

TÉCNICAS AVANZADAS PARA LA GENERACION DE IMAGENES

CARLOS ENRIQUE ALVAREZ GUERRERO

Facultad de Ingenieria, Dpto. de Sistemas, Grupo de Investigación DFAC.

Universidad de Los Andes, Bogotá D.E., Colombia.

e-mail: DFAC@ANDESCOL.BITNET AA: 4976 Télex: 42343 Unand Co

Fax: 57(1)2841890 Tel: 57(1)28666185

Resumen - Se discuten los principales algoritmos para implementar un generador de imágenes por rastreo de rayos. Primero se analizan los algoritmos básicos de la técnica de rastreo. Posteriormente se presentan los métodos acelerados de intersección rayo-objetos y el algoritmo de generación de imágenes por duplicaciones sucesivas. Los objetivos son mostrar diversos métodos para implementar un algoritmo eficiente de rastreo de rayos, determinar los problemas que deben ser resueltos y proponer futuras líneas de trabajo en el área.

Palabras Claves: Geometría Constructiva de Sólidos, Caras de un sólido, Imágenes, Rastreo de Rayos, Inclusión en Sólidos, Envoltentes Convexas, Complejidad de Algoritmos.

INTRODUCCION

Las técnicas avanzadas de generación de imágenes de objetos tridimensionales como la de rastreo de rayos supera en calidad a las imágenes generadas con técnicas de estructura de alambre, pero también involucran un orden de complejidad mucho mayor en el número de operaciones requeridas.

En los modeladores basados en el esquema de representación por Geometría Constructiva de Sólidos (GCS) partiendo de superficies cuadráticas, es muy costoso y complejo manejar la representación de la frontera. En este caso se aplica la técnica de rastreo de rayos, con su relativo alto costo en tiempo debido al número de operaciones involucradas.

Por esta razón se inició la investigación de métodos para mejorar los tiempos de respuesta de los algoritmos basados en la técnica de rastreo de rayos, logrando, como se muestra a continuación, resultados positivos y abriendo nuevas posibilidades para trabajos futuros.

DESARROLLO DE LA INVESTIGACION

La investigación comienza con el estudio de las ideas principales que explota la técnica de rastreo de rayos. Esta técnica modela el fenómeno físico de la formación de imágenes en la retina del ojo, que consiste en un rayo de luz que viaja desde una fuente y tras sucesivos choques con objetos 'rebota' hasta llegar al ojo.

El fenómeno real ocurre en paralelo involucrando millones de rayos de luz, que parten de las distintas fuentes, de los cuales sólo una porción llega hasta la retina del ojo contribuyendo en la formación de la imagen. Para aplicar herramientas computacionales en la generación de imágenes se modela el fenómeno invirtiendo el orden causa-efecto. Primero se discretiza el área sobre la retina, donde se forma la imagen, con una matriz de puntos, y para cada punto se calcula el rayo como si partiera de la retina rebotando sucesivamente con los objetos hasta llegar a las fuentes de luz. El modelo mostrado para generar imágenes se conoce como técnica de rastreo de rayos (ray tracing).

El primer resultado que se obtiene del anterior análisis es que el número de operaciones de los algoritmos que implementan la técnica de rastreo de rayos resulta de la cantidad de puntos necesarios para formar la imagen y la complejidad para calcular el color de cada punto. Resultados empíricos muestran que para generar imágenes de mediana resolución es necesario calcular 300 filas por 300 columnas de puntos, lo que da un total de 90.000 puntos. Disminuir esta cantidad es sacrificar la calidad de la imagen generada, anulando la principal característica de la técnica de producir imágenes 'reales'.

Visto que el número de puntos requeridos para generar una imagen no se puede disminuir, a continuación se analiza la complejidad del cálculo del color de cada punto de la imagen, que es el segundo factor en la complejidad total de los algoritmos de generación de imágenes.

