines conservan el mismo orden de continuidad que los *splines naturales*, pero tienen la desventaja de que la curva aproximada no atraviesa los puntos de control.

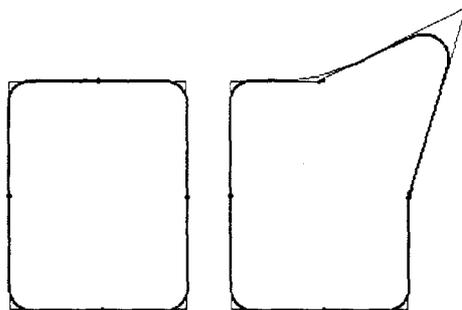


figura 3.11. Las curvas B-Spline tienen control local. Si se mueve uno de los puntos de control no afecta a los demás que conforman la curva.

Existen tres tipos de B-Splines: los B-Splines no-rationales uniformes, los B-Splines no-rationales no-uniformes y los B-Splines racionales no-uniformes. El término *no-razional* se utiliza para diferenciarlos de las curvas cúbicas polinomiales racionales, donde $x(t)$, $y(t)$ y $z(t)$, se definen como la razón de dos polinomios cúbicos ($x(t) = X(t)/W(t)$) cuyos puntos de control son definidos en un espacio homogéneo (x, y, z, W): curvas conocidas como NURBS. El término *uniforme* significa que los intervalos del parámetro t son constantes. La B se incluye por *basis*, ya que pueden ser representados como la suma pesada de funciones base de polinomios, en contraste con los *splines naturales* donde esto no se cumple. Existen dos formas de escoger los nodos de control: la formulación uniforme B-Spline no periódica

y la formulación uniforme B-Spline periódica. La primera se utiliza para modelar curvas abiertas, la segunda modela curvas cerradas como lo es un círculo.

Cada segmento de curva B-Spline no-racional uniforme Q_i se define por cuatro puntos de control (vector geométrico) y se representa en forma vectorial como

$$Q_i(t) = T_i \cdot M_{B_s} \cdot G_{B_s}, \quad \text{para } t_i \leq t \leq t_{i+1},$$

donde

$$T_i = \begin{bmatrix} (t-t_i)^3 & (t-t_i)^2 & (t-t_i) & 1 \end{bmatrix}, \quad M_{B_s} = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \quad \text{y} \quad G_{B_s} = \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix}.$$

Las funciones de mezcla de las curvas B-Spline son calculadas por el producto $B_{B_s} = T_i \cdot M_{B_s}$ y se representan en la figura 3.12.

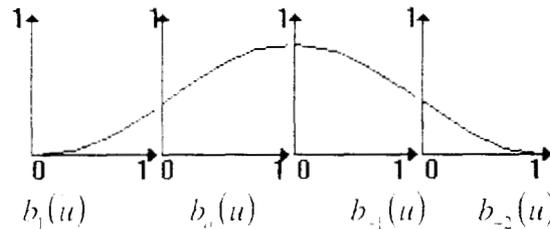


figura 3.12. Gráfica de las funciones base para la formulación B-Spline cúbica.

En este tipo de curva puede ocurrir que exista continuidad paramétrica, pero no así continuidad geométrica del mismo orden, sino menor. Esto sucede cuando existen múltiples puntos de control dentro de una curva. Puede demostrarse con relativa facilidad que la segunda derivada de dos segmentos que se unen en un punto es

$$\frac{d^2}{dt^2} x_i|_{t=t_i} = \frac{d^2}{dt^2} x_{i+1}|_{t=t_i} = P_{i-2} - 2P_{i-1} + P_i,$$

asegurando de esta forma continuidad C^1 . Sin embargo, cuando tres puntos son iguales sea $P_{i-2} = P_{i-1} = P_i$, el segmento de curva $Q_i(t) = T \cdot M_{B_s} \cdot G_{B_s} = B_{B_s} \cdot G_{B_s}$ se convierte en una recta $Q_i(t) = B_{B_{s-3}} \cdot P_{i-3} + (B_{B_{s-2}} + B_{B_{s-1}} + B_{B_s}) \cdot P_i$ (fig. 3.13).

Según se demostró anteriormente las curvas B-Spline uniformes no-racionales no atraviesan los puntos de control, sin embargo, Yamaguchi[38] y Barsky y Greenberg[4], realizan una inversión en el algoritmo de manera que obtengan unos nuevos puntos de control que al ser interpolados, aseguren el paso de la curva aproximada a través de los puntos de control iniciales. Para obtener estos vértices de control es necesario resolver un sistema de n ecuaciones lineales para los n vértices de control que se deseen.

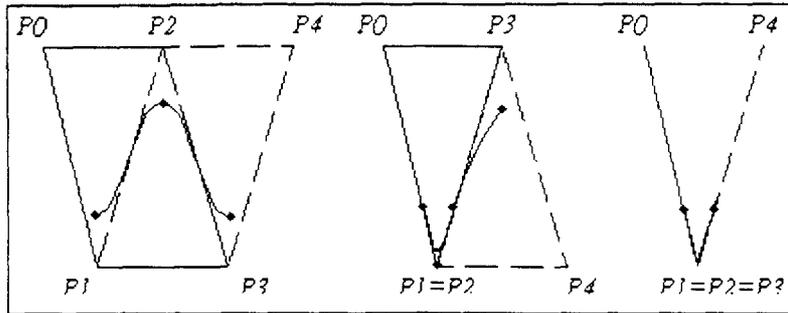


figura 3.13. Efecto de múltiples puntos de control sobre la curva B-Spline uniforme.

La diferencia entre estos dos métodos es la forma en que resuelve el sistema de ecuaciones lineales. Yamaguchi utiliza el método de Gauss-Seidel donde su tiempo de cálculo depende de la velocidad de convergencia a un margen de error. Barsky emplea un manejo de matrices de donde obtiene una matriz triangular de la cual resultan los vértices de control deseados por medio de una sustitución hacia atrás.

Para el caso de aproximaciones en superficies por B-Splines (fig. 3.14), se requieren 16 parches o facetas vecinas para interpolar un parche. La superficie B-Spline requiere la resolución de $m \times n$ ecuaciones lineales cada una con 16 variables.

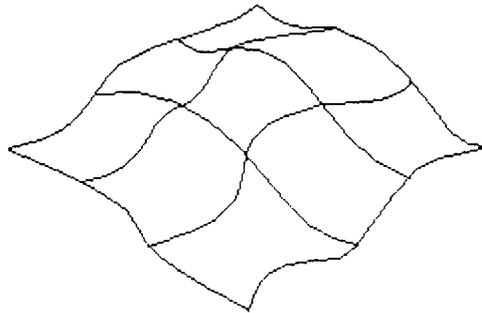


figura 3.14. Una superficie B-Spline es un mosaico de parches o facetas.

Las curvas B-Spline no-uniformes no-rationales ofrecen algunas ventajas sobre las uniformes. La continuidad en los puntos de unión puede reducirse de C_2 , a C_1 , a C_0 , a ninguna. Si la continuidad es reducida a C_0 , entonces la curva interpola un punto de control, pero sin el efecto indeseable de los B-Splines uniformes donde ambos segmentos son líneas rectas.

Las funciones de mezcla de las curvas no uniformes no son simples, ya que dependen de los intervalos que forman el segmento y además actúan recursivamente en términos de funciones de mezcla de orden más bajo.

En cuanto a las B-Splines no-uniformes racionales son muy útiles por dos razones. La primera y más importante es que son invariantes a rotaciones, escalamientos, traslaciones y a proyecciones de los puntos de control (las curvas no-rationales son invariantes a todas las anteriores menos a las proyecciones). La segunda ventaja es que pueden definir muy

precisamente cualquier sección cónica a diferencia de los no-rationales donde se necesitarían múltiples puntos de control.

3.4. ELIMINACION DE SUPERFICIES OCULTAS

Uno de los problemas más desafiantes en la reconstrucción tridimensional es la eliminación de partes escondidas de imágenes de objetos sólidos, con la finalidad de evitar cálculos innecesarios de aquellas partes del objeto que no son visibles.

Aunque existen muchos algoritmos, no es fácil determinar si existe alguno mejor que otro. Muchas de las diferencias entre ellos provienen de diversos requerimientos, por ejemplo, los algoritmos operan sobre diferentes estructuras (en 3D), generan diferentes formas de salida o soportan imágenes de diferentes complejidades. Sin embargo, todos utilizan formas de separación geométrica para distinguir partes visibles de un objeto. Esta separación geométrica localiza los objetos más cercanos al observador y que por lo tanto son visibles.

3.4.1. Visibilidad de un objeto simple o convexo².

El problema de visibilidad de un objeto simple puede resolverse examinando las direcciones de las normales de las facetas que componen el objeto. La normal de la faceta se define como un vector perpendicular al plano de la faceta.

Al desarrollar la prueba de visibilidad, se introducen dos vectores asociados con cada superficie, el vector normal y el vector de la línea de visión. El vector normal está definido como un vector apuntando hacia afuera de la superficie (\mathbf{n}). El vector de visión (\mathbf{l}) representa la línea que pasa a través del punto de visión en el espacio del observador e inicia en el vector normal (fig. 3.15.). Si el ángulo Φ entre los vectores \mathbf{n} y \mathbf{l} se encuentra entre 0° y 90° , la superficie es visible y debe ser desplegada.

Para calcular el *vector normal* a cada superficie se necesitan el número de vértices de cada superficie en dirección de las manecillas del reloj, donde el vértice de inicio es arbitrario. Después de numerar los vértices, se identifica el vector u dirigido desde el vértice número uno (V1) al número 2 (V2) y el vector v , dirigido desde V1 hasta V3. Entonces el vector normal n se define como:

$$\mathbf{n} = \mathbf{u} \times \mathbf{v} = (bf - ce, cd - af, ae - bd)$$

donde

$$\mathbf{u} = (a, b, c), \quad \text{y} \quad \mathbf{v} = (d, e, f).$$

Para calcular el *vector de visión* se extiende una línea desde el punto de visión $P(D, \theta, \Phi)$, hasta cualquier punto sobre la superficie del objeto (por conveniencia se extiende hasta el vector normal).

² Un objeto convexo se define como aquel en el cual los segmentos que unen un par de puntos en el objeto están enteramente contenidos en el objeto (i.e., cubo, esfera, pirámide, rectángulo, triángulo, círculo, etc.).

El punto de visión se identifica en términos de sus coordenadas esféricas.

$$l = (x, y, z) - (x_1, y_1, z_1)$$

donde

$$x = D \cdot \text{sen}\phi \cdot \text{cos}\theta, \quad y = D \cdot \text{sen}\phi \cdot \text{sen}\theta, \quad z = D \cdot \text{cos}\phi;$$

y

$$D = \sqrt{x^2 + y^2 + z^2}, \quad \theta = \tan^{-1}(y/x), \quad \phi = \cos^{-1}(z/D).$$

La *prueba de visibilidad* se realiza al encontrar el ángulo entre los vectores \mathbf{n} y \mathbf{l} por medio del producto punto entre ellos,

$$\mathbf{n} \cdot \mathbf{l} = |\mathbf{n}| \cdot |\mathbf{l}| \cdot \cos\Phi$$

$$\Phi = \cos^{-1}\left(\frac{\mathbf{n} \cdot \mathbf{l}}{|\mathbf{n}| \cdot |\mathbf{l}|}\right),$$

donde

$$\mathbf{n} \cdot \mathbf{l} = (n_1, n_2, n_3) \cdot (l_1, l_2, l_3) = n_1 l_1 + n_2 l_2 + n_3 l_3.$$

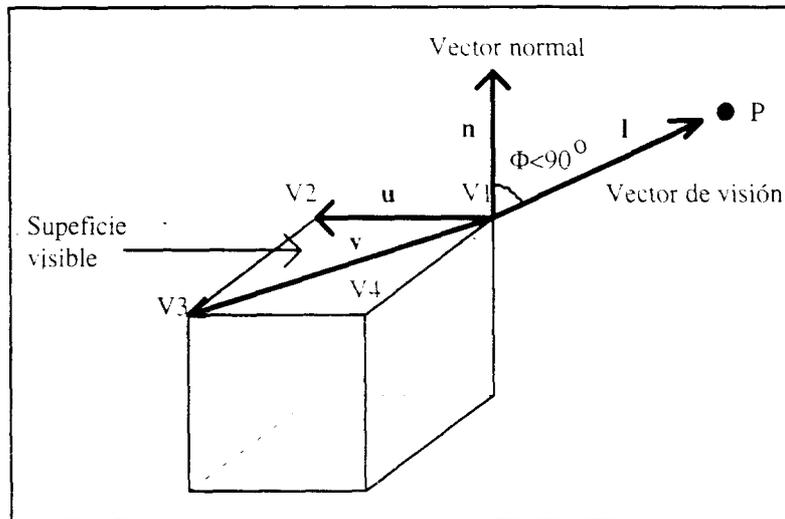


figura 3.15. Vectores normal y visión.

