

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

MAESTRÍA EN CIENCIAS DE LA INGENIERÍA



Rogelio A. Campos Serapio

TESIS

RENDER BASADO EN FÍSICA, UTILIZANDO CONOS Y  
ESTRUCTURAS HBVH.

TESIS PRESENTADA POR:  
ING. ROGELIO ALFREDO CAMPOS SERAPIO

PARA OBTENER EL GRADO DE:  
MAESTRO EN CIENCIAS DE LA INGENIERÍA

ASESOR:  
DR. MARCO ANTONIO RAMOS CORCHADO



*Nadie se hace capitán en aguas someras.*  
*Raúl A. Campos V.*

# A GRADECIMIENTOS

*A mis padres*

*Raúl Campos y Gloria Serapio.*

*Quienes con su amor y sabiduría me han apoyado incondicionalmente*

*A mi Lucerito*

*“Cuando te canses de andar... yo seré tu bastón.”*

*A mi abuela Paula*

*Quien es un gran ejemplo de vida, bondad y lucha.*

*A mis amigos*

*Quienes me apoyaron en esta aventura y juntos crecimos tanto personal como profesionalmente: Iván F. Valencia , Ruth Montes de Oca.*

*“Los sobrevivientes del Olimpo serán los titanes”.*

*Agradecimiento especial a:*

*Dr. Manuel I. Martín Ortiz, Dr. J. Arturo Olvera L., Dra. Rosa M. Valdovinos R., Mtra. M. Carmen Santiago D., quienes han sido amigos y mentores.*

*Agradecimiento a las instituciones*

*Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo con la beca 647544*

*Al Laboratorio Nacional de Supercómputo del Sureste de México (LNS) por el apoyo brindado.*

**En memoria de**

*Ari y Mamá Nely  
Siempre vivirán en nuestros corazones.*



# TABLA DE CONTENIDO

<b>Introducción</b>	<b>VI</b>
Problema . . . . .	VIII
Motivación . . . . .	IX
Objetivo . . . . .	IX
Objetivos particulares . . . . .	X
Metodología . . . . .	X
Estructura del documento . . . . .	X
<b>1. Estructuras utilizadas en el Raytracing</b>	<b>1</b>
1.1. Jerarquías Espaciales . . . . .	1
1.1.1. Z-buffer . . . . .	2
1.1.2. Kd-Tree . . . . .	4
1.1.3. Árbol de particionamiento binario del espacio . . . . .	6
1.1.4. Quadtree . . . . .	8
1.2. Octree . . . . .	11
1.3. Jerarquías de Objetos . . . . .	11
1.3.1. Objetos acotados por volumen . . . . .	12
1.3.2. Jerarquía de Volúmenes Acotados . . . . .	13
1.3.3. Jerarquías Híbridas . . . . .	16
1.3.4. Heurísticas basadas en áreas de superficie . . . . .	17



Rogelio Alfredo Campos Serapio



## ÍNDICE DE FIGURAS

1.1.	Planos delimitadores del espacio de trabajo, dirección del eje $z$ .	2
1.2.	Representación de los polígonos almacenados en el buffer de datos	3
1.3.	Aplicando Z-buffer a los modelos 3D	3
1.4.	Coordenadas en un espacio bidimensional manejado por un Kd-tree.	5
1.5.	Representación de un espacio bidimensional ejemplificado por la Figura 1.4, el resultado de los datos ordenados en un árbol Kd-tree.	5
1.6.	Polígonos en espacio 3D	6
1.7.	Árbol BSP	7
1.8.	Cada línea $h_n$ divide el espacio en dos, esto se realiza de forma recursiva.	7
1.9.	Cada nodo reserva cuatro espacios de memoria.	9
1.10.	Representación de la información en una imagen y su correspondiente quadtree.	9
1.11.	División del espacio en cuadrantes.	10
1.12.	Árbol resultante de la Figura 1.11.	10
1.13.	Representación de un espacio tridimensional y su respectivo árbol.	11
1.14.	Esquema del espacio que ocupa un objeto en $\mathbb{R}^2$ , la distancia entre $d_i^{Near}$ y $d_i^{Far}$ se conoce como <i>slab</i> .	12
1.15.	Objetos comúnmente usados como delimitadores.	14
1.16.	Árbol BVH.	14
1.17.	Objeto delimitado por OBB, se ejemplifica que varios volúmenes son requeridos dependiendo de la resolución deseada.	15
1.18.	Combinación de AABB y esferas.	16
1.19.	Esquema de superficies de SAH.	17



# ÍNDICE DE CUADROS



Rogelio Alfredo Campos Serapio



## INTRODUCCIÓN

ACTUALMENTE el ser humano ha integrado a su vida cotidiana el uso de sistemas de procesamiento masivo, estos van desde simples dispositivos móviles hasta equipos sofisticados de alto desempeño, así mismo las ciencias computacionales proponen nuevas técnicas y metodologías para el tratamiento de la información en específico la parte de la interacción humano computadora. Es en esta área donde la representación gráfica toma importancia. Este trabajo de investigación aborda el problema de la representación gráfica de los entornos virtuales y los objetos que los componen esta es un área de las ciencias computacionales denominada “graficación por computadora”.

La interacción humano computadora es natural cuando la información es presentada de manera gráfica [4], la información puede ser interpretada de mejor forma y de manera mucho mas rápida, en comparación a ser presentadas como cadenas de caracteres que el usuario tendrá que traducir e interpretar. De ahí la importancia de construir ambientes virtuales que permitan al usuario tomar decisiones en condiciones de incertidumbre. A medida que incrementa el volumen de la información es necesario encontrar mejores técnicas de procesamiento, en el caso particular de los gráficos por computadora se puede apreciar que una imagen está saturada de múltiples características. Hoy en día existen herramientas que permiten generar gráficos con complejas relaciones entre datos y su visualización. Sin embargo estos gráficos carecen de los fenómenos físicos observados en los ambientes naturales por el ser humano [5]. Los gráficos por computadora se han posicionado como una herramienta útil en la interpretación de información. En la actualidad se utiliza ampliamente la representación de objetos tridimensionales en diferentes áreas, como: medicina, arquitectura, industria de manufactura, desarrollo de vídeo juegos, animación e industria del cine, entre otros.

El realismo en las imágenes toma importancia, ya que permite crear sistemas de simulación complejos que pueden ser utilizados en las diferentes áreas del conocimiento, como en la industria aeroespacial en donde se requiere que el piloto esté familiarizado con la aeronave antes de realizar el primer vuelo; en el caso de la medicina se utilizan las imágenes realistas en el entrenamiento de procedimientos quirúrgicos o el estudio anatómico; en la industria del cine se usa para generar escenarios sintéticos y filmar escenas de alto riesgo para mantener seguros a los actores y personal

involucrado, ejemplo de esto se puede apreciar en algunos títulos, como en el caso de películas como *Avatar*, *X-Men:First Class*, *Fast and Furious* [6], entre otras.

Las ciencias computacionales, en específico la graficación por computadora, propone diferentes técnicas para la creación de imágenes sintéticas, sin embargo la complejidad de la imagen y el tiempo de producción son elevadas, por ello se hace uso del cálculo de la interacción de la luz conocido como *Raytracing* [7]. La técnica de raytracing consiste en la emisión de rayos que al colisionar con el objeto este obtiene una coloración en el área en cuestión, esta técnica ofrece la reconstrucción de objetos sintéticos con cierto realismo, sin embargo el tiempo para obtener los efectos visuales deseados es alto. El uso de algoritmos de cálculo son utilizados para acelerar la construcción de los objetos. En este trabajo se estudian las diferentes estructuras del tratamiento de la información y se propone una alternativa para obtener imágenes en el menor tiempo posible, pero sobre todo observar fenómenos que los métodos tradicionales no consiguen obtener.

La creación de imágenes foto-realistas depende mucho de las estructuras de datos que manejan la información de los objetos a renderizar, las cuales son cruciales para una rápida reconstrucción o actualización de la escena. La carencia de una estructura de datos o la inadecuada selección de ésta, puede ser un cuello de botella en el desempeño del motor de raytracing [8]. Desde los orígenes de las imágenes por computadora se han formulado distintas estructuras de datos para manejar las primitivas de las escenas, esto ha ocupado a los investigadores interesados en estas áreas a desarrollar nuevas y mejores formas de calcular las superficies visibles de los objetos a partir del punto de observación (la cámara).

Utilizar el raytracing y algoritmos involucrados como técnica para la generación de imágenes sintéticas crea un amplio campo de estudio, en donde el objetivo principal es minimizar el tiempo de espera y obtener una imagen que pueda ser utilizada.

Hoy en día es común hacer uso del cómputo paralelo en raytracing, lo que reduce el tiempo de creación de imágenes sintéticas. Sí bien es cierto que los tiempos de procesamiento y aumento de la memoria permite procesar los algoritmos actuales del raytracing, aún existen problemas en poder observar fenómenos en las imágenes obtenidas, en donde la demanda actual es obtener escenas con mayor grado de complejidad y realismo.

## PROBLEMA

---

En un escenario dinámico virtual, el ángulo de visualización se modifica constantemente, lo que significa que el objeto a visualizar requiere calcular diferentes fenómenos luminosos en tiempos breves para no perder la sensación de: continuidad, perspectiva, volumen y textura.

Las técnicas actuales para la generación de imágenes realistas basadas en raytracing utilizan algoritmos recursivos tomando tiempos largos de procesamiento. El presente trabajo propone una técnica para no perder la sensación de realismo en las imágenes obtenidas, sin sacrificar características foto-realistas como sombras, reflejos, transparencias, etc.

Para darse una idea del cálculo necesario para renderizar una imagen se tiene que tomar en cuenta la cantidad de primitivas ( $N$ ), el número de fuentes de luz ( $L$ ) y el tamaño de la imagen deseada ( $M$ ) [9]. Suponiendo que se calcule solo la primera intersección del rayo y el objeto, para ejemplificar, los cálculos necesarios de la primera interacción entre la luz y el objeto se tiene que calcular:

$$O(N \cdot M \cdot L)$$

Sí se considera que:

$$N = 252,000$$

$$M = 1,024 \cdot 1,024 = 1,048,576$$

$$L = 2$$

$$O(N \cdot M \cdot L) = (252,000)(1,048,576)(2) = 528,480,304,000 \text{ rayos.}$$

Cada rayo requiere que se calcule el punto de origen, dirección, ángulo de incidencia, ángulo de salida, entre otros, lo que consume tiempo de procesador.

## MOTIVACIÓN

---

La interacción humano computadora requiere de interfaces gráficas que permitan tomar decisiones y acciones en una condición dada. Por otra parte las imágenes foto-realistas resuelven problemas de visualización para distintas disciplinas. La posibilidad de jerarquizar los objetos permite la paralelización de los algoritmos involucrados en la generación de imágenes basadas en raytracing.

## OBJETIVO

---

La propuesta de investigación busca generar imágenes basadas en objetos 3D con un alto realismo, utilizando la técnica de raytracing, optimizando los tiempos de las estructuras de datos que permitan reducir los tiempos de obtención de la imagen final.

Generar imágenes de superficies cristalinas, caracterizando sus propiedades físicas, ya que este tipo de textura involucra una gran cantidad de tiempo de cómputo para

generar una imagen aceptable.

Utilizar los modelos tridimensionales propuestos por la comunidad especializada en gráficos por computadora.

## OBJETIVOS PARTICULARES

---

- Estudiar y analizar los principales algoritmos de raytracing.
- Evaluar las estructuras propuestas en la literatura de raytracing.
- Implementar los algoritmos raytracing más eficientes para el caso de estudio.
- Seleccionar casos de uso para validar los algoritmos implementados.

## METODOLOGÍA

---

- Revisión constante de el estado del arte sobre el raytracing.
- Estudio de las principales estructuras algorítmicas para el uso de raytracing en el área de las ciencias computacionales.
- Clasificación de las estructuras susceptibles a paralelizar.
- Analizar las características y propiedades de los principales métodos de raytracing destacando cualidades y debilidades.
- Implementar los principales y validar con los objetos 3D de la literatura la propuesta realizada para la obtención de objetos foto-realistas.
- Describir detalladamente las mejoras de la propuesta desarrollada.

## ESTRUCTURA DEL DOCUMENTO

---

El presente trabajo de investigación está compuesto por cuatro capítulos, en los cuales se detallan ideas y conceptos necesarios para el desarrollo de la presente investigación.

El primer capítulo aborda a detalle las principales estructuras de datos que son utilizadas con la técnica del raytracing, estructuras que trabajan sobre las dimensiones del espacio y las que trabajan el volumen de objeto. Tomará particular interés sobre

las estructuras híbridas que permitirán formalizar la propuesta de investigación.

El segundo capítulo aborda los conceptos de óptica los cuales permiten modelar interacción entre la luz y las superficies de los objetos, de forma que crean distintos efectos a visualizar, cada uno de ellos con un requerimiento computacional diferente.

En el capítulo tres se estudian los algoritmos de renderizado que toman en cuenta gran parte de las características que se desean obtener en la imagen final.

En el capítulo cuatro se desarrolla la propuesta de investigación utilizando los algoritmos y conceptos de los capítulos previos y la muestra de resultados, haciendo una comparación entre los distintos tiempos de cómputo, así como de la cantidad de polígonos utilizados por los modelos propuestos por la comunidad de gráficos por computadora.

Rogelio Alfredo Campos Serapio



# Capítulo

# 1

## ESTRUCTURAS UTILIZADAS EN EL RAYTRACING

Las estructuras de datos se han utilizado con distintos propósitos, ya que permiten llevar un mayor control de la información, en los gráficos por computadora también han tenido cabida al optimizar la consulta de los datos que compone la escena. Existen distintos enfoques de estructuras orientadas a los gráficos por computadora, computar la información necesaria para ser mostrada en pantalla es de gran interés, ya que de esta manera se puede descartar información innecesaria, optimizando el tiempo de renderizado.

Hablando desde un sentido de computación, un árbol es una colección jerárquica de información, donde cada uno de sus elementos (nodos) está ordenado de acuerdo al dato o datos que lo componen. La búsqueda de datos tiene aplicación en una gran variedad de campos de investigación, en este caso interesa su aplicación en gráficos por computadora, ya que un árbol de búsqueda permite localizar la información de forma rápida. En los gráficos por computadora existen dos grandes enfoques de estructuras de datos; Las *jerarquías espaciales* las cuales dividen el espacio en sub-regiones; Las *jerarquías de objetos* las cuales dividen el objeto recursivamente en objetos cada vez más pequeños. En los siguientes párrafos se aborda cada uno de estos.

### 1.1 JERARQUÍAS ESPACIALES

---

Estructuras de datos orientadas a la jerarquía espacial, en donde la dimensión del espacio en el que se está trabajando es un criterio para ordenar los datos que aparecen en este, dependiendo de los usos a los que se enfoque la estructura de datos será la forma en la cual se trabaje la división del espacio, así como el criterio con el cual se compararán los puntos para poderse almacenar, en estos casos se puede encontrar estructuras como *z-buffer*, *BSP*, *Kd-tree* y *Octree*.

1.1.1 Z-BUFFER

Un buffer de datos es un espacio de memoria reservada para almacenar temporalmente la información a procesar, en este espacio se almacena el resultado de la ecuación 1.1 en lo que se conoce como Z-buffer [10] es un método que elimina las superficies no visibles de la escena, transformando los objetos de la escena de tal forma que la dirección de visualización en el eje  $z$  sea positiva. Los objetos en la escena se exploran de forma arbitraria, a medida que se alejen del plano de visión se apreciarán de un menor tamaño en la escena, la conversión de proyección permite determinar si un objeto es cubierto por el espacio de visualización de la cámara, por tanto potencialmente visible.

$$z' = -\frac{f+n}{f-n} - \frac{1}{z} \cdot \frac{2nf}{f-n} \quad (1.1)$$

El buffer determina si los objetos que conforman el escenario se encuentran dentro del ángulo de visión de la cámara, es decir, si existe una correspondencia de los objetos y el plano de visión en donde se generan los píxeles de la imagen, de ser así, se calcula la profundidad a la que se encuentran los puntos de interés del objeto.

El Z-buffer almacena la distancia entre el objeto y la cámara. La profundidad de  $z$ , donde el espacio de trabajo debe ser acotado entre dos planos, el plano lejano  $f$  y el plano cercano  $n$  ( ver Figura 1.1 ).

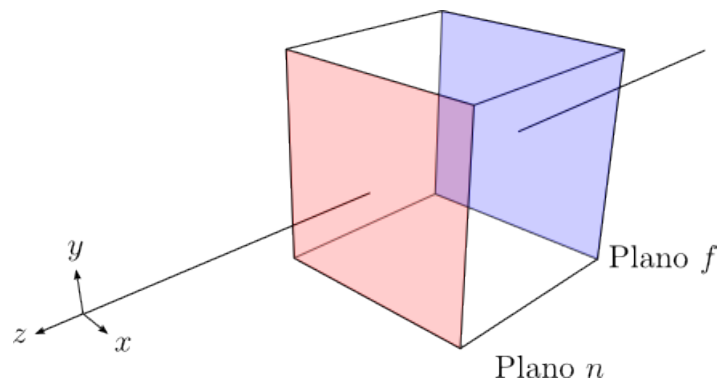


Figura 1.1: Planos delimitadores del espacio de trabajo, dirección del eje  $z$ .

Para generar la imagen, se recorre el Z-buffer, empezando por los puntos más alejados, si la coordenada  $z$  del objeto de interés es menor a la coordenada  $z$  del polígono  $P_n$  se almacena en la memoria del buffer  $M_n$  ( ver Figura 1.2 ).

## 1.1. JERARQUÍAS ESPACIALES

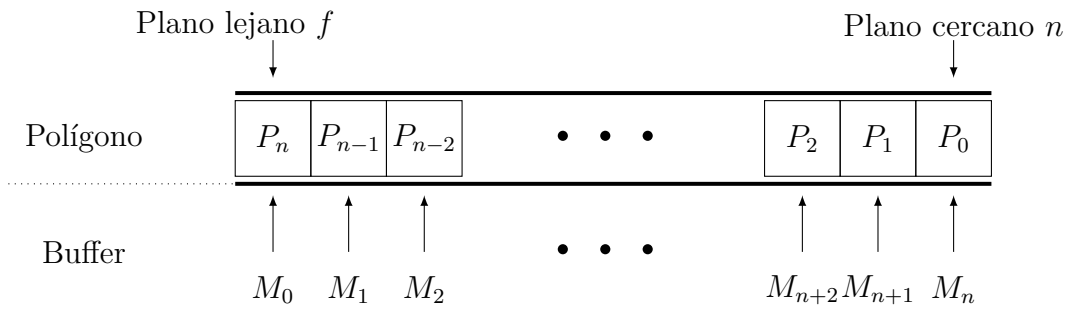
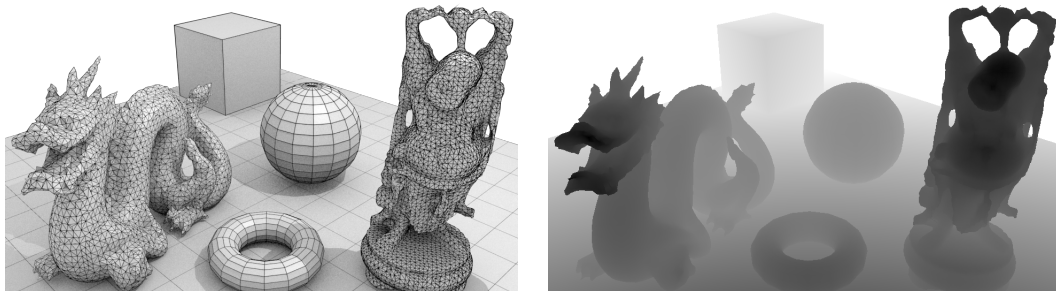


Figura 1.2: Representación de los polígonos almacenados en el buffer de datos

Esto permite determinar la profundidad entre los objetos que conforman el escenario tridimensional y el plano de visión. Al recorrer el buffer los objetos que se encuentran distantes al plano se pintan primero en la imagen, finalizando con los objetos que se encuentran cerca del plano de visión, esto provoca que se dibuje sobre los objetos previamente mostrados.

El método es rápido y sencillo, la desventaja es que se requiere memoria extra para el Z-buffer, y una prueba adicional en el plano  $z$  para comprobar si cada pixel es cubierto por un objeto. En la Figura 1.3(a) se muestran modelos poligonales que conforman el escenario y la implementación de Z-buffer se muestra en la Figura 1.3(b), donde se resalta la profundidad en escala de grises, las áreas oscuras se encuentran cercanos a la cámara y áreas claras para el caso contrario.



(a) Composición del mallado de objetos tridimensionales.

(b) Implementación utilizando Z-buffer.

Figura 1.3: Aplicando Z-buffer a los modelos 3D

En la Figura 1.3 se muestra el resultado de implementar Z-buffer, cada pixel de la coordenada  $z$  del objeto visible se muestra como una intensidad de color, los objetos se van oscureciendo a medida que se acercan al plano de visión.

El frame buffer calcula la correspondencia del objeto y el plano de visión, mientras que Z-buffer la distancia a la que se encuentran de éste, denominándose como profundidad del plano de visión, simplemente profundidad. Z-buffer es uno de los principales algoritmos para determinar las superficies visibles en el escenario al ordenar la profundidad a la que se encuentran éstas. La sencillez del algoritmo lo hace

uno de los más utilizados para remover superficies ocultas, sin embargo el algoritmo tiene el inconveniente de requerir almacenamiento adicional para el Z-buffer y una prueba extra en la coordenada  $z$ . A esto se le conoce como *el algoritmo del pintor* donde los objetos más alejados al plano de visión se “pintan” primero en pantalla, sobre estos se sobre ponen los que están a menor profundidad.

Z-Buffer opera la dimensión  $z$  de los objetos, sin tomar en cuenta como parte del criterio de ordenamiento las demás dimensiones;  $x$  y  $y$ , lo que facilita la implementación del algoritmo, pero esta misma falta de criterio hace que Z-buffer tenga que almacenar y consultar toda la información de los objetos que se encuentran dentro del escenario, así que los polígonos no visibles también ocupan espacio de memoria, sin embargo el mayor costo es que tengan que ser procesados a pesar de no ser presentados en la imagen final.

### 1.1.2 KD-TREE

Kd-Tree [11] es un árbol multidimensional, descompone el espacio conforme se desciende por cada nivel del árbol, los cuales representan la división del plano de trabajo, dependiendo de la dimensionalidad de los datos que se requiera almacenar será el grado del árbol.

La ventaja de *Kd-tree* respecto a otras estructuras como octree y quadtrees es que Kd-tree permite hacer consultas por niveles del árbol, lo que no sucede con estructuras que necesitan recorrer todo el árbol antes de poder procesar la información. Lo que es más costoso en términos de espacio debido a múltiples nodos nulos o vacíos, en particular el tamaño de un nodo es mayor para un árbol  $d$ -dimensional cuyo tamaño de registro es al menos  $d + 2^d$  (asumiendo que cada puntero es almacenado en un registro) [12].

Suponiendo que se tiene un punto  $P$  en un espacio bidimensional, este tiene un discriminador  $x$ , todos los nodos que contengan una coordenada  $x$ , cuyo valor sea menor que el discriminador  $x$  de  $P$ , serán hijos izquierdos de  $P$ , de lo contrario serán hijos derechos de  $P$ , un criterio similar se usa para espacios como  $y$  y  $z$ .

#### DEFINICIÓN 1.1:

Sea  $D$  un conjunto de puntos en  $\mathbb{R}^k$ , donde todos los puntos son distintos. Cada elemento de  $\mathbb{R}^k$  es una  $k$ -tupla  $(v_0, v_1, \dots, v_{k-1})$ , donde cada valor es un atributo del registro.

Sea  $P$  un punto con  $k$  elementos, y cada uno apuntara a otro nodo como los cuales serán  $LOSON(P)$  y  $HISON(P)$ , y un discriminador  $DISC(P)$ , el cual indica el orden del árbol.  $j$  será el discriminador para cada nodo  $Q$  en  $LOSON(P)$  cuando  $K_j(Q) < K_j(P)$ , de la misma forma para cualquier nodo  $R$  en  $HISON(P)$  cuando  $K_j(Q) < K_j(P)$ . El discriminador es calculado como:

$$NEXTDISC(i) = (i + 1) \bmod k$$

## 1.1. JERARQUÍAS ESPACIALES

De los puntos :  $A(50, 50)$ ,  $B(10, 70)$ ,  $C(80, 15)$ ,  $D(25, 20)$ ,  $E(40, 85)$ ,  $F(70, 85)$  y  $G(10, 60)$ , se obtiene la Figura 1.4 al localizarlos en un espacio bidimensional.

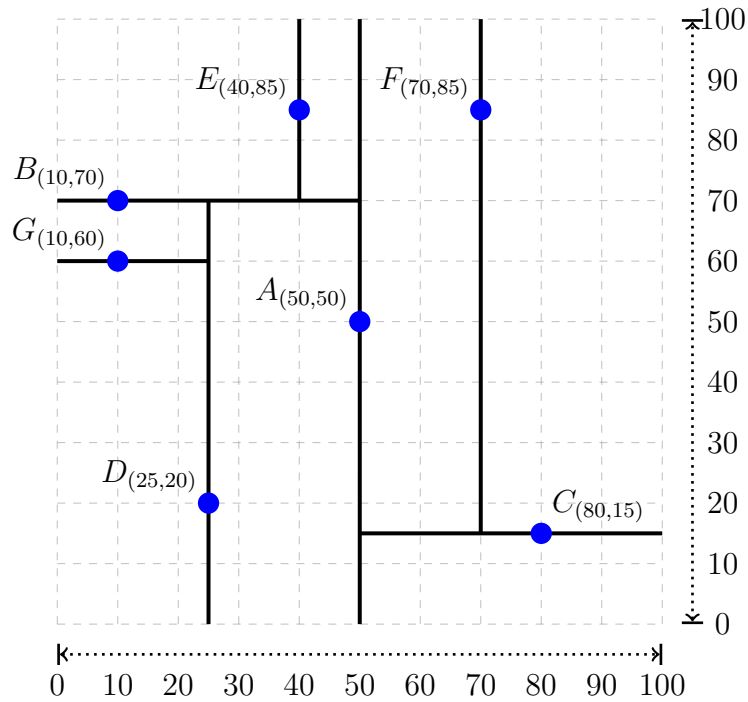


Figura 1.4: Coordenadas en un espacio bidimensional manejado por un Kd-tree.

Con esto orden se logra una subdivisión recursiva del espacio en un tiempo  $O(\log n)$  y su tiempo de construcción es de  $O(n \log n)$ . Cada nivel del árbol Kd-tree es un espacio dimensional ( ver Figura 1.5 ).

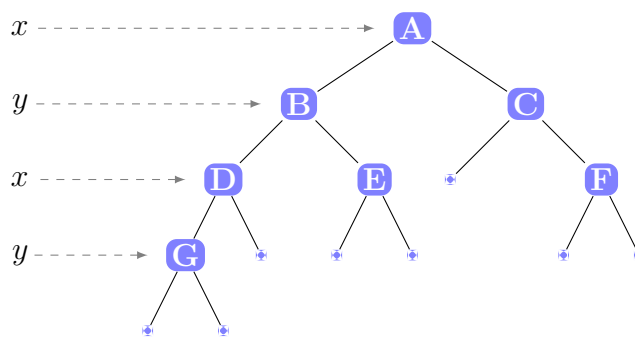


Figura 1.5: Representación de un espacio bidimensional ejemplificado por la Figura 1.4, el resultado de los datos ordenados en un árbol Kd-tree.