Para calcular el color de cada punto de la imagen se modelan varios fenómenos de la física óptica: reflexión, refracción y reflexión difusa. Cuando un rayo de luz choca contra un objeto, una porción se refleja si la superficie del objeto tiene brillo y otra porción se refracta atravesándolo si la superficie tiene transparencia. El fenómeno de reflexión difusa se produce por rayos provenientes de las fuentes de luz y que se reflejan en todas direcciones al chocar contra el objeto. Modelando estos tres fenómenos se calcula el color para cada punto de la imagen, construyendo un rayo inicial que parte de la imagen y calculando las contribuciones por reflexión, refracción y reflexión difusa en el punto en donde el rayo choca con el objeto. Esta técnica para calcular el color de cada punto (figuras 1 y 2) es la que propiamente se conoce como rastreo de rayos.

El algoritmo básico para rastrear los rayos consiste en encontrar el siguiente punto en donde el rayo choca con un objeto de la escena. Este algoritmo se resuelve en dos etapas, en la primera se calculan los puntos de intersección del rayo con todos los objetos que se están visualizando, y en la segunda se escoge el punto de intersección con el que realmente choca el rayo.

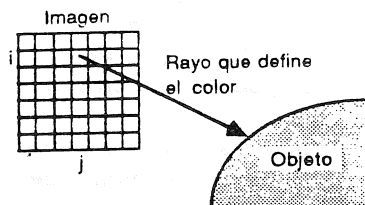


Fig 1. Cálculo del color de un punto de la imagen

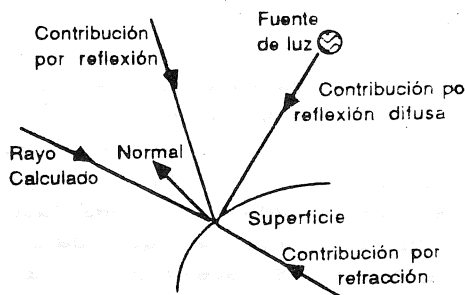


Fig 2. Cálculo del color de un rayo

En el esquema GCS no se mantiene la representación por la frontera de los sólidos, la intersección de un rayo con los objetos no se puede realizar a nivel de caras sino de superficies (entendiendo que una superficie es ilimitada mientras que una cara es una porción de superficie delimitada por contornos que determinan su frontera). Por esta razón el número de puntos de intersección de un rayo con los objetos es directamente proporcional al número de superficies de los objetos (figuras 3 y 4) incrementando la complejidad en la escogencia del verdadero punto de choque.

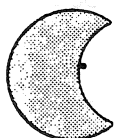


Fig 3. Intersección rayo-objeto

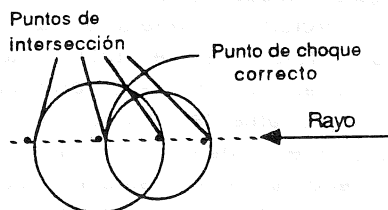
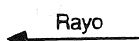


Fig 4. Intersección rayo-superficies

El algoritmo de escogencia del punto de choque correcto se realiza en dos pasos: primero se ordenan los puntos de intersección de menor a mayor con respecto a su distancia al origen del rayo, después se escoge el punto de choque como el primer punto que pertenece a la frontera del objeto.

Los algoritmos de intersección (rayo-objeto) e inclusión (para evaluar si un punto pertenece a la frontera de un objeto) se

mejoran para disminuir sus tiempos de respuesta, pero el tiempo total de generación de imágenes no disminuye considerablemente debido a que la complejidad en el número de operaciones es la misma.

Una mejora se realiza introduciendo envolventes convexos para delimitar el espacio ocupado por los objetos, los cuales se definen a nivel de las primitivas del esquema de representación y se operan a medida que se realizan las operaciones de construcción. Los envolventes sirven como criterio para decidir si una evaluación sobre un objeto es aplicable o no sobre sus componentes.

El criterio para intersectar un rayo con un sólido es el siguiente: si el rayo NO intersecta el envolvente convexo del sólido entonces NO intersecta el sólido; de lo contrario, el cálculo recurre sobre los sólidos hijo-izquierdo e hijo-derecho. El criterio se aplica recursivamente hasta llegar a las primitivas en donde se calcula la intersección entre el rayo y las superficies.

La mejora consiste en detectar pronta y económicamente los casos en que el rayo no intersecta los objetos, ahorrando el cálculo infructuoso de intersecciones entre el rayo y todas las primitivas (superficies) de los objetos.