3.4.2. Visibilidad de diversos objetos.

En esta sección se tratan en forma global algunos algoritmos para desplegar diversos objetos con las líneas ocultas removidas. Estos algoritmos operan en el *espacio objeto* y en el *espacio imagen*. En el espacio objeto los algoritmos utilizan las relaciones geométricas entre los objetos en la escena para determinar qué objeto es visible y cuál no. Por otro lado, los algoritmos en el espacio imagen, operan en la imagen final para determinar qué pixel es visible al observador.

Las dos aproximaciones se utilizan con diferentes características de empleo y con requerimientos de hardware diferentes. Los métodos en el espacio objeto son empleados para eliminar objetos completos dentro de la imagen y el tiempo de cálculo se incrementa con el número de objetos en la escena. Los algoritmos en el espacio imagen se utilizan para eliminación de superficies y el tiempo de cálculo se incrementa con la complejidad de las partes visibles de la imagen.

3.4.3. Algoritmo Z-Buffer.

De todos los algoritmos en el espacio imagen el Z-Buffer es el más sencillo. Para cada pixel que conforma al objeto, se tiene que registrar su profundidad con respecto al usuario a fin de determinar cuál es el visible.

El algoritmo consiste en:

1. Para todos los pixeles en la escena, inicializa los arreglos de profundidad a un valor constante ($\text{profundidad}[x,y]=1$) y de intensidad ($\text{intensidad}[x,y]=\text{gris_de_fondo}$).
2. Para cada polígono en la escena, encuentra todos los pixeles (x,y) que caen dentro del polígono. Para cada uno de esos pixeles:
 - 2.1. Calcula la profundidad z del polígono en (x,y) .
 - 2.2. Si z es menor que $\text{profundidad}[x,y]$, este polígono está mas cerca del usuario y entonces $\text{profundidad}[x,y] = z$, $\text{intensidad}[x,y]=\text{al nivel de gris calculado para ese punto}$. Si z no es menor que $\text{profundidad}[x,y]$, ya se ha introducido un valor al arreglo que está más cercano al observador.

En muchos casos este algoritmo no es muy empleado debido a que requiere un gran espacio en memoria para almacenar la intensidad y la profundidad de cada pixel de la imagen.

3.5. ILUMINACION Y SOMBREADO.

227468

Finalizado el proceso de reconstrucción, se dispone ya de una malla formada por polígonos o facetas que pueden ser rellenadas con algoritmos de sombreado. Los algoritmos de sombreado modelan los tipos de iluminación que se reflejan en el objeto de interés y su distribución a través del mismo. La intensidad luminosa en cierto punto sobre el objeto de interés se codifica en niveles de gris y su valor depende de la posición, orientación y propiedades de la superficie, además de las características de la fuente de iluminación.

3.5.1. Modelos de iluminación.

El modelo de sombreado tiene dos ingredientes principales. El primero representa las propiedades de la superficie y el otro las propiedades de la iluminación que lo invade. La principal propiedad de superficie es su *reflectancia*, la cual determina qué cantidad de la luz incidente es reflejada. Otra propiedad de superficie que tiene un papel muy importante es la *transparencia*, donde una superficie puede permitir que alguna luz sea transmitida a través de ella.

En el cálculo de intensidad, la iluminación del objeto es tan importante como las propiedades de la superficie del mismo objeto. La escena puede tener alguna iluminación que sea uniforme desde todas las direcciones y puede ser producto de múltiples reflexiones de luz de muchas superficies en el medio. En este caso tenemos una iluminación conocida como *luz ambiental*:

$$E_{pa} = R_p I_a.$$

donde E_{pa} , es la energía que irradia el objeto en el punto P debido a una iluminación difusa del ambiente, R_p es el coeficiente de reflexión de todas las fuentes difusas e I_a es la intensidad luminosa que es constante para todas las fuentes (fig. 3.5.1).

Las contribuciones de iluminación desde fuentes de luz específicas, causarán que el sombreado de una superficie cambie con respecto a la orientación de la fuente de luz. esto incluirá efectos de reflexión especulares. El primero de esos efectos es debido a la *ley de Lambert*, la cual establece que la energía que invade una superficie varía con el coseno del ángulo de incidencia de la luz. Este tipo de iluminación conocido como *iluminación difusa* toma en cuenta las propiedades del objeto, el cual se considera con una *reflexión difusa* en toda su área. La energía que refleja el objeto se determina como:

$$E_{pd} = I_p R_p \cos \iota.$$

donde I_p es la iluminación de la fuente,
 R_p es la reflexión del objeto, e
 ι es el ángulo entre el rayo incidente y la normal.

La cantidad de energía que detecta el observador debido a la *iluminación especular* depende del *coeficiente de reflexión especular* del material $W(i)$ y el ángulo s entre el rayo reflejado y el vector de visión (fig. 3.5.1.),

$$E_{ps} = [W(\iota)(\cos s)^n] I_{ps}.$$

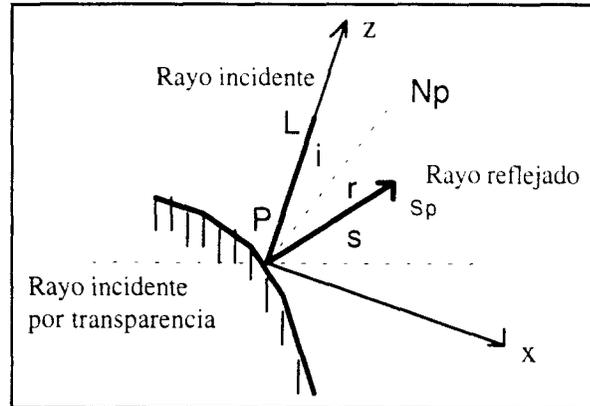


figura 3.5.1. La intensidad luminosa del punto P se determina por la luz reflejada y transmitida. El vector \mathbf{N}_p es normal a la superficie en P.

La propiedad de *transparencia* agrega otra contribución a la energía reflejada en el punto P sobre el objeto,

$$E_{pt} = T_p E_{pb},$$

donde T_p es el coeficiente de transmisión, que varía de 0 a 1 y determina cuanta luz es transmitida en el punto P. E_{pb} es la energía que llega a P.

Además de las energías mencionadas puede haber diversas fuentes de luz en la escena, en donde cada una contribuye con diferentes energías. Por último, la intensidad en el punto puede verse afectada por algunas sombras en el medio que pueden bloquear parcialmente la intensidad de luz que invade un objeto. En total la energía que detecta el ojo a partir del punto P, es la suma de las energías de los efectos individuales.

Energía Total:

$$E = E_{pd} + \sum E_{ps} + E_{pt}.$$

Los ángulos requeridos en el modelo de sombreado pueden determinarse completamente por el *vector normal* a la superficie. Ya que fué utilizado en la eliminación de superficies ocultas (sección 3.4), no es necesario volver a calcularlo. Por otra parte, todos los cálculos angulares son simplificados si se asume que la fuente luminosa se encuentra lo suficientemente retirada del objeto. Por conveniencia estos vectores son normalizados para tener longitud unitaria.

Los ángulos i y s pueden calcularse fácilmente, considerando que los vectores son unitarios:

$$\begin{aligned} \cos i &= \mathbf{L} \cdot \mathbf{N}_p \\ \cos s &= [x_s \quad y_s \quad z_s] \cdot \mathbf{V}' \end{aligned}$$

donde

$$x_s = 2z_n x_n, \quad y_s = 2z_n y_n, \quad z_s = 2z_n^2 - 1,$$

y \mathbf{V}' es un vector que apunta hacia el punto de visión.

3.5.2. Sombreado constante.

Se utiliza un modelo de iluminación para asignar un nivel de gris constante a todo el polígono. Esta asignación es válida si se cumple que:

1. La fuente de luz se encuentre situada en el infinito, para que $\bar{N} \cdot \bar{L}$ sea constante a través del polígono.
2. El observador se sitúe en el infinito, para que $\bar{N} \cdot \bar{V}$ sea constante a través del polígono.
3. El polígono represente la superficie actual a ser modelada, no una aproximación a una superficie curva.

3.5.3. Método de Gouraud.

Gouraud introdujo un sombreado suave con un método de interpolación lineal. Primero se calculan las normales en los vértices de cada faceta por medio de una aproximación a la normal. Cada normal en el vértice es utilizada para calcular la sombra en el vértice, este valor es precisamente la sombra en el vértice.

La distribución de luz dentro de la faceta es interpolada desde el nivel de gris o intensidad en el vértice. La figura 3.5.2. muestra una línea de rastreo que intersecta la faceta a ser sombreada en L y R. El sombreado en L es una interpolación lineal entre las sombras en A y B; la sombra en R es una interpolación entre las sombras C y D. Finalmente la sombra en el punto P sobre la línea de rastreo es linealmente interpolada entre L y R.

Las normales en los vectores pueden calcularse a partir de un modelo de superficie o por una técnica de promediación basada en las normales de las facetas vecinas (fig. 3.5.3.a). La normal en el vértice V es el promedio de las normales de las facetas a su alrededor.

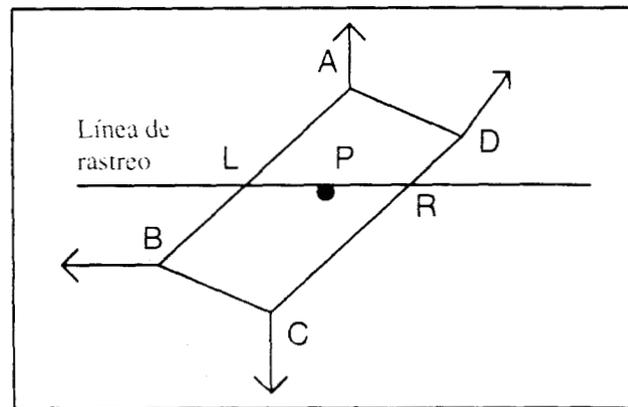


figura 3.5.2 Sombreado de Gouraud.

Algunas veces el sombreado suave a través del límite de una superficie (componente de alta frecuencia) puede ser modificado para mostrar algún borde o filo. En la figura 3.5.3.a, si se promediaran las cuatro facetas se perdería la información del borde muy marcada entre los pares de facetas A-B y C-D. En este caso, se calculan dos vectores (3.5.3.b), uno se obtiene con el promedio de las normales de las facetas A y B, y el segundo vector se obtiene a partir de las normales de C y D.

Los promedios de normales algunas veces producen ciertas anomalías que deben ser corregidas. La figura 3.5.4. muestra una serie de superficies con cierto patrón de ordenamiento, donde la normal promedio en las facetas sería idéntica en todas y resultaría en

un sombreado constante. Para remediar este fenómeno se pueden agregar pequeñas superficies cerca de los límites entre facetas.

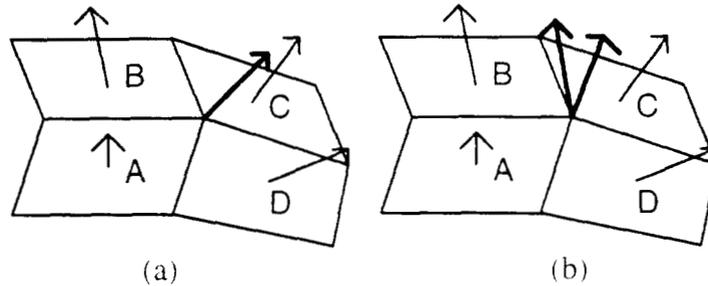


figura 3.5.3. Promedio de vectores normal. (a) Calcula un vector promedio. (b) Calcula dos vectores promedio.

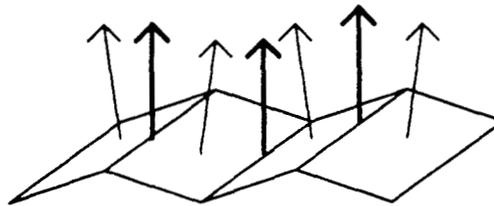


figura 3.5.4. Anomalías al promediar normales. Los vectores en negritas permanecen constantes.

La técnica de Gouraud aunque es simple tiene dos marcadas deficiencias:

1. Si se utiliza en movimiento, el sombreado cambia grandemente.
2. Este método también produce el efecto *Mach Band*¹ [13], que es un efecto perceptual debido al procesamiento en el sistema visual humano. Las discontinuidades en el sombreado causan que el ojo perciba bandas de luz o de oscuridad.

3.5.4. Método de Phong.

Las técnicas de Phong remedian muchos de los problemas introducidos por la técnica de Gouraud con la desventaja de que requiere de mayor cálculo.