El Kd-tree es considerado por muchos como la mejor estructura de datos para escenarios estáticos en el raytracing [3]. Su desventaja radica en los escenarios dinámicos. Esto implica que se necesite reconstruir la estructura desde el principio para cada cuadro a mostrar, ya que se requiere una constante actualización de la estructura, por lo que se vuelve costoso computacionalmente. Kd-tree se adapta mejor a

la implementación a nuevas arquitecturas de hardware con múltiples procesadores, aplicando paralelización se logra un mejor desempeño.

### 1.1.3 ÁRBOL DE PARTICIONAMIENTO BINARIO DEL ESPACIO

Un árbol de particionamiento binario del espacio abreviado (*binary space partitioning tree*, BSP) [13] se puede interpretar como una generalización de Kd-tree, con la diferencia de que en un BSP el plano de división puede tener una orientación y posición arbitraria [14]. La división sobre un espacio  $\mathbb{R}^3$  se realiza desde el punto de visión por medio de un plano<sup>1</sup>, el cual crea dos particiones del espacio visible [15].

Un ambiente 3D está conformado por un conjunto de polígonos  $P = \{p_1, p_2, \dots, p_n\}$ , seleccionar un polígono arbitrario  $p_k$  funge como nodo raíz ( ver Figura 1.6 ). El plano de corte definido por  $p_k$  particiona el espacio en dos,  $S_k$  y  $S'_k$ , las cuales se identifican como positivo y negativo, tomando la ecuación del plano  $ax + by + cz + d = 0$ ,  $S_k = \{(x, y, z) | ax + by + cz + d \geq 0\}$ . Sí un polígono en  $P - \{p_k\}$  se encuentra completamente en  $S_k$ , entonces es parte del sub-árbol positivo, de lo contrario se encuentra en el sub-árbol negativo  $S'_k$

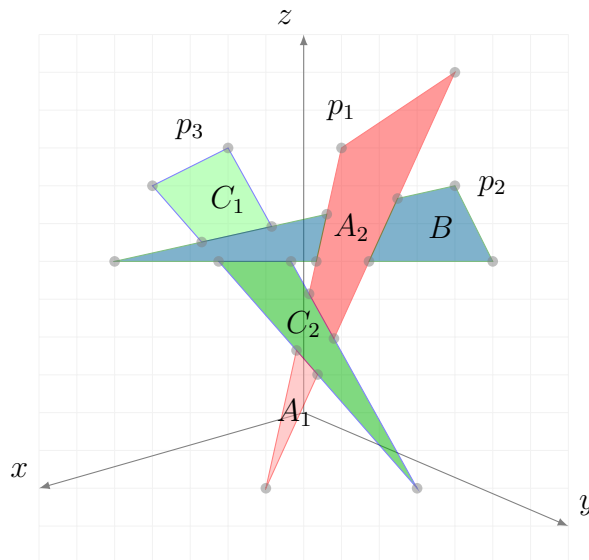


Figura 1.6: Polígonos en espacio 3D

Cada polígono en el escenario es dividido en dos, el polígono de referencia seleccionado en la Figura 1.6 es el  $B$ , el cual funge como la raíz, el árbol se muestra en la Figura 1.7.

<sup>1</sup>El concepto se puede extender a  $\mathbb{R}^n$ , en el cual se divide el espacio por hiperplanos.

## 1.1. JERARQUÍAS ESPACIALES

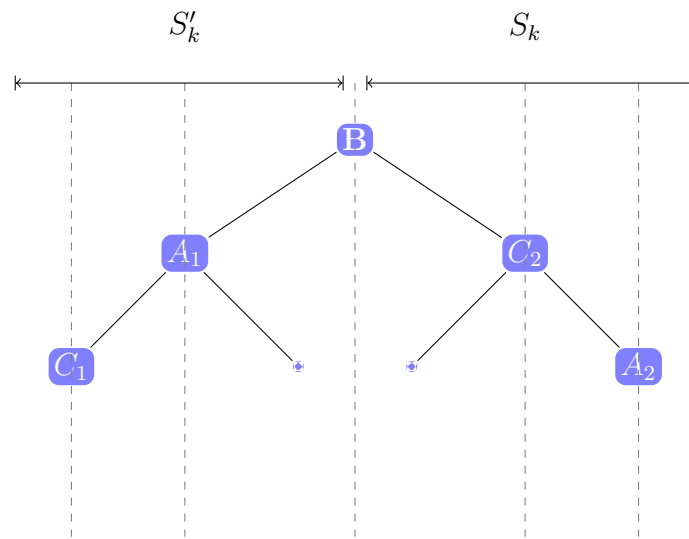


Figura 1.7: Árbol BSP

### DEFINICIÓN 1.2:

Sea  $S$  un grupo de objetos ( puntos, líneas, polígonos, grupo de polígonos o cualquier otro tipo de objetos en el espacio), y sea  $S(v)$  el conjunto de objetos asociados con el nodo  $v$ . La definición de BSP [14]  $T(S)$  es:

- Si  $|S| \leq 1$ , entonces  $T$  es una hoja  $S(v) := S$ .
- Si  $|S| > 1$ , entonces  $T$  es un nodo  $v$ ;  $v$  almacena al plano  $h_v$  y al conjunto  $S(v) := \{x \in S | x \subseteq h_v\}$ . Cada nodo  $v$  tiene dos hijos  $T^-$  y  $T^+$ . Donde  $T^-$  es  $S^- := \{x \cap h_v^- | x \in S\}$  y  $T^+$  es  $S^+ := \{x \cap h_v^+ | x \in S\}$ .

Con BSP se tiene mayor libertad de dividir el espacio en comparación con Kd-tree, el auto-particionado de BSP crea  $\Omega(n^2)$  fragmentos ( ver Figura 1.7 ).

### LEMA 1:

En un conjunto  $S$  con  $n$  planos de división, se espera fragmentar  $T(S)$  en  $O(n \log n)$ , en tiempo  $O(n^2 \log n)$

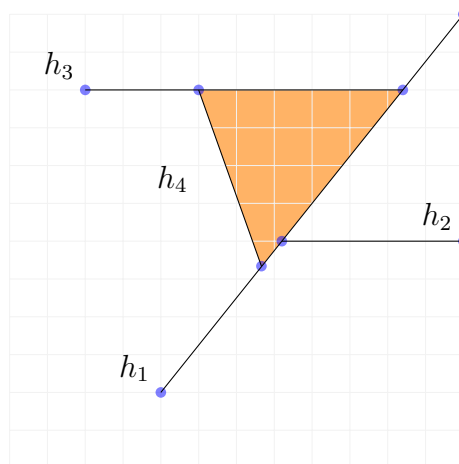


Figura 1.8: Cada línea  $h_n$  divide el espacio en dos, esto se realiza de forma recursiva.

Este enfoque no sólo divide el espacio, sino que también divide los polígonos que conforman los objetos, fragmentándose dentro de la escena ( ver Figura 1.8 ), en caso de que el polígono sea dividido una parte se encontrará en  $S_k$  y el otro en  $S'_k$ . El proceso continua hasta que quede un elemento en cada región, a esto se le denomina auto-partición.

Una vez creado el árbol BSP, se puede recorrer de dos maneras; de forma positiva donde se recorre *sub-árbol positivo*  $\rightarrow$  *nodo*  $\rightarrow$  *sub-árbol negativo*, o de forma negativa, en cuyo caso el recorrido es *sub-árbol negativo*  $\rightarrow$  *nodo*  $\rightarrow$  *sub-árbol positivo*.

BSP puede renderizar de *atrás hacia adelante*, donde primero se accede a los nodos más lejano del árbol, por lo que se muestran los polígonos más lejanos, después el nodo<sup>2</sup>, posteriormente el nodo más cercano [16]. En el renderizado de *adelante hacia atrás* se muestran los polígonos más cercanos, y posteriormente los polígonos más lejanos, así sólo los polígonos visibles son mostrados, descartando los que se encuentren en oclusión, como consecuencia se tiene un mejor rendimiento.

### 1.1.4 QUADTREE

---

Extender el concepto del árbol binario, el cual maneja datos en una dimensión, dividiéndola en dos partes, ordenándolas de acuerdo a posición relativa a estas pudiendo ser un hijo izquierdo o derecho. El *quadtree* fue desarrollado por Bentley [17], extendiendo el concepto de árbol binario a un espacio bidimensional, creando un árbol que divide el espacio en cuadrantes, reduciendo el factor de ramificación. Cada nodo de un quadtree está compuesto de siete campos: coordenada  $(x, y)$ , cuadrante ( $NW, NE, SW, SE$ ) y Nombre. El quadtree es inherentemente una estructura de datos que maneja espacios bidimensionales, como los que podemos encontrar en un mapa, donde la información se expresa por medio de coordenadas.

Un quadtree es una estructura de datos en forma de árbol cuyos nodos tienen cuatro hijos, cada nodo en el árbol corresponde a un cuadrante el cual se divide recursivamente.

- Sí  $|P| \leq 1$ , entonces el quadtree está compuesto por un solo nodo, cuyo elemento es raíz y hoja.
- En otro caso  $Q_{NE}, Q_{NW}, Q_{SE}$  y  $Q_{SW}$  denotan los cuatro cuadrantes. Donde se tiene que:

---

<sup>2</sup>Cada nodo en el árbol guarda información de un polígono, por lo que cada nodo en el árbol representa un polígono.

## 1.1. JERARQUÍAS ESPACIALES

$$\begin{aligned}
 P_{NE} &:= \{p \in P : p_x > x_{mid} \wedge p_y > y_{mid}\} \\
 P_{NW} &:= \{p \in P : p_x \leq x_{mid} \wedge p_y > y_{mid}\} \\
 P_{SW} &:= \{p \in P : p_x \leq x_{mid} \wedge p_y \leq y_{mid}\} \\
 P_{SE} &:= \{p \in P : p_x > x_{mid} \wedge p_y \leq y_{mid}\}
 \end{aligned} \tag{1.2}$$

Sea  $P$  un conjunto de puntos. La definición de un quadtree para el conjunto  $P$  dentro del cuadro  $\sigma = [x_\sigma : x'_\sigma] \times [y_\sigma : y'_\sigma]$ .

$$\begin{aligned}
 x_{mid} &= \frac{x_\sigma + x'_\sigma}{2} \\
 y_{mid} &= \frac{y_\sigma + y'_\sigma}{2}
 \end{aligned} \tag{1.3}$$

La información se organiza en forma de árbol ( ver Figura 1.9 ), en la cual cada nodo de información tiene cuatro sucesores, representados de la siguiente manera:

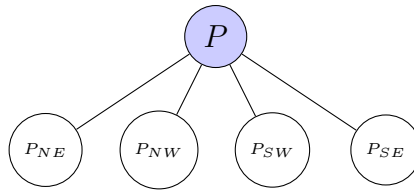
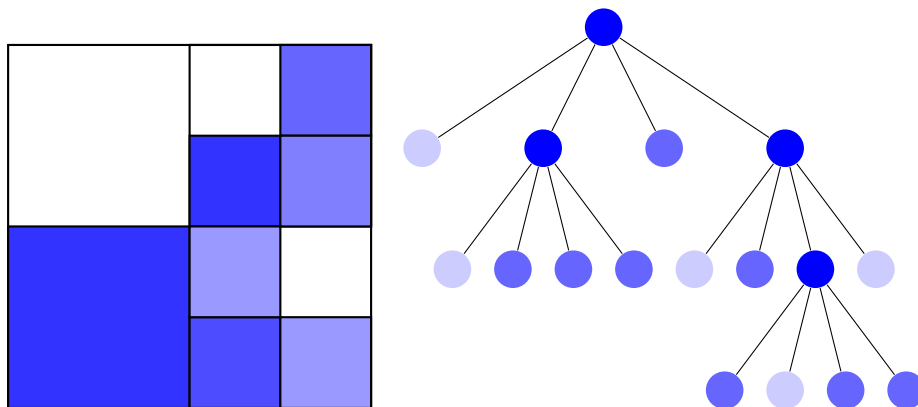


Figura 1.9: Cada nodo reserva cuatro espacios de memoria.

Sí se tiene información en un espacio de dos dimensiones, la información se encuentra dispersa dentro de este espacio finito, es necesario descartar las partes donde no se encuentra información como se muestra en la Figura 1.10.



(a) Información dispersa en un espacio bidimensional.

(b) La información es organizada en un árbol quadtree.

Figura 1.10: Representación del información en una imagen y su correspondiente quadtree.

**DEFINICIÓN 1.3:**

La profundidad de un quadtree para un conjunto  $P$  de puntos en el plano es a lo más  $\log \frac{s}{c} + \frac{2}{3}$ , donde  $c$  es la distancia más pequeña entre cualquier par de puntos en  $P$ ,  $s$  es la longitud del cuadrado inicial que contiene  $P$ . El costo de la construcción recursiva de un quadtree y su complejidad depende de la profundidad del árbol [18].

**DEFINICIÓN 1.4:**

Un quadtree con una profundidad  $d$  que almacena un conjunto de puntos  $n$  tiene  $O((d + 1)n)$  nodos y puede ser construida en un tiempo  $O((d + 1)n)$  [15].

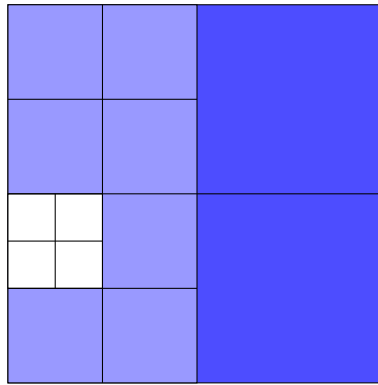


Figura 1.11: División del espacio en cuadrantes.

A medida que se subdivide el espacio se abarca información con mayor detalle, sí la información de interés está concentrada en una sola región ocasiona que el árbol se cargue de información en una rama, lo que afecta al desempeño de éste, en la Figura 1.11 se muestra un cuadrante que tiene una mayor concentración de información.

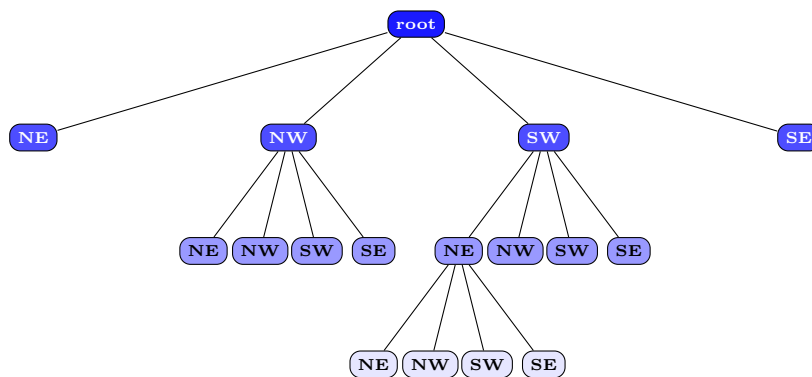


Figura 1.12: Árbol resultante de la Figura 1.11.

## 1.2. OCTREE

### 1.2 OCTREE

---

El *octree* [19] es el equivalente tridimensional del *quadtree*, solo que el *octree* divide el espacio en octantes y cada nodo tiene ocho hijos ( ver Figura 1.13 ). Este tipo de estructuras es ideal para ambientes tridimensionales estáticos, donde las posiciones de las primitivas se mantienen constantes respecto a su punto de referencia. Si se incrementa el dimensionamiento del espacio, cada nivel del árbol tiene como resultado nuevas celdas las cuales amplían la dimensión del árbol.

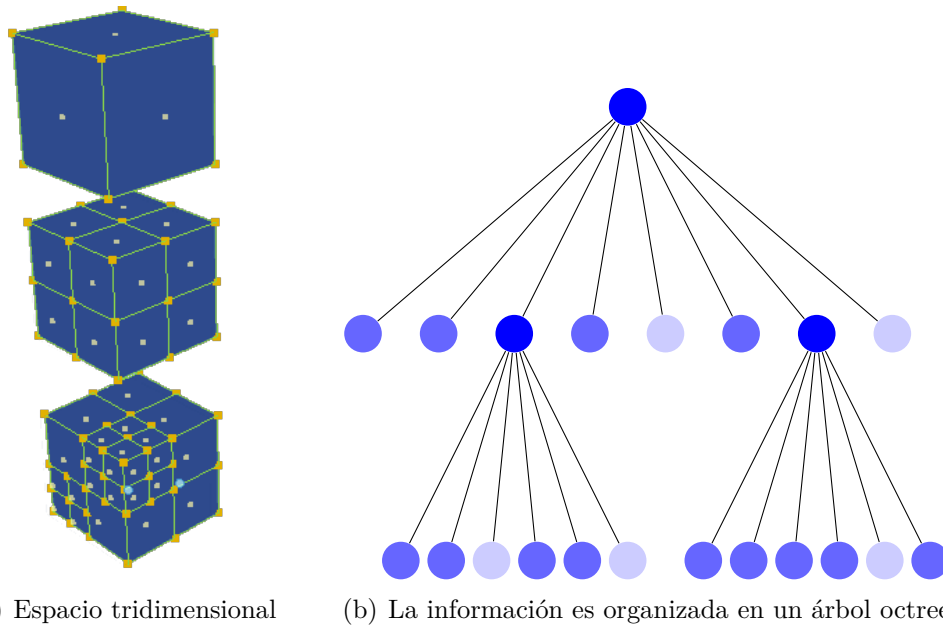


Figura 1.13: Representación de un espacio tridimensional y su respectivo árbol.

## 1.3 JERARQUÍAS DE OBJETOS

---

En las jerarquías por objetos estos son clasificados por su forma y volumen. Diseñar estructuras de datos que almacenen más que puntos es complicado, aunque su eficiencia es mayor ya que evita que información se referencie más de una vez [20]. Una estructura que pueda manejar distintos tipos de datos, tiene una ventaja al hacer una consulta ya que no da una respuesta, sino un conjunto de ellas, de las cuales se puede seleccionar el candidato idóneo.

Los objetos en el escenario son geometrías que pueden ser almacenadas y consultadas en la estructura de datos. Para almacenar en una estructura la información que compone un escenario de interés es necesario conocer cierta característica de la geometría como el volumen que ocupa, permitiendo ordenar la información de forma jerárquica [21].

1.3.1 OBJETOS ACOTADOS POR VOLUMEN

La prioridad de renderizar un objeto está dada por tres condiciones: la cercanía a la cámara, su volumen, y si entra dentro del campo de visión de ésta. La representación jerárquica descompone cada objeto recursivamente para poder ser almacenado dentro de la estructura de datos.

Conocer el volumen de un objeto y la cercanía al punto de visión es necesario para poder comparar objetos a partir del espacio que ocupa el mallado que lo compone y así poder determinar la prioridad que tienen sobre los demás. El espacio está dado por la distancia entre su punto más cercano ( $d_i^{near}$ ) al punto de referencia y el punto más alejado del objeto ( $d_i^{far}$ ), el espacio producido entre estas distancias se le denomina *slab* [3] o bloque. Una de las primeras formas para delimitar los objetos fue el uso de planos ( ver Figura 1.14 ), los diferentes conjuntos de planos producen bloques delimitadores, la intersección de los distintos bloques genera el volumen delimitador, este enfoque se puede usar tanto en  $\mathbb{R}^2$  como para  $\mathbb{R}^3$ .

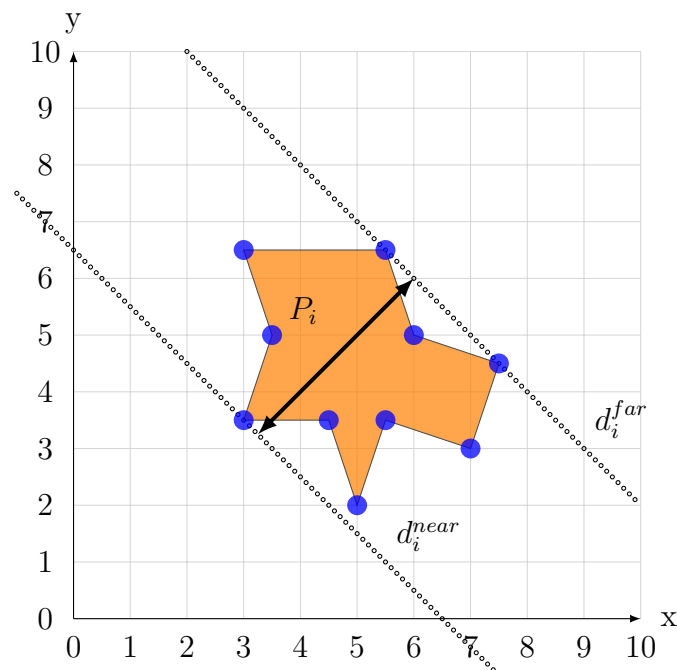


Figura 1.14: Esquema del espacio que ocupa un objeto en  $\mathbb{R}^2$ , la distancia entre  $d_i^{Near}$  y  $d_i^{Far}$  se conoce como *slab*.

El volumen acotado (*Bounding Volume*, **BV**) de un objeto permite disminuir el espacio de búsqueda de interacción rayo-objeto, sí el volumen límite no es el adecuado los rayos se pierden ya que no logran una interacción con el objeto de interés por lo tanto son cálculos desaprovechados para el resultado final.

### 1.3. JERARQUÍAS DE OBJETOS

#### 1.3.2 JERARQUÍA DE VOLÚMENES ACOTADOS

---

La jerarquía de volúmenes acotados (*Bounding Volume Hierarchy*, **BVH**) es un árbol de datos que almacena los objetos que han sido clasificados por su volumen, donde cada nodo se conecta con otros por medio de aristas. Las primitivas son almacenadas como hojas en el árbol, un grupo  $g$  está conformado por los nodos interiores. Los grupos siempre almacenan BV, mientras que las primitivas pueden almacenar ya sea BV o la geometría completa como un solo nodo, formalmente una jerarquía delimitada por volumen se define [14] como:

#### DEFINICIÓN 1.5:

Sea  $P = \{p_1, p_2, \dots, p_n\}$  un conjunto de  $n$  primitivas, y sea  $e$  un entero donde  $e > 0$ , el cual es el factor de ramificación, por tanto un BVH para  $P$  está definido como:

- Si  $|P| \leq e$  entonces se tiene un solo grupo  $g$  con  $P$  como sus nodos hijos.
- Si  $|P| > e$ , entonces se tiene un grupo  $g$  con un conjunto de nodos  $C = \{c_1, c_2, \dots, c_k\}$  como nodos hijos. Cada hijo  $c_i$  es un subconjunto  $p_i \subset P$ , la unión de todos subconjuntos componen  $P$ .

En un espacio  $\mathbb{R}^3$  regularmente se utilizan esferas, cubos, elipsoides o una combinación de éstos para ajustar tanto como sea posible los límites al volumen del objeto, con esto se intenta abarcar cualquier modelo tridimensional geométricos (objetos definidos por ecuaciones) u orgánicos (modelos sin una forma definida).

Es importante que el volumen delimitador sea un objeto simple ( ver Figura 1.15 ), esto mejora el desempeño de la intersección y da mejores resultados, además de ser sencillos de implementar [21]. Objetos comúnmente utilizados para estos propósitos son:

- **AABB** Cajas delimitadoras alineadas a los ejes (*Axis Aligned Bounding Boxes*), estas son comúnmente representadas con un par de puntos: coordenadas mínima y máxima.
- **OBB** Cajas delimitadoras orientadas (*Oriented Bounding Boxes*), éstas regularmente se extienden sobre un eje en el espacio, ya sea  $x$ ,  $y$  o  $z$
- **Esferas** Éstas están definidas por una coordenada en el espacio y un radio.
- **Cápsulas** son representadas por un vector y radios en sus extremos

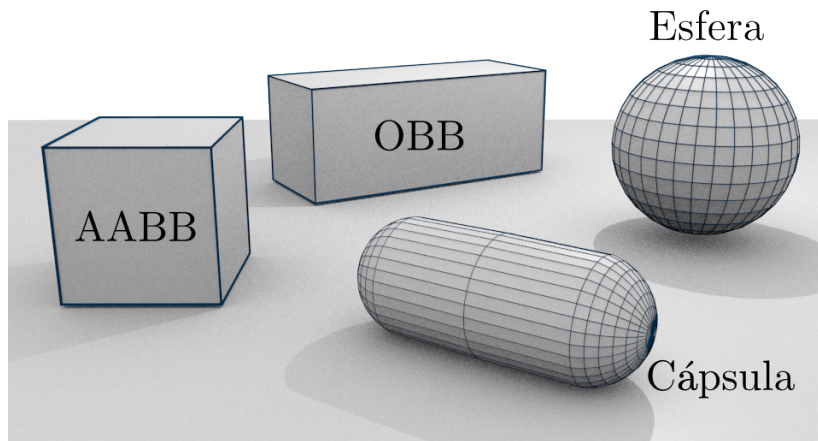


Figura 1.15: Objetos comúnmente usados como delimitadores.

Objetos que se encuentren cerca del punto de visión muestran más detalle del mallado, es por ello que se requiere subdividir el objeto de forma recursiva para poder aproximarse con mayor exactitud al volumen del mallado.

**DEFINICIÓN 1.6:**

Las primitivas cubiertas por un nodo  $n$ , en un BVH  $coverage(n)$ , son los nodos hoja que pueden ser consultados al recorrer todos los nodos de una subjerarquía con raíz en  $n$ .

Es importante que la estructura del árbol esté balanceada con respecto al volumen ( ver Figura 1.16 ), una estructura balanceada representa un alto desempeño en la consulta.

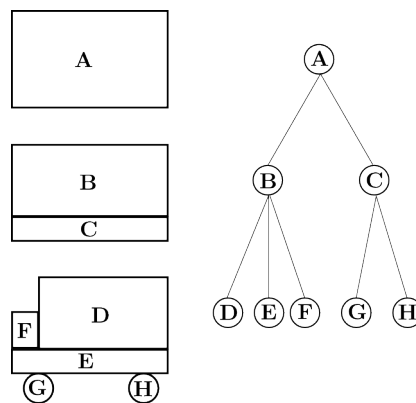


Figura 1.16: Árbol BVH.

**DEFINICIÓN 1.7:**

Sea  $B$  un objeto,  $volumen(B)$  es el volumen delimitado del objeto. Sea  $C(B)$  el conjunto de nodos hijos del nodo  $B$ , el ajuste o *tightness*  $\tau$  está definido como:

### 1.3. JERARQUÍAS DE OBJETOS

$$\tau = \frac{\text{volume}(B)}{\sum_{c \in C(B)} \text{volume}(C)} \quad (1.4)$$

Un ajuste  $\tau$  que tiende a uno entre el volumen del objeto  $B$  y la suma de los volúmenes  $C(B)$  que componen la estructura  $C$ , lo cual repercute en un traslape mínimo de los BV y los nodos contiguos, y así se evita cubrir el mismo espacio múltiples veces por la estructura.