Los mismos envolventes se utilizan en los algoritmos de inclusión con el siguiente criterio: si un punto está por fuera del envolvente convexo de un sólido entonces está por fuera del sólido; en caso contrario, aplique el algoritmo de inclusión sobre los sólidos hijo-izquierdo e hijo-derecho, y determine la respuesta según la operación de construcción entre estos.

En el caso de la inclusión los envolventes permiten detectar rápidamente los casos en que los puntos están por fuera de los objetos, evitando calcular la inclusión con todas las primitivas que los componen.

Las estructuras de los envolventes convexos son útiles para detectar los casos en que NO es necesario recurrir en la evaluación de un objeto, pero no modifica la forma en que finalmente se realiza. En particular, no se obtiene ningún ahorro

en la evaluación cuando el criterio "no recurra" no es aplicable, y en cambio se recarga el número de operaciones por evaluar dicho criterio a los envolventes. Esto se tomó en cuenta para escoger a la esfera como envolvente convexo, que aunque no es el envolvente más pequeño para algunos casos, si tiene la operación más sencilla para evaluar el criterio.

CONTINUIDAD DE LA FRONTERA DE UN OBJETO

El aporte importante a la disminución de la complejidad de la generación de imágenes se realizó modificando el algoritmo general para calcular el color de cada punto de la imagen. Este aporte resultó de observar que la frontera de los objetos cambian con relativa suavidad, es decir, mantiene continua su derivada con muy pocos puntos de discontinuidad en donde se intersectan sus caras (aristas).

Al mirar la foto de un objeto, se encuentra que el porcentaje de los puntos que caen en las aristas es mucho menor que los que caen totalmente dentro de alguna cara. Visto de otra forma, se encuentra que el número de puntos que conforman las aristas de una cara es casi insignificante comparado con el número de puntos que conforman su interior.

Como cada cara del sólido es generada por SOLO UNA superficie, el párrafo anterior implica que las mayores áreas de las imágenes se pueden generar intersectando los rayos con una sola superficie de los objetos. Más aún, en los puntos sobre las aristas basta con intersectar el rayo con las caras vecinas a la arista, que de todos modos es un número pequeño de caras, generalmente constante.

La mejora introducida consiste en que para calcular el choque de un rayo con los objetos visualizables sólo se intersecta el rayo con UNA superficie de un objeto, generando únicamente dos puntos de intersección. El algoritmo de selección del punto de choque correcto también cambia: dado que sólo uno de los dos puntos de intersección tiene su normal dirigida hacia el observador, pues pertenecen a la misma superficie, el punto que cumpla esta condición es el punto de choque correcto. En total se obtiene que para calcular el choque de un rayo con los objetos visualizables

se realiza un número constante de operaciones que es realmente el MINIMO número posible, ya que de otra forma no se pueden modelar los fenómenos físicos de reflexión, refracción y reflexión difusa.

La nueva estrategia para calcular el punto de choque de un rayo con los objetos se basa en poder determinar de antemano si el rayo se dirige al interior de una cara y conocer la superficie que genera esa cara.

Esta estrategia se puede implementar fácilmente si se tiene información acerca del punto de choque de los rayos vecinos. Suponiendo la configuración de la figura 5, donde cada celda contiene la información de la superficie que generó el punto de choque correcto para el respectivo rayo, la superficie de choque para el rayo de la mitad representado con '?' se puede calcular así: si las ocho celdas vecinas indican que los respectivos rayos chocaron con la misma superficie, $S1 = \dots = S8$, se supone que el rayo se dirige al interior de una cara y se escoge la superficie S1 para calcular el punto de choque de ?; en caso contrario se escogen todas las superficies S1 hasta S8 como candidatas para generar el punto de choque de ?.

S1	S2	S3
S4	?	S5
S6	S7	S8

Fig 5. Información de Vecindad

En el párrafo anterior se muestran los dos casos mencionados al comienzo de esta sección: cuando un rayo choca al interior de una cara, o sobre la intersección de varias caras (aristas o vértices). Cuando las superficies S1 hasta S8 son la misma, el rayo de la mitad se dirige al interior de la cara generada por la superficie S1. En caso de que algunas de las superficies Si sean diferentes, probablemente el rayo se dirige a la intersección de varias caras y el punto de choque se calcula intersectando el rayo con las ocho superficies S1 hasta S8, que en el peor caso serán ocho superficies diferentes.