La idea es interpolar los vectores normal en vez de las intensidades para aplicar el modelo de sombreado a cada pixel. La normal a lo largo de una orilla de la faceta puede ser expresada como $N_1 + N_2$, una ecuación vectorial que puede implementarse con adiciones incrementales (fig. 3.5.5). Después de determinado el vector normal para cada punto, se emplea el modelo de iluminación deseado para calcular una intensidad en el pixel.

El método de Phong reduce (no las anula) las bandas del efecto Mach, produce mejor los destellos y remueve pieza por pieza las discontinuidades. También comparte un problema con el método de Gouraud: un polígono concavo puede introducir anomalías en el sombreado. En la figura 3.5.6. se muestra un ejemplo. La sombra en el punto P calculada de

¹ Gouraud justifica este efecto por la discontinuidad de la interpolación lineal provocando que la retina actúe como un filtro espacial en dos dimensiones. Dicho filtro atenúa las bajas frecuencias y amplifica las altas frecuencias presentes en la distribución de sombreado.

la interpolación entre L y R, puede variar considerablemente en el punto V. Esta discrepancia en los vectores normal causará un sombreado discontinuo.

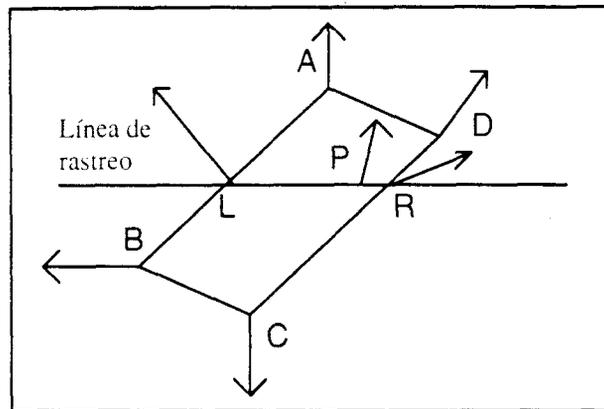


figura 3.5.5. Sombreado por método de Phong.

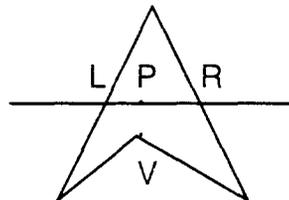


figura 3.5.6. Anomalía en el sombreado de Phong.

3.5.5. Problemas con interpolación en sombreado.

Además de las anomalías que se mencionaron anteriormente en los métodos de Gouraud y Phong, en esta sección se agregan algunas otras que deben tomarse en cuenta.

Silüeta poligonal. No importa qué bien realizada sea la aproximación por el modelo de sombreado a una superficie curva, la silüeta de la orilla sigue siendo poligonal. Este problema puede ser suavizado con un mayor número de facetas y de menor tamaño, pero con su debido incremento en costo computacional.

Distorsión en perspectiva. Se introducen anomalías si la interpolación es llevada a cabo después de una transformación en perspectiva. Este problema también puede mejorarse reduciendo el tamaño de la faceta.

Dependencia con la orientación. Los resultados del sombreado no son independientes de la perspectiva de orientación. Como los valores son interpolados entre vértices y a través de líneas de rastreo, los resultados pueden diferir cuando el polígono es rotado.

Vértices divididos. Pueden existir discontinuidades en el sombreado cuando un vértice está formado por dos polígonos adyacentes de un lado y un polígono mayor que no contribuye en el vértice (fig. 3.5.7).

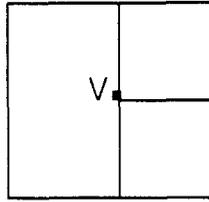


figura 3.5.7. El vértice V está formado por dos polígonos a la derecha, pero no por el izquierdo.

3.6. SOMBREADO EN IMAGENES MEDICAS.

Actualmente los cortes tomográficos que han proporcionado imágenes médicas de mejor calidad son los obtenidos por tomografía computarizada (CT) y resonancia magnética nuclear (MRI). Por esta razón, la mayoría de los trabajos realizados sobre reconstrucción tridimensional a partir de contornos se realizan en base a imágenes de estos dos tipos.

Con esta idea de calidad, algunos investigadores han recapitado en los últimos años (desde 1985) en la posibilidad de estar presentando información irreal en sus imágenes, representaciones que sin embargo ofrecen resultados agradables al observador. En aplicaciones médicas es esencial asegurar la fidelidad de la imagen o por lo menos conocer sus restricciones. Esta es la justificación para el desarrollo de algoritmos que se asemejen aún más al mundo real, involucrando en forma más cercana la información que proporciona la imagen base (imagen anterior a la segmentación).

La descripción de estos algoritmos está basada en el cálculo de gradientes sobre la imagen base ya sea para obtener la intensidad en el pixel o para obtener la normal a la faceta. A diferencia de los métodos de sombreado tradicionales mencionados en el capítulo anterior donde se realiza una estimación de la normal a la superficie por medio de vectores de posición, aquí se introducen otros conceptos para obtener la distribución de luz a través del objeto. En este capítulo se recopilan los principales algoritmos de sombreado utilizados para representación en 3D de imágenes médicas.

3.6.1. Gradientes por niveles de gris [16].

En este método de sombreado se considera que el nivel de gris del voxel en la imagen de RMN o CT, representa la densidad media dentro del elemento de volumen. Por ejemplo, si se encuentran dos tipos de tejido dentro de un voxel, el nivel de gris de dicho voxel será el promedio de los dos tejidos. Asumiendo que la superficie es debida a un borde entre dos regiones de material uniforme, el nivel de gris es una medida media del material en el voxel (fig. 3.6.1). Este efecto se conoce como el efecto de volumen parcial que normalmente deteriora la imagen, pero de la cual se obtiene información acerca de la estructura de la superficie. El nivel de gris por sí solo no proporciona información sobre la dirección de la superficie, pero el gradiente de los niveles de gris vecinos a los voxeles de superficie, si lo lleva a cabo.

Experimentalmente la intensidad del voxel (i,j) en el plano imagen (fig. 3.6.1), se calcula siguiendo la Ley de Lambert para la iluminación difusa como:

$$I(i,j) = A \cos \left[\frac{g(i,j+1,k_T) - g(i,j-1,k_T)}{B} \right]$$

donde, $g(i,j+1,k_T)$ son los niveles de gris en la profundidad $k_T(i,j)$ a la cual se encuentra la superficie. A y B son factores de escala que dependen del rango de niveles de gris de las imágenes originales.

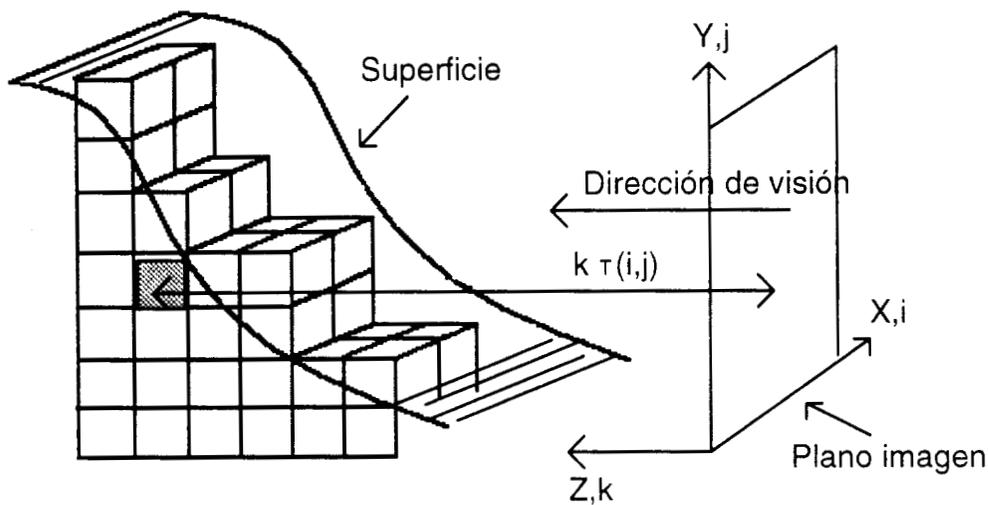


figura 3.6.1. Voxel y sus vecinos a lo largo de la superficie.

El efecto de la incidencia de la luz se logra escogiendo los gradientes en diferentes direcciones. La intensidad del voxel puede ser calculada tomando el cambio en la dirección z.

$$I(i,j) = A \cos \left[\frac{g(i,j,k_T+1) - g(i,j,k_T-1)}{B} \right]$$

En este tipo de representación pueden existir efectos de *aliasing* debidos al tamaño de la ventana. Estos efectos pueden reducirse si se aumenta la ventana gradiente tomando en cuenta las tres direcciones del plano del objeto.

$$G_{(x,y,z)} = 1/9 \sum_{s=-1}^{+1} \sum_{l=-1}^{+1} \sum_{m=-1}^{+1} \left(g(i+l,j+s,k_T+m) - g(i+1,j-s,k_T+m) \right)$$

La intensidad del plano imagen para cada pixel se obtiene con,

$$I(i, j) = A \cos \left[\frac{G_{(x, y, z)}}{B} \right]$$

Características:

- El efecto de diversas fuentes luminosas puede modelarse escogiendo los gradientes en diferentes direcciones.
- Impresión más realista comparado con un sombreado constante debido a un mayor rango dinámico en cada voxel.
- Requiere de mayor espacio en memoria RAM para almacenar las imágenes de RMN o CT.

3.6.2. Sombreado por gradiente z-buffer [5, 12, 31].

Considerando que previamente se obtuvo un ordenamiento según la distancia hacia el observador con el algoritmo z-buffer ($Z(x,y)$), los componentes de la normal a la superficie pueden ser aproximados por el vector gradiente,

$$\nabla z = \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, 1 \right) = (\varphi, \theta, r).$$

La derivada parcial $\frac{\partial z}{\partial x}$ es aproximada como la suma de diferencias ponderadas hacia atrás y hacia adelante de los vecinos (x,y) en el buffer-z,

$$\frac{\partial z}{\partial x} \approx \frac{W_b \delta_b + W_f \delta_f}{W_b + W_f},$$

donde

$$W(\delta) = \begin{cases} 1 & \text{si } \delta \leq a \\ \varepsilon & \text{si } \delta \geq b \\ \frac{1+\varepsilon}{2} + \frac{1-\varepsilon}{2} \cos\left(\frac{\delta-a}{b-a} \pi\right) & \text{otro.} \end{cases}$$

Por otro lado, Tiede y cols. (1990)[35] en su revisión de algoritmos de sombreado reportan un cálculo de normales de la misma forma que Gordon y Reynolds pero con una variación en la estimación de los pesos. Las siguientes ecuaciones muestran esos cambios:

$$\frac{\partial z}{\partial x} \approx W(|\delta_{b_x}|) \delta_{b_x} + W(|\delta_{f_x}|) \delta_{f_x}$$

donde

$$\delta_{b_i} = Z(x, y) - Z(x - 1, y)$$

$$\delta_{f_i} = Z(x + 1, y) - Z(x, y)$$

El vector en coordenadas esféricas resultante se convierte a coordenadas cartesianas. De este algoritmo resultan imágenes muy realistas, pero sus puntos débiles son el bajo rango dinámico y la baja resolución de los ángulos de superficie. Esto es a causa de que son calculados utilizando la posición de los voxels de superficie con una ventana de 3 x 3.

Características:

- Los objetos delgados no son detectados.
- Las normales a la superficie son inexactas.
- Si se trabaja con este modelo, se puede observar un bajo rango dinámico y discretización en la superficie.

3.6.3. Sombreado por gradiente de niveles de gris (GG) [5, 17, 35].

Este algoritmo es similar al reportado por Hoehne y Bernstein (1986), a diferencia de que éste se utiliza para calcular las normales en la superficie y no las intensidades. El procedimiento es el siguiente, conocido el nivel de gris i en la localización (i,j,k) el gradiente se calcula como,

$$G_x = i_{(i+1,j,k)} - i_{(i-1,j,k)}$$

$$G_y = i_{(i,j-1,k)} - i_{(i,j+1,k)}$$

$$G_z = i_{(i,j,k+1)} - i_{(i,j,k-1)}$$

Los componentes de la normal a la superficie son normalizados de la siguiente forma,

$$N_{(x,y,z)} = \frac{G_{(x,y,z)}}{\sqrt{(G_x)^2 + (G_y)^2 + (G_z)^2}}$$

Con aumento en el ancho de la ventana a 26 vecinos, se mejora la calidad de la imagen realizando un promedio entre más vecinos, sin embargo, con superficies delgadas se puede perder información en la imagen resultante.

Características:

- Alto rango dinámico de los niveles de gris.
- Ofrece una impresión más realista que los anteriores algoritmos.
- En superficies delgadas el error se incrementa.
- El umbral obtenido no es representativo de la superficie ni de su inclinación (por la acción de promediación llevada a cabo por la ventana).