Objetos que se encuentren con mayor lejanía a la cámara no requieren tanta subdivisión, ya que los detalles mostrados en algunos casos no son perceptibles. Generalmente se usan de cinco a seis niveles de subdivisión para cubrir escenarios complejos ( ver Figura 1.17 ), cada nivel contiene un orden de magnitud con mayor detalle, lo que incrementa el tiempo de acceso a causa de los niveles adicionales, ocasionando un aumento exponencial de subdivisión del mallado.

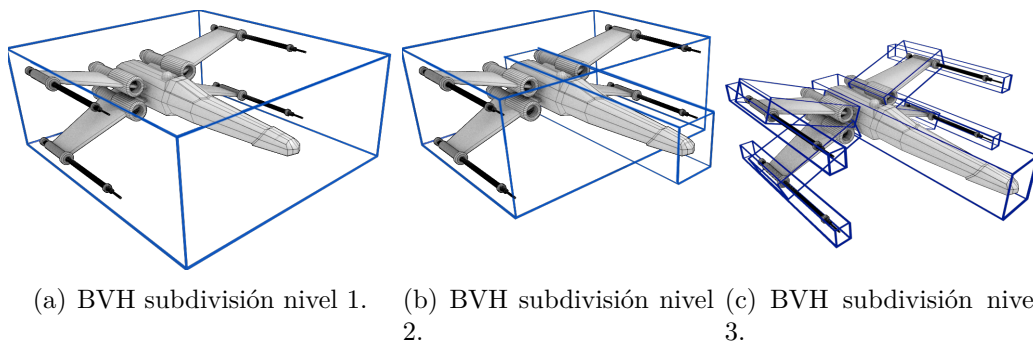


Figura 1.17: Objeto delimitado por OBB, se ejemplifica que varios volúmenes son requeridos dependiendo de la resolución deseada.

#### DEFINICIÓN 1.8:

El costo de la jerarquía está dada por el conjunto de nodos hijo de  $n$  multiplicado por el volumen que ocupa cada uno de esos nodos:

$$\text{cost}(H) = \sum_{n \in H} \cdot \text{volume}(n) \quad (1.5)$$

En este tipo de estructuras se desea que sea corta y amplia, de lo contrario se vuelve lenta. Esto depende del factor de ramificación así como de los BV usados. Muchas de las estructuras de datos BVH se orientan específicamente a un tipo de objeto, y así un árbol está conformado por sólo OBB, lo que no se ajusta apropiadamente a superficies curvas, por tanto el ajuste no es el adecuado, esto provoca una mayor subdivisión del objeto, lo que a su vez provoca un mayor consumo de memoria.

1.3.3 JERARQUÍAS HÍBRIDAS

Las jerarquías híbridas son estructuras de datos que hacen uso de distintos tipos de BV sobre un mismo árbol, por ello se puede encontrar conformado por AABB, OBB, esferas y cápsulas. Los árboles de jerarquías híbridas de volúmenes acotados (*Hybrid Bounding Volume Hierarchy*, **HBVH**) [22] tienen un mayor factor de ramificación al hacer uso de múltiples geometrías que en conjunto se aproximan mejor al objeto de interés que contiene superficies planas y curvas. El tipo de combinación comúnmente usada es AABB y esferas como se puede apreciar en la Figura 1.18. El tiempo de ejecución de un HBVH es de  $O(n \log^2 n)$ , donde  $n$  es el número de primitivas.

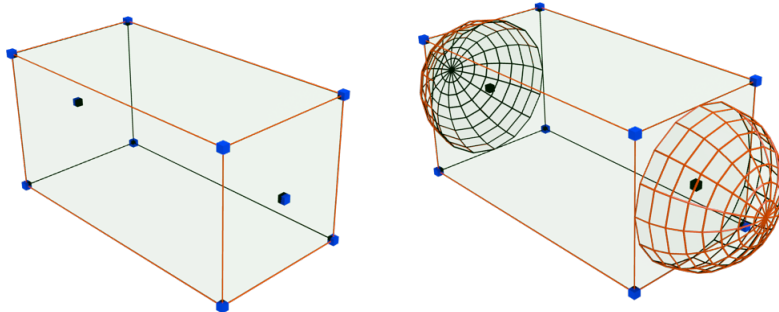
Calcular la coordenada centro de los extremos de un AABB se realiza de la siguiente forma [23]:

$$O_i = \begin{cases} x = \frac{x_a + x_b}{2} \\ y = \frac{y_a + y_b}{2} \\ z = \frac{z_a + z_b}{2} \end{cases} \quad (1.6)$$

Con ésto se pueden calcular los  $R_x$ ,  $R_y$  y  $R_z$  los cuales son tres radios del AABB.

$$\begin{cases} R_x = \frac{1}{2} \cdot [(Max_{(x_A, x_B)} + R) - (Min_{(x_A, x_B)} - R)] \\ R_y = \frac{1}{2} \cdot [(Max_{(y_A, y_B)} + R) - (Min_{(y_A, y_B)} - R)] \\ R_z = \frac{1}{2} \cdot [(Max_{(z_A, z_B)} + R) - (Min_{(z_A, z_B)} - R)] \end{cases} \quad (1.7)$$

Se establece  $A(X_A, Y_A, Z_A)$  y  $A(X_B, Y_B, Z_B)$  como los extremos del AABB y en cuya posición se crearán las esferas delimitadoras, el cual da como resultado:



(a) Caja delimitadora orientada. (b) Esferas en los extremos de AABB.

Figura 1.18: Combinación de AABB y esferas.

Los HBVH al hacer uso de esferas reduce la necesidad de subdividir la superficie

### 1.3. JERARQUÍAS DE OBJETOS

curva en AABB cada vez más pequeños, por tanto los niveles del árbol son menores. La desventaja de este tipo de estructura es que su construcción depende del orden de inserción de los nodos, teniendo una alta probabilidad de que el árbol no quede balanceado.

La construcción de la estructura se puede hacer de arriba-abajo o de abajo-arriba.

- Arriba-abajo: El BV se establece alrededor del conjunto de primitivas, posteriormente se divide este conjunto en dos o más subconjuntos usando un plano de división, este proceso se repite sobre los subconjuntos deteniéndose hasta que se encuentre la condición de paro.
- Abajo-arriba: Aquí los objetos se fusionan como un grupo, el algoritmo se detiene cuando sólo queda un grupo siendo el último grupo la raíz del árbol.

#### 1.3.4 HEURÍSTICAS BASADAS EN ÁREAS DE SUPERFICIE

Tanto las estructuras de división espacial como las estructuras de división jerárquica comparten un problema en común. Dependiendo del criterio de almacenamiento, un objeto puede estar almacenado más de una vez o puede no estar cubierto en su totalidad, a esto se le conoce como fragmentación. Las heurísticas basadas en el área de superficie (Surface Area Heuristic) [24] son utilizadas en raytracing para estimar la probabilidad de que un rayo interactúe con un objeto  $B$  dentro de un espacio  $A$  correspondiente a la escena. Distribuciones de probabilidad que tengan una rápida convergencia permiten reducir el tiempo de procesamiento.

El uso común de las estructuras jerárquicas es la subdivisión de los objetos de manera recursiva, los árboles pueden ser compuestos por un solo tipo de objeto delimitador o una combinación varios, donde el nodo raíz corresponde al escenario completo. El costo de un nodo interior  $C_i$ , es la probabilidad de interacción de un rayo en la superficie de un nodo hijo  $SA(v')$  sobre la probabilidad de la superficie del nodo padre  $SA(v)$  ( ver Figura 1.19 ).

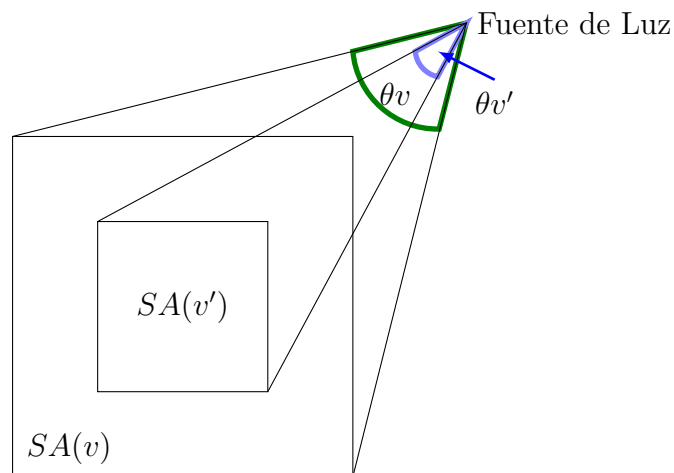


Figura 1.19: Esquema de superficies de SAH.

## CAPÍTULO 1. ESTRUCTURAS UTILIZADAS EN EL RAYTRACING

Los objetos en el escenario han sido ordenados por una estructura de datos, para así calcular la probabilidad de interacción de los rayos sobre cada una de las superficies de los objetos que componen el escenario, por ello es necesario visitar el objeto hijo dentro del nodo padre, asumiendo que todos los rayos tengan la misma probabilidad de que el rayo  $S$  golpee al nodo hijo  $v'$  que se encuentra dentro del nodo padre, el costo de total  $C_t$  es:

$$C_i = P(SA(v')|SA(v)) = \frac{\theta v'}{\theta v} \quad (1.8)$$

Para calcular el costo total del árbol es necesario calcular el costo de cada uno de sus componentes, la cual es la suma del número de nodos interiores  $N_i$ , número de hojas  $N_l$ , número de objetos almacenadas en la hoja  $N(l)$  está dada por:

$$C_t = \frac{1}{SA(v)} \left( C_i \cdot \sum_{i=1}^{N_i} SA(i) + C_l \cdot \sum_{l=1}^{N_l} SA(l) + C_o \cdot \sum_{l=1}^{N_l} SA(l) \cdot N(l) \right) \quad (1.9)$$

Donde:

- $N_i$       Número de nodos interiores.
- $N_l$       Número de hojas.
- $N(l)$     Número de objetos almacenados en la hoja  $l$ .
- $SA(i)$     Área de la superficie de los nodos interiores.
- $SA(l)$     Área de la superficie de los nodos hoja.

La interacción entre la superficie de cada uno de los objeto que se encuentra dentro del escenario ( $N_i$ ) y la superficie total ( $SA(v)$ ) es el costo de lanzar rayos sobre la superficie total del escenario, la cual es calculada por la ecuación 1.9, suponiendo que la distribución de estos rayos se dé uniformemente. La interacción de la luz dado en un punto en particular de la superficie del objeto, forma un ángulo entre la fuente de luz y la cámara, es necesario conocer la interacción entre éstos tres elementos, como la incidencia de la luz sobre un área en particular, la proyección de energía a causa de esta interacción.

# Capítulo

# 2

## MANEJO DE MATERIALES SOBRE LAS SUPERFICIES

Los materiales tienen distintos tipos de comportamientos dependiendo de sus propiedades y la composición de su superficie, pudiendo reflejar la luz total o parcialmente, en otros casos atravesar dicho material. Éste tipo de características se calculan con las leyes de reflexión y refracción; para calcular la luz incidente en la superficie del objeto es necesaria conocer cómo es proyectada y los cálculos involucrados para lograr el efecto deseado.

### 2.1 LONGITUD DE ARCO

La longitud del arco [25]  $ds$  está dada por la ecuación ??, donde  $r$  es el radio y  $d\theta$  es el ángulo de apertura entre  $A$  y  $B$ , como podemos ver en la Figura ??.

$$ds = r \cdot d\theta \quad (2.1)$$

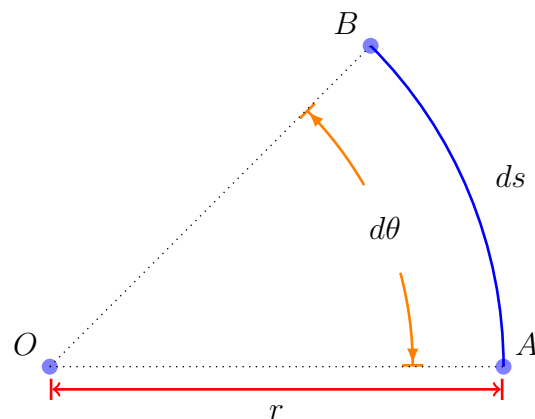


Figura 2.1: Longitud de arco entre el punto  $A$  y  $B$  con origen en  $O$ .

## 2.2 ÁNGULO SÓLIDO

El *ángulo sólido* [26] también conocido como *esteriorradián* es la representación tridimensional de la longitud de arco, es el volumen que ocupa  $d\omega$  ( ecuación ?? ), el cual se determina dividiendo el área de la superficie que ocupa la sección de la esfera  $dA$  por el cuadrado del radio de la punta a la base ( ver Figura ?? ).

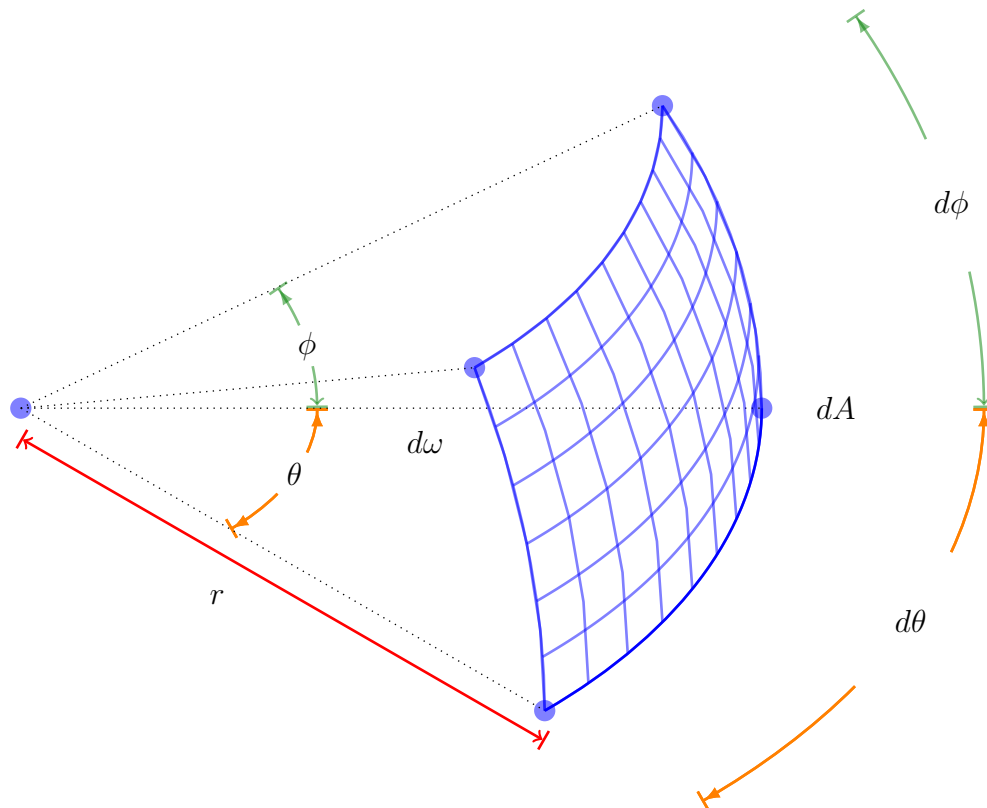


Figura 2.2: Ángulo sólido.

El área de  $dA$  se calcula con la ecuación ??.

$$dA = r d\theta \cdot r \sin \theta \cdot d\phi \quad (2.2)$$

Una hemiesfera tiene una superficie  $2\pi r^2$  y su ángulo sólido se subtiende<sup>1</sup> en  $2\pi$  estereorradianes( $sr$ ) cuyo centro es la base con  $r = 1$ . El ángulo sólido desde el centro a la base es igual al área de la hemiesfera unitaria, la dirección es medida por los ángulos cenital  $\theta$  y azimutal  $\phi$  sobre el plano ( ver Figura ?? ). La posición angular para  $\phi = 0$  es arbitraria, pero usualmente se mide desde el eje  $x$ . El ángulo sólido de cualquier superficie  $dA$  es igual a la intersección del área en la hemiesfera unitaria, cuyo elemento se calcula :

<sup>1</sup>R.A.E.

Subtender: Del lat. subtendĕre ‘extender por debajo’, tr. Geom. Unir con una línea recta los extremos de un arco de curva o de una línea quebrada.

### 2.3. INTENSIDAD LUMINOSA

$$d\omega = \frac{dA}{r^2} = \frac{r d\theta \cdot r \sin \theta \cdot d\phi}{r^2} = d\theta \sin \theta \cdot d\phi \quad (2.3)$$

La integral de  $d\Omega$  se refiere a los ángulos alrededor de la hemiesfera es  $2\pi \cdot d\omega$ .

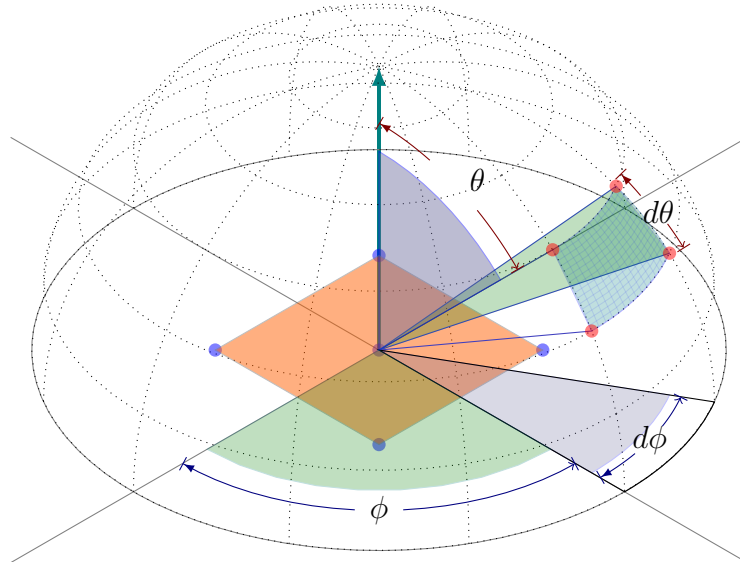


Figura 2.3: Ángulos involucrados en el cálculo de la hemiesfera.

La hemiesfera expresada en integrales es:

$$2\pi = \int_{4\pi} d\Omega = \int_{\theta=0}^{\frac{\pi}{2}} \int_{\phi=0}^{2\pi} d\theta \sin \theta d\phi \quad (2.4)$$

De esta forma se puede calcular la proyección del ángulo sólido sobre una esfera con radio  $r$ . Trabajar en un espacio tridimensional permite entender mejor el comportamiento de modelos como es el flujo luminoso, refracción, reflejo, entre otros.

### 2.3 INTENSIDAD LUMINOSA

---

El intensidad radiante ( $\Phi$ ) emitido por unidad de ángulo sólido ( $\omega$ ) por una fuente de luz puntual (ecuación ??), es conocido como intensidad radiante [26].

$$I = \frac{d\Phi}{d\omega} \quad (2.5)$$

Una fuente de luz puntual proyecta una intensidad radiante  $I$  por área de ángulo sólido  $d\Omega$ , la potencia radiante [25] está dada por la ecuación ??.

$$\Phi = \int I d\Omega \quad (2.6)$$

## CAPÍTULO 2. MANEJO DE MATERIALES SOBRE LAS SUPERFICIES

El reflejo de la radiación incidente  $N_i(\Psi)$  es reflejado en dirección  $N_r(\theta, \phi)$ , el resultado de este reflejo es:

$$dN_r = (\Psi; \theta, \phi) \quad (2.7)$$

La dirección de la luz reflejada requiere de dos ángulos; el ángulo *cenit*  $\theta$  y el ángulo *azimut*<sup>2</sup>  $\phi$ . El *ángulo de reflexión* se denota como  $(\Psi; \theta, \phi)$ , donde como se mencionó anteriormente  $\Psi$  es el ángulo de incidencia del rayo de luz, y  $(\theta, \phi)$  son los ángulos con los que el rayo abandona la superficie de contacto.

La radiación  $N$  es el flujo radiante  $\Phi$  por unidad de ángulo sólido, en dirección  $\theta$  ó  $\Psi$  por unidad de área de proyección  $A$  perpendicular al rayo de incidencia. La superficie de un objeto tiene determinada cantidad de radiación incidente  $N$  la cual es inducida por una fuente de luz. El resplandor de la superficie se mide como el flujo o cantidad de radiación  $\Phi$  emitida por el ángulo sólido  $\omega$  en dirección  $\cos \theta$  ó  $\cos \Psi$  perpendicular al rayo. Por tanto el flujo de radiación emitida por un rayo en algún punto arbitrario es:

$$d\Phi_i(\Psi) = N_i(\Psi) \cos \Psi dA d\omega_i \quad (2.8)$$

Donde  $d\omega_i$  es el ángulo sólido de la fuente de luz cuando es visualizada por  $dA$ , la radiación reflejada por  $dA$  en dirección  $(\theta, \phi)$  se designa como  $dN_r(\Psi; \theta, \phi)$ , para pequeñas superficies se calcula la integral de  $dN_r(\Psi; \theta, \phi)$

$$d\Phi_r(\Psi; \theta, \phi) = dN_r(\Psi; \theta, \phi) \cos \theta dA d\omega_r \quad (2.9)$$

La función de reflectancia se define como la radiación reflejada  $dN_r$  en dirección  $(\theta, \phi)$ , dividida por el flujo de *radiación incidente* por la superficie del área<sup>3</sup> ( $d\Phi_i/dA$ ).

$$E(\Psi) = \frac{d\Phi_i(\theta_i, \phi_i)}{dA} \quad (2.10)$$

$$\rho(\Psi; \theta, \phi) = \frac{dN_r(\Psi; \theta, \phi)}{dE(\Psi)} = \frac{dN_r(\Psi; \theta, \phi)}{N_i(\Psi) \cos \Psi d\omega_i} \quad (2.11)$$

La ecuación de reflectancia bidireccional ?? relaciona las ecuaciones ?? y ??, también conocida como *reflectancia biangular*, la cual denota la dependencia de iluminación incidente y la dirección de reflectancia.

### 2.4 FUNCIÓN DE REFLECTANCIA BIDIRECCIONAL

La *función de distribución reflectancia bidireccional* (Bidirectional reflectance distribution function, **BRDF**) [27] calcula la cantidad de radiación que es reflejada por una superficie en un punto específico en determinada dirección. La designación de

<sup>2</sup>se puede emplear indistintamente **acimut** ó **azimut** ya que tienen el mismo significado

<sup>3</sup>En algunos textos se puede encontrar  $\Psi$  para referirse a los ángulos de incidencia  $(\theta_i, \phi_i)$

## 2.4. FUNCIÓN DE REFLECTANCIA BIDIRECCIONAL

*bidireccional* se refiere a que dos ángulos están involucrados, el ángulo de incidencia  $\Psi$  sobre la superficie de interés y el ángulo de salida con la que es reflejada dicha radiación, éste reflejo está compuesto por dos valores  $\theta$  y  $\phi$ . En la Figura ?? se ilustra un elemento de la superficie del objeto, la cual se conoce como primitiva del objeto.  $\phi$  es el ángulo de entrada, mientras que  $\theta$  es el ángulo de salida, es decir, el reflejo que produce la superficie.

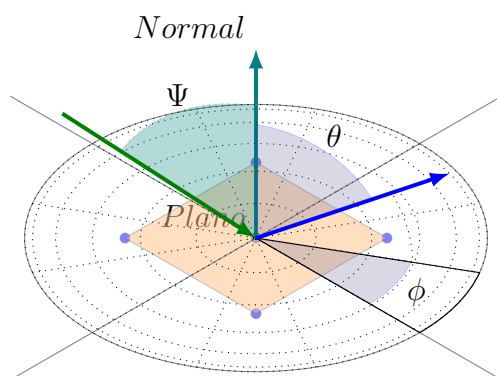
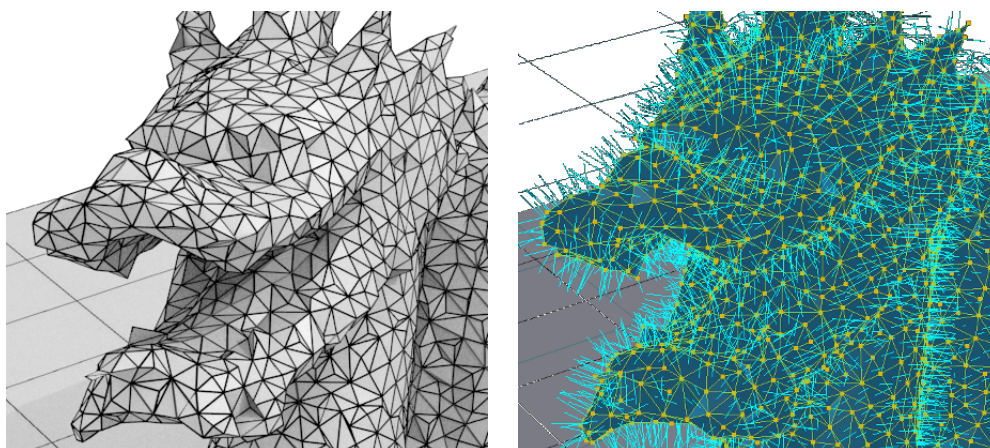


Figura 2.4: Vector normal del plano de la superficie.

$$\rho(\Psi; \theta, \phi) = \frac{dN_r(\Psi; \theta, \phi)}{N_i(\Psi) \cos \Psi d\omega_i} \quad (2.12)$$

El BRDF es ampliamente utilizado para describir el comportamiento de la luz que incide sobre la superficie de los objetos. Tomando en cuenta que los objetos tridimensionales están compuestos por triángulos ( ver Figura ?? ), los cuales se pueden visualizar individualmente como planos, cada plano tiene un vector normal, el cual es fundamental para calcular los rayos de incidencia y el reflejo de estos.



(a) Objeto tridimensional compuesto por triángulos (b) Visualización de las normales de cada uno de los triángulos que componen al objeto

Figura 2.5: Objetos 3D compuestos por triángulos.

## CAPÍTULO 2. MANEJO DE MATERIALES SOBRE LAS SUPERFICIES

Los efectos de iluminación dependen del material con el que estén compuestos los objetos en la escena. Los materiales reflejan de distintas formas la luz, el raytracing haciendo uso de la reflectancia bidireccional simula distintos comportamientos de estos materiales. La función de reflectancia bidireccional hace uso del ángulo sólido para calcular la cantidad de radiación que es reflejada en determinada dirección.

### 2.5 LEY DE REFLEXIÓN

---

La ley de reflexión dice que cuando un rayo incide sobre una superficie de un medio uniforme, la luz es reflejada por el plano de interés, el ángulo de reflexión  $\theta_r$  es igual al ángulo de incidencia  $\theta_i$  ( ecuación ?? ), medido desde la normal del plano en el punto de incidencia ( ver Figura ?? ).

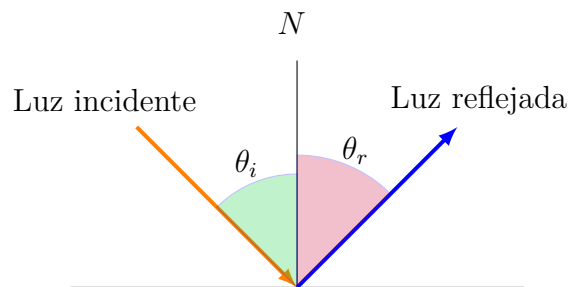


Figura 2.6: Relación entre la luz entrante y la reflejada

$$\theta_i = \theta_r \quad (2.13)$$

La reflexión es una propiedad de los materiales con superficies altamente pulidas cuya textura es de espejo, donde se puede apreciar el reflejo de los objetos que se encuentran alrededor.