PROBLEMAS DE LA INFORMACION DE VECINDAD

En cualquiera de los dos casos mencionados, la **suposición** de encontrarse en el interior de una cara o en la intersección de máximo ocho caras puede fallar ocasionalmente cuando sobre el punto en el que choca un rayo con el objeto exista un accidente geométrico no detectado por los rayos vecinos. Un accidente así puede ser una superficie nueva muy pequeña que sobresale o un hueco dentro del objeto cuyo diámetro sea más pequeño que la distancia entre los puntos de choque de los rayos vecinos.

En las figuras 6 y 7, la superficie Sup2 es la base de una depresión del objeto ubicada sobre la superficie Sup1. La suposición de vecindad falla porque los rayos vecinos de R2, R1 y R3, chocan ambos contra la superficie Sup1.

Este inconveniente es fácilmente ignorado si se acepta que la probabilidad de que ocurra un accidente geométrico es muy pequeño y el error que se comete afecta sólo a un punto entre sus ocho vecinos. Lo anterior quiere decir que el error cometido sólo afecta ocasional y localmente puntos dispersos sobre la imagen, lo que no disminuye considerablemente su calidad, y en cambio la complejidad de los algoritmos se hace constante.

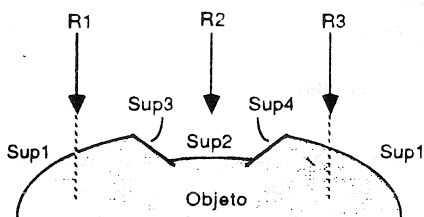


Fig 6. Accidente Geométrico sobre Sup1

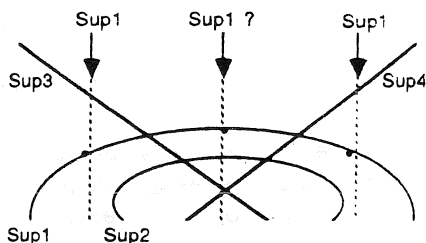


Fig 7. Error de la suposición de vecindad

Lo anterior es cierto, pero pierde sustentación cuando se implementa eficientemente la nueva estrategia. Para acelerar el cálculo de los puntos de choque de los rayos se tiene que disponer apriori de la información de los rayos vecinos. Esta información tiene que ser calculada en algún momento utilizando los algoritmos pesados de rastreo de rayos. Una buena

implementación debe garantizar que el número de rayos iniciales sea tan pequeño como sea posible y estén lo más uniformemente repartidos sobre la imagen.

El algoritmo utilizado para implementar la técnica de rastreo de rayos utilizando la información de vecindad trabaja por duplicación de imágenes. La idea es contruir las imágenes por duplicaciones sucesivas comenzando con otras tan pequeñas como sea posible y aumentándolas hasta conseguir el tamaño requerido. El nuevo esquema de trabajo explota los algoritmos acelerados en cada etapa de duplicación de la imagen, pero introduce fuertemente los problemas causados por los supuestos sobre la información de vecindad.

En las dos siguientes secciones se describen el algoritmo de duplicación de imágenes y se presentan soluciones al problema de la información de vecindad.

DUPLICACION DE IMAGENES

El problema de determinar apriori las superficies de choque de los rayos de una vecindad se puede definir recursivamente haciendo que estos rayos se calculen a partir de otro vecindario, hasta terminar finalmente en un conjunto de rayos que se calculan con los algoritmos pesados de rastreo de rayos. El objetivo de esta sección es definir un algoritmo para duplicar el tamaño de una imagen utilizando la información de vecindad y los algoritmos acelerados, para calcular los nuevos puntos de la imagen.

Para cumplir los anteriores objetivos se desarrollaron los siguientes pasos. Suponga una imagen de 3 filas y 3 columnas (figura 8). Si se conoce las superficies con la que choca el rayo correspondiente a cada punto de la imagen (por cada punto una superficie), es posible duplicar el tamaño de la imagen utilizando los puntos conocidos como vecindad de los nuevos puntos intermedios. La estrategia se muestra en las siguientes figuras en donde los asteriscos "*" indican puntos conocidos y los interrogantes "?" los puntos que se calculan.

.	.	.
.	.	.
.	.	.