3.6.4. Sombreado por gradiente de niveles de gris adaptivo (AGG) [35].

En algunas superficies una ventana con seis vecinos puede resultar muy grande y causar problemas a la representación. Este algoritmo se adapta al grosor del objeto utilizando solamente 3 de 6 voxeles adyacentes.

Se obtiene G_x (G_y y G_z) de la siguiente forma, si el nivel de gris en la superficie del voxel (i,j,k) es mayor (o menor) que el nivel de gris de sus dos vecinos $(i-1,j,k)$ e $(i+1,j,k)$, se realiza el gradiente entre el voxel (i,j,k) y el vecino menor (mayor); de otra manera este toma el gradiente entre ambos vecinos. Esto se realiza para mantener las altas frecuencias de la imagen.

Características:

- Para objetos sólidos se presentan los mismos resultados que el algoritmo no adaptivo.
- La calidad en los objetos delgados mejora significativamente.

3.7. RECONSTRUCCION A PARTIR DE CORTES EN 2D.

En forma general, los algoritmos de reconstrucción en 3D consisten en obtener una malla formada por polígonos donde cada vértice del polígono contiene información representativa de su origen (imagen en dos dimensiones) y sus vecindades. Esto significa que solo se extraen algunos datos del conjunto que conforma a un volumen o a la superficie de un volumen. Esta reducción de información permite ocupar menor espacio en memoria de disco y una mejor interacción usuario-computadora. Como siempre, la calidad de la representación va de la mano con el tiempo de cálculo, por lo tanto, mientras mayor reducción se obtenga de los contornos originales, la imagen resultante perderá mas detalle. Sin embargo, simplemente al conectar toda la densidad de datos que representan el contorno podemos obtener una representación de superficie del objeto, lo cual podemos también lograr con un número menor de datos cuidadosamente seleccionados.

En este capítulo se trata un método de reconstrucción en 3D para representación de superficie de estructuras biológicas a partir de datos discretos en dos dimensiones desarrollado por Olalekan Odesanya y cols. (1993)[27]. Este método fué como un paso inicial en la evolución de modelos matemáticos más completos para modelar estructuras biológicas. Basado en el algoritmo para reducción de contornos desarrollado por Lozober y Preiss (1983)[28], O. Odesanya y cols. obtienen una mejora en el sentido de que logran una mayor reducción de información, con la misma exactitud. En este sentido es un algoritmo muy práctico para el rápido despliegue de la imagen reconstruida.

La tarea del algoritmo es identificar una colección mínima de datos o puntos del contorno, de manera que la información faltante se aproxime por algún método de interpolación. Este procedimiento incluye un error de aproximación que se obtiene con el cálculo de la diferencia espacial entre la información faltante (contorno no reducido) y los datos aproximados para cada contorno (curva interpolada).

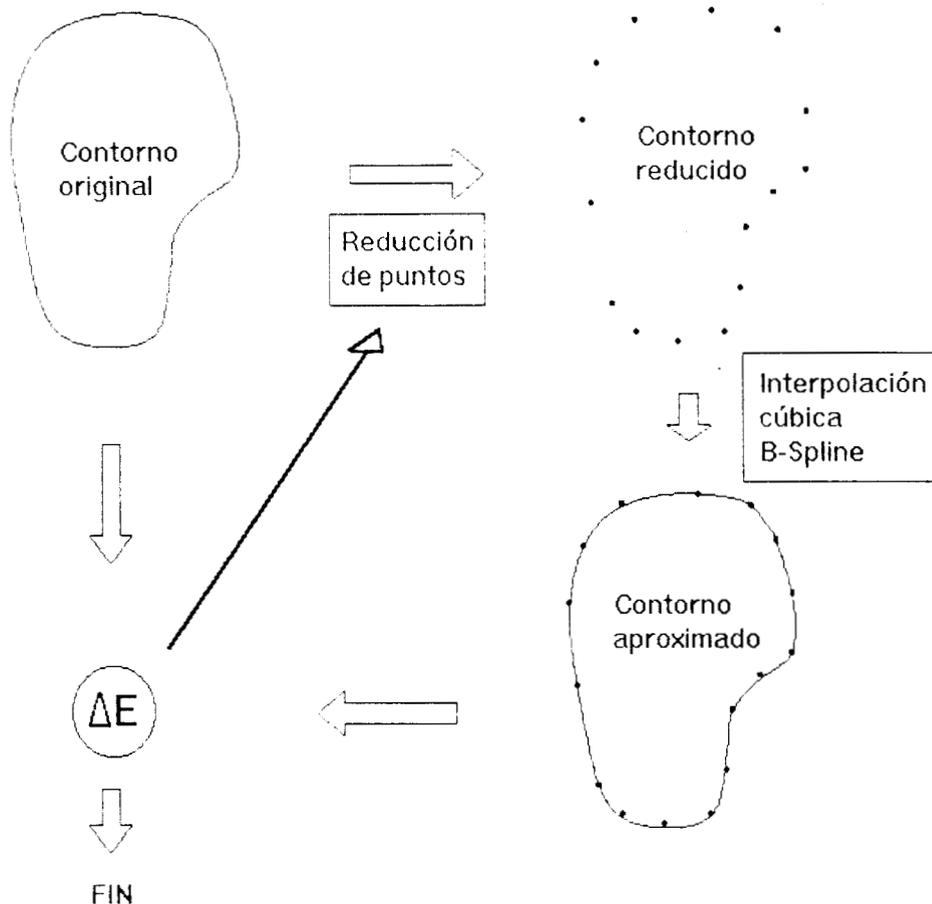


figura 3.7.1. Algoritmo iterativo de reconstrucción a partir de contornos desarrollado por Odesanya. La iteración termina cuando se encontró el error deseado.

El algoritmo en forma general consiste de las siguientes etapas (fig. 3.7.1):

1. Se utilizan los puntos del contorno de la imagen segmentada como puntos de base y se aplica una máscara con ciertos pixeles de libertad (en un dato inicial cualquiera) que trata de localizar a los candidatos a puntos de control. Estos puntos de control formarán un nuevo contorno reducido representativo del contorno base.
2. Se define una nueva curva cerrada con una interpolación de orden tres a través de los puntos de control representativos del contorno base.
3. Se estima un error de aproximación entre el contorno base y la curva aproximada.
4. Se repiten los pasos 1-3 utilizando como dato inicial el subsecuente de la anterior iteración. La iteración termina cuando se encuentra un mínimo error total o algún valor deseado por el usuario, o cuando el dato inicial regresó a la primera iteración.

1) En la reducción de información se aplica una máscara rectangular inicial con tamaño de 3 pixeles por tantos pixeles como se deseen de ancho (A en figura 3.7.3), de forma que su línea central coincida con el primer y tercer punto en el segmento seleccionado (fig. 3.7.2). La ventana inicia con p_0 y si todos los puntos intermedios se encuentran adentro de la máscara se aumenta un pixel al largo de la misma ($n = n+1$), en caso contrario se

reposiciona la máscara en el último punto que incluye la máscara ($p_0' = p_n$, $p_n' = p_n + 2$ puntos) para iniciar una nueva reducción (fig. 3.7.3). Esto se repite para todos los puntos del contorno.

El ancho de la ventana (A) se mantiene constante según la tasa de reducción deseada. En caso de desear mayor reducción de información el ancho de la ventana se amplía y viceversa si se requiere menor reducción. Se reinicia el proceso tomando el segundo dato del contorno base como valor inicial (p_0) y así sucesivamente se reinicia el proceso con cada dato siguiente hasta que se encuentra el error deseado.

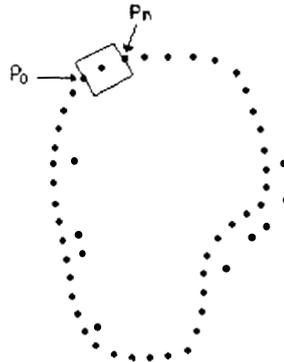


fig. 3.7.2. Máscara inicial que incluye tres puntos del contorno base.

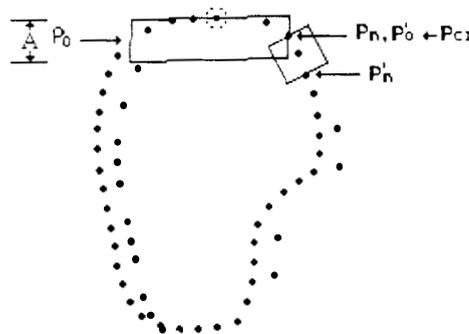


figura 3.7.3. Aplicación de la máscara para reducción de puntos. La máscara se amplía en un punto hasta que alguno de los puntos intermedios cae fuera de la máscara (punto adentro del círculo). Se seleccionan los puntos de control en el punto final de la máscara antes de ampliarla.

Entonces, el contorno reducido consiste del conjunto de puntos de control ($p_{c1}, p_{c2}, \dots, p_{cm}$) donde m es el número de puntos de control del contorno reducido y p_{ci} es el punto donde se reposiciona la ventana.

2) Ya seleccionados los puntos de control del contorno reducido, se aplica el algoritmo de interpolación B-Spline de orden cúbico, desarrollado por Yamaguchi[38]. Este algoritmo B-Spline uniforme no-racional, tiene la particularidad de que la curva aproximada pasa a través de los puntos de control.

Como puede revisarse (ver sección 3.3.4), este tipo de curvas B-Spline se aproximan al punto de control, más no pasan exactamente por éste. Barsky crea una solución bastante ingeniosa para el éxito de su algoritmo. En general, el algoritmo parte en sentido inverso para encontrar unos nuevos puntos de control temporales que proporcionarán una curva que si atraviere los puntos del contorno reducido.

3) El error es calculado con la diferencia cuadrática media entre cada punto aproximado obtenido por interpolación entre los puntos de control del contorno reducido y el punto más cercano o semejante en el contorno base. Obsérvese que la magnitud de este error también dependerá del número de puntos intermedios que aproximarán un segmento de curva entre dos puntos de control.

4) La curva óptima se determina cuando se obtuvo la diferencia deseada entre el contorno base y el contorno reducido. Esto significa que se analizan todas las posibles reducciones en un contorno, si lo que se quiere es obtener el error óptimo.

Finalizado el proceso de reducción, ahora se procede a crear la malla que dará forma al objeto. Para esto es necesario igualar cada contorno a un mismo tamaño para que cada punto en el contorno n identifique un punto en el contorno $n+1$ con el cual formará el polígono.

Para igualar al mismo número de puntos cada contorno, se considera que existe un segmento de arco entre puntos que se subdivide según sea necesario. Todos los contornos se igualan al que obtuvo mayor cantidad de datos. La subdivisión de cada segmento del contorno se realiza mediante la siguiente ecuación,

$$NS = \frac{LA * MD}{LAT} \dots (1)$$

donde NS es el número de subdivisiones del segmento.

LA es la longitud de arco del segmento.

MD es el número de divisiones del contorno mayor, y

LAT es la longitud de arco de todo el contorno.

CAPITULO 4.

METODO Y DESARROLLO

En esta sección se describen las implementaciones realizadas para la obtención de un modelo de superficie formado por polígonos de cuatro vértices (facetas) y su consecuente representación de superficie por los algoritmos de sombreado de Gouraud y Phong (sección 3.5). Por otro lado también se describirá la representación de superficie con el algoritmo de sombreado por gradientes de niveles de gris (sección 3.6), que trabaja sobre los contornos originales producto de la segmentación.

Para iniciar este estudio primero se probaron los algoritmos de Gouraud y Phong en un cuerpo sintético (cubo) y a continuación se obtuvieron resultados en imágenes médicas, específicamente en representación tridimensional de superficie de cerebro y piel.

El trabajo se desarrolló en una computadora personal de fácil acceso, bajo costo y comercialmente difundida con capacidad de despliegue gráfico para 64 niveles de gris en 640 x 480 pixeles. Todos los programas ejecutables se realizaron en Borland C++ 3.1 con programación orientada a objetos.

Se emplearon imágenes de resonancia magnética nuclear (RMN) de 256 x 256 pixeles con 256 niveles de gris proporcionadas vía Internet por el Departamento de Ciencias de la Computación de la Universidad de Carolina del Norte (<ftp.cs.unc.edu/pub/projects/softlab/CHVRTD>). Las imágenes consisten en un conjunto de 109 cortes sagitales de cabeza humana de los cuales se desconoce su relación espacial cuantitativa en los tres ejes coordenados.