### 2.6 LEY DE SNELL

---

La ley de Snell permite calcular tanto el reflejo como la desviación de la luz al atravesar un material, a esto se le conoce como refracción incidente o índice de refracción (IR), calculándose la desviación de la luz desde la normal del plano en el punto de incidencia. El seno del ángulo de refracción es directamente proporcional al seno del ángulo de incidencia.

Esta propiedad se puede ver propiamente en los objetos transparentes como: agua, cristal, incluso sobre gases debido a que la luz cambia su ángulo de dirección cuando

## 2.7. LEY DE FRESNEL

atraviesa un medio. La refracción de la luz se calcula con la ecuación ??.

El ángulo de refracción  $\theta_t$  depende de la velocidad con la cual la luz atraviesa dos medios  $n_i$  y su ángulo de incidencia ( ver Figura ?? ), a está relación se le conoce como *ley de Snell*.

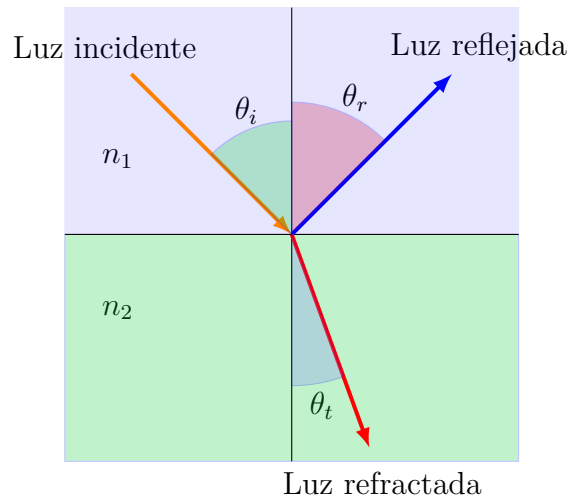


Figura 2.7: Relación del ángulo incidencia y el de refracción.

$$n_1 \sin \theta_i = n_2 \sin \theta_t \quad (2.14)$$

Cuando los medios por los que viaja la luz son distintos, esta es desviada por las distintas densidades del medio en el que viaja, por lo que también tienen un índice de refracción (IR)<sup>4</sup> distinta, si los materiales con los que el haz de luz interactúa tienen el mismo índice de refracción, significa que son el mismo tipo de material, efectos ópticos interesantes se pueden ver cuando los objetos tienen un IR semejante.

$$\frac{\sin \theta_i}{\sin \theta_t} = \frac{n_1}{n_2} = n \quad (2.15)$$

Si  $n_t > n_i$  entonces  $\theta_t < \theta_i$ , la luz entra a un medio en donde su velocidad disminuye, entonces el rayo se dobla hacia la normal. Si  $n_t < n_i$ , entonces  $\theta_t > \theta_i$ , el haz de luz se dobla alejándose de la normal. Otros materiales reflejan o transmiten la luz de forma difusa, por lo que la ley de reflexión y de refracción no pueden modelar su comportamiento de forma precisa.

## 2.7 LEY DE FRESNEL

La ley de Fresnel describe la fracción de la luz incidente reflejada o transmitida sobre la superficie del material, la cantidad de luz perceptible no sólo depende del índice de refracción y el cambio del ángulo de incidencia, sino que también de la

<sup>4</sup>En inglés se encuentra como *index of refraction* (IOR)

## CAPÍTULO 2. MANEJO DE MATERIALES SOBRE LAS SUPERFICIES

polarización de la luz incidente.

El índice de refracción relativo  $n \equiv n_1/n_2$ , el índice de refracción y el ángulo de transmisión se relacionan con la ley de Snell,  $\sin \theta = n \sin \theta_t$ . Una parte de la luz incidente es reflejada y la otra parte transmitida, quedando la ecuación en términos sólo del ángulo de incidencia y un índice de refracción, las ecuaciones de reflexión eléctrica y magnética [26].

$$\begin{aligned}
 R = r^2 &= \frac{n_1 \cos \theta_i - n_2 \cos \theta_t}{n_1 \cos \theta_i + n_2 \cos \theta_t} = \frac{\cos \theta_i - n \cos \theta_t}{\cos \theta_i + n \cos \theta_t} \\
 &= \frac{\cos \theta - \sqrt{n^2 - \sin^2 \theta}}{\cos \theta + \sqrt{n^2 - \sin^2 \theta}}
 \end{aligned} \tag{2.16}$$

$$\begin{aligned}
 TM : r &= \frac{n_2 \cos \theta_i - n_1 \cos \theta_t}{n_2 \cos \theta_i + n_1 \cos \theta_t} = \frac{n \cos \theta - \cos \theta_t}{n \cos \theta + \cos \theta_t} \\
 &= \frac{n^2 \cos \theta - \sqrt{n^2 - \sin^2 \theta}}{n^2 \cos \theta + \sqrt{n^2 - \sin^2 \theta}}
 \end{aligned} \tag{2.17}$$

Las ecuaciones anteriores se pueden simplificar para obtener el porcentaje de refracción como:

$$R = \left( \frac{n_1 - n_2}{n_1 + n_2} \right)^2 \tag{2.18}$$

Ecuación de transmisión eléctrica por:

$$\begin{aligned}
 TE : t &= \frac{2n_1 \cos \theta_i}{n_1 \cos \theta_i + n_2 \cos \theta_t} = \frac{2 \cos \theta}{\cos \theta + n \cos \theta} \\
 &= \frac{2 \cos \theta}{\cos \theta + \sqrt{n^2 - \sin^2 \theta}}
 \end{aligned} \tag{2.19}$$

Ecuación de transmisión magnética:

$$\begin{aligned}
 TM : t &= \frac{2n_i \cos \theta_i}{n_2 \cos \theta_i + n_1 \cos \theta_t} = \frac{2n \cos \theta}{n^2 \cos \theta + n \cos \theta} \\
 &= \frac{2n \cos \theta}{n^2 \cos \theta + \sqrt{n^2 - \sin^2 \theta}}
 \end{aligned} \tag{2.20}$$

## 2.8. SUPERFICIES LAMBERTIANAS

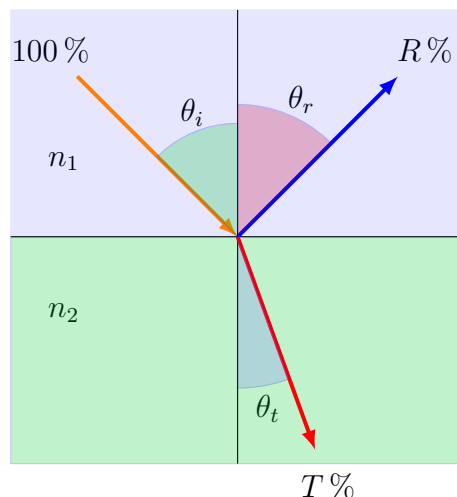


Figura 2.8: Refracción de la luz.

Simplificando el factor de transmisión:

$$T = 1 - \left( \frac{n_1 - n_2}{n_1 + n_2} \right)^2 = \frac{4n_1n_2}{(n_1 + n_2)^2} \quad (2.21)$$

Un medio transparente transmite la mayor parte de la luz incidente; sin embargo parte de ésta luz escapa creando un efecto de reflexión-refracción, este efecto crea las cáusticas en los materiales cristalinos, como es el agua, miel, cristal, etc., incluso se puede apreciar en la interacción del aire a distintas temperaturas. En el caso de las superficies altamente pulidas, el reflejo se acentúa en determinado ángulo, el cual se calcula con la ley del coseno de Lambert.

## 2.8 SUPERFICIES LAMBERTIANAS

---

Las superficies altamente pulidas como espejos y cristales reflejan la luz directamente, son conocidas como superficies lambertianas, ya que su brillo es independiente de la dirección de visualización.

### 2.8.1 LEY DEL COSENO DE LAMBERT

---

En los gráficos por computadora se busca generar materiales los cuales simulen características ópticas del mundo real, el renderizado basado en física se basa en simular estos materiales los cuales están clasificados generalmente en dos categorías: superficies de reflectancia, también conocidas como superficies lambertianas y superficies de dispersión, conocidos como superficies no-lambertianas.

La intensidad de la luz incidente sobre una superficie decrece cuando el ángulo de incidencia se incrementa respecto a la normal (cuando el ángulo tiende a 90 grados), la intensidad de la luz se incrementa cuando el ángulo tiende a cero, está carac-

terística se le conoce como la ley de Lambert.

**Ley del coseno de Lambert:** la luz incidente sobre una superficie es directamente proporcional al coseno del ángulo de incidencia entre la dirección de la fuente de luz y la normal de la superficie de incidencia [26].

Las superficies lambertianas reflejan la misma cantidad de energía en todas direcciones. Así una superficie que obedezca a esta característica se le conoce como superficie lambertiana, por tanto se dice que tiene una reflectancia lambertiana.

La radiación  $L_\lambda$  incidente es constante. La radiación lambertiana depende del coseno del ángulo de incidencia sobre una superficie  $dA$ .

$$L_\lambda(\Psi; \theta, \phi) = L_\lambda \cos \theta \quad (2.22)$$

El comportamiento de una superficie lambertiana refleja la luz con mayor o menor intensidad dependiendo del ángulo que se este visualizando ( ver Figura ?? ), el valor máximo de luz reflejada es en 0 grados respecto a la normal de la superficie y su punto más bajo es en los 90 grados.

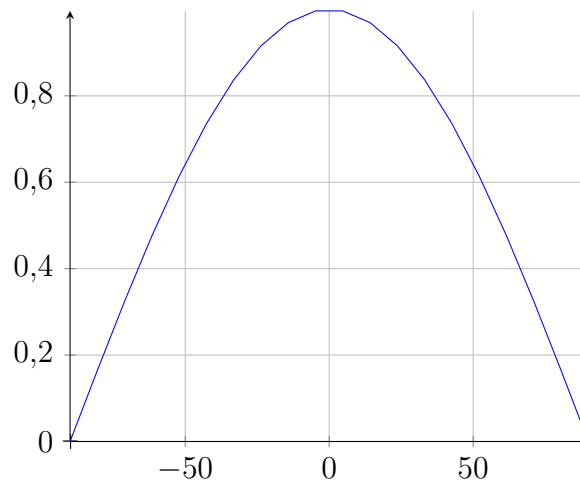


Figura 2.9: Valor máximo de luz reflejada es en 0 grados respecto a la normal de la superficie.

La luz incidente sobre una superficie lambertiana ( ver Figura ?? ) puede ser reflejada; para determinar la potencia con la que esta energía abandona dicha superficie  $dA$  es necesario tomar en cuenta el ángulo  $d\omega$  con el que es proyectada [25]. La intensidad de la luz se incrementara cuanto el ángulo de la fuente de luz y la normal de la superficie  $dA$  tiendan a cero.

## 2.8. SUPERFICIES LAMBERTIANAS

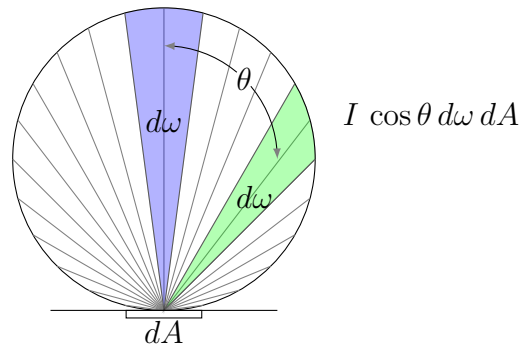


Figura 2.10: Esquema de una superficie lambertiana.

La integración del ángulo sólido.

$$\int_0^{\pi/2} I \cos \theta d\omega \quad (2.23)$$

Una superficie lambertiana refleja la luz incidente en todas direcciones sobre la superficie, la potencia emitida dentro de una hemiesfera es calculada por:

$$\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} I \cos \theta \sin \theta d\omega d\phi \quad (2.24)$$

Gran parte de los materiales reflejan con mayor intensidad la luz cuando el ángulo incidente tiende a alinearse con la normal, estos materiales tienen un comportamiento que se puede modelar con la ecuación:

$$I \cos^n \theta \quad (2.25)$$

El cual para una superficie de la hemiesfera se tiene la ecuación ??, donde  $n$  expresa la característica especular del material como los metales, los cuales tienen un valor especular alto.

$$\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} I \cos^n \theta \sin \theta d\omega d\phi \quad (2.26)$$

La iluminación de cada cara depende del ángulo de incidencia con respecto a la normal de la superficie, por tanto cada cara es iluminada de manera uniforme, en los primeros gráficos por computadora se lograba apreciar un fuerte reflejo en los polígonos, quedando el polígono con un color uniforme, y dando un efecto de que el objeto está conformado por cuadrados.

### 2.8.2 SUPERFICIES NO LAMBERTIANAS

Las superficies rugosas conocidas como superficies **No Lambertianas** están constituidas por micro-cavidades que reflejan la luz en distintos ángulos, y en muchos

casos no están dispuestos uniformemente. Cuando una gran cantidad de estas micro-superficies conocidas como facetas inciden sobre el un mismo pixel la intensidad lumínica se incrementa con el ángulo de incidencia.

Una superficie  $dA$  que contiene una colección de estas micro-superficies, las cuales reflejan la luz en un ángulo sólido  $d\omega_r$ , cada una de estas se inclinaran en un ángulo  $\alpha$  respecto a la superficie ( ver Figura ?? ). Sea  $Pd\omega_r$ , el número de facetas por unidad de superficie, cuyas normales están contenidas en  $d\omega'$  para superficies isotrópicas las normales tienen una distribución normal. El modelo de reflectancia de Torrance y Sparrow [28] asume que está distribución es uniforme y simétrica.

$$P = P(\alpha) = b \exp(-c^2 \alpha^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.27)$$

Por tanto, calcular la luz reflejada por la superficie  $dA$  con ángulo sólido  $d\omega$  y distribución normal de facetas que están orientadas en un ángulo  $\alpha$  hacia el pixel con un promedio normal de la superficie  $P(\alpha)$  es:

$$f P(\alpha) d\omega dA \quad (2.28)$$

La superficie principal  $dA$  está constituida por pequeñas superficies, las cuáles forman cavidades, éstas están constituidas por dos facetas adyacentes.

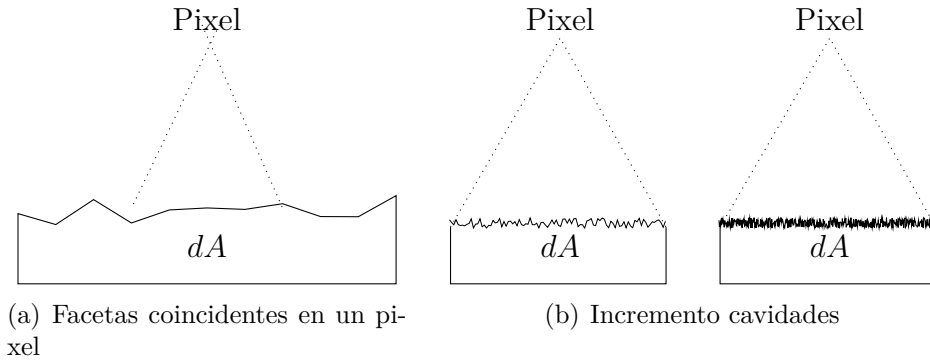


Figura 2.11: Micro-superficies sobre una superficie mayor.

Para calcular la cantidad de luz que refleja por la superficie respecto a la fuente de luz, es necesario tomar en cuenta el ángulo de incidencia, como se ha mencionado anteriormente, el ángulo de incidencia de la luz sobre esta superficie es  $\cos \Psi$ , de la ecuación ?? se obtiene la luz incidente sobre la micro-superficies calculadas por la ecuación ??.

$$f \cos \Psi P(\alpha) d\omega dA \quad (2.29)$$

$$d\Phi_i = f N_i \cos \Psi' P(\alpha) d\omega' dA d\omega_i \quad (2.30)$$

## 2.8. SUPERFICIES LAMBERTIANAS

Debido a que una fracción de la luz incidente es reflejada por la superficie de cada una de las facetas, la superficie no refleja perfectamente la luz.

### 2.8.3 FACTOR GEOMÉTRICO DE ATENUACIÓN

Las superficies rugosas están compuestas por micro-cavidades, cuando todas las cavidades son iluminadas y visibles por grandes ángulos de incidencia y reflexión se producen dos fenómenos conocidos como *sombreado* y *enmascaramiento* los cuales repercuten en la cantidad total de luz reflejada por la superficie [29].

El sombreado se refiere a que las facetas dentro de la cavidad son parcialmente iluminada, causada por la faceta adyacente la cual ocluye la luz sobre la faceta de interés. El enmascaramiento se da cuando las facetas dentro de la cavidad son parcialmente visibles, debido a que la faceta adyacente la ocluye. Ambos efectos de iluminación deben ser tomados en cuenta al calcular la energía total proyectada por la superficie, a esta interacción se le denomina *factor geométrico de atenuación* o simplemente conocido como factor geométrico [30].

Cuando un observador se mueve alrededor de la fuente de luz que incide sobre el objeto, este la refleja dependiendo de como estén distribuidas las cavidades sobre su superficie, creando un efecto de incremento de brillo, mientras que las áreas oscuras decrecen, es decir, las áreas se distinguen con un mayor brillo cuando el observador se alinea con la fuente de luz.

Al incidir la luz sobre superficies irregulares se pueden dar uno de los tres tipos de traslape:

- **Unidireccional.** Todas las cavidades en este tipo de traslape se encuentran alineadas en una sola dirección sobre la superficie.
- **Isotrópico.** Este tipo de superficies tienen la característica de que las cavidades están distribuidas uniformemente y sobre la superficie.
- **Gaussiana.** El caso general donde la distribución el traslape tiene una media respecto a la normal, la cual tiende a cero.

El factor geométrico de atenuación  $G(\Psi, \theta)$  se define como la fracción de luz reflejada por las facetas de la superficie, la intensidad de la luz es proporcional al número de facetas incidentes sobre el pixel [31]. La longitud de la faceta  $l$  es sombreada o enmascarada por una longitud  $m$  ( ver Figura ?? ).

$$\frac{m}{l} = \frac{\sin \gamma}{\sin \beta} \quad (2.31)$$

$$G(\Psi, \theta) = 1 - \frac{m}{l} \quad (2.32)$$

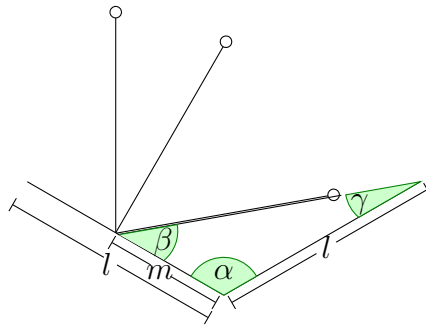


Figura 2.12: Relación del factor geométrico de las cavidades.

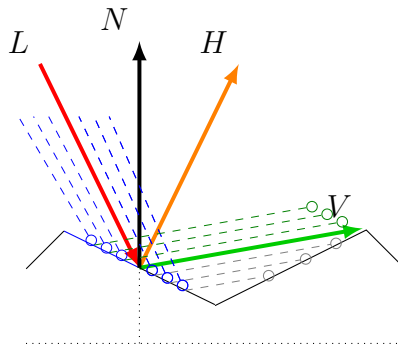


Figura 2.13: Diagrama de enmascaramiento.

La luz incidente por el vector  $L$  es enmascarado ( ver Figura ?? ) por la faceta adyacente con dirección al vector  $V$ , es decir, parte de la luz es bloqueada y es calculada por la ecuación ?? en términos de sus vectores.

Donde el factor de enmascaramiento  $G_e$  está dado por:

$$G_e = \frac{2(N \cdot H)(N \cdot V)}{V \cdot H} \quad (2.33)$$

Otro efecto es el de sombreado, en el cual la luz incidente ha sido parcialmente bloqueado por la faceta adyacente, como se muestra en la Figura ??, al evaluar el producto punto de dos vectores la normalización tiene una longitud de 1.

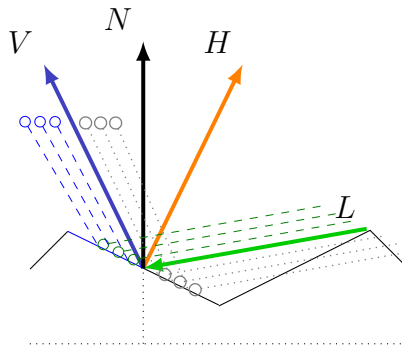


Figura 2.14: Diagrama de sombreado.

## 2.8. SUPERFICIES LAMBERTIANAS

El factor de sombreado  $G_s$  es:

$$G_s = \frac{2(N \cdot H)(N \cdot L)}{V \cdot H} \quad (2.34)$$

Si este producto punto es negativo significa que la superficie se está visualizando desde el lado opuesto. Por tanto el factor de atenuación se expresa de la siguiente manera:

$$G = \min \{1, G_e, G_s\}$$

Cada una de estas características fueron tomadas en cuenta y modeladas por Oren y Nayar [30] tomando en cuenta ambas disposiciones de las facetas y dando como resultado la ecuación ?? la cual se apega de mejor forma a distintos tipos de materiales.

$$f_r(\omega_i, \omega_o) = \frac{\rho}{\pi} (A + B \max[0, \cos(\phi_r - \phi_i)] \sin \alpha \tan \beta) \quad (2.35)$$

Donde:

$$A = 1,0 - 0,5 \frac{\sigma^2}{\sigma^2 + 0,33}$$

$$B = 0,45 \frac{\sigma^2}{\sigma^2 + 0,09}$$

$$\alpha = \max(\theta_i, \theta_o)$$

$$\beta = \min(\theta_i, \theta_o)$$

$$\frac{\rho}{\pi} = \text{albedo}$$

Una superficie no lambertiana refleja con mayor intensidad la luz en cierta dirección, este reflejo anisotrópico es calculado por el término geométrico, este tipo de textura se puede apreciar en materiales como el metal bruñido, discos de vinilo, seda, cabello, etc. Conociendo el manejo de la luz sobre la superficie de los objetos se puede hacer uso de las distintas propiedades de ésta para lograr diferentes efectos físicos de los materiales, tarea que es realizada por el proceso de render.

## *CAPÍTULO 2. MANEJO DE MATERIALES SOBRE LAS SUPERFICIES*

# Capítulo

## PROCESO DE RENDER

# 3

Render es un término anglosajón utilizado en computación para describir los cálculos necesarios para generar una imagen o video, partiendo de un modelo 3D sólo con el cálculo de la iluminación del entorno. El proceso de renderizado tiene el objetivo de generar un espacio 3D formado por estructuras poligonales. Como hemos visto anteriormente para lograr una simulación realista se deben trabajar las superficies que contendrán las texturas, como agua, madera, metal, plástico, etc. El cálculo de la iluminación del ambiente permite observar colisiones y comportamientos físicos en las estructuras, logrando escenas con alta calidad, pero con carencias de varios comportamientos físicos observados en la naturaleza.

A mediados de los años 60 se comenzó a cuestionar la posibilidad de realizar los cálculos de la luminosidad, a partir de la emisión de múltiples rayos que incidan directamente sobre un objeto, a primera vista la respuesta fue que sería prácticamente imposible, ya que los rayos incidentes sobre el objeto son millones y de forma continua. La forma en que se resolvió este problema fue utilizar una cámara virtual en un espacio tridimensional, tomando en cuenta sólo aquellos rayos que incidan en la superficie del objeto y dicha superficie sea observable por la cámara. Esto dio origen a lo que hoy se conoce como *Raytracing*.

El interés de utilizar un trazador de rayos es la posibilidad de calcular los canales alfas y gamas, permitiendo observar fenómenos físicos como transparencia, reflejo, sombras e iluminación global haciendo que las escenas sean difíciles de diferenciar entre sintéticas y reales.

Los objetos 3D están compuestos por coordenadas cuyos componentes están en  $\mathbb{R}^3$ ; denotados como  $x$ ,  $y$  y  $z$ ; esto permite manejar el volumen del objeto representados por estas componentes. En el espacio tridimensional la primitiva mínima del objeto es un triángulo, a estas primitivas se les puede asignar ciertas características de interés como son textura y color.

## CAPÍTULO 3. PROCESO DE RENDER

La textura de un objeto depende del material del que esté hecho, los materiales tienen distintos tipos de comportamiento y por tanto distintas ecuaciones que los modelan. Para lograr visualizar los objetos en el escenario tridimensional es necesario que estos se encuentren iluminados, ya que sin una fuente de luz los polígonos que componen a los objetos no pueden ser visualizados.

### 3.1 TÉCNICAS DE ILUMINACIÓN

---

Los efectos de iluminación dan la sensación de profundidad, ya que la degradación de la sombra sobre la superficie 3D hace una mayor descripción del objeto observado, esto es claro en el caso en donde se tienen superficies curvas como cilindros y esferas, donde la superficie con mayor cercanía a la fuente de luz presenta una mayor concentración de ésta [32].

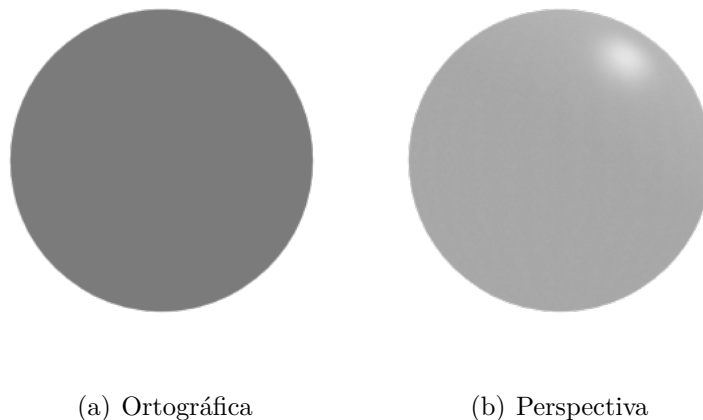


Figura 3.1: Distintas proyecciones de objetos curvos

Como se puede apreciar en la Figura ?? no se obtiene información relevante sobre la profundidad, en algunos casos parecen no tener volumen, la falta de los claros-oscuros crea confusión y da la sensación que son objetos de dos dimensiones, lo cual confunden al observador al no saber de que ángulo se está mostrando el objeto. Por ello es necesario dotar con efectos de iluminación, sombreado y proyección a las primitivas.

#### 3.1.1 RAYCASTING

---

El modelo de cámara de *pinhole* es el modelo comúnmente utilizado para la generación de una imagen bidimensional, el cual consiste en la convergencia de haces de luz en un punto focal en un arreglo matricial de píxeles, los objetos visibles se encuentran “frente” a la cámara (encerrados o truncados por la pirámide de cuatro lados generada por la cámara) [33]. El raycasting se refiere a trazar los rayos los

### 3.1. TÉCNICAS DE ILUMINACIÓN

cuales pasan a través de cada uno de los píxeles en el arreglo, los cuales conectan al punto focal con la escena (un rayo por pixel), este punto focal es “la posición” del observador ( ver Figura ?? ), los rayos incidentes en la matriz permite determinar la intensidad con la que un punto de intersección es iluminado.