Fig 8. Imagen Original (3x3)

.		.		.
.		.		.
.		.		.

Fig 9. Imagen duplicada incompleta

.		.		.
	?		?	
.		.		.
	?		?	
.		.		.

Fig 10. Primer paso

.	?	.	?	.
?	.	?	.	?
.	?	.	?	.
?	.	?	.	?
.	?	.	?	.

Fig 11. Segundo paso

.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Fig 12. Imagen duplicada completa (5x5)

Primero que todo se duplica el tamaño de la imagen con lo que quedan puntos intermedios indeterminados (celdas en blanco). Estos puntos se calculan utilizando los puntos conocidos como 'vecindad' en la forma mostrada en las figuras 10 y 11. En el primer paso, figura 10, se calculan los puntos sobre las filas y columnas pares de la imagen, utilizando los puntos vecinos sobre las filas y columnas impares (vecino superior-izquierdo, superior-derecho, inferior-izquierdo e inferior-derecho). En el segundo paso, figura 11, se calculan los puntos restantes utilizando los puntos vecinos en cruz (vecino superior, izquierdo, derecho e inferior).

Este es el proceso completo para duplicar el tamaño de una imagen. De esta forma es posible generar imágenes de cualquier tamaño comenzando con unas más pequeñas y repitiendo el proceso de duplicación tantas veces como sea necesario. Queda también claro como se utilizan los algoritmos acelerados que se basan en la información de vecindad para calcular los puntos nuevos de cada duplicación.

El problema debido a los supuestos por la información de vecindad fué mencionado a causa de la ocurrencia de accidentes

geométricos. Cuando las imágenes se generan por duplicaciones sucesivas, el efecto visual después de cada duplicación es el aumento en la resolución de la imagen. El algoritmo puede explotarse al máximo comenzando con imágenes rectangulares pequeñas, de por ejemplo 16 a 25 puntos iniciales, utilizando al mínimo los algoritmos pesados de cálculo de choque de rayos.

Cuando el número de puntos de la imagen inicial es muy pequeño, el nivel de resolución de la imagen es muy baja y todas las caras de los objetos se vuelven accidentes geométricos. Los errores cometidos se acumulan por las duplicaciones sucesivas produciendo el ocultamiento de caras. La manera más obvia de solucionar este problema es comenzar con imágenes tan sólo dos o tres veces más pequeñas que el tamaño requerido final, con el fin de disminuir la probabilidad de encontrar accidentes geométricos y localizar los posibles errores en áreas pequeñas de la imagen que no sean significativas en comparación con la imagen total.

Esta solución disminuye la probabilidad de cometer errores por la suposición de vecindad, pero no los evita. De otra parte incrementa el número de puntos que se calculan con los algoritmos pesados de cálculo de choque de rayos. Las soluciones que se proponen a continuación no evitan que se presente el problema de los accidentes geométricos, sino que los detecta. De esta forma se refuerza la suposición de vecindad con criterios más eficaces que determinan cuando es o no correcta la suposición.

SOLUCIONES AL PROBLEMA DE LA SUPOSICION DE VECINDAD

La suposición de vecindad establece que en la vecindad de un rayo se encuentra toda la información necesaria para calcular la superficie con la cual choca el rayo. Los errores debidos a esta suposición ocurren cuando se presentan accidentes geométricos, figuras 6 y 7. Para reforzar la suposición de vecindad se presenta una idea sencilla pero útil para detectar los accidentes geométricos.

Los accidentes geométricos son huecos en las caras de los objetos donde sobresale o se descubre otra cara. Estos huecos tienen un tamaño tal que aunque un rayo se dirija hacia él, ninguno de los rayos vecinos lo hace. Esta situación corresponde a una discontinuidad en la cara de un objeto que se 'oculta' por la suposición de vecindad. Si tal discontinuidad ocurre, entonces ninguno de los puntos de intersección calculados, entre el rayo y las superficies chocadas por los respectivos rayos vecinos, se localizan sobre la frontera del objeto (figuras 6 y 7). Esta condición sencilla pero importante se evalúa con los algoritmos de inclusión.