Las segmentaciones de piel se lograron fácilmente por identificación de umbrales de niveles de gris. La imagen de RMN tiene un fondo negro que limita completamente el tejido de la piel. En el software empleado, únicamente se discrimina entre los valores más bajos de niveles de gris (0 al 20) y los niveles restantes (20 al 255) que representan la imagen binaria; esto lo aplica para todo el conjunto de imágenes que forman al objeto en forma automática. A partir de la imagen binaria se implementó un algoritmo reconocedor de límites que genera como resultado un arreglo de datos representativos del contorno de la imagen. La segmentación de cerebro por umbral no es posible en forma automática, por tal razón actualmente se desarrollan otro tipo de métodos para lograr una extracción exacta y automática de esta estructura compleja.

Para la obtención de la malla de polígonos representativa de la superficie del objeto tridimensional, se implementó el algoritmo de reconstrucción a partir de cortes tomográficos descrito en la sección 3.7, con la inclusión de algunas modificaciones que se describen en el capítulo de resultados.

Para el manejo de los objetos en 3D se empleó la proyección Cavalier con opción para el manejo de una proyección central¹. Para la eliminación de superficies ocultas se emplearon el método de normales para una discriminación gruesa de las facetas no visibles y el método Z-buffer para una eliminación fina a nivel pixel (sección 3.4). Se emplearon los modelos de iluminación ambiental y difusa (sección 3.5) para los tres tipos de

¹ Cuando un objeto se despliega con proyección Cavalier y se realiza una rotación, la imagen resultante se deforma.

sombreado (Gouraud, Phong y Gradientes por niveles de gris). Para el manejo de la imagen resultante se implementaron las transformaciones descritas en la sección 3.1, a fin de observar el objeto reconstruido desde cualquier ángulo.

En las siguientes páginas únicamente se muestran las clases de los objetos manejados en el algoritmo. El desarrollo de cada una de las funciones miembro no se incluye debido a la extensión de la mismo. Se espera que el nombre de cada una de las funciones miembro indique su labor.

CLASES Y FUNCIONES MIEMBRO DEL ALGORITMO DE RECONSTRUCCION A PARTIR DE CONTORNOS

```

class objeto
/* El objeto reconstruido se compone de contornos. */
-----
uint num_contornos,mayor.num_primer_cno;
char arch_obj[90];
contorno *CONTORNO;
public:
objeto(); //Constructor.
void lee_archivos(); //Lee contornos originales.
void despliega(); //Pinta en pantalla
void encuentra_mayor(); //Mayor número de ptos. en c/contorno.
void guarda(); //Contornos reducidos.
void iguala_contornos(); //Iguala al mayor número de ptos.
void reduce(); //Aplica la máscara de reducción.
void busca_mas_cercano(); //Reacomoda el punto de inicio.
void deshecha_ptos_inutiles(uint); //Según modificación al
//algoritmo de Odesanya.
uint mayorLAT(); //Encuentra el contorno con mayor longitud de arco.
void guarda_error_reduccion(); //Error total
~objeto(); //Destructor

```

```

class mascara
/* Abre un ventana para reducción.*/
-----
int x1,x2,y1,y2,MARGEN;
float pendiente;
public:
mascara(uint ancho);
void define(int,int,int,int);
int enmascara(voxel dato);

```

```

class bicontorno
/*Se instancia esta clase para formas un
listón de polígonos con dos contornos*/
-----
contorno *C1,*C2;
uint num_datos;
voxel *dat_red1,*dat_red2,*nvo;
public:
bicontorno(contorno *,contorno *);
void deshecha_ptos_inutiles();
void guarda_nuevos();
~bicontorno();

```

```

class contorno
/*Representa al contorno original, al reducido y al la curva
interpolada.*/

```

```

FILE *fp_cno;
memoria *dat3D; //Memoria expandida.
pila *PILA;
uint num_sp,num_red,num_ptos;
double error, LAT;
public:
contorno();
void lee_archivo(char *);
void calculaLAT();
void despliega();
void despliega(int);
void despliega(uint);
void reduce_datos(mascara,uint);
void guarda_puntos(char *);
double calcula_error();
uint calcula_splines();
void reacomoda_datos(uint);
void libera_mem_exp();
void ordena_segmentos(uint);
void calcula_subdiv(uint);
void intro_nvos_segmentos(uint);
voxel primer_dato();
uint busca_mas_cercano(voxel);
voxel *datos_reducidos(uint *);
void guarda_nvos_datos(voxel *);
~contorno();

```

```

class segmento
/*Une dos voxeles*/

```

```

public:
voxel dato1;
voxel dato2;
float long_arco;
segmento();
segmento(voxel ,voxel );
void despliega();
double longitud_arco();

```

```

class memoria
/*Maneja la memoria expandida*/

```

```

union REGS in,out;
struct SREGS seg;
uint num_seg,num_pag,handler; //Handler de la
//memoria expandida.
uint *frame; //Direccion de page frame.
public:
memoria();
void guarda(voxel,uint);
void guarda(uint,uint);
voxel lee(uint);
void libera(void);
~memoria();

```

```

class nodo
/*Son los nodos de la pila*/

```

```

segmento dato;
public:
uint posicion, no_subdiv;
class nodo *sig,*ant;
nodo(uint,segmento);
uint inutil();
uint mayor(nodo);
void calcula_subdiv(float *,uint *,int *);
void intro_nvos_segmentos(memoria *,uint *,uint);

```

CLASES Y FUNCIONES MIEMBRO EN EL ALGORITMO PARA REPRESENTACION DE LOS METODOS DE GOURAUD Y PHONG.

```
class objeto
/*Conforma la representación de superficie del objeto*/
-----
uint num_fac,num_datos,num_franjas;
  memems *face;
  buffer huge *z_buffer;
public:
  objeto();
  ~objeto();
  void lee_facetas();
  void calcula_normal();
  void despliega_facetas();
  void despliega_phong();
  void despliega_gouraud();
  void rotaX(int);
  void rotaY(int);
  void rotaZ(int);
  void desplaza(uint,uint,uint);
```

```
class voxel;
/*Elemento de la escena en 3D*/
-----
public:
  float x,y,z;
  voxel(){}
  voxel(float xx,float yy,float zz);
  voxel operator = (voxel A);
  voxel operator - (voxel A);
  voxel operator + (voxel A){
  voxel operator * (float A){
  voxel operator / (float A){
  double operator * (voxel A){
  pixel proyeccion();
  void guarda(FILE *fp);
  double eqm();
  voxel normaliza();
  float ec_plano(int,int);
  float mayor();
  void rotaX(float);
  void rotaY(float);
  void rotaZ(float);
  void desplaza(int,int,int);
```

```
class buffer
/*Maneja el z-buffer para eliminación de superficies escondidas.*/
-----
union REGS in,out;
  struct SREGS seg;
  ulong num_pag;
  void *frame; //Direccion de page frame.
  uint handler; //Handler de la memoria expandida.
public:
  buffer(ulong);
  void guarda(byte,voxel);
  ~buffer();
```

```
class vertice
/*Contiene la normal y la posicion en
cada vertice de faceta*/
-----
public:
  voxel pos;
  voxel normal;
  vertice(){};
  vertice(voxel d,voxel n);
  vertice operator = (vertice A);
```

```

class faceta
/*Contiene cuatros vértices y la normal */
-----

voxel ver[4];
friend memems;
public:
voxel normal;
faceta(){}
faceta(voxel a,voxel b,voxel c,voxel d);
faceta operator =(faceta FAC);
~faceta();
void calcula_normal();
faceta cal_normal_en_ver(faceta *);
void despliega(uint);
int visible(voxel);
void interpola_phong(voxel,buffer huge *,faceta);
void interpola_gouraud(voxel,buffer huge
*,faceta);
void rotaX(float);
void rotaY(float);
void rotaZ(float);
void desplaza(uint,uint,uint);

```

```

class memoria
/*Maneja la memoria expandida*/
-----

union REGS in,out;
struct SREGS seg;
uint num_seg,num_pag;
uint *frame; //Direccion de page frame.
uint handler; //Handler de la memoria expandida.
public:
memoria(uint); //Localiza pages requeridas.
void moveCE(uint *,uint,uint); //Traspasa datos de conv-exp.
void moveEC(uint *,uint,uint); //Traspasa datos de exp-conv.
~memoria(void);

```

```

class segmento
-----

voxel dato1;
voxel dato2;
public:
float long_arco;
segmento(){};
segmento(voxel a,voxel b);
double longitud_arco();
void interpola_gouraud(byte,byte,buffer huge *);
void interpola_phong(segmento,voxel,buffer huge *);
void despliega();
voxel paso(float *);
voxel paso(float mayor);

```

```

class pila
/*Se diseña una pila doblemente ligada para
ordenamiento según longitud del segmento. Se utiliza
para igualación de tamaño de contornos.*/
-----

nodo *ppio,*fin;
public:
pila(nodo *);
void inserta(nodo *);
void calcula_subdiv(float,uint,uint);
void libera_inutiles();
nodo *pertenece(uint);
~pila();

```

```
class ventana
/*Ventana gráfica*/
```

```
int x1,x2,y1,y2;
byte huge *p; //Buffer.
char *titulo;
public:
ventana(char *nom,int xx1,int yy1,int xx2,int yy2,int
color);
FILE *Despliega(void);
char Despliega(char *Menu[],int);
char *Despliega(char *);
void Despliega(char *,int *);
~ventana();
```

```
class bicontorno
/*Compone un listón de facetas*/
```

```
contorno *CONTORNO;
uint num_datos;
public:
bicontorno(uint,char *);
faceta forma_faceta(uint);
~bicontorno();
```

```
class pixel
/*Elemento de pantalla*/
```

```
public:
unsigned x,y;
pixel(void){};
pixel(unsigned xx,unsigned yy);
```

```
class trio
/*Contiene la posicion y la intensidad
de la escena*/
```

```
public:
uint posicion;
byte intensidad;
public:trio(uint pos,byte I);
```

CLASES Y FUNCIONES MIEMBRO DE LA REPRESENTACION DE SUPERFICIE POR EL METODO DE GRADIENTES POR NIVELES DE GRIS.

```
class objeto
/*Contiene lal objeto formado por contornos*/
-----
uint num_contornos,y_mayor;
contorno *CONTORNO;
buffer *z_buffer;
public:
objeto();
~objeto();
void lee_archivo();
void calcula_normal();
void despliega();
void rotaX(int);
void rotaY(int);
void rotaZ(int);
void desplaza(int,int,int);
void mayor();
```

```
class emsvert
/*Manejo de memoria expandida para vertices*/
-----
union REGS in,out;
struct SREGS seg;
ulong num_pag;
vertice *frame; //Direccion de page frame.
uint handler; //Handler de la memoria expandida.
public:
emsvert(ulong);
void guarda(vertice,uint);
vertice *lee(uint);
~emsvert();
```

```
class contorno
/*Almacena las características de cada contorno*/
-----
uint num_ptos; //No.puntos originales
emsvert *dat3D;
public:
contorno();
void lee_archivo(char *);
void calcula_normal(char *,char *,char *);
void despliega(voxel,buffer *,uint);
void rotaX(float);
void rotaY(float);
void rotaZ(float);
void desplaza(int,int,int);
uint mayor();
~contorno();
```

Las clases faltantes se encuentran incluidas en las anteriores implementaciones.

CAPITULO 5.

RESULTADOS Y DISCUSION

Se inicia este capítulo con la presentación de los resultados obtenidos al implantar el algoritmo de reconstrucción para modelar la superficie de un objeto por medio de polígonos. Se trabajó con contornos de cerebro y piel de los cuales partimos previa segmentación de la imagen de RMN. En la figura 5.1a y 5.1b se presentan los contornos de cerebro y piel del corte No.48 del conjunto de 109 cortes sagitales de cabeza humana. A manera de ilustración se muestran los puntos reducidos sobrepuestos al contorno base.

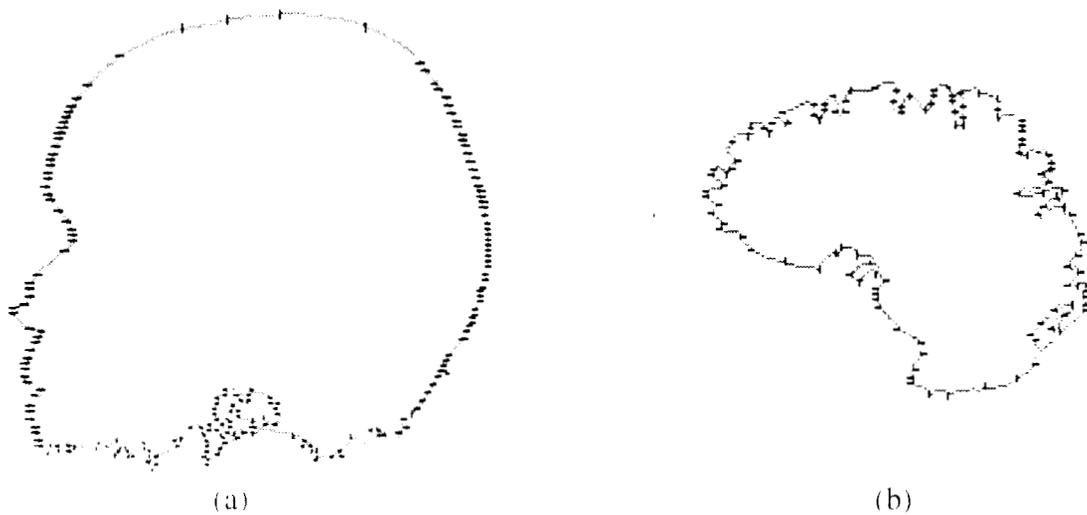


figura 5.1. Contornos originales y puntos reducidos (en marcas diagonales) para (a) piel y (b) cerebro.