Cuando la distancia focal converge en un punto se crea la vista en *perspectiva*, la cual es la forma natural de visión del ojo humano, en caso de que la distancia focal sea infinita los rayos no convergen por tanto son paralelos a la matriz de visión, también conocida como plano de visión, a este tipo de visualización se le conoce como vista *paralela*. En la Figura ?? se puede apreciar que no existe variación en el eje de profundidad  $z$ . En la Figura ?? el punto focal converge en el origen del sistema de coordenadas.

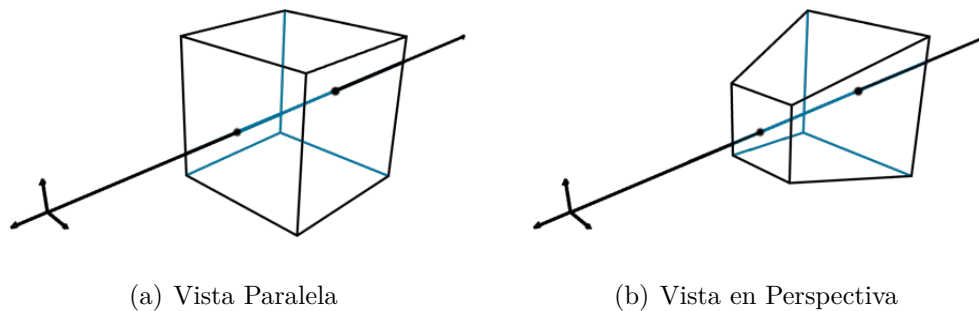


Figura 3.2: Esquema de distintos puntos focales.

Por conveniencia el sistema de coordenadas se estandariza que la pantalla se encuentra en un plano  $xy$ , y el escenario en el eje  $+z$ , mientras que el punto focal en el eje  $-z$ . Hacer una correspondencia de un espacio  $\mathbb{R}^3$  a  $\mathbb{R}^2$  y mantener la proporción de la lejanía de los objetos da la sensación que se encuentran a mayor distancia, se visualizan de menor tamaño o mayor tamaño dependiendo de su posición en el eje  $z$ .

Para hacer esto se usa la matriz de proyección, donde un punto  $p = (x, y, z, 1)$  es transformado por la matriz de proyección [34], en donde  $z$  es la distancia entre el plano y el objeto, la ecuación ?? debe satisfacer:  $(-1 \leq z \leq 1)$ ,  $1 \leq x \leq r$  y  $b \leq y \leq t$ , donde  $n$  es la distancia menor y  $f$  es la mayor distancia al punto de visión, a este espacio se le conoce comúnmente como *frustum* ( ver Figura ?? )

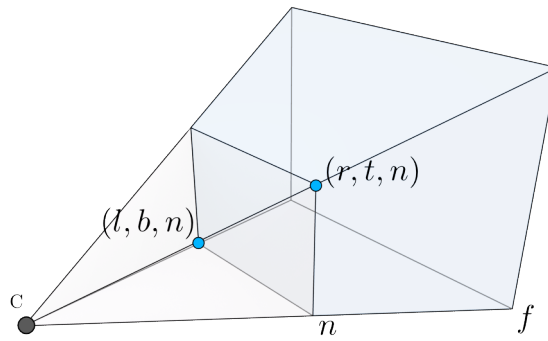
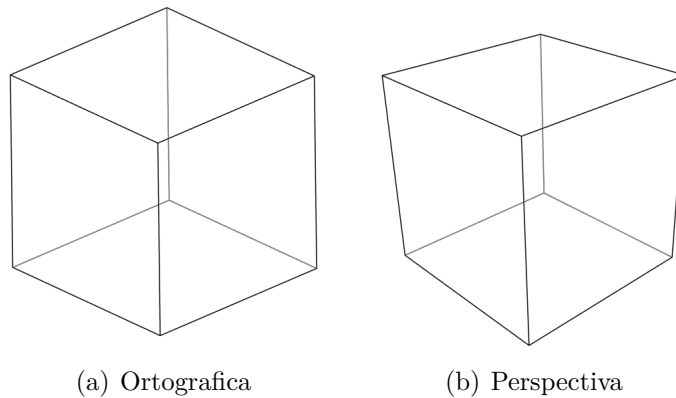


Figura 3.3: Plano de visión cercano  $n$  y el plano de visión lejano  $f$ .

$$\begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t-b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{2n}{r-l}p_x + \frac{r+l}{r-l}p_z \\ \frac{2n}{t-b}p_y + \frac{t-b}{t-b}p_z \\ -\frac{f+n}{f-n}p_z - \frac{2n-f}{t-n} \\ -p_z \end{pmatrix} \quad (3.1)$$

Las primeras computadoras no representaban la profundidad haciendo uso de la perspectiva, solo se limitaban a dibujar las primitivas es lo que conoce como proyección ortográfica. En algunas aplicaciones de gráficos por computadora se requiere una forma geométrica con mayor descripción, es decir, representar la profundidad con claro-oscuros, los cuales se generan en objetos con volumen ( ver Figura ?? ).



(a) Ortografica (b) Perspectiva

Figura 3.4: Vista plana vs perspectiva

La imagen generada debe representar características mínimas propias de volumen, como son altura, anchura y profundidad. Los objetos al estar conformados por coordenadas  $(x, y, z)$  que al ser mapeadas sobre un espacio bidimensional  $(x, y)$  se mantiene la descripción de profundidad [34] con la ecuación ??, a medida que los objetos se alejan del plano de visualización, estos se ven más pequeños, ya que al alejarse el

### 3.1. TÉCNICAS DE ILUMINACIÓN

plano de visión es mayor ( ver Figura ?? ). Las aristas del cubo se perciben iguales en la Figura ??, mientras que en la Figura ?? se perciben más cortas conforme éstas se alejan.

#### 3.1.2 SOMBREADO PLANO

---

Arthur Appel [1] desarrolla un algoritmo para plasmar los efectos de claro-oscuros ( ver Figura ?? ) al imprimir planos de piezas de ingeniería. Esta técnica permite crear distintas intensidades de color, sí la luz incidente sobre un punto de interés es bloqueada o no, por tanto el punto de incidencia es iluminado si existe dicha incidencia, esto logra una mayor velocidad para mostrar el objeto de interés, pero con carencias, puesto que esta técnica no permite crear reflejos, cáusticas o transparencias. El raycasting calcula sólo un punto de intersección entre el objeto y el rayo.

Para lograr un efecto con mayor degradación de sombreado es necesario aumentar el número de polígonos, por tanto también aumenta la cantidad de memoria requerida para almacenar el mallado. Representar la profundidad tiene una gran importancia en una escena tridimensional, ya que con ella se puede identificar la dirección de iluminación, la posición, intensidad, tamaño de la fuente de luz y eliminar la ambigüedad.

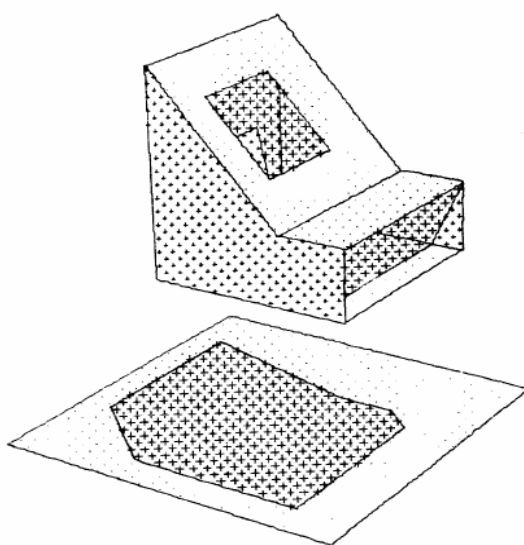


Figura 3.5: La presencia de claro-oscuros, *Arthur Appel, 1968* [1]

Esto genera sombras bien definidas dentro del escenario debido al cálculo de **fuen- te de luz-objeto-cámara**, en caso de que se produzca un ángulo entre éstos tres objetos, se tiene una superficie iluminada, de lo contrario ésta se encuentra en las sombras.

En el estudio de imágenes por computadora se ha encontrado que la generación de sombra tiene un gran impacto en la percepción de realismo de la escena [35].

## CAPÍTULO 3. PROCESO DE RENDER

En ambientes 3D se intenta simular distintos tipos de sombra que existen en la naturaleza, como son *hard shadows*, *soft shadows* [36]. La sombra es una región relativamente oscurecida por causa de un objeto el cual ocluye parcial o totalmente la superficie respecto a la fuente de luz. El sombreado de las caras que conforman el modelo tridimensional son iluminadas dependiendo del ángulo de incidencia de la luz y de la intensidad de ésta, la ecuación ?? calcula la intensidad con que es iluminado cada polígono que conforma el objeto. Usando la ley del coseno y tomando en cuenta tanto la orientación de observación, como la distancia hacia el punto de interés se tiene:

$$I = \Phi \frac{\cos \theta}{r^2} \quad (3.2)$$

Donde la intensidad  $\Phi$  de la fuente de luz, el ángulo de  $\theta$  ésta dado entre la normal  $N$  del polígono y la Fuente de luz  $L$ .

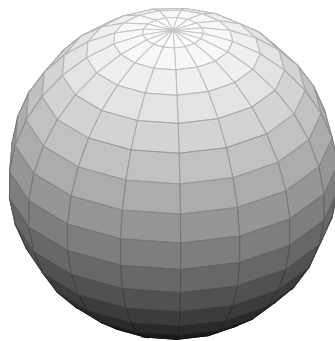


Figura 3.6: Efecto de iluminación en una esfera.

Cada polígono tiene un sólo color como se puede apreciar en la Figura ?. Se sabe que una de las características físicas de la iluminación es que obedece a la ley del cuadrado inverso, es decir, decrece con el cuadrado de la distancia a la fuente de luz.

### 3.1.3 SOMBREADO GOURAUD

---

Este tipo de sombreado crea una interpolación de color entre cada uno de los polígonos, con lo que crea un efecto de suavidad del sombreado [37]. Por ello es necesario calcular el nivel de degradación entre cada vértice, así como la profundidad dentro del escenario. Una forma básica de calcular la perspectiva de un punto  $P(x, y, z)$  es:

$$P(x, y, z) = \left( \frac{x}{z}, \frac{y}{z}, \frac{1}{z} \right) \quad (3.3)$$

### 3.1. TÉCNICAS DE ILUMINACIÓN

Suponiendo que el observador se encuentra en el origen del sistema de coordenadas, éste tiene la dirección de observación en el eje  $z$ , la coordenada  $1/z$  es una buena aproximación de la profundidad a la que se encuentra el punto de interés. Por tanto el sombreado es:

$$S = \cos^2 \theta \left( \frac{1}{z} \right) \quad (3.4)$$

El factor de sombreado en el polígono es la interpolación entre dos puntos de los vértices que lo conforman ( ver Figura ?? ), en este caso el polígono está conformado por cuatro vértices:

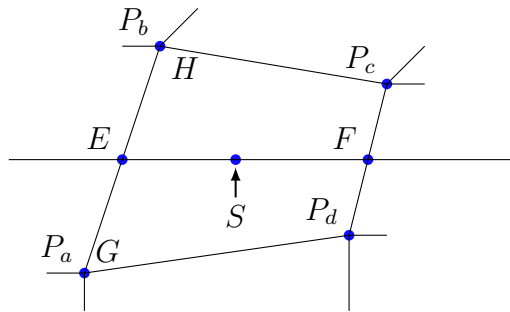


Figura 3.7: El sombreado puntual se encuentra dentro de los límites del polígono.

El coeficiente de atenuación  $\alpha$  se encuentra acotado ( $0 \leq \alpha \leq 1$ ). La posición de  $S$  se encuentra entre el segmento formado por  $E$  y  $F$ .

$$\alpha = \frac{S - E}{F - E} \quad (3.5)$$

De forma similar se calcula  $\beta$ , sobre el segmento  $GH$

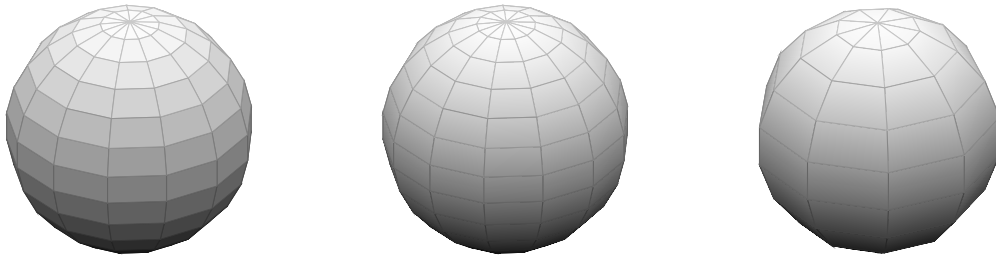
$$\beta = \frac{E - G}{H - G} \quad (3.6)$$

El sombreado que se da en el segmento  $EF$  es calculado por la ecuación ??, y el sombreado en el segmento  $GH$  por la ecuación ??.

$$S_\alpha = (1 - \alpha) \frac{1}{S_E} \cos^2 \theta + \alpha \frac{1}{S_F} \cos^2 \theta \quad (3.7)$$

$$S_\beta = (1 - \beta) \frac{1}{S_G} \cos^2 \theta + \alpha \frac{1}{S_H} \cos^2 \theta \quad (3.8)$$

En la Figura ?? se muestra la comparación entre el sombreado plano y el sombreado Gouraud.



(a) Efecto de iluminación plana (b) Efecto de iluminación Gouraud (c) Esfera con iluminación Gouraud y menor número de polígonos

Figura 3.8: Comparación entre iluminación plana y Gouraud.

Interpolar el sombreado permite crear un efecto de sombra continua, lo que en gráficos por computadora se traduce en una reducción de polígonos en el modelo, por tanto menor consumo de memoria y mayor velocidad al procesar dicho modelo.

#### 3.1.4 BLINN-PHONG

La calidad de las imágenes dependen de las técnicas de sombreado e iluminación. La percepción visual de realismo se fundamentan en las leyes de la óptica para incrementar esta sensación. La ley del coseno de Lambert ignora la posición del observador, así como la normal de la superficie en cada punto. Tomando en cuenta la posición de la cámara la cual recibe parte de reflexión difusa y una parte de la reflexión especular de la luz incidente.

El modelo de sombreado Gouraud crea un efecto de bandas de color sobre superficies que tienen una diferencia de ángulo notable entre los polígonos, a este efecto se le conoce como *bandas de Mach* [38], el cual dice “Cualquier descenso en la intensidad de luz en una superficie curva (la cual varía en un eje de dirección) ya sea cóncavo o convexa, en un punto en particular aparece una región brillante u oscura las cuales se encuentran al rededor de dicha región”.

En gráficos por computadora el sombreado es la obtención de la intensidad de la luz en cada punto de la superficie, calculando la relación de posición de la fuente de iluminación y la del observador. Phong [39] toma en cuenta estas características, se basa en el cálculo de varios ángulos: la incidencia de la luz; el ángulo de reflexión y el ángulo de visualización, los cuales crean el efecto completo de reflexión de superficies pulidas. El modelo de iluminación Phong es la suma de la luz ambiental, luz difusa y luz especular.

### 3.1. TÉCNICAS DE ILUMINACIÓN

$$L = A(L_a + L_d + L_s)$$

La ecuación ?? es el modelo de intensidad modelada por Phong.

$$I = I_a + K_d \sum_{j=1}^{j=ls} (N \cdot L_j) + k_s \sum_{j=1}^{j=ls} (N \cdot L'_j)^n \quad (3.9)$$

Donde:

- $I_a$  Reflexión debido a la iluminación ambiental
- $K_d$  Coeficiente de reflexión difusa
- $N$  Normal de la superficie
- $L_j$  Vector de dirección hacia la j-ésima fuente de luz
- $K_s$  Coeficiente de reflexión especular
- $L'_j$  Vector de dirección entre el punto de visión y la j-ésima fuente de luz
- $n$  Coeficiente de reflexión de la superficie del material

Este tipo de iluminación proporciona mejores resultados que Gouraud sobre superficies lisas y también con superficies con un alto índice de reflexión. El aumento de la calidad de representación del material también aumenta la cantidad de tiempo requerido para lograr el resultado final.

#### 3.1.5 ILUMINACIÓN DIRECTA

---

La iluminación directa total es la suma de la radiación emitida por una superficie, más la integral de la radiación entrante, cada punto de la superficie  $x$ . Cada radiación entrante en dirección  $w_i$  tiene un término coseno  $\theta_i$ .

La iluminación directa crea sombras bien definidas, conocidas como sombras fuertes, un rayo de sombra es disparado hacia un punto de intersección y la fuente de luz. La iluminación directa permite determinar la existencia de oclusión. Sobre la superficie de interés se trazan rayos de sombra, sí el rayo intersecta cualquier otro objeto entre el origen y la fuente de luz, entonces es una sombra, en caso contrario es ignorado. El efecto que produce son penumbras [36].

Los rayos de iluminación y los rayos de sombra se lanzan juntos, se tienen rayos primarios los cuales determinan si el punto a iluminar es visible por la cámara, y el rayo secundario o rayo de sombra determina si dicho punto es iluminado por la fuente de luz [40]. En este caso solo interesa la luz incidente directamente por la fuente de luz ( ver Figura ?? ) la cual se denota  $L_d(x, w)$ , el ángulo entre el rayo primario y la sombra es  $\cos \theta_i$ . La superficie de interés  $L_e(x, w)$  es conocido como el punto de intersección [41]. Por tanto la ecuación de iluminación directa es:

$$L_o(x, w_o) = L_e(x, w_o) + \int_{\Omega} f(x, w_o, w_i) L_d(x, w_i) \cos \theta_i d\omega_i \quad (3.10)$$

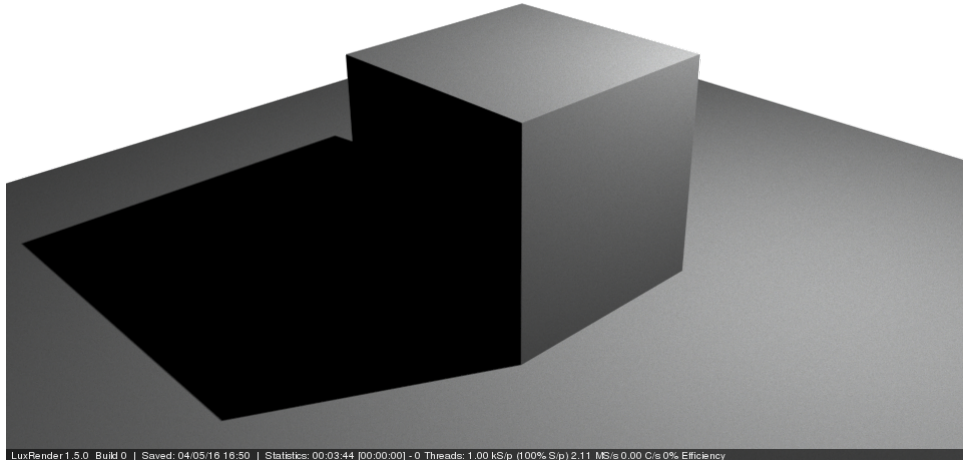


Figura 3.9: La iluminación directa genera sombras bien definidas.

### 3.1.6 ILUMINACIÓN INDIRECTA

La iluminación indirecta también es conocida como iluminación global toma en cuenta todas las superficies que tienen una reflexión difusa, este modelo relaciona la luz reflejada por una superficie hacia otra, si se asume que las superficies que conforman el escenario son superficies lambertianas, entonces se requiere que la distancia entre estas superficies se considere, así como la oclusión que pudiera suscitarse. Por ello es necesario el *factor de forma*.

$$B(x) = E(x) + \rho(x) \int_{\Omega} B(x) F_{ij} dA \quad (3.11)$$

Donde:

Radiocidad $B$	Energía total que deja una superficie, la cual es la suma de la energía emitida y la energía reflejada.
Emisión $E$	Energía emitida por una superficie.
Factor de forma $F$	Fracción de energía que deja una superficie y llega a otra.
$N$	Es el número de superficies discretizadas como “losas”.

Donde  $F_{ij}$  es el factor de forma entre un número finito de superficies ( ver Figura ??). El factor de forma representa la fracción de la energía que deja un elemento  $i$  y se dirige a otro elemento  $j$ .

### 3.1. TÉCNICAS DE ILUMINACIÓN

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \phi_i \cos \phi_j}{\pi r^2} dA_i dA_j \quad (3.12)$$

La luz interacciona de forma distinta sobre las superficies con alta rugosidad, las cuales, como se ha mencionado anteriormente dispersan la luz de forma difusa, esta dispersión de luz provoca que otras superficies sean iluminadas, la ventaja de este método es que la intensidad es independiente del punto de observación.

$$B_i = E_i + \rho_i \sum_{j=1}^N B_j F_{ij} A_j \quad (3.13)$$

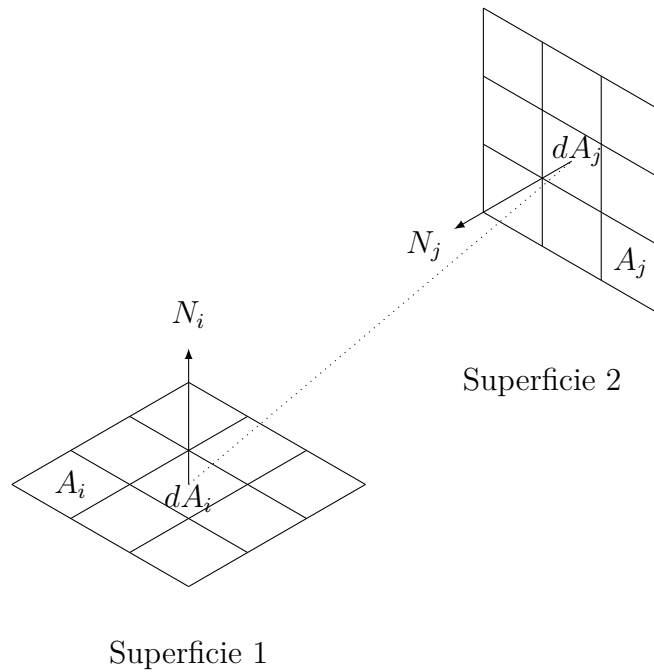


Figura 3.10: El factor de forma relaciona la intensidad entre las superficies.

La iluminación indirecta se puede apreciar independientemente de donde se encuentre el punto de observación, ya que en principio la radiación es dispersa de forma uniforme en todas direcciones ( ver Figura ??), la ecuación de radiocidad puede crear inter-reflexiones difusas, agregando a las sombra el color de la superficie primaria con la cual la luz interaccionó. La ecuación discretizada se puede expresar como una matriz.

$$\begin{bmatrix} 1 - \rho_1 F_{1,1} & -\rho_1 F_{1,2} & \cdots & -\rho_1 F_{1,n} \\ -\rho_2 F_{2,1} & -\rho_2 F_{2,2} & \cdots & -\rho_2 F_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ -\rho_n F_{n,1} & -\rho_n F_{n,2} & \cdots & 1 - \rho_n F_{n,n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix} \quad (3.14)$$

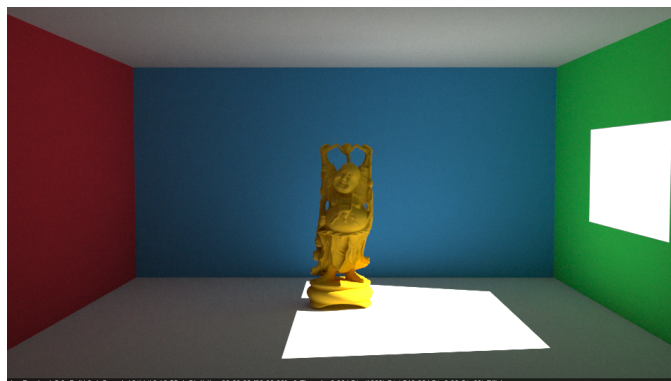


Figura 3.11: El modelo presenta características de iluminación indirecta.

A pesar de que es aceptable para algunas aplicaciones, muchas otras aplicaciones de imágenes requieren de la habilidad de cambiar o mover los objetos, por ende, la solución puede siempre reiniciar desde el principio después del cambio, pero esto requiere desechar información potencialmente útil sobre todo en cambios pequeños.

## 3.2 RENDER

---

El renderizado es la interpretación de la información a mostrar, su principal objetivo es crear una imagen de dos dimensiones, a partir de una cámara virtual, objetos tridimensionales, luces, texturas, etc. Renderizar las primitivas requiere realizar una detección de colisión entre *cámara-objeto-fuente de luz* ( ver Figura ?? ), lo que se busca, es determinar las superficies visibles del objeto y al mismo tiempo que dichas superficies sean iluminadas, los objetos opacos bloquean completamente la luz, mientras que los objetos translucidos la atenúan, a medida que se ha avanzado en el mejoramiento de los gráficos por computadora, también se han ido incrementando la exigencias para generar imágenes con características realistas.

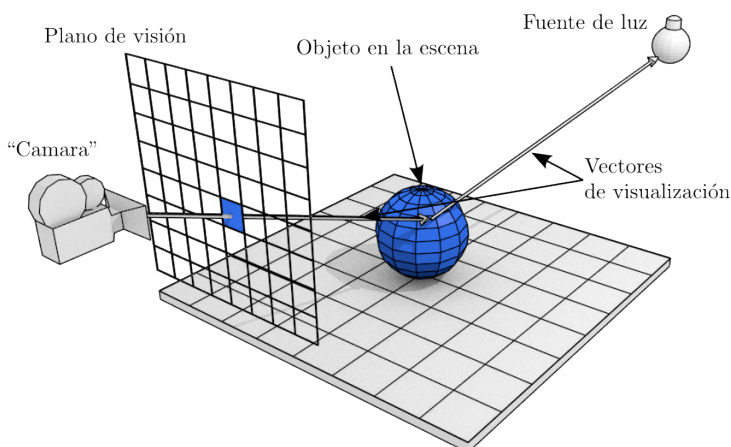


Figura 3.12: Esquema de conceptual de raytracing.

### 3.2. RENDER

Para que una imagen aparezca en el plano de visión (cámara) se debe generar un “rayo”, con el cual determina el punto de contacto de la luz, este punto de intersección es visible sí sigue una trayectoria sin obstrucción entre la cámara y el objeto, permitiendo crear imágenes por medio de “disparo” de rayos (vectores) y calcular la intersección que se genera con los objetos que conforman la escena ( ver Figura ?? ). Estos objetos se encuentran en un espacio  $\mathbb{R}^3$ , al ser calculada la intensidad de luz y color pasa a un espacio  $\mathbb{R}^2$ , el cual se conoce como plano de imagen, este es grabado como una imagen digital [42].

El render se divide principalmente en tres pasos:

- Etapa de aplicación; se establecen las características de la escena, como son la posición de los objetos, textura, parámetros de cámara e iluminación.
- Etapa de geometría, en esta parte se llevan acabo las operaciones sobre los objetos en la escena, como son transformación, escalado, rotación, etc.
- Rasterización, aquí todos los cálculos realizados sobre los objetos con los parámetros establecidos se convierten en píxeles sobre la pantalla.