Con la anterior propuesta se modifica el algoritmo general de generación de imágenes por rastreo de rayos quedando de la siguiente forma:

1. Genere una imagen inicial con los algoritmos pesados de cálculo de choque de rayos:
 - a. Genere la matriz de puntos que conforman la imagen inicial.
 - b. Para cada punto de la matriz genere el rayo respectivo (el rayo que proviene de la escena y determina el color)
 - c. Calcule el punto de choque (determinando la respectiva superficie) de cada rayo:
 - i. Calcule los puntos de intersección con el algoritmo de intersección de rayo con todas las superficies de los objetos
 - ii. Ordene los puntos de menor a mayor de acuerdo a su distancia al origen del rayo
 - iii. Seleccione el punto de choque correcto evaluando el algoritmo de inclusión sobre los puntos
2. Duplique sucesivamente la imagen hasta obtener el tamaño requerido. Para cada duplicación:
 - a. Duplique la matriz de la imagen.
 - b. Para calcular los puntos nuevos siga los pasos mostrados en las figuras 9 a 12.
 - c. Para calcular el punto de choque del rayo respectivo a cada punto:
 - i. Calcule los puntos de intersección entre el rayo y las superficies de choque de los rayos vecinos.

- ii. Ordene los puntos de menor a mayor de acuerdo a su distancia al origen del rayo.
- iii. Seleccione el punto de choque correcto evaluando el algoritmo de inclusión sobre los puntos.
- iv. Si ningún punto se localiza sobre la frontera de los objetos, recalcule el punto de choque utilizando los pasos del numeral c del primer punto (1.c).

RESULTADOS OBTENIDOS

Se implementaron todos los algoritmos y las mejoras propuestas y se realizaron pruebas de generación de imágenes con diferentes objetos. Se utilizó el esquema de representación GCS con envoltentes convexos esféricos para mejorar los algoritmos de intersección e inclusión.

Se modelaron cinco objetos (sólidos tri-dimensionales) diferentes, nombrados y descritos en la tabla-1. Las primeras cuatro columnas describen el número de **superficies** planas, esféricas, cilíndricas y cónicas utilizadas para representar cada objeto, sumando el total en la quinta columna. Las últimas cinco describen el número de **caras** siguiendo la misma clasificación.

Objeto	s pln	s esf	s cil	s con	t sup	c pln	c esf	c cil	c con	t car
ansi	62	-	7	-	69	15	-	7	-	22
botella	20	2	4	2	28	2	2	4	2	10
decaucho	32	-	4	-	36	8	-	8	-	16
dif cilin	8	4	2	-	14	-	2	4	-	6
empaque	86	-	22	-	108	5	-	24	-	29

Tabla 1. Descripción de los objetos modelados

El algoritmo general produce imágenes de rastreo de rayos con un nivel de exploración: cada rayo parte de la imagen hasta chocar contra un objeto de la escena en donde se calcula el color modelando el fenómeno de reflexión difusa. Primero se construye una imagen inicial utilizando el algoritmo original de **rastreo de rayos** (pasos mostrados en el numeral uno de la sección anterior). Para duplicar las imágenes se utilizaron tres tipos de algoritmos basados en las técnicas propuestas. El primer algoritmo, **vecindad-inclusión**, implementa los pasos mostrados en el

numeral dos de la sección precedente. El segundo, **vecindad**, difiere del anterior en que suprime el paso 2.c.iv. El último algoritmo, **interpolación**, utiliza directamente la información de color de los puntos vecinos para interpolar linealmente.

Para cada objeto modelado se generaron doce imágenes de 321 filas por 321 columnas, empleando diferentes combinaciones de algoritmos durante las duplicaciones. Como es de esperarse, los algoritmos más exactos se utilizaron en los primeros niveles de duplicación y los más rápidos en los últimos. Siguiendo este orden, se utilizó siempre primero el algoritmo de vecindad-inclusión, luego el de vecindad y por último el de interpolación. En la tabla 2 se resumen los tiempos y calidades de las imágenes obtenidas de cada objeto y cada tipo de combinación.

La primera columna de la tabla 2, **código**, indica la combinación de algoritmos utilizada en los diferentes niveles de duplicación de la imagen. El código se compone de tres dígitos, que leídos de izquierda a derecha indican el número de veces que se empleó el algoritmo de vecindad-inclusión, de vecindad, y de interpolación respectivamente. Para un tamaño de 321 puntos de lado, la imagen puede ser generada hasta por seis niveles de duplicación, siendo esta cantidad la suma máxima de los dígitos del código. El código '000', primera fila, indica que no se usó ningún nivel de duplicación, en otras palabras, toda la imagen fué generada inicialmente del tamaño total utilizando el algoritmo original de rastreo de rayos.