En las figuras 5.2a y 5.2b se muestran las mallas de superficie para piel y cerebro respectivamente, tal como resulta de la implementación del algoritmo de Odesanya descrito en la sección 3.7, con eliminación de superficies escondidas por método de normales para observar más claramente las facetas más cercanas al observador.

Se puede observar un efecto de torcimiento entre facetas producto de la independencia en la búsqueda de los puntos óptimos representativos del contorno, con respecto a los restantes que conforman al objeto. La malla se obtuvo con un ancho de máscara de tres pixeles (el pixel medio ± 1 pixel en el eje y), dos puntos de unión¹ como aproximación a la curva y con las características reportadas en las tablas 5.1 y 5.2. En algunas zonas se aprecian las facetas aproximadamente rectangulares, en otras triangulares, torcidas, y en las manchas negras no se aprecian los polígonos debido a su amontonamiento sin orden, mayormente acentuado en la zona de la nariz, en el caso de la cara y en diversas zonas en el cerebro.

¹ Son los puntos entre dos puntos de control con los que se aproximará la curva B-Spline.

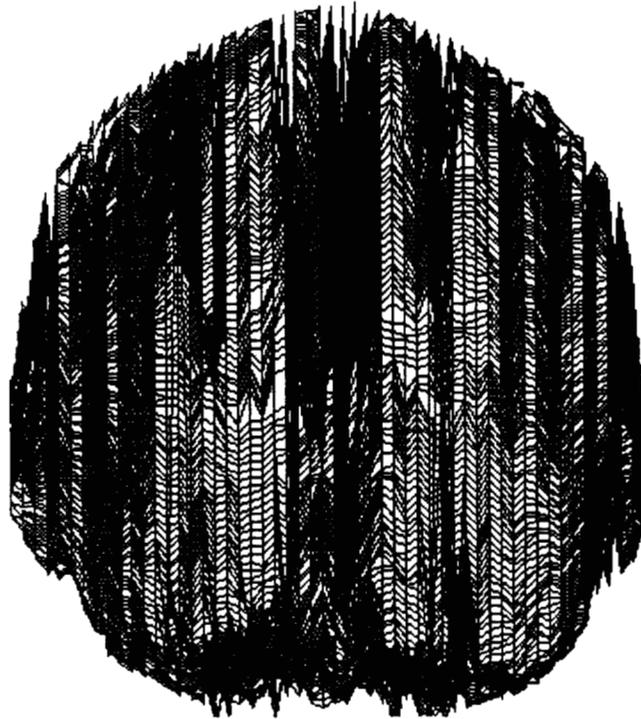


figura 5.2a. Modelo de superficie de Odesanya en piel formado por polígonos con 54 contornos, cada uno con 300 puntos de control.



figura 5.2b. Modelo de superficie de Odesanya en cerebro formado por polígonos con 37 contornos, cada uno con 184 puntos de control.

El algoritmo de Odesanya reportó resultados óptimos en representación de dedos de la mano, donde la representación de superficie del modelo producto se aproxima fácilmente con una curva de interpolación de orden cúbico, con un mínimo de error [27]. En nuestro

caso pretendemos tratar superficies más complicadas que el dedo como lo son la piel y aún más, el cerebro.

A partir de los contornos reducidos (después de aplicar la máscara) se realizó una rectangulación uniendo punto a punto consecuente de cada contorno. Esto funciona bien cuando las variaciones de un contorno al siguiente no son importantes. En el caso de cerebro, algunos contornos siguen fielmente las circonvoluciones cerebrales mientras que en el contorno siguiente estas no se presentan. En el caso de la piel existen problemas con los contornos que contienen la nariz y los ojos con respecto a los que no. Estos cambios bruscos de un contorno a otro provoca un desfase entre contornos contiguos, y por consiguiente, cada faceta se presenta con cierto grado de torcimiento.

Proponemos entonces, una modificación al algoritmo de Odesanya que consiste en asociar siempre un punto del contorno i al punto más cercano en el contorno $(i+1)$. Esto crea una malla de elementos rectangulares o triangulares si un punto resulta ser el más cercano a varios otros en el contorno siguiente (fig. 5.3). Esta nueva estructura aparece mucho más flexible a la superficie del objeto que si fuera únicamente por polígonos rectangulares, con la inconveniencia de que algunos puntos de la reducción inicial se eliminarán a fin de mantener mismo número de puntos entre contornos. A fin de que la nueva pérdida de información no sea grave (perder los puntos más representativos de las estructura), se inicia comparando con el contorno que contiene la mayor longitud de arco total, el cual contiene el mayor ancho de banda de frecuencia espacial del objeto.

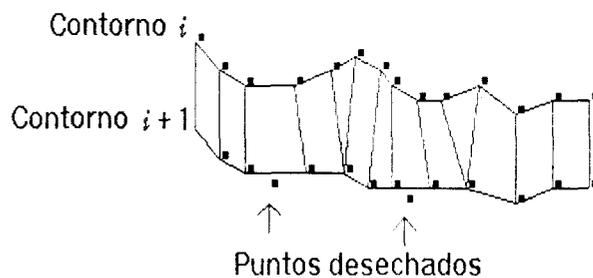


figura 5.3. Modificación al algoritmo de Odesanya para la formación de facetas entre contornos. Las facetas pueden ser rectangulares o triangulares. El contorno i representa el contorno con mayor longitud de arco total.

En las figuras 5.4a y 5.4b se aprecia la disminución considerable de traslapamientos y torcimientos entre facetas logrado con la modificación, que al igual que en los casos a y b, conservan 300 (en piel) y 184 (en cerebro) puntos de control por cada contorno.

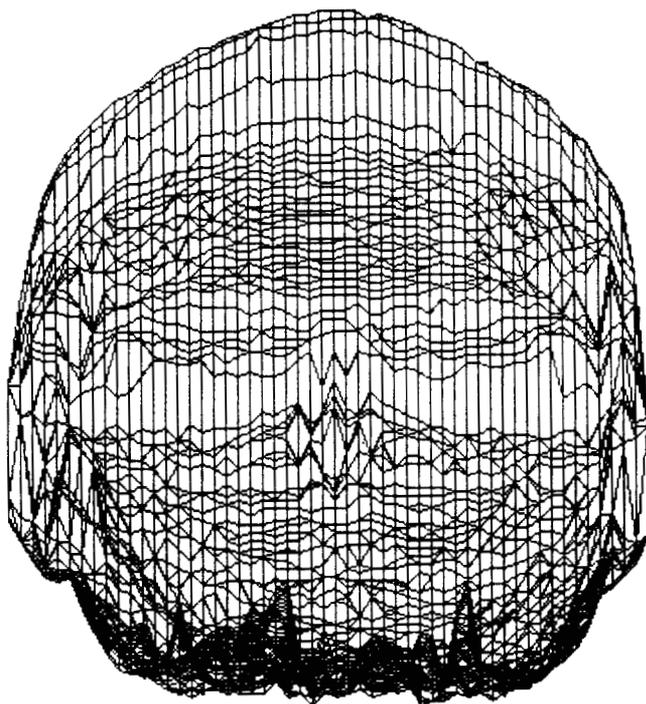


figura 5.4a. Modelo de superficie modificada de piel formado por polígonos con 54 contornos. Se obtienen 300 puntos de control en cada contorno reducido.

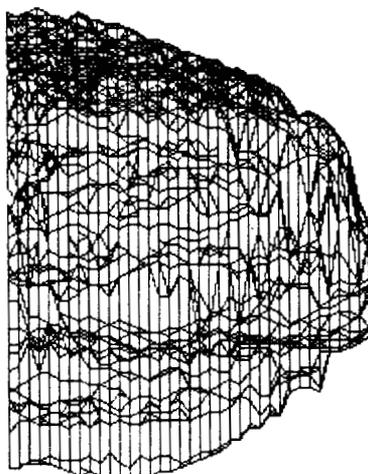


figura 5.4b. Modelo de superficie modificada de cerebro formado por polígonos con 54 contornos. Se obtienen 184 puntos de control en cada contorno reducido.

En la tabla 5.1 se muestran los resultados de la implementación del algoritmo de reconstrucción llevada a cabo con las segmentaciones de 37 cortes tomográficos (no se dispone de mayor número de segmentaciones) representativos del cerebro. La tabla incluye el número de iteración a la cual logra el error óptimo (mínimo) de entre 10 que realizó, el número de puntos del contorno original, número de puntos de control del contorno reducido, errores del modelo antes y después de la modificación señalada anteriormente (fig.

5.3), y por último, el porcentaje de reducción² obtenido con número de puntos iguales en cada contorno. Se presenta la misma información en la tabla 5.2 para el caso de la piel con 54 contornos representativos de la cara completa. Desafortunadamente se desconoce la información cuantitativa en milímetros de los errores reportados.

Debe tomarse en cuenta que la magnitud de estos errores es dependiente de la cantidad de puntos de unión entre vértices de control. Este valor es difícil de definir ya que se desconoce la cantidad de información perdida en el proceso de reducción entre lo que se definió como dos vértices de control. En nuestro caso únicamente empleamos un punto de unión debido a que el tiempo del proceso se incrementa considerablemente.

No. de contorno	No. de iteración con mínimo error	No. de puntos originales	No. de puntos reducidos	Error según algoritmo de Odesanya	Error final según modificación	Porcentaje de reducción
0	4	649	158	10.644582	13.771641	71.64
1	0	710	168	10.784883	14.119134	74.08
2	0	687	161	11.062171	14.776321	73.21
3	0	612	142	10.652947	13.946950	69.93
4	8	641	144	11.006459	12.597956	71.29
5	1	671	167	10.250418	13.617548	72.57
6	0	643	153	10.407387	12.184004	71.38
7	5	730	184	9.604303	9.604303	74.79
8	0	698	177	10.987327	11.411484	73.63
9	1	629	161	10.688435	12.021086	70.74
10	1	501	142	11.006459	11.762930	69.38
11	1	652	161	10.395724	12.756313	71.77
12	0	598	155	10.719118	12.942209	69.23
13	0	554	137	10.605598	12.633626	66.78
14	0	580	139	11.231302	11.767560	68.27
15	0	573	157	10.683425	12.673456	67.88
16	0	551	141	10.876908	12.192897	66.60
17	0	553	141	10.550220	11.897723	66.72
18	0	576	136	10.922781	12.268372	68.05
19	0	694	177	10.949863	13.209248	73.48
20	4	654	164	10.730430	13.498440	71.86
21	4	585	143	10.299081	12.133534	68.54
22	3	558	131	10.664021	11.762457	67.02
23	0	498	133	10.253884	12.115423	63.05
24	0	437	104	10.839063	12.868509	57.89
25	0	307	99	10.649611	13.163397	40.06
26	0	425	101	10.463521	11.518659	56.70
27	6	480	129	10.547163	12.808739	61.66
28	2	361	108	10.032789	12.047845	49.03
29	0	341	99	10.968761	12.607442	46.04
30	4	303	80	10.451920	11.893613	39.27
31	5	291	82	9.768020	12.649977	36.76
32	2	251	64	10.169278	12.427764	26.69
33	1	240	59	10.762848	11.500663	23.33
34	5	502	118	10.806174	11.150075	63.34
35	7	396	102	11.115720	10.731660	53.53
36	0	262	65	10.340177	9.369410	29.77

tabla 5.1. Errores y resultados al aplicar el algoritmo de reconstrucción a 37 contornos de cerebro, con un punto de unión. Se realizaron 10 iteraciones por contorno.

² Los porcentajes negativos representan incremento del número de puntos originales.