El trazo de un rayo  $R$  es una línea en el espacio tridimensional, el cual es calculado de forma paramétrica como un vector ( ver Figura ??), el extremo del vector es el punto de intersección, si una superficie es altamente pulida este rayo es reflejado en otra dirección.

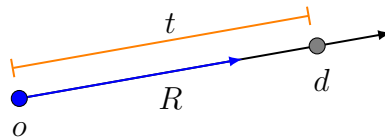


Figura 3.13:  $o$  es el origen,  $d$  es el vector de dirección,  $t$  es un parámetro  $[0, \infty)$ .

La ecuación ??, calculá el recorrido del vector.

$$r(t) = o + td \quad (3.15)$$

El punto de origen y el vector dirección están constituidos por tres componentes, los cuales deben ser calculados independientemente ( ecuación ?? ), permitiendo encontrar la intersección entre un rayo  $R$  y una superficie definida por una función  $F(x, y, z)$ .

$$\begin{aligned} x &= x_o + td_x \\ y &= y_o + td_y \\ z &= z_o + td_z \end{aligned} \quad (3.16)$$

## CAPÍTULO 3. PROCESO DE RENDER

Para encontrar la intersección de un rayo y la superficie de una esfera la cual tiene la forma :

$$x^2 + y^2 + z^2 = r^2 \quad (3.17)$$

Se sustituye la ecuación del rayo (ecuación ??) en la ecuación ??, la cual queda de la forma:

$$(x_o + td_x)^2 + (y_o + td_y)^2 + (z_o + td_z)^2 - r^2 = 0 \quad (3.18)$$

Calcular el punto de intersección no proporciona suficiente información sobre el objeto, además es necesario calcular la normal del punto de intersección, el ángulo de reflexión, transmisión o absorción, entre otros.

### 3.3 RAYTRACING RECURSIVO

---

La luz incidente sobre los materiales afectan a los objetos que se encuentran alrededor como en el caso del reflejo, a diferencia de las técnicas de iluminación donde la intensidad de la luz se calcula directamente sobre el punto a iluminar, el raytracing toma en cuenta la luz reflejada o absorbida por los demás objetos. Turner Whitted [2] introduce propiamente el raytracing recursivo con la ecuación ?? ya que a diferencia de los efectos de iluminación crea rayos, los cuales tiene las propiedades de poder ser reflejados o transmitidos por los objetos circundantes (ver Figura ??), creando efectos con mayor grado de realismo.

Comúnmente para crear el efecto de transparencia se pinta la superficie de forma inversa, es decir, el fondo se vuelve a pintar parcialmente sobre los objetos que se quieren simular como transparentes, pero esta técnica no logra simular los reflejos entre los objetos. La intensidad de la luz  $I$  que pasa entre el observador y el punto de la superficie consiste en los rayos primarios del reflejo  $S$ , y el componente de transmisión  $T$ , éstas intensidades representan la luz que se propaga en dirección  $V$ ,  $R$  y  $P$ , la cantidad de reflejos que pueden darse en el escenario eleva considerablemente el tiempo de computo, ya que cada reflejo eleva exponencialmente el número de cálculos sobre la superficie.

$$I = I_a + k_d \sum_{j=1}^{j=ls} (N \cdot L_j) + K_s S + K_t T. \quad (3.19)$$

$S$  es la intensidad de la luz incidente causada por el reflejo,  $k_t$  es el coeficiente de transmisión,  $T$  es la intensidad de la luz en dirección  $P$ .

### 3.3. RAYTRACING RECURSIVO

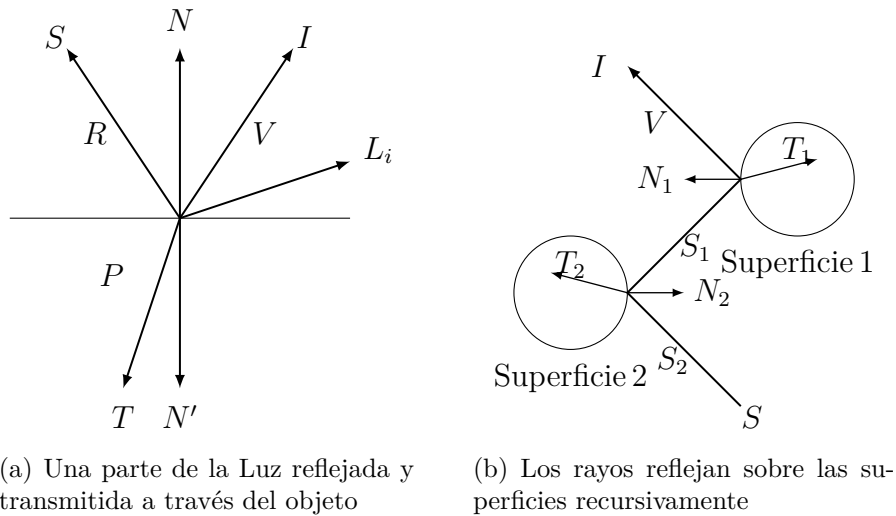


Figura 3.14: La luz incidente en los objetos puede ser reflejada o transmitida, en el caso del reflejo puede ser recursivo hasta que no rebote con más superficies reflectantes.

El modelo Phong no toma en cuenta la luz reflejada por los objetos circundantes. Este inconveniente no afecta el comportamiento de la reflexión difusa, sin embargo se puede apreciar su impacto en superficies con un alto grado de reflectancia. Los reflejos y la simulación de sombras son características deseables en los modelos de iluminación. Una superficie se encuentra en sombra cuando ocurre simultáneamente dos características; debe ser vista directamente por el observador, pero no iluminada directamente por la fuente de luz.

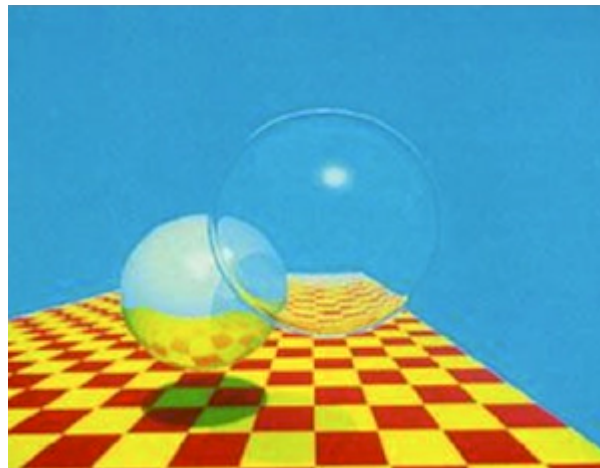


Figura 3.15: Imagen producida por Turner Whitted con su algoritmo de renderizado recursivo en 1980 [2].

Generalmente una escena con moderada complejidad a menudo la luz se refleja varias veces sobre las superficies antes de poder ser captada por el observador. Esto crea un árbol en el cual se calculan las intersecciones de los rayos producidos por cada una de las superficies. El proceso de reflexión se repite recursivamente, en el

peor caso es cuando dos superficies reflectivas se encuentran alineadas de forma paralela, lo que produce un árbol cuyas ramas se extienden de forma infinita, así que es necesario que la profundidad sea truncada en algún punto. Los cálculos de reflejo producen rayos secundarios, este proceso se repite hasta que ya no se creen nuevos rayos secundarios por parte de ningún otro objeto.

### 3.4 ECUACIÓN DE RENDERIZADO

La ecuación de renderizado calcula los rayos incidentes sobre las superficies punto a punto, las cuales se trazan desde el punto de observación hasta la fuente de iluminación, esta es modelada por *Kajiya* [42] a partir de las ecuaciones de electromagnetismo de Maxwell. La ecuación ?? es una aproximación de óptica geométrica, la cual modela el transporte de la luz de un punto a otro. La suma de la luz emitida sobre una superficie es igual a la intensidad total que es dirigida desde  $x$  hacia todas las demás direcciones  $w$ . Esta ecuación sólo maneja efectos como reflexión, refracción e iluminación directa.

$$L_o(x, w_o) = L_e(x, w_o) + L_r(x, w_o) \quad (3.20)$$

La idea detrás de la ecuación de renderizado es calcular la radiación de salida  $L_o$  (??) como la suma de la radiación emitida  $L_e$ , más la radiación reflejada  $L_r$ , esta última esta compuesta por el BRDF multiplicado por la iluminación difusa  $L_d$ .

$$\begin{aligned} L_r(x, w_o) &= \int_{\Omega} f_r(x, w_i, w_o) dE(x, w_o) \\ &= \int_{\Omega} f_r(x, w_i, w_o) L_d(x, w_o) (w_i, n) d\omega_i \end{aligned} \quad (3.21)$$

La ecuación de renderizado ?? fue propuesta por *Kajiya* [42], utilizando la notación de Pharr [43] queda de la siguiente manera:

$$L_o(x, w_o) = L_e(x, w_o) + \int_{\Omega} f(x, w_o, w_i) L_d(x, w_o) \cos \theta_i d\omega_i \quad (3.22)$$

Donde :

- $L_o(x, w_o)$  Intensidad de la luz que pasa del punto  $x$  en dirección  $w_o$ .
- $L_e(x, w_o)$  Es la relación de la intensidad de la luz emitida de  $x$  en dirección  $w_o$ .
- $f(x, w_o, w_i)$  Función de reflectancia bidireccional.
- $L_d(x, w_o)$  Es la iluminación difusa en un punto  $x$ , con un ángulo  $w_i$ .
- $\cos \theta_i$  Es al ángulo entre la fuente de luz y la normal de la superficie.
- $d\omega_i$  Es el ángulo solido.

La ecuación de renderizado calcula la incidencia de los rayos punto a punto, esto es

### 3.5. IMÁGENES BASADAS EN FOTONES

que por cada punto de la superficie iluminada se ha calculado la intensidad entre la fuente de iluminación y el observador, lo que ocasiona que las sombras que se generan sean nítidas, este efecto se le llama iluminación directa, efectos como reflexión y refracción son fácilmente alcanzados con este modelo ( ver Figura ?? ).

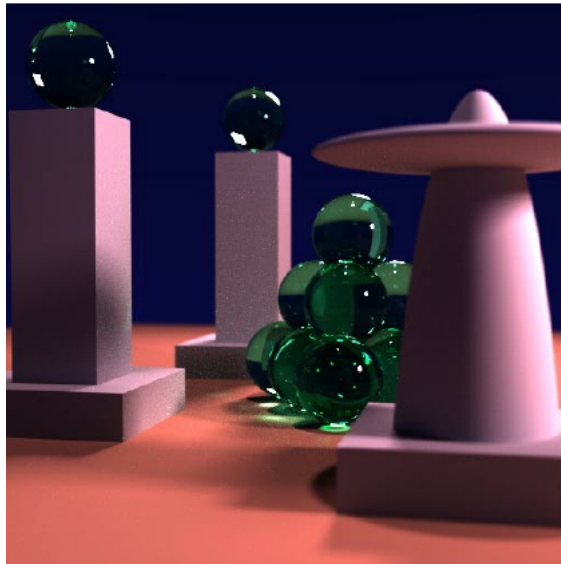


Figura 3.16: Imagen creada por J. Kajiya, en el año 1986 [3]

El proceso de renderizado en los últimos años ha proporcionado un profundo y mejor entendimiento de la interacción de la luz y la materia [40]. Las imágenes que se basan en el cálculo del comportamiento óptico, se conocen como imágenes basadas en física, las cuales crean renderizados realistas al simular la propagación de la luz en el ambiente, tomando en cuenta la energía emitida por las fuentes de luz sobre los objetos en el escenario y modelar el comportamiento de estos, ya sea que la luz por éstos sea reflejada, transmitida o absorbida.

### 3.5 IMÁGENES BASADAS EN FOTONES

---

La ecuación de renderizado haciendo uso de métodos estocásticos extiende su concepto llamado *path tracing* con el que en teoría simula todo tipo de reflejos, y la cual tiene problemas para simular cáusticas, por tanto está limitado a superficies lambertianas, escenarios con un grado considerable de superficies reflectantes requieren un número considerable de operaciones y cálculos matemáticos para poder observar cáusticas. Los métodos con mayor éxito de hoy día combinan la capacidad de la radiocidad y del raytracing para generar imágenes con un alto grado de realismo.

El concepto de las imágenes creada por fotones se divide en dos tipos de mapas, uno conocido como *mapa de cáusticas de fotones* (caustics photon map) y *mapa global de*

## CAPÍTULO 3. PROCESO DE RENDER

*fotones* (global photon map) [41], la creación de imágenes por fotones consiste en que los rayos ahora son vistos como fotones los cuales son paquetes de energía para ser emitidos sobre la escena desde una fuente de luz y golpear la superficie de los objetos.

Separar los métodos garantiza mayor exactitud, mejora el rendimiento y reduce la cantidad de memoria requerida. El resplandor  $L_s(x, \Psi_r)$  de los objetos es calculada con la ecuación de renderizado (??), esto se calcula por cada uno de los rayos que golpean la superficie de los objetos.

$$L_s(x, \Psi_r) = L_e(x, \Psi_r) + \int_{\Omega} f_r(x, \Psi_i; \Psi_r) L_i(x, \Psi_i) \cos \theta_i d\omega_i \quad (3.23)$$

Donde  $L_e$  es el resplandor emitido por la superficie del objeto,  $L_i$  es el resplandor entrante en dirección  $\Psi_i$ , mientras que  $f_r$  es el **BRDF** y  $\Omega$  es la hemiesfera de vectores entrantes.  $L_e$  es tomada directamente de la información que posee la superficie, así que no requiere cálculos adicionales.  $L_r$  depende del resplandor emitido por el resto del escenario, aplicando el método Monte Carlo [44] se puede resolver directamente. La ecuación ?? se puede dividir como la suma de sus componentes, donde  $L_r$  es expresada como:

$$\begin{aligned} L_r = & \int_{\Omega} f_r(x, w_o, w_i) L_{i,l}(x, w_o) \cos \theta_i d\omega_i + \\ & \int_{\Omega} f_r(x, w_i, w_o) [L_{i,c}(x, w_i) + L_{i,d}(x, w_i)] \cos \theta_i d\omega_i + \\ & \int_{\Omega} f_{r,d} L_{i,c}(x, w_o, w_i) \cos \theta_i d\omega_i + \\ & \int_{\Omega} f_{r,d} L_{i,d}(x, w_o, w_i) \cos \theta_i d\omega_i \end{aligned} \quad (3.24)$$

### 3.5.1 MAPA GLOBAL DE FOTONES

Cada rayo de luz es trazado en la escena hasta que el rayo sea absorbido por la superficie, ó ya no golpea con otro objeto. Cada vez que un rayo golpea una superficie la energía es depositada en un mapa de fotones, este mapa solo almacena los rayos secundarios de las reflexiones.

En el trazado de luz se emiten una cierta cantidad de fotones, los cuales son emitidos desde distintas posiciones por las fuentes de luz, la cantidad es determinada por el tamaño de la fuente de luz y su relación respecto a las dimensiones de la escena.

Sí la proyección de fotones en cierta dirección solo contiene superficies difusas la cantidad de estos será menor, si por el contrario, esa dirección sobre la que se proyectaron contiene superficies lambertianas con alto grado de reflectividad, la cantidad de fotones se incrementará. Cada rayo proporciona una fracción de la energía de la fuente de luz, cada vez que un rayo golpea la superficie de un objeto se decide si este

### 3.5. IMÁGENES BASADAS EN FOTONES

fotón se absorbe o se refleja, la nueva dirección de un fotón es calculada utilizando el BRDF de la superficie.

El mapa global de fotones es usado para renderizar superficies rugosas, esto es, emitir fotones hacia todos los objetos en la escena, y si este no es visualizado directamente, entonces no requiere una alta densidad de fotones. La primera intersección fotón-objeto (iluminación directa) es almacenada, La siguiente intersección del fotón es almacenada como sombra, estos fotones de sombra son utilizados para reducir el número rayos requeridos ( ver Figura ?? ).

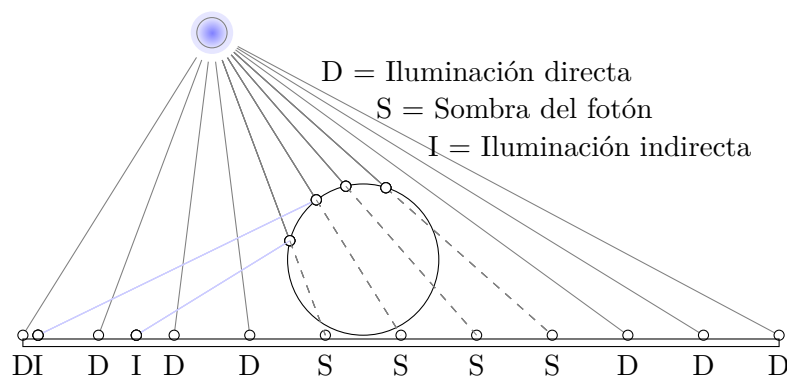


Figura 3.17: *Global Photon Map* es usado para reducir el número de rayos requeridos.

#### 3.5.2 MAPA DE CÁUSTICAS

El mapa de cáusticas de fotones es usado para almacenar sólo la información relacionada a las cáusticas producidas por los fotones, este tipo de mapas requiere una gran densidad de fotones. La información computada de estos fotones es almacenada en un *mapa global de fotones* (global photon map). Efectos como las cáusticas son difíciles de generar con un trazador de rayos, habitualmente se utilizan métodos estocásticos como lo es Monte Carlo [45], utilizando la combinación de rayos con un mapa de fotones se obtiene de forma sencilla, sin embargo las cáusticas no se logran simular de forma adecuada usando solamente radiosidad, así que es necesario tomar distintos métodos para crear un híbrido.

Las cáusticas se generan cuando los rayos de luz son reflejados o refractados por un objeto, esto crea concentraciones de luz en determinada área. La formación de cáusticas requiere una gran cantidad de fotones para converger adecuadamente, debido a esta característica, computacionalmente se requiere tiempos considerables para generarlas. Las cáusticas solo pueden ser visualizadas directamente cuando se proyectan sobre la superficie, lo requieren alta precisión, contrario a lo que sucede con la iluminación global.

3.5.3 ILUMINACIÓN DIRECTA

Siguiendo las ecuaciones del mapeado de fotones de Jensen [41] al implementarlas muestra los distintos comportamientos que tiene la luz, el resultado de cada una de estas ecuaciones se combina para generar una imagen que muestre la mayor cantidad de efectos involucrados en el escenario. La ecuación de mapeado de fotones ?? esta constituida por cuatro términos, los cuales contribuyen en el color y la intensidad de iluminación del pixel final, el primer término se refiere a la iluminación directa. La ecuación ?? normalmente se envía rayos de sombra junto a los rayos de iluminación, esto permite saber si un objeto es visible o no por el punto de observación.

$$\int_{\Omega} f_r(x, w_i, w_o) L_{i,l}(x, w_i) \cos \theta_i d\omega_i \quad (3.25)$$

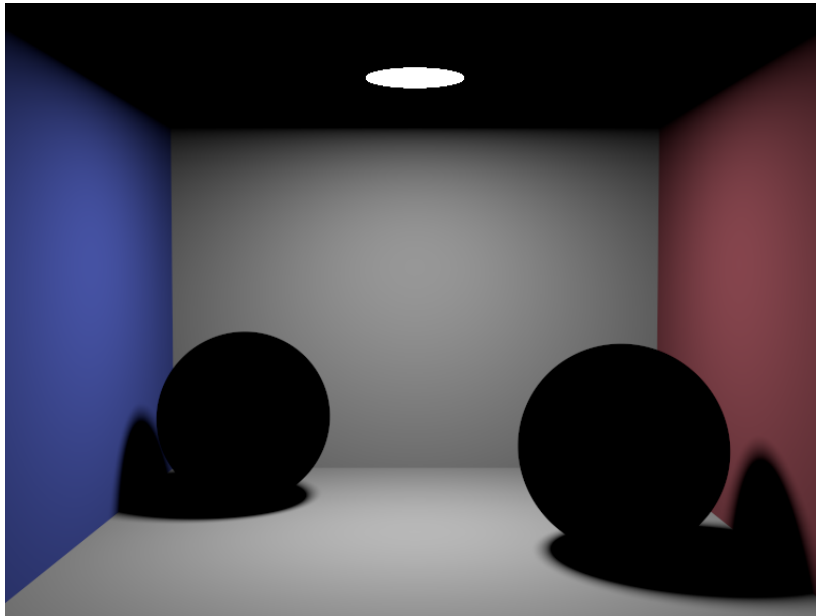


Figura 3.18: Iluminación directa.

En la Figura ?? se muestra solo un rebote de la luz hacia la cámara generado por los objetos; se aprecian objetos con textura rugosa. La iluminación directa además de calcular las superficies que son directamente iluminadas, identificar estas áreas disminuye el uso de rayos de sombra, si este rayo no golpea la superficie de interés, entonces la contribución de la iluminación se incluye en la integral y la contribución de la sombra es cero, en el caso de que el rayo de sombra golpee la superficie la iluminación es atenuada, el uso de sombras es costoso. El uso de rayos de sombra son utilizados sólo en situaciones donde existe una mezcla de superficies sombreadas e iluminadas o donde la cantidad de superficies sombreadas es baja.

### 3.5. IMÁGENES BASADAS EN FOTONES

#### 3.5.4 REFLEJO ESPECULAR

---

La ecuación ?? pertenece al segundo término de la ecuación ??, el cual se refiere al efecto de reflejo de superficies especulares, las superficies que tienen esta característica requieren de un alto número de fotones de forma recursiva, ya que en éste tipo de superficies en reflejo con mayor nitidez de los objetos circundantes crea un efecto convincente de superficie pulida.

$$\int_{\Omega} f_r(x, w_i, w_o) [L_{i,c}(x, w_i) + L_{i,d}(x, w_i)] \cos \theta_i d\omega_i \quad (3.26)$$

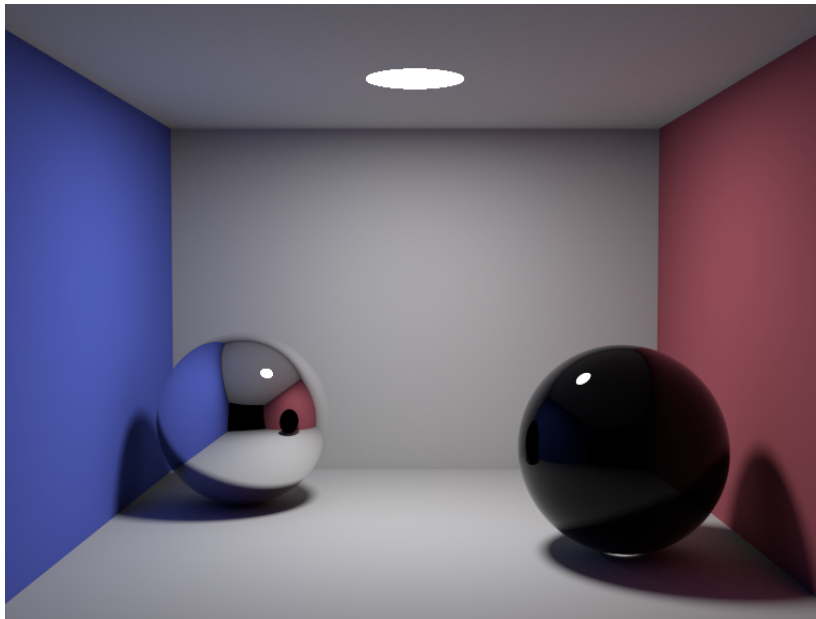


Figura 3.19: Superficies especulares.

Este término representa al modelo para los reflejos lambertianos, el cual requiere una cantidad menor de fotones para obtener buenos resultados en comparación con otros efectos.

#### 3.5.5 CÁUSTICAS

---

La ecuación ?? pertenece al tercer término de ??, y se refiere al cálculo de cáusticas, las cuales son generadas por materiales como líquidos y/o cristales, como se ha mencionado anteriormente estas requieren una gran cantidad de fotones, lo cual consume un tiempo de computo considerable.

$$\int_{\Omega} f_{r,d} L_{i,c}(x, w_o, w_i) \cos \theta_i d\omega_i \quad (3.27)$$

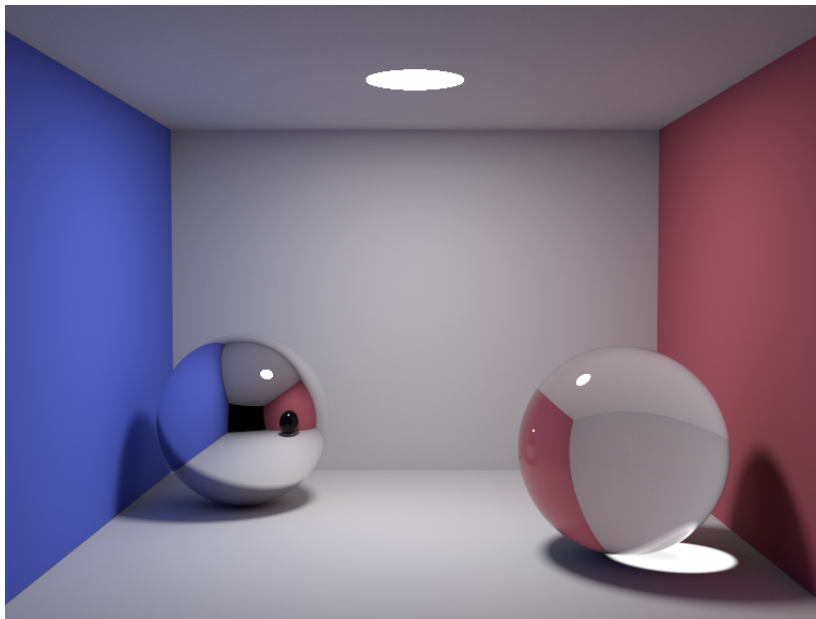


Figura 3.20: Aparición de cáusticas.

Jenssen [41] no recomienda hacer uso del método Monte Carlo para generar las cáusticas ya que resulta ineficiente, esto significa que la cáustica es visualizada directamente, una aproximación de la radiación de la cáustica se incluye en el mapa de global de fotones.

### 3.5.6 ILUMINACIÓN INDIRECTA

La ecuación ?? es el cuarto término de ??, este calcula la luz reflejada al menos una vez, desde que fue emitida por la fuente de luz, este reflejo crea una iluminación suave sobre la superficie del objeto.

$$\int_{\Omega} f_{r,d} L_{i,d}(x, w_o, w_i) \cos \theta_i d\omega_i \quad (3.28)$$

La iluminación indirecta proporciona parte del color de la superficie primaria con la que interaccionó la luz. Esta iluminación puede combinarse con las cáusticas ( ver Figura ?? ).