Para cada objeto, según la combinación utilizada, se describe el tiempo total de generación de la imagen, el porcentaje comparativo de tiempo con la imagen ideal (correspondiente a la combinación de la primera fila) y la calidad visual. Para todos los objetos se obtuvieron máximo tres niveles distintos de calidad de imagen calificadas de uno a tres, uno para la imagen "perfecta" y tres para la más errada.

En el anexo A se muestran los tipos de imágenes obtenidos, identificadas en correspondencia con la tabla 3.

imagen	ansi			botella			decaucho			empaque			dif cilind		
Código	Tiemp	%	Cl	Tiemp	%	Cl	Tiemp	%	Cl	Tiemp	%	Cl	Tiemp	%	Cl
000	1163	100	1	550	100	1	1975	100	1	1651	100	1	297	100	1
600	651	60	3	339	62	3	1146	58	1	1358	82	1	272	92	1
060	95	8.2	3	55	10	3	161	8.2	1	107	6.5	3	49	16	1
003	93	7.9	2	51	9.3	1	144	7.3	1	189	11	1	36	12	2
222	83	7.1	3	46	8.4	3	124	6.3	1	177	10	1	34	11	1
123	82	7.0	3	45	8.2	3	125	6.3	1	176	10	1	33	11	2
004	83	7.1	2	45	8.2	2	127	6.4	1	178	10	1	33	11	2
114	82	7.0	3	45	8.2	3	120	6.1	1	175	10	1	33	11	2
105	83	7.1	3	46	8.4	3	123	6.2	1	175	10	1	33	11	3
051	85	7.3	3	47	8.5	3	132	6.7	1	101	6.1	3	37	12	1
015	83	7.1	3	46	8.4	3	123	6.2	1	99	5.9	3	33	11	3
006	82	7.0	3	45	8.2	3	120	6.1	1	99	5.9	3	32	11	3

Tabla 2. Resultados en tiempo (segundos) y calidad de imagen

Objeto	Calidad	Objeto	Calidad	Objeto	Calidad
botella	3	botella	2	botella	1
ansi	3	ansi	2	ansi	1
decaucho	1	empaque	3	empaque	1
dif cilin	3	dif cilin	2	dif cilin	1

Tabla 3. Identificación de la imágenes

CONCLUSIONES Y TRABAJOS FUTUROS

En la práctica se diseñaron algoritmos para implementar las técnicas presentadas en este artículo y se obtuvieron resultados muy alentadores en cuanto a los tiempos de respuesta de generación de imágenes.

Los mayores avances se logran combinando varios métodos de duplicación de imágenes, utilizando sólo en los primeros niveles de duplicación, cuando la resolución es muy baja, los criterios para detectar accidentes geométricos y utilizando en los últimos niveles de duplicación la suposición de vecindad.

Otras formas de jugar con los niveles de duplicación es utilizar métodos de interpolación lineal o de segundo grado para duplicar las imágenes utilizando tan sólo la información del color. Si estos métodos se incluyen en los últimos niveles de duplicación,

la imagen no sufre mayor deterioro en cuanto a calidad y en cambio se obtienen tiempos de respuesta más cortos.

Combinar diferentes métodos en los niveles de duplicación ha reportado resultados muy interesantes en cuanto a la calidad de las imágenes generadas y los tiempos de respuesta. En este sentido se juega con un problema ya muy conocido expresado en la siguiente ecuación:

$$\text{calidad_imagen} / \text{tiempo_generación} = \text{constante.}$$

Sin embargo la generación de imágenes por duplicaciones sucesivas tiene dos características importantes muy favorables:

1. Para duplicar una imagen, el número de puntos nuevos que se calculan

es tres veces mayor que todos los puntos ya calculados.

2. En un solo nivel de duplicación, los errores que se cometen por ocultamiento de caras ocurren con poca frecuencia y están lo suficientemente dispersos de modo que no afectan significativamente la calidad de la imagen total duplicada.