No. de contorno	No. de iteración con mínimo error	No. de puntos originales	No. de puntos reducidos	Error según algoritmo de Odesanya	Error final según modificación	Porcentaje de reducción
0	0	147	44	11.782548	11.312117	-104.08
1	0	315	85	11.072739	8.547513	4.76
2	6	335	93	10.927681	10.361833	10.44
3	2	431	126	11.466703	12.157637	30.39
4	1	369	98	12.199526	9.772668	18.69
5	9	415	108	11.194361	10.479835	27.71
6	2	453	109	11.264968	9.552649	33.77
7	5	513	115	11.578906	11.199730	41.52
8	1	563	119	11.272580	11.797163	46.71
9	6	597	137	11.951085	12.020752	49.74
10	7	685	163	19.505950	11.246275	56.20
11	6	713	172	10.649611	11.960569	57.92
12	2	675	158	11.154112	11.810389	55.55
13	6	757	171	11.747340	12.311358	60.36
14	3	837	213	16.557624	12.380416	64.15
15	1	801	189	11.593714	12.511351	62.54
16	2	889	222	11.217327	12.682312	66.25
17	1	1023	261	10.555293	12.630722	70.67
18	1	983	235	11.217327	12.845167	69.48
19	0	1003	253	10.743101	12.783879	70.08
20	6	1109	276	11.224972	12.286473	72.94
21	6	1093	274	10.900845	12.256420	72.55
22	7	1131	271	10.900845	13.419829	73.47
23	1	1205	300	10.555293	10.555293	75.10
24	2	935	210	10.954451	12.161394	67.91
25	1	1011	228	11.217327	12.880710	70.32
26	2	1011	226	11.419904	12.342830	70.32
27	1	905	209	10.723805	11.725282	66.85
28	1	1041	233	9.848858	11.664669	71.18
29	2	1093	240	10.973341	11.966897	72.55
30	0	1009	244	11.235775	12.607463	70.26
31	1	1105	258	11.018812	12.146855	72.85
32	6	1109	256	10.954451	12.707161	72.94
33	6	1065	262	10.862780	13.764057	71.83
34	1	963	230	11.269428	12.811540	68.84
35	7	999	266	10.160839	12.993802	69.96
36	0	881	210	15.580043	12.173149	65.94
37	1	1107	293	11.368493	13.331217	72.89
38	0	937	228	10.827864	13.551579	67.98
39	6	895	209	10.762362	12.423102	66.48
40	6	759	180	11.463604	11.817799	60.47
41	0	785	184	10.649611	12.942484	61.78
42	0	679	152	11.789826	12.852302	55.81
43	1	641	146	10.762362	11.305712	53.19
44	1	593	136	11.135529	11.712474	49.40
45	2	579	133	11.045361	12.254215	48.18
46	0	553	122	11.543078	12.257626	45.75
47	0	547	121	11.757663	11.686016	45.15
48	2	541	129	11.459219	12.688416	44.54
49	0	567	136	11.817650	11.752928	47.08
50	0	465	123	11.466417	10.703849	35.48
51	5	371	111	11.629387	12.764855	19.13
52	2	265	83	12.010106	11.785153	-13.20
53	2	123	36	12.158471	8.395160	-143.90

tabla 5.2. Errores y resultados al aplicar el algoritmo de reconstrucción a 54 contornos de piel, con un punto de unión. Se realizaron 20 iteraciones por contorno.

Con respecto a los métodos de sombreado, se utilizó un objeto geométrico de prueba e imágenes médicas reales. En las figuras 5.5a y 5.5b se muestran la aplicación de los sombreados de Gouraud y de Phong a un cubo (seis facetas). En el método de Gouraud claramente se observan las bandas conocidas como Mach que en el método de Phong se ven mucho más atenuadas. En general, ambos métodos proporcionan al observador una impresión de profundidad con una distribución de la iluminación según la Ley de Lambert (sección 3.5), sin embargo, puede apreciarse una mayor uniformidad en el sombreado de Phong debido a una mayor resolución en la intensidad que asocia a cada pixel en el objeto.

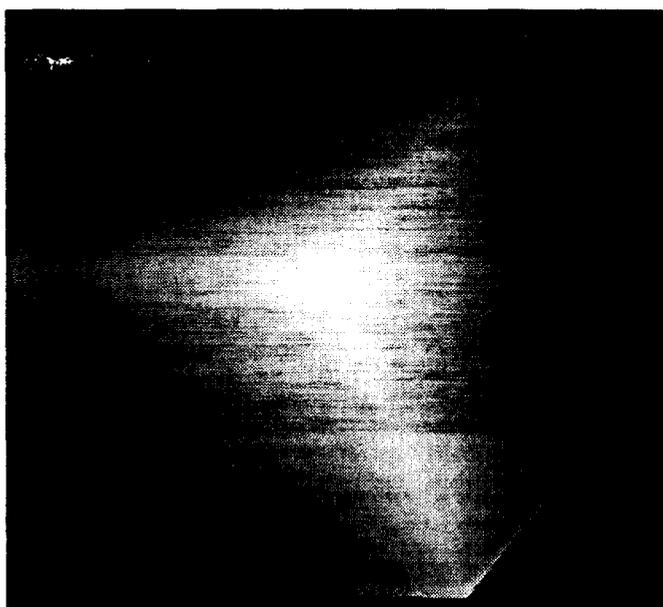


figura 5.5a. Método de Gouraud aplicado a un cubo.

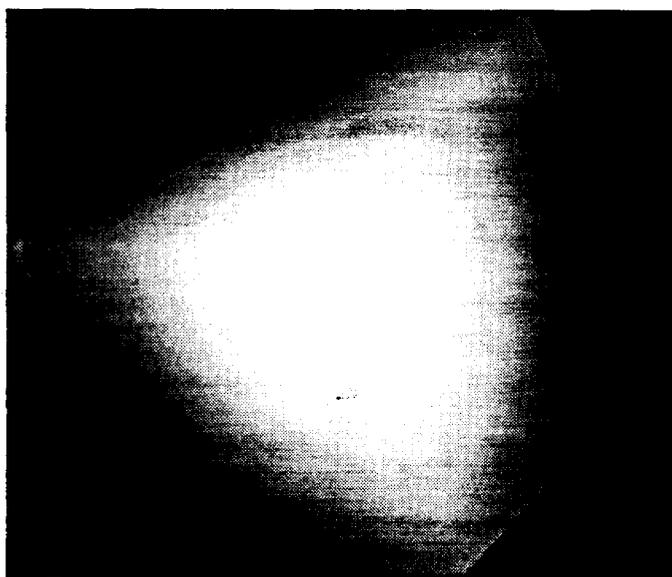


figura 5.5b. Método de Phong aplicado a un cubo.

En la figuras 5.6a y 5.6b se observa el sombreado de Gouraud para la mitad de la cabeza (54 contornos) y para el total de la cabeza (con 54 contornos espaciados al doble), respectivamente. El mismo procedimiento para sombreado de Phong se muestra en la figura 5.7a y 5.7b.

Una vez más se observa una clara diferencia entre los dos tipos de sombreado e indudablemente ofrece una mejor representación visual el método de Phong. Por otro lado, el objeto reconstruido pierde detalle conforme menor es la información que se representa, sin embargo, el algoritmo de Phong siempre trata de 'suavizar' la información que representa, por lo que visualmente puede llegar a 'verse mejor' una representación con menor cantidad de facetas o con menor distancia entre los contornos (comparaciones entre casos a y b en figuras 5.6 y 5.7), y no necesariamente desplegar información real.

Otro punto a tomar en cuenta en este tipo de representación es la dependencia total del método de sombreado con el algoritmo que construye las facetas. Siempre que se utilice un modelo de superficie integrado por polígonos lleva inherente una pérdida de información que resultará en óptima o no, dependiendo de su capacidad para encontrar los puntos de alta frecuencia espacial en el contorno.

Por otro lado, no se descartan todos los errores que pueden producirse por una mala aproximación del vector normal. Primeramente un error en la aproximación inicial de la normal a la faceta y posteriormente el error introducido por la aproximación a la normal de cada vértice por el promedio sobre tres vecinos, comentados en la sección 3.5.

Se aprecia en la figura 5.7a y 5.7b que es más suave la representación cuando el espacio es mayor entre contornos y probablemente, sea mas atractivo para el observador este tipo de representación.



figura 5.6a. Sombreado de Gouraud con 54 contornos espaciados doblemente. La imagen de la derecha se encuentra rotada 20 grados con respecto a la izquierda.



figura 5.6b. Sombreado de Gouraud para la mitad de la cabeza con 54 contornos.

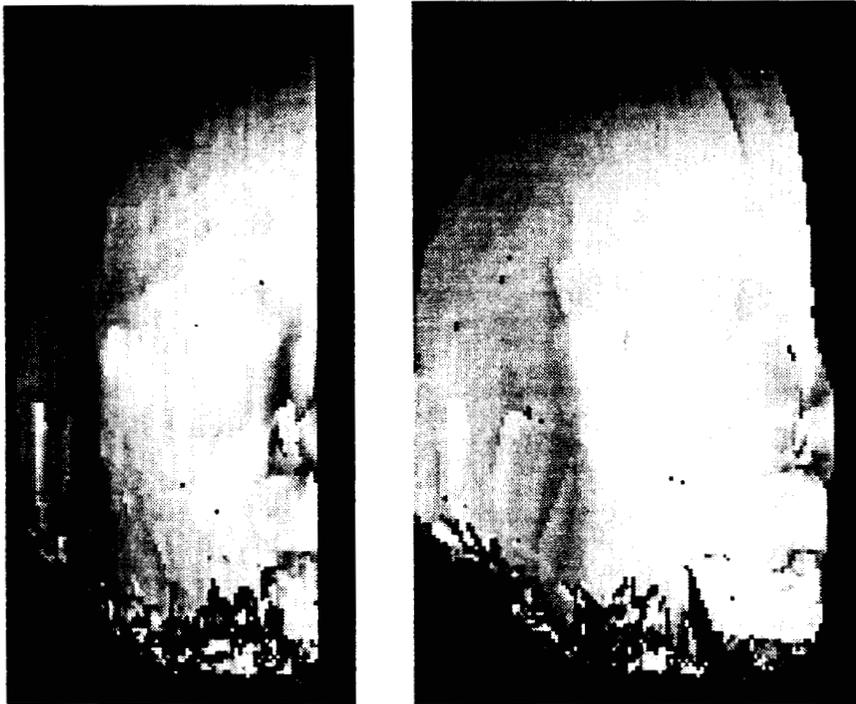


figura 5.7a. Sombreado de Phong para la mitad de la cabeza con 54 contornos.



figura 5.7b. Sombreado de Phong con 54 contornos espaciados doblemente. La imagen inferior se encuentra rotada 20 grados con respecto a la imagen superior.

En las figuras 5.8a y 5.8b se presenta el resultado de la implementación del método de sombreado por gradientes de niveles de gris, para 60 y 54 contornos, respectivamente. Existe una diferencia notoria con respecto a la anterior representación (Gouraud y Phong).

Iniciando con que no existe reducción de información (no hay pérdida), la aproximación a la normal es más exacta y como resultado la imagen aparece más real debido a que no se pierden los componentes de alta frecuencia espacial. Realizando una comparación entre los casos a y b, es notable en los extremos (mayormente) de la imagen la baja resolución espacial de la representación del caso b (líneas con un mismo nivel de gris), sin embargo, para fines no tan detallados se considera aceptable este tipo de representación con menor número de contornos.

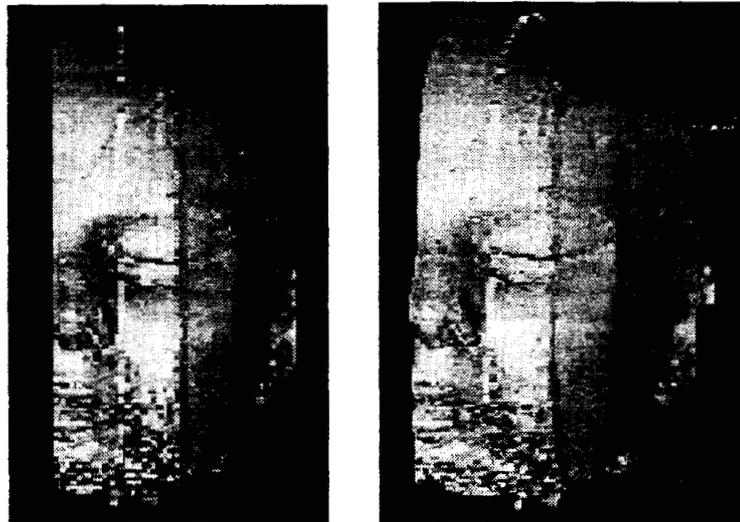


figura 5.8a. Sombreado por gradientes de niveles de gris en 60 contornos. La normal a la superficie se calcula con seis vecinos.