### 3.5. IMÁGENES BASADAS EN FOTONES

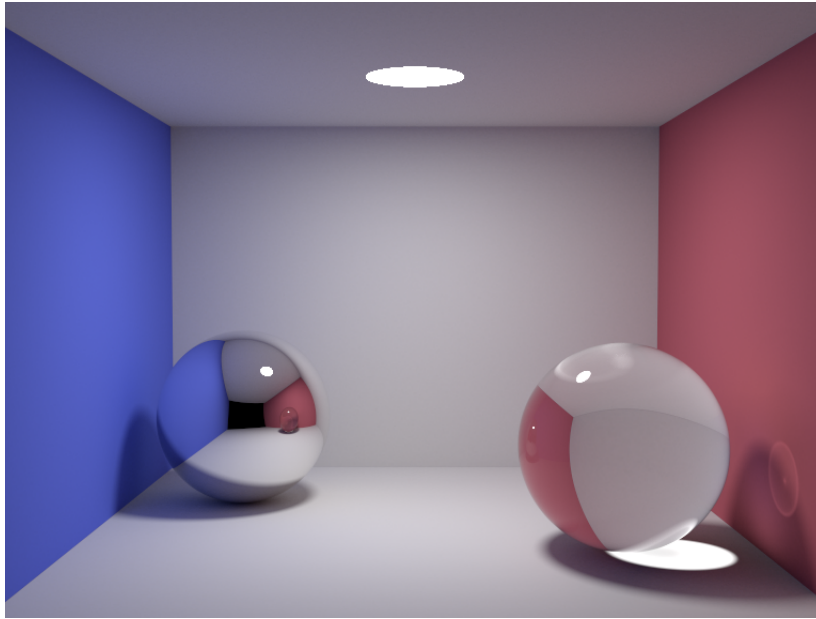


Figura 3.21: Iluminación indirecta

Al combinar las ecuaciones en una sola imagen se puede apreciar cómo los distintos objetos dentro del escenario crean un efecto visual con mayor realismo. Como puede apreciarse la luz interactúa de distintas maneras sobre los objetos dependiendo de su material, un solo modelo no puede modelar todos los comportamientos de la luz. El mapeado de fotones es un modelo muy robusto, el cual considera la mayor parte de interacciones.

La luz en la mayoría de los casos es reflejada por la superficie con la que ha colisionado o hasta que la energía es absorbida y no puede ser reflejada. Por lo que escenarios con mayor realismo requieren un número considerable de recursión, aumentando exponencialmente el tiempo de renderizado.

### *CAPÍTULO 3. PROCESO DE RENDER*

# Capítulo

## PROPUESTA Y ANÁLISIS DE RESULTADOS

# 4

Los sistemas de renderizado se han valido de innumerables técnicas y combinaciones de ellas para encontrar un equilibrio entre velocidad y exactitud. El realismo en imágenes generadas por computadora requiere de estas dos características, por un lado generar una imagen en tiempos reducidos disminuye el realismo, por otro lado si se requiere mayor detalle el tiempo para alcanzarlo se incrementa. Las superficies a calcular debe ser delimitadas, puesto que no todas las superficies que conforman el escenario son visibles, las estructuras de datos permiten delimitarlos y sólo calcular aquellas superficies de interés.

Las estructuras de datos involucradas en la generación de imágenes por computadora tienen distintas características y son eficientes bajo ciertas condiciones, por lo que es un problema seleccionar la estructura de datos adecuada para generar una imagen y que ésta se obtenga con una alta calidad de renderizado y en el menor tiempo posible, por lo que dentro de la propuesta se hace el uso de la técnica de raytracing conocida por conos, ésta permite obtener eficiencia en la obtención de la imagen a renderizar y minimizando los tiempos de obtención lo que nos permite imágenes con una alta calidad que es uno de los objetivos planteados en este trabajo.

Generar imágenes por computadora con alto grado de realismo involucra distintas tareas y operaciones, las cuales deben trabajar en conjunto para lograr el objetivo deseado, en la Figura ?? se muestra un diagrama general, donde se relacionan cada una de las tareas, éstas contribuyen tanto a mejorar la velocidad de procesamiento, como la creación de efectos físicos visuales en la imagen final.

La siguiente sección muestra la metodología utilizada para la generación de raytracing usando el esquema de conos y fotones, con un control híbrido (HBVH), así mismo el cálculo de los materiales utilizados en la textura de los objetos a renderizar, tomando en consideración los conceptos antes vistos.

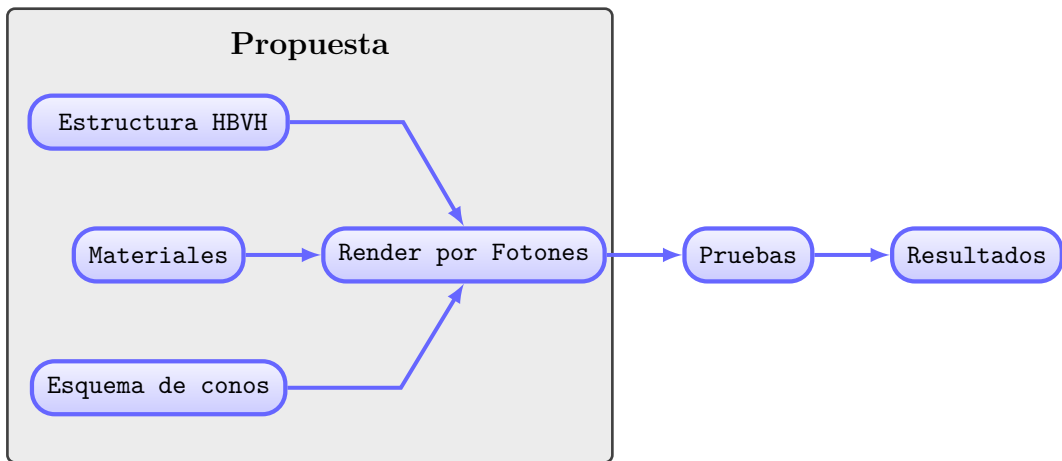


Figura 4.1: Diagrama de metodología.

El manejo de las primitivas que componen los objetos dentro del escenario tridimensional, se realiza por medio de la estructura de datos de volúmenes acotados por jerarquías híbridas ( ver Figura ?? ).

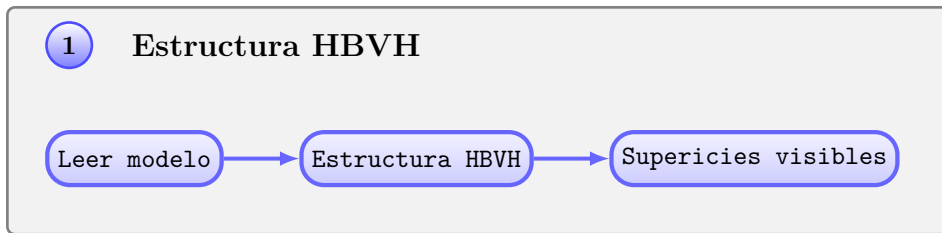


Figura 4.2: Diagrama de la estructura HBVH.

Las texturas a mostrar en una imagen dependen del color y de la composición de su superficie, la cual puede estar constituida por micro-cavidades, dependiendo de la densidad de estas cavidades sobre la superficie del objeto, será la apariencia de la textura pudiendo ser una apariencia rugosa, satinada o lisa, en la Figura ?? se muestra el diagrama a seguir para al cálculo de los materiales.

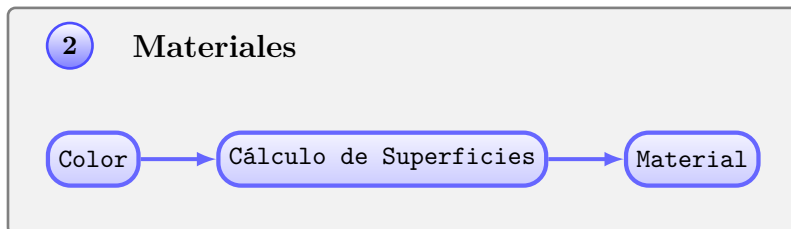


Figura 4.3: Superficie de los materiales

En la Figura ?? se muestra el diagrama a seguir para cada rayo saliente de la superficie, los cuales son proyectados en forma de cono, el cual tiene un punto de mayor intensidad en el centro de este, desde donde se calcula la intensidad con la que son afectados los píxeles vecinos.

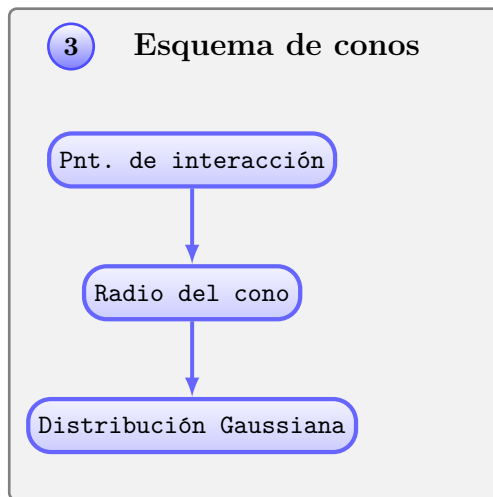


Figura 4.4: Esquema de conos

Al obtener la imagen con las características visuales deseadas es necesario compararla con una imagen base, en la Figura ?? se muestra el diagrama a seguir para la validación de las imágenes obtenidas.

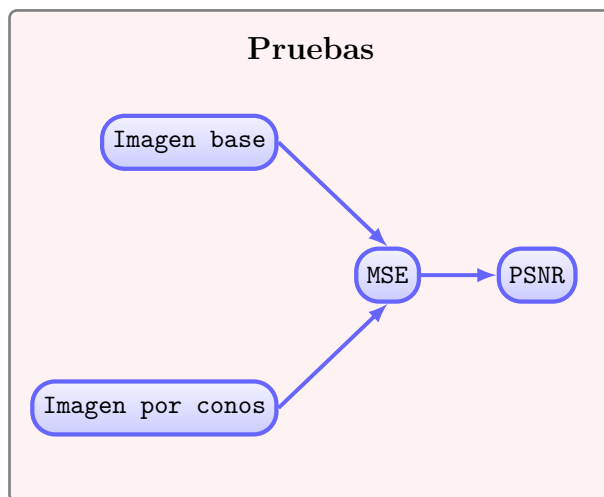


Figura 4.5: Pruebas

La Figura ?? muestra el diagrama de flujo de la propuesta realizada para la obtención de imágenes foto-realistas, pudiendo observar cáusticas y fenómenos físicos difíciles de obtener mediante métodos tradicionales.

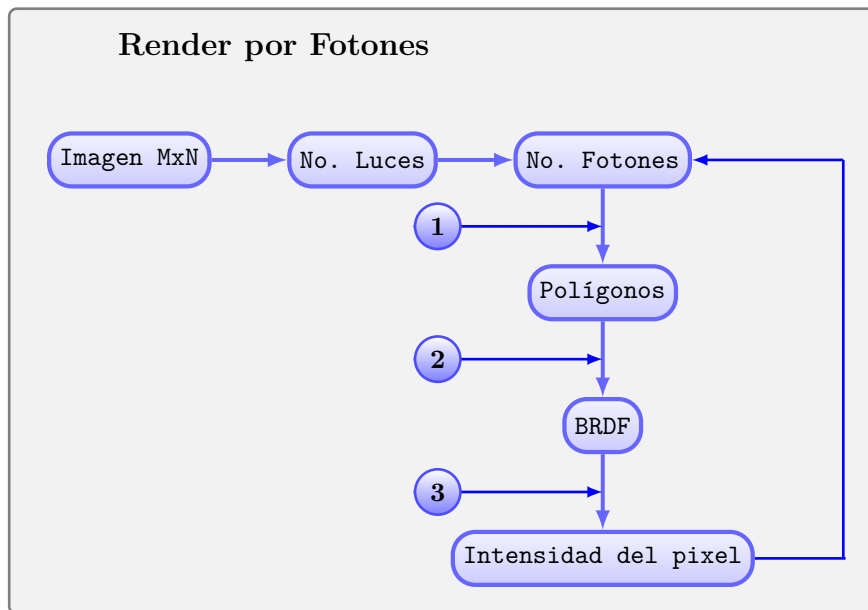


Figura 4.6: Raytracing basado en conos, fotones y HBVH.

#### 4.1 RAYTRACING POR CONOS

Alcanzar altas tasas de cuadros por segundo es complicado, dependiendo de los efectos que se quieran mostrar en la imagen. Crassin [46] utiliza el esquema de conos, iluminación directa e indirecta para crear renderizados en tiempo real, la iluminación directa es discretizada como puntos de iluminación virtual, debido a que este tipo de iluminación es costosa, generalmente utilizada en escenarios que contienen iluminación estática. Los sistemas de renderizado más eficientes hoy solo trabajan con un solo punto de observación, desde el cual se han calculado los efectos de iluminación de la escena, a medida que el punto de observación cambia, se actualiza la información de iluminación de la escena, se debe tomar en cuenta que una superficie iluminada reflejan la misma cantidad de luz en todas direcciones, por ello es que independientemente de donde se esté observando la superficie se visualiza con la misma cantidad de luz.

El renderizado por conos originalmente fue propuesto por Amanatides [47], en el cual se intenta que el rayo de incidencia abarque una cierta área, para lograr esto es necesario dividir el escenario en primitivas, es decir un plano se divide en pequeños elementos conocidos como voxels ( ver Figura ?? ), los cuales son elementos de volumen que contienen información como posición, color y un vector normal.

Un rayo se extiende en forma de pirámide circular ( cono, ver Figura ?? ). Ésto se logra sí se incluye el ángulo de extensión, el punto donde interacciona es tomado como el origen del rayo virtual, la forma tradicional de un rayo es que tiene un origen y una dirección, los cuales sólo dan un punto de interacción, difícilmente la luz mantendrá una forma puntual, es decir, la luz tiende a dispersarse mientras ésta recorre

## 4.2. REPRESENTACIÓN DE VOXELES

el escenario, creando un área de iluminación, como lo es la iluminación difusa. Al propagarse la luz en los bordes del cono se crean áreas de traslape, mientras que el centro tiene una mayor intensidad.

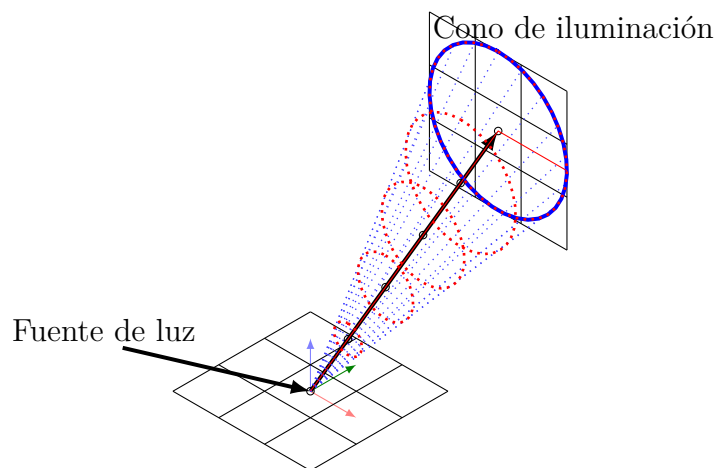


Figura 4.7: Proyección de la luz en forma de cono.

## 4.2 REPRESENTACIÓN DE VOXELES

Dividir el espacio requiere hacer un mayor uso de la memoria, sin embargo tiene la ventaja de una mayor exactitud. El rayo que coincide con el centro del voxel ( ver Figura ?? ) almacena la mayor intensidad de la luz incidente, a partir de este punto se distribuye hacia los vecinos de forma gaussiana lo que facilita la interpolación de la información.

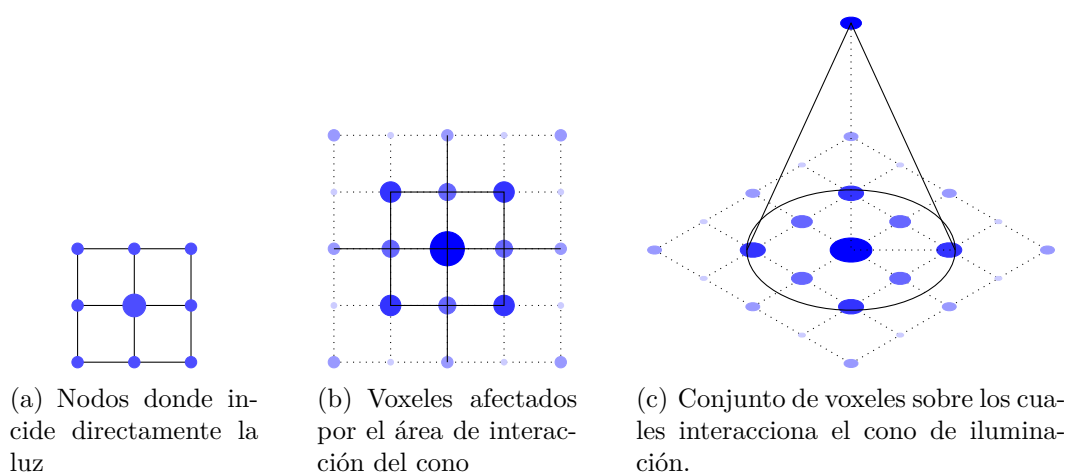


Figura 4.8: Representación de píxeles y su vecindad.

Los voxeles se almacenan en un árbol octree, de esta forma se puede calcular el escenario de forma paralela, permitiendo adquirir mayor velocidad de muestreo, lo

cual es conveniente para algunos usos donde se requiere tiempos de respuesta cortos, también considerando que el dispositivo donde se muestra la imagen tenga las características deseadas para dichos propósitos. Lo deseable es que la velocidad de muestreo sea mayor a la del dispositivo.

Al aumentar la cantidad de voxeles afectados por el cono de iluminación el efecto de suavizado es más natural, pero este incremento también crea un mayor requerimiento de memoria y tiempo de computo. La incidencia de la luz en el centro del área de interacción crea una distribución gaussiana ( ecuación ?? ) de iluminación sobre los voxeles adyacentes, con lo cual los bordes tienen una menor intensidad luminosa, y el traslape no crea intensidades de luz indeseadas.

La radiación emitida  $L$  producida por el cono es calculada con la integral sobre la superficie la cual contiene el volumen, como se sabe la radiación emitida depende de la radiación entrante, lo cual genera los efectos que percibe el observador, la ecuación ?? toma los valores dependientes del cono  $x_c$  es el centro del cono,  $r_c$  es el radio que abarca y  $v_c$  es la dirección con la que se proyecta.  $R_\Omega$  es el BRDF de la superficie.

$$L(x_c, r_c, v_c) = R_\Omega V_\Omega(x_c, r_c) N_\Omega(v_c) \quad (4.1)$$

A partir del radio  $r$  que tiene el cono y el punto de interacción  $x_c$ , es necesario calcular un punto  $x$  y determinar si se encuentra dentro de éste, la ecuación ?? permite determinar esto.

$$V_\Omega(x_c, r_c) = \int_{\Omega} H(r_c - \|x - x_c\|) d\omega \quad (4.2)$$

$V_\Omega(x_c, r_c)$  determina el porcentaje de rayos que caen dentro del cono, procesar la profundidad  $z$  del punto  $x_c$  y el cuadrado de la distancia ( $z^2$ ). Calculando una distribución gaussiana sobre el punto  $x_c$  ( ver Figura ?? ), para esto es necesario calcular la media  $\mu$  ( ecuación ?? ) y la varianza  $\sigma^2$  ( ecuación ?? ) sobre el ángulo sólido que genera el cono. La ecuación ?? toma en cuenta estas tres características.

$$\mu = \int_{\Omega} z d\omega \quad (4.3)$$

$$\sigma^2 = \int_{\Omega} z^2 d\omega - \mu^2 \quad (4.4)$$

$$V_\Omega(z_c, r_c) \approx \int_{z_c - r_c}^{z_c + r_c} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} dz \quad (4.5)$$

$N_\Omega(v_c)$  determina si un punto es potencialmente visible, por el método *backface-culling* [48], esto descarta todos los triángulos cuando el producto punto sobre la superficie de la normal y la cámara es mayor o igual a cero.

$$N_\Omega(v_c) = \langle \vec{n} \cdot v_c \rangle \quad (4.6)$$

Para asegurar una transición progresiva entre los conos es necesario calcular la región de traslape que se da entre éstos como una *función de decaimiento* [49].

## 4.2. REPRESENTACIÓN DE VOXELES

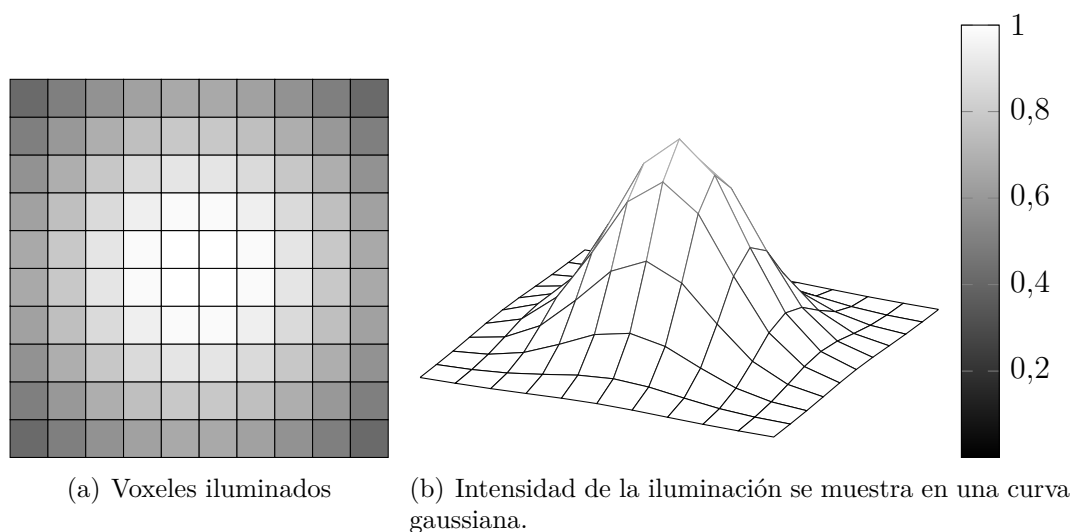
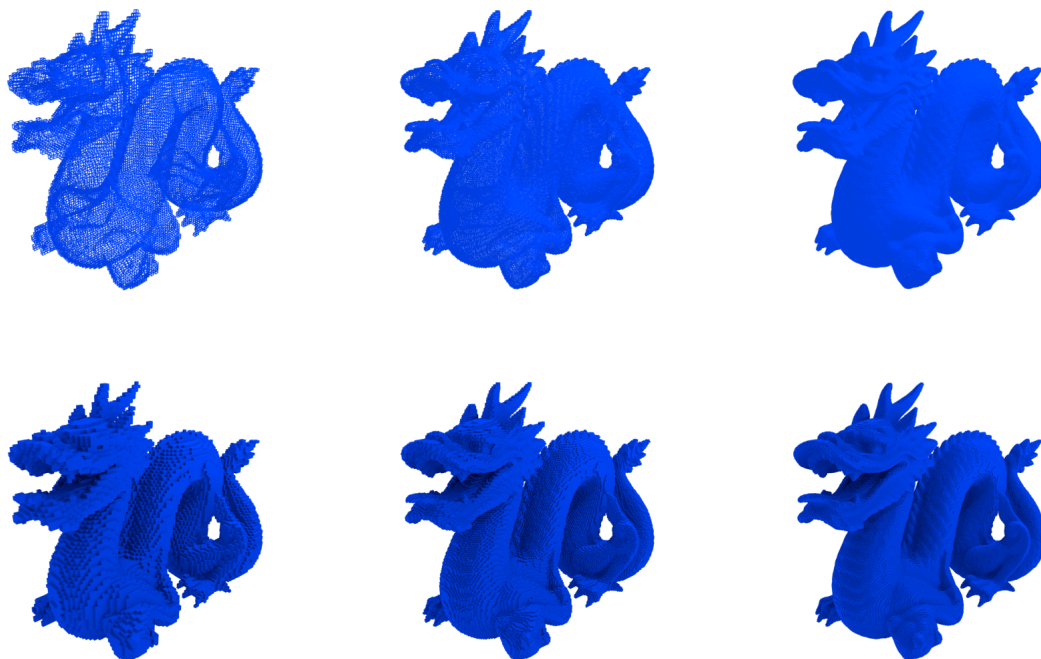


Figura 4.9: En la imagen se muestran los voxeles iluminados, y en la imagen se muestra curva de intensidad.



Subdivisión No.8  
Memoria 228.06 MB

Subdivisión No.9  
Memoria 476.31 MB

Subdivisión No.10  
Memoria 1469.22 MB.

Figura 4.10: Subdivisión del modelo dragón.

Detalles en los objetos requieren una mayor subdivisión del mallado ( ver Figura ?? ), de no ser así, el efecto de luz y sombras son vistas como grietas en la imagen final, por tanto el costo de subdividir crea un árbol con un mayor profundidad y

requerimiento de memoria como se muestra en la Figura ?? . Con el enfoque de conos sólo un rayo es requerido por un conjunto de ellos, lo que incrementa el desempeño, decidir cual es el nivel correcto de subdivisión depende mucho del escenario y de los objetos involucrados, ya que para percibir detalles en el objeto requieren una mayor subdivisión del mallado.

La técnica de conos está enfocada sobre todo a superficies difusas, área en la que se desempeña de forma muy exitosa ya que al dividir el escenario, se puede grabar la información de iluminación en cada uno de los elementos que lo componen, para después ser leído y mostrado directamente, es decir, no requiere que la información de iluminación sobre el escenario sea computado nuevamente sí el punto de observación se mantiene constante.

La iluminación directa define un área de interacción mientras que la iluminación indirecta es la que se vuelve a calcular a menos que el objeto que genera esos rayos secundarios se mueva. La iluminación global requiere una gran cantidad de rayos sobre la escena, lo que resulta costoso.

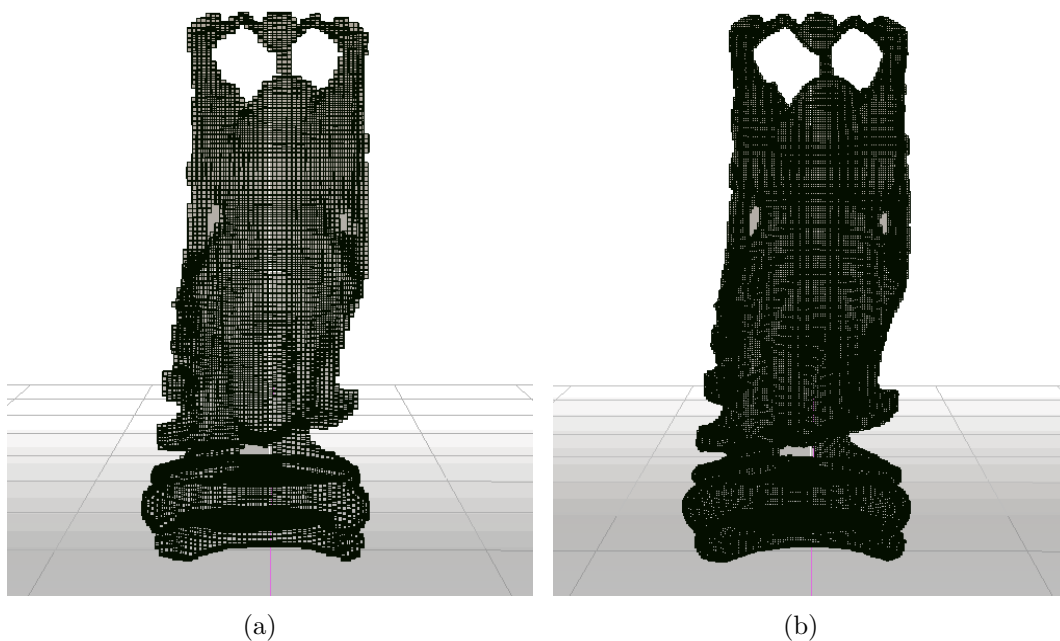


Figura 4.11: El modelo Happy buddha con distintos niveles de sub-división muestra mayor definición de la figura.