Las dos características mencionadas abren muchas alternativas para aplicar diversos métodos en los diferentes niveles de duplicación de las imágenes. La estrategia global se orienta a usar métodos más efectivos pero menos eficientes en los primeros niveles de duplicación, y métodos menos efectivos pero mas eficientes en los últimos. La conclusión de las dos características mencionadas es, que la calidad de las imágenes generadas por duplicaciones sucesivas se define en los primeros niveles de duplicación, mientras que el tiempo de respuesta total se define en los últimos niveles.

Otra alternativa que se abre para disminuir el tiempo de respuesta de generación de imágenes, independiente de la decisión de los métodos que se usan en los diferentes niveles de duplicación, es dividir la imagen en sub-imágenes y usar procesamiento en paralelo. El problema de generar una imagen de un tamaño dado es perfectamente divisible en el problema de generar n sub-imágenes independientes, dividiendo el tiempo de respuesta por el factor n , donde n es el número de procesadores que pueden actuar en paralelo.

En el presente trabajo se muestran varios algoritmos para calcular los puntos de choque de los rayos con los objetos de una escena. Para los algoritmos acelerados, basados en la suposición de vecindad, se muestra igualmente que el criterio de inclusión los hace más efectivos porque detectan los accidentes geométricos. Existe sin embargo un caso patológico que el criterio de inclusión no detecta cuando se aplica la suposición de vecindad.

El caso patológico se describe con las figuras 13 y 14, donde R1, R2 y R3 son rayos y Sup1, Sup2, Sup3, Sup4 y Sup5 son superficies de un objeto de la escena. Si se está calculando la superficie de choque de R2, los rayos vecinos R1 y R3 sólo mencionan las superficies Sup1 y Sup5. El problema es que aunque R2 corta a Sup5, y el punto de intersección pertenece a la frontera del objeto, la verdadera superficie de choque de R2 es Sup2.

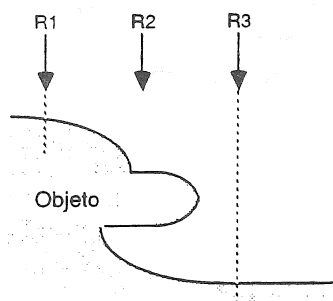


Fig 13. Caso Patológico de un accidente geométrico

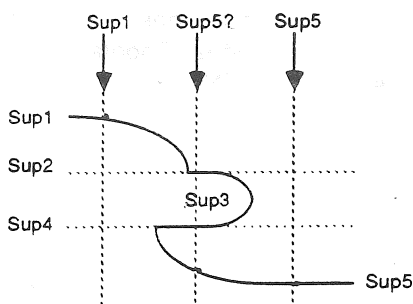


Fig 14. Error por la suposición de vecindad no detectado con inclusión

La anterior situación se muestra como un caso patológico que tiene que ser tratado y solucionado en trabajos posteriores. También se propone como trabajos futuros la corrección de imágenes, mediante barridos que detecten puntos con cambios bruscos en el color o que pertenezcan a superficies diferentes a las superficies de todos los puntos vecinos.

BIBLIOGRAFIA

- [LEV76] LEVIN, J. A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces. Comm. ACM. 19, 10. (Oct 1976) Págs 555-563.
- [MAN88] MANTILA, M. An introduction to Solid Modeling. Comput. Science Press, Inc. Rockville, Maryland. 1988.
- [MIL88] MILLER, J. R. Analysis of Quadric-Surface-Based Solid Models. IEEE, Comp. Graphics & Applications. January, 1988. págs 28-42.
- [YAN87] YANG, C-G. Illumination Models for generating images of curved surfaces. Computer-aided design. Vol 29, Nº 10. (Dec 1987) Butterworth & Co Ltda. págs 544-554.
- [HEC87] HECKBERT, P. S. Writing a Ray Tracer. Introduction to Ray Tracing Tutorial Notes. SIGGRAPH'87, July 1987.
- [ALV90] ALVAREZ, C. Superficies Cuadráticas para el Modelaje de Sólidos. Tesis de Grado, CIFI, Universidad de Los Andes, Bogotá DE, Colombia. 1990.
- [HAI] HAINES, E. Essential Ray Tracing Algorithms.

Anexo A. Niveles de Calidad de las imágenes generadas