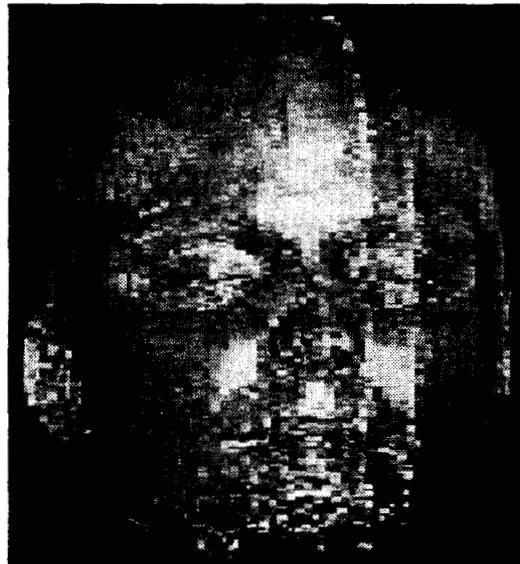


figura 5.8b. Sombreado por gradientes de niveles de gris con 54 contornos. La normal a la superficie se calcula con seis vecinos.

MEMORIA DE GRADUACIÓN

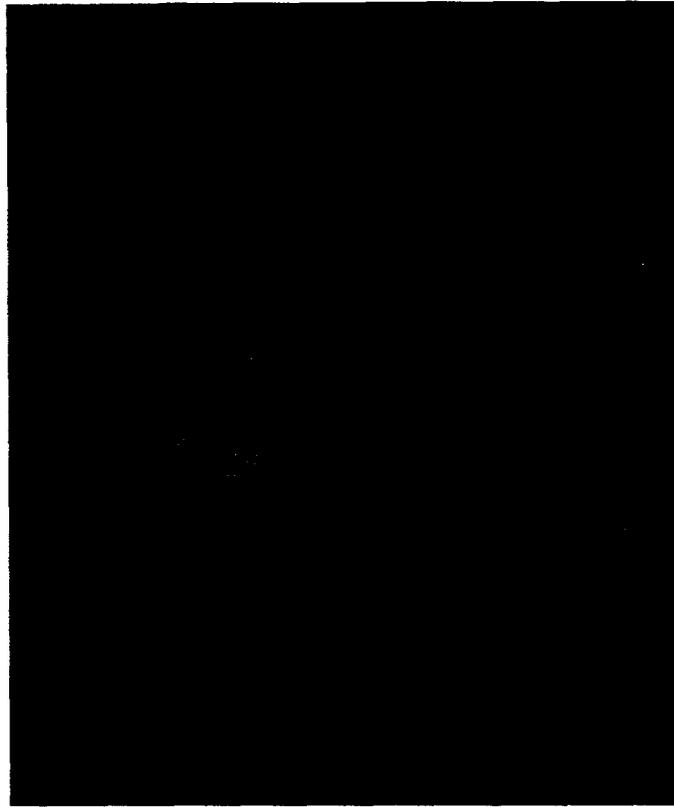


figura 5.9. Representación de superficie mediante sombreado por niveles de gris con 60 contornos. La normal a la superficie se calcula con 10 vecinos.

En la figura 5.10 se muestra la imagen de la cara representada por una paquete comercial desarrollado en el Departamento de radiología en la Universidad de Pennsylvania. El objeto reconstruido se conforma de 109 contornos y se emplean 8 vecinos para el cálculo de normales. Este software corre en una estación de trabajo con monitor de alta resolución con capacidad de manejo de 256 niveles de gris.



figura 5.10. Representación de la cara según paquete comercial con 109 contornos. La normal a la superficie se calcula con 8 vecinos.

CAPITULO 6.

CONCLUSIONES

La reconstrucción tridimensional de imágenes médicas en sus representaciones de superficie y volumen, ha continuado con la visualización no invasiva del cuerpo humano que se inició con el nacimiento de la radiología. Se han reportado ya grandes avances con resultados bastante reales en representación de estructuras anatómicas, se han diseñado algoritmos de diferentes complejidades para diferentes aplicaciones, todos persiguiendo la meta de modelar cada vez con mayor exactitud, el cuerpo humano.

Según los análisis realizados se llegó a comprobar que un mal modelo de superficie produce resultados probablemente agradables al observador, más no siempre reales y exactos. En el caso del modelo de superficie formado por polígonos se considera como simple para el modelo de estructuras complejas. No se recomienda su empleo en el caso de representación de superficies como la del cerebro, ya que puede distorsionar la imagen resultante. De las imágenes reconstruidas obtenidas por gradientes en niveles de gris, se concluyen dos cosas, primero si se requiere una aproximación más exacta se debe evitar la reducción de información en el contorno y segundo, el vector normal es más exacto si se calcula por gradientes en los niveles de gris de la imagen en 2D original, producto (en nuestro caso) del sistema RMN.

Se ha comprobado la dificultad de los algoritmos de sombreado de Gouraud y Phong para aproximarse a la realidad por su inherente característica de suavizamiento dentro de una faceta, que por otro lado, se encuentra limitado grandemente por el promedio entre las facetas vecinas para obtener la normal en el vértice. A diferencia de este, se notó la facultad del algoritmo de sombreado por niveles de gris para representar superficies con un alto grado de exactitud (cualitativamente analizada) debido a una buena aproximación a la normal a la superficie[35]. Desafortunadamente no es posible obtener una representación en tiempo real en una PC, pero es posible reducir el tiempo de cálculo disminuyendo el número de contornos que conforman al objeto. Como siempre, el costo computacional va de la mano con la calidad de la representación gráfica.

Con respecto al sombreado de Phong y Gouraud, esta visto que este tipo de representación no es óptima ni en calidad, ni en tiempo de ejecución. El sombreado por gradientes puede obtener una imagen en el mismo tiempo con mayor exactitud. Sin embargo, en este análisis se toma en cuenta que este método es altamente dependiente del algoritmo que modela al objeto con polígonos. En dicho algoritmo, si se desea un mínimo de error se debe esperar a probar todos los casos posibles con cada iteración en cada contorno (sección 3.7), que a reserva del resultado el proceso puede llegar a ser muy lento, pero que logra un objeto reconstruido para futuros manejos. Según los resultados se observa que el error obtenido no mejora lejos de las primeras iteraciones debido a la sencillez de la curva que describe el contorno de la piel. Otro punto que debe tomarse en cuenta en este método, es que el error encontrado depende a su vez de la calidad de aproximación de la curva de interpolación empleada. Como se describió en la sección 3.3 si la estructura es de mayor complejidad que alguna otra, y se utiliza una curva de orden bajo, la aproximación es pobre. Si por el contrario se encuentra un orden de óptima

interpolación para la curva compleja, en la estructura sencilla puede causar ondulaciones indeseables contribuyendo con un error adicional.

Cabe remarcar que la calidad de la representación depende grandemente de la precisión del vector normal a la superficie, la distribución de la luz que refleja la superficie del objeto y por su puesto , la segmentación, que es la base de una buena representación tridimensional de superficie.

BIBLIOGRAFIA

1. Allen, B.A. and C Levinthal, "Semi-automated nerve tracing: effective three-dimensional reconstruction from serial section micrographs", Three-Dimensional Neuroimaging. (New York: Raven Press, 1990).
2. Apodaca Anthony, Mantle M. W., "Renderman: Pursuing the future of graphics", IEEE Comput. Graphics. App., julio 1990.
3. Barillot Christian, "Surface and Volume rendering techniques to display 3D data". IEEE Eng. Med. & Biol., marzo 1993.
4. Barsky Brian A., Greenberg Donald P., "Determining a set of B-Spline control vertices to generate an interpolating surface", Comput. Graphics Image Procces., 14:203-226, 1980.
5. Chen Lih-Shyang, Herman G. T., Reynolds R. A., "Surface shading in the cuberille environment", IEEE CG&A, diciembre 1988.
6. Chen Lih-Shyang, Sontag M. R., "Representation, Display, and Manipulation of 3D Digital Scenes and their Medical Applications", Comput. Vision Graphics Image Procces., 48:190-216, 1989.
7. Dealere D. Smets C, Suetens P, Marchal G, Van de werf F, "A knowledge-based system for the 3D reconstruction of blood vessels from two angiographic projections", J Med Biomed Eng Comp 1991; 29:27-36.
8. Delckeer A, Diener HC, "Quantification of atherosclerotic plaques in carotid arteries by three-dimensional ultrasound", The British Journal of Radiology, 1994, 67, 672-678.
9. Fishman EK, Magid D, Ney DR, Kuhlman JE, "Three-dimensional CT imaging in orthopedics: state of the art 1988". Orthopedics 1988; 2:1021-1026
10. Foley van Dam, Feiner, Hughes, "Computer graphics: Principles and Practice", 2a. Ed., Addison Wesley, USA, 1992.
11. Geiger, B., "Three-dimensional modeling of human organs and its application to diagnosis and surgical planning", Technical Report 2105, Institut National de Recherche en Informatique et Automatique, (France), Dec 1993.
12. Gordon Dan, Reynolds R. A., "Image space shading of 3D objects", Comput. Vision Graphics Image Procces., 29:361-376, 1985.

13. Gouraud H., "Continuous Shading of curved surfaces", IEEE Trans. Comput., Vol. c-20(6):623-629, 1971.
14. Herman Gabor T., Liu H. K., "Display of three dimensional information in computed tomography", Journal of Computed Assisted Tomography, 1(1), 1977.
15. Herman Gabor T., Udupa J. K., "Display of 3D digital images: Computational foundations and medical applications", IEEE CG&A, agosto 1983.
16. Hoehne K. H., Bernstein R., "Shading 3D-Images from CT using gray level gradients", IEEE Trans. Medical Imag., MI-5(1):45-47, 1986.
17. Hoehne K.H., Riemer M., Tiede U., "Viewing operations for 3D tomographic gray level data", International Symposium CAR'87 Computer Assisted Radiology (Berlin), Springer-Verlag, 1987.
18. Johnson Ellen, Capowski J. J., "A system for the three dimensional reconstruction of biological structures", Comput. and Biomedical Research., vol. 16, 1983.
19. Levoy Marc, "Display of surfaces from volume data". IEEE CG&A, mayo 1988.
20. Levoy Marc, "A hybrid ray tracer for rendering polygon and volume data", IEEE CG&A, marzo 1990.
21. Levoy Marc, "Display of surfaces from volume data". IEEE CG&A, mayo 1988.
22. Magid D, Fishman EK, "Imaging musculoskeletal trauma in three dimensions", Radiol Clin North Am 1989; 27:945-956.
23. Matthew Wong Y., Thompson, Loren, Fitzharris. "Computer reconstruction of serial sections", Comput. Biomedical Research, vol. 16, 1983.
24. Montgomery K, and M D Ross. "Improvements in semiautomated serial section reconstruction and visualization of neural tissue from TEM images", (San Jose, CA: SPIE, 1994).
25. Newman M. William, Sproull Robert F., "Principles of interactive computer graphics", 2a. Ed., Mc. Graw Hill Co. Japan., 1981.
26. Ney Derek, Fishman Elliot, Magid Donna. "Volumetric rendering of computed tomography data: Principles and techniques", IEEE CG&A, marzo 1990.
27. Odesanya S. Olalekan, Warren N. W., "Construction of biological surface models from cross-sections", IEEE Trans. Medical Imag., 40(4), 1993.

28. Lozover, K. Preiss, "Automatic construction of a cubic B-Spline representation for a general curve", *Comput. and Graphics*, 7(2):149-163, 1983.
29. Park, Chan S., "Interactive microcomputer graphics", Addison Wesley Co., USA, 1985.
30. Prothero Jeffrey, Prothero John, "Three dimensional reconstruction from serial sections", *Comput. and Biomedical Res.*, vol. 15, 1982.
31. Reynolds R. A., Gordon D., Chen Lih-Shyang, "A dynamic screen technique for shaded graphics display of slice-represented objects", *Comput. Vision, Graphics Image Proces.*, 38:275-298, 1987.
32. Soyer-P, Roche-A, "Three-dimensional imaging of the liver", *Acta-Radiol* 1991 Sep, VOL: 32(5), P:432-5, ISSN: 0284-1851.
33. Stytz Martin R., Ophir Frieder, "Computer systems for three-dimensional diagnostic imaging: An examination of the state of the art", *CRC*, 19(1):1-45, 1991.
34. Tello-R, Scholz-E, Finn-J-P, Costello-P, "Subclavian vein thrombosis detected with spiral CT and three-dimensional reconstruction", *AJR-Am-J-Roentgenol* 1993 Jan, VOL: 160 (1), P:33-4, ISSN: 0361-803X.
35. Tiede, Ulf y cols., "Investigation of medical 3D-Rendering algorithms", *IEEE CG&A*, Marzo 1990.
36. Wislow James, Bjercknes Mattew, Cheng Hazel, "Three dimensional reconstruction of biological objetcs using a graphics engine", *Comput. and Biomedical Res.*, vol 20, 1987.
37. Wood Sally L, Napel Sandy, "Artifacts and illusions in surface and volume rendering", *IEEE*.
38. Yamaguchi Fujio, "A new fitting method using a CRT computer display", *Comput. Graphics Image Proces.*, 7(1):425-437, 1978.