### 4.3 MODELOS

---

Calcular la iluminación en cada una de las primitivas que componen la escena requiere un mayor nivel de detalle, por tanto es necesario subdividir los objetos en el

### 4.3. MODELOS

escenario. El área que abarca el cono sobre las primitivas es mayor a medida que la distancia entre la fuente de luz y el objeto de interés se incrementa, por ello los objetos que se encuentran a menor distancia requieren mayor subdivisión de la superficie para mayor detalle.

Como se ha mencionado anteriormente, los objetos almacenados en el árbol requiere que se subdividan, lo que tiende a crear árboles des-balanceados, con ramas cargadas, esto crea problemas de desempeño, ya que para acceder a un nodo en específico se requiere recorrer los nodos previos, en el peor de los casos.

Los modelos utilizados son los propuestos por la comunidad de gráficos por computadora ( ver Figura ?? ), estos modelos han sido ampliamente estudiados debido a que están compuestos por distintas cantidades de polígonos, los cuales son bien conocidos.



Figura 4.12: Modelos comúnmente utilizados por la comunidad de gráficos por computadora.

En el Cuadro ?? se muestra especificaciones sobre los modelos utilizados en las pruebas de renderizados con el algoritmo propuesto.

## CAPÍTULO 4. PROPUESTA Y ANÁLISIS DE RESULTADOS

Modelo	Nombre	Vértices	Polígonos	MB.
1	Bunny	35,947	69,451	2,9
2	Armadillo	172,974	345,944	6,6
3	Dragon	437,645	871,414	32,3
4	Budha	543,652	1,087,716	40,6
5	Blade	882,954	1,765,388	80,1
6	xyz-Dragon	3,609,600	7,219,045	130,8
7	Statuette	4,999,996	10,000,000	181,2

Cuadro 4.1: Características de los objetos.

De el cuadro anterior se puede observar la comparación de la cantidad de vértices y polígonos que componen los modelos es significativa ( ver Figura ?? ). Lo que los hace ideales para probar algoritmos bajo distintos requerimientos de carga de memoria.

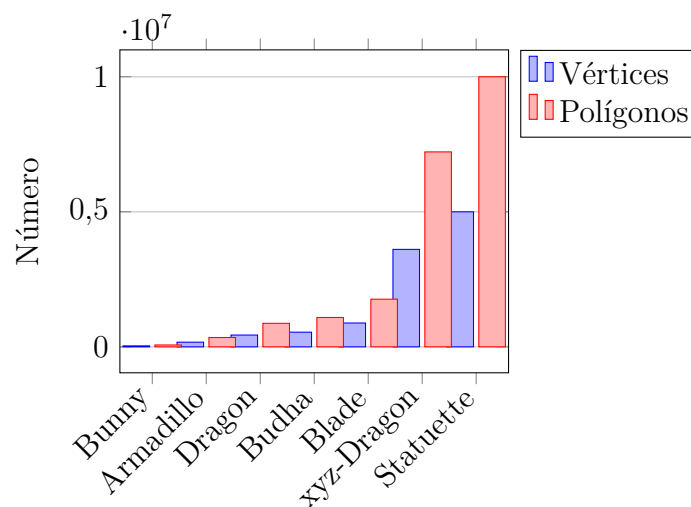


Figura 4.13: Los polígonos duplican en cantidad a los vértices

Los modelos tridimensionales son constituidos por polígonos, como se puede apreciar en la comparativa cada modelo tiene una mayor cantidad en comparación con los demás. Los modelos de la Figura ?? se renderizaron con el escenario Cornellbox ( Figura ?? ), esto se realizó para establecer los escenarios base, los cuales serán comparados con el algoritmo aquí propuesto, en los cuales se utilizaron las estructuras de datos estudiadas para las pruebas de desempeño presentaron las siguientes características.

### 4.3. MODELOS

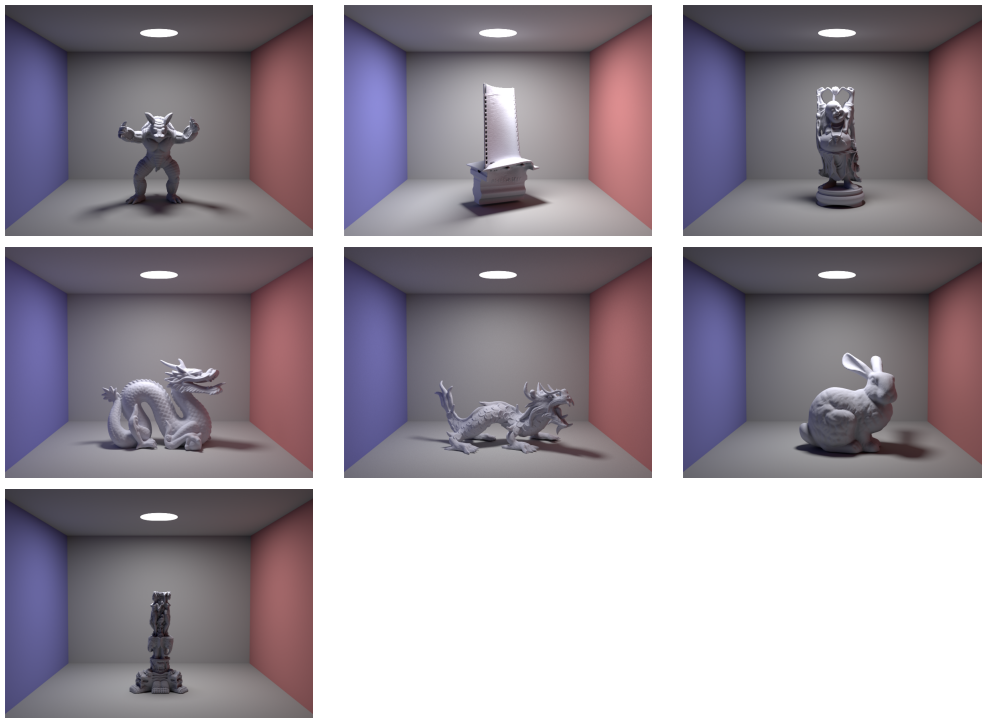


Figura 4.14: Modelos en el escenario CornellBox.

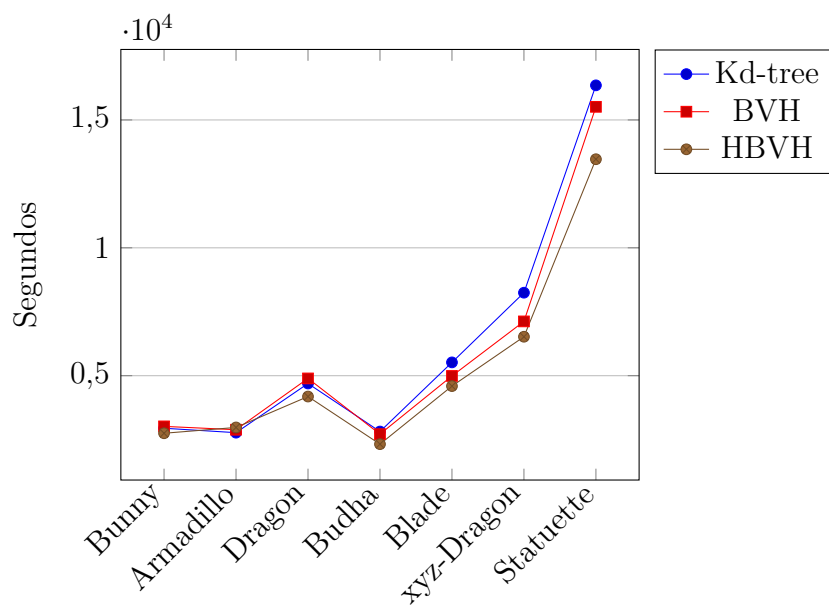


Figura 4.15: Comparación de tiempos entre las estructuras.

En cada uno de los escenarios de la Figura ?? se utilizaron 10 mil fotones por pixel, con las estructuras de datos estudiadas, lo que se muestra en la Figura ??.

La estructura de datos HBVH permite una menor cantidad de tiempo en comparación con las estructuras de datos Kd-tree y BVH, a partir de esta se establece la imagen de comparación base. La memoria que cada uno de estos modelos ocupa se

muestra en la Figura ??.

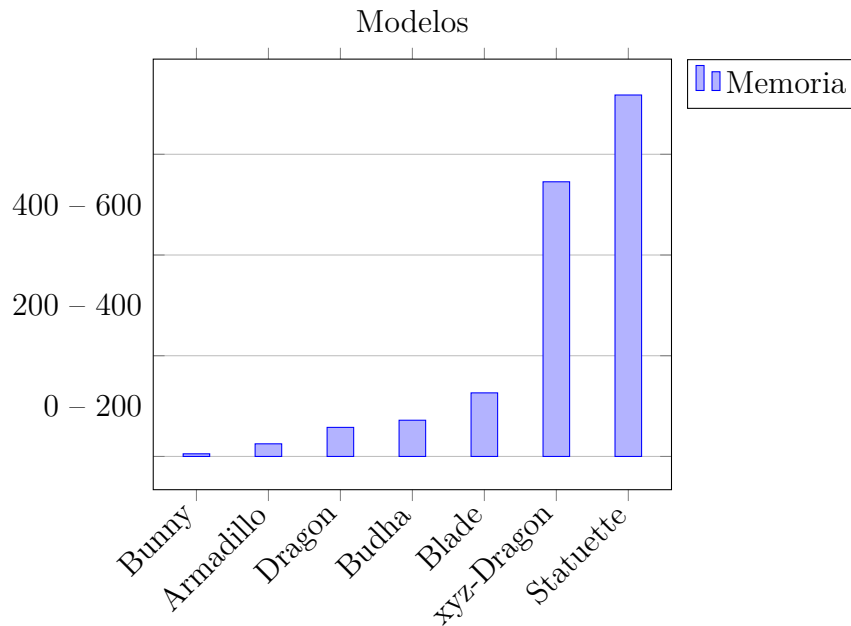


Figura 4.16: Los escenarios de la Figura ?? con las distintas estructuras de datos.

La estructura de datos utilizada para manejar estos modelos fue HBVH, la cual como se ha visto en comparación con otras estructuras de datos requiere menor tiempo de construcción, otro factor a considerar es la memoria que se requiere para cargar cada uno de los modelos que se han utilizado en esta investigación.

## 4.4 MÉTRICAS

Las imágenes obtenidas con el método de renderizado aquí propuesto ha sido logrado con distintas cantidades de fotones vertidos sobre el escenario, por ello lograr un número aceptable de exposición, es necesario establecer una imagen base, la cual ha sido expuesta a una cantidad considerable de fotones, logrando así los efectos y la imagen deseada. Debido a que la generación de fotones es de forma aleatoria, se genera ruido en la imagen, con una exposición suficiente se logra disminuir dicho ruido y esta converge en una calidad deseada, una sobre exposición consume tiempo de cálculo.

### 4.4.1 ERROR CUADRÁTICO MEDIO

Para encontrar las diferencias entre dos imágenes es útil usar el error cuadrático medio ( *Mean Square Error* ) [50] el cual mide la diferencia entre el estimador y lo que se estima, es decir una imagen con una gran cantidad de fotones la cual se

#### 4.4. MÉTRICAS

establece como imagen base y las demás que son con menores cantidad.

Sean  $C$  y  $S$  las matrices de las imágenes a comparar, donde  $N$  y  $M$  son valores correspondientes a las filas y columnas de la imagen.

$$MSE = \frac{1}{N \cdot M} \sum_{i=1}^N \sum_{j=1}^M [C_{i,j} - S_{i,j}]^2 \quad (4.7)$$

Entre más pequeño sea el error cuadrático medio entre las dos imágenes, la similitud es mayor, por lo que en caso de que el MSE sea igual a cero, significa que la imagen es la misma.

##### 4.4.2 RELACIÓN SEÑAL A RUIDO DE PICO

---

La relación señal a ruido de pico *PSNR* (*Peak Signal to Noise Ratio*) [50]. Esto permite relacionar el máximo valor posible del pixel en comparación con el valor de la imagen de interés.

Para calcular el PSNR, es necesario calcular previamente el MSE, que como se ha mencionando, indica el error cuadrático entre las dos imágenes, en las cuales en cuanto mayor sea el error, la diferencia entre estas es más evidente. Los valores típicos que adopta PSNR están entre 30 y 50 dB, siendo mayor cuanto mejor es la calidad de la imagen. Un pequeño MSE resulta en un bajo rango de ruido, sí el MSE tiende a cero, entonces el PSNR tiende a infinito, es decir el incremento del valor PSNR indica mejoramiento en la calidad de la imagen.

$P$  indica las intensidades de los valores del pixel para cada canal, los cuales están entre 0 y 255, por tanto  $P = 2^8 - 1$ .

$$PSNR = 10 \cdot \log \frac{P^2}{MSE} \quad (4.8)$$

El ruido de la imagen disminuye al aumentar la cantidad de fotones a calcular para la generación de la imagen final, este incremento de fotones repercute directamente, tanto en la calidad de la imagen como el tiempo necesario para que sean calculados y por tanto en la obtención de la imagen final. Cuanto mayor sea el número de fotones en los algoritmos de renderizado tradicional.

La Figura ?? del modelo *Statuette* es establecida como imagen base, al igual que la Figura ?? del modelo *dragón*, las cuales son comparadas con distintas cantidades de fotones, en la Figura ?? se muestra dicha comparación para el modelo *Statuette* y en la Figura ?? para el modelo *dragón*, el resultado del MSE se muestra en la Figura ?? y el PSNR correspondiente en la Figura ??.

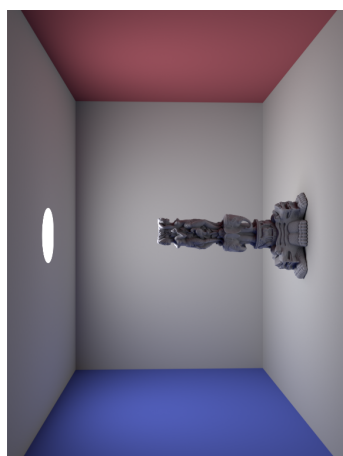


Figura 4.17: Imagen base de statuette con 10 mil fotones.

Modelo Statuette.

	<p>Fotones: 1000 MSE: 0.236920 PSNR: 54.384779</p>
	<p>Fotones: 2000 MSE: 0.235060 PSNR: 54.419017</p>
	<p>Fotones: 5000 MSE: 0.010509 PSNR: 67.915045</p>

Figura 4.18: Modelo Statuette con distintas cantidades de fotones, medición de MSE y PSNR.

#### 4.4. MÉTRICAS

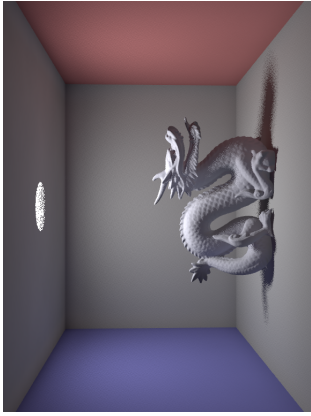


Figura 4.19: Imagen base dragón.

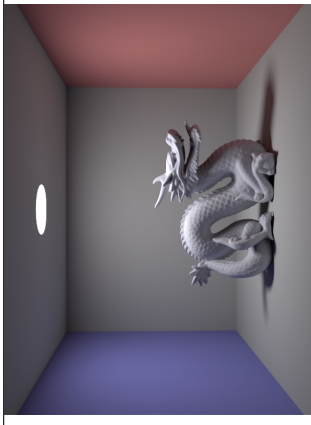
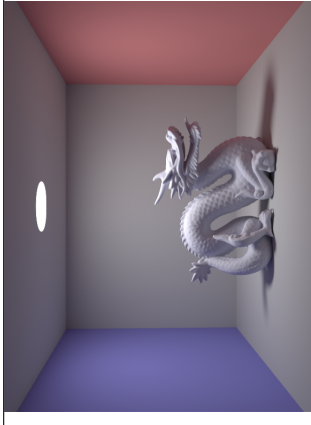
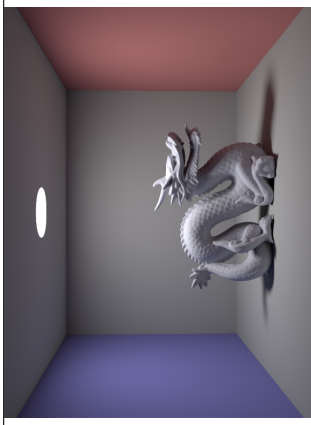
Modelo Dragón.		
		
Fotones: 1000 MSE: 0.236920 PSNR: 54.384779	Fotones: 2000 MSE: 0.235060 PSNR: 54.419017	Fotones: 5000 MSE: 0.010509 PSNR: 67.915045

Figura 4.20: MSE y PSNR del modelo dragón.

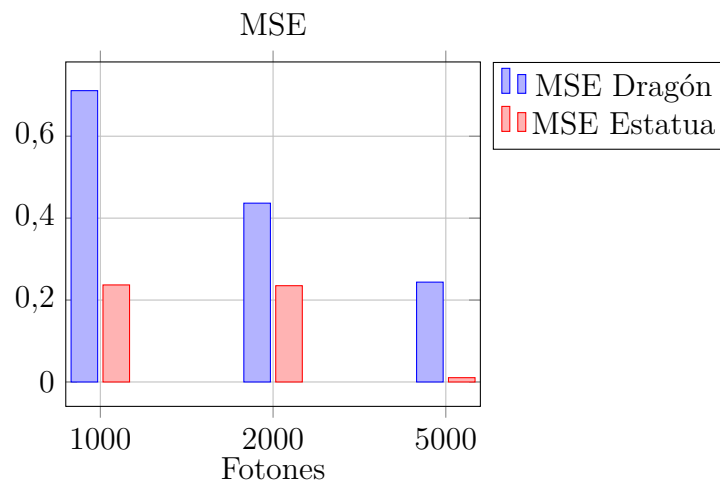


Figura 4.21: El error cuadrático disminuye al aumentar la cantidad de fotones sobre el escenario.

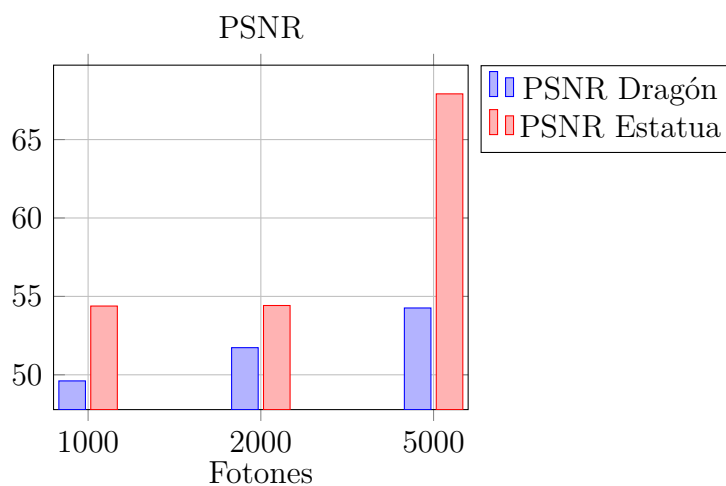


Figura 4.22: Una mayor cantidad de fotones aumenta la calidad de la imagen.

El renderizado de las superficies difusas no logra mostrar todas las características del comportamiento de la luz que se esperan ver en la naturaleza, particularmente efectos como cáusticas y reflejos, por ello la comunidad de gráficos por computadora propuso el escenario *CornellBox* [29], el cual fue estudiado en la universidad de Cornell, donde la luz incidente sobre determinada superficie afectaba el resultado de iluminación de sobre las demás.

La Figura ?? es tomada como imagen base, ya que en ella se utilizaron una gran cantidad de fotones, logrando el efecto deseado, imperfecciones de la iluminación del escenario son nulas, y los detalles de reflejos, iluminación directa e indirecta, así como cáusticas se pueden apreciar en el escenario, la imagen es renderizada con 10 mil fotones, en los cuadros azules se hace un acercamiento a las regiones donde comúnmente se requiere una mayor cantidad de fotones.

#### 4.4. MÉTRICAS

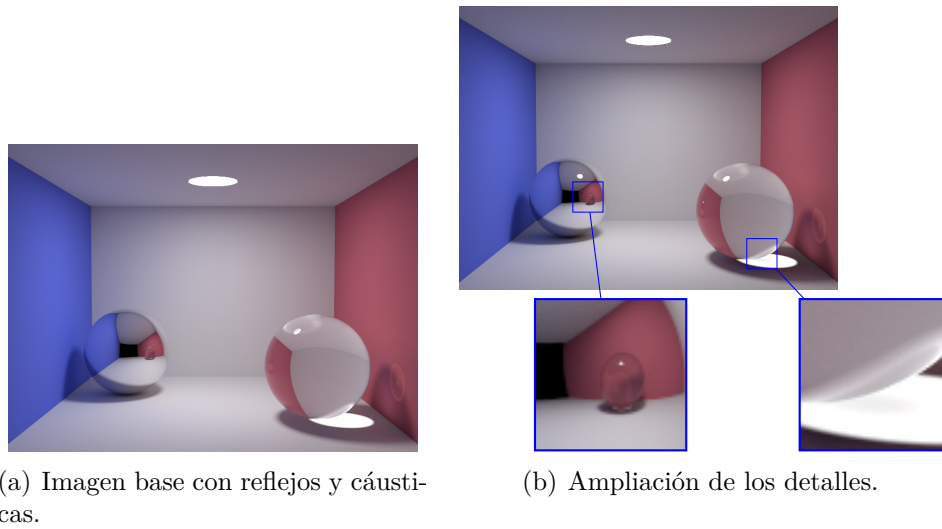


Figura 4.23: Imagen base renderizada con 10 mil fotones

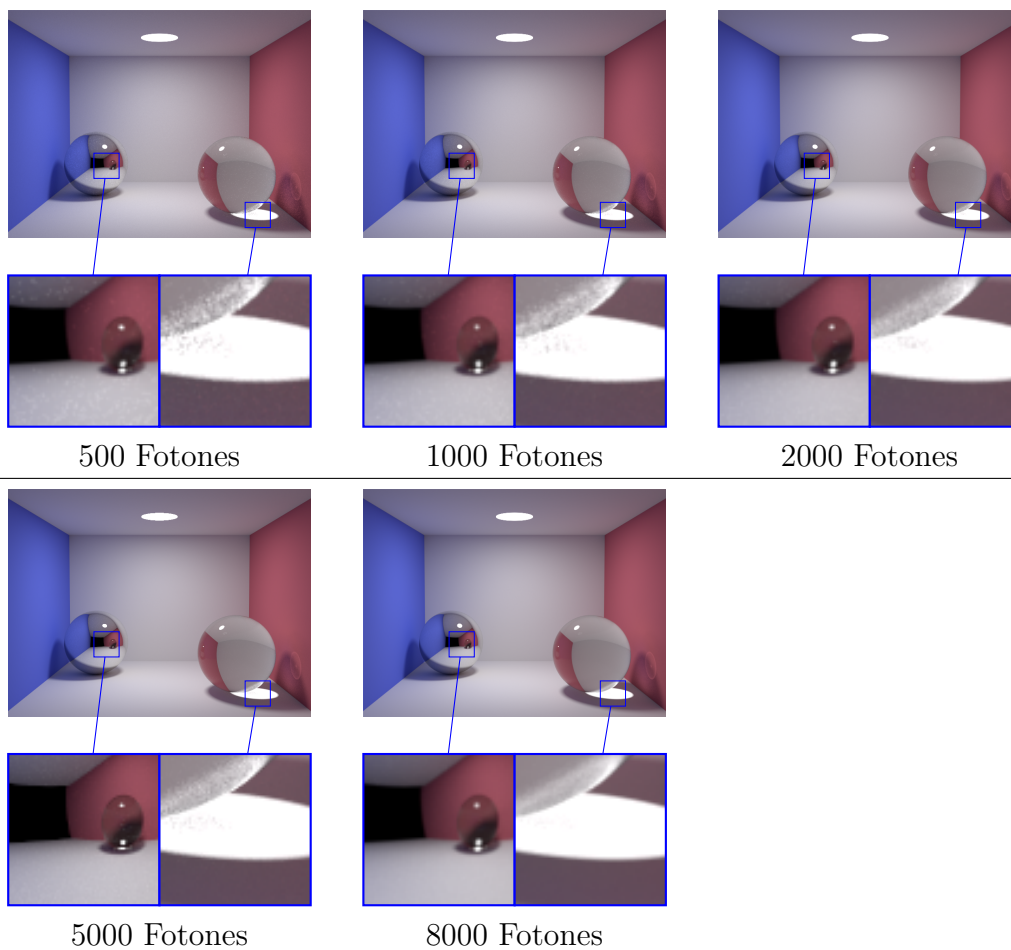


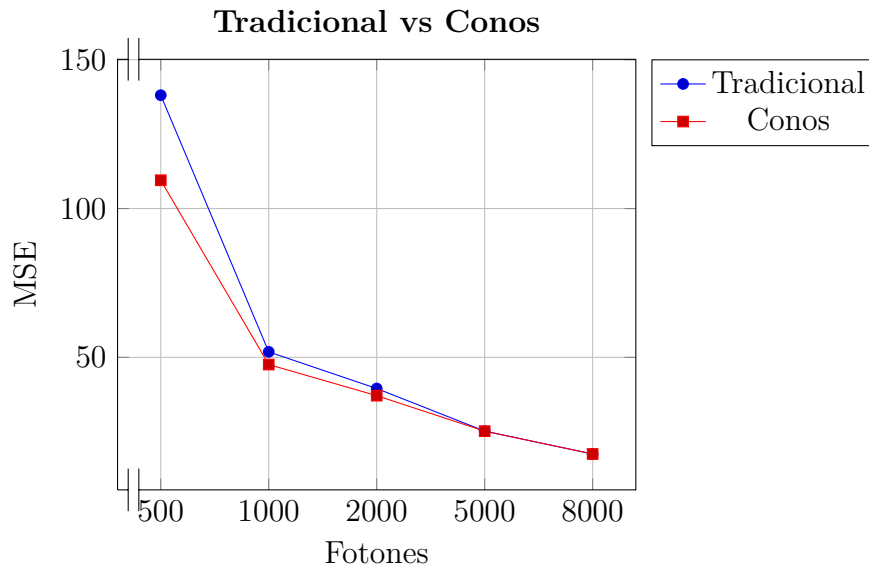
Figura 4.24: Imágenes del escenario cornell box, con distintas cantidades de fotones.

Las imágenes de obtenidas ( ver Figura ?? ) son comparadas con la imagen base, las cuales deben de alcanzar con la menor cantidad de fotones y en el menor tiempo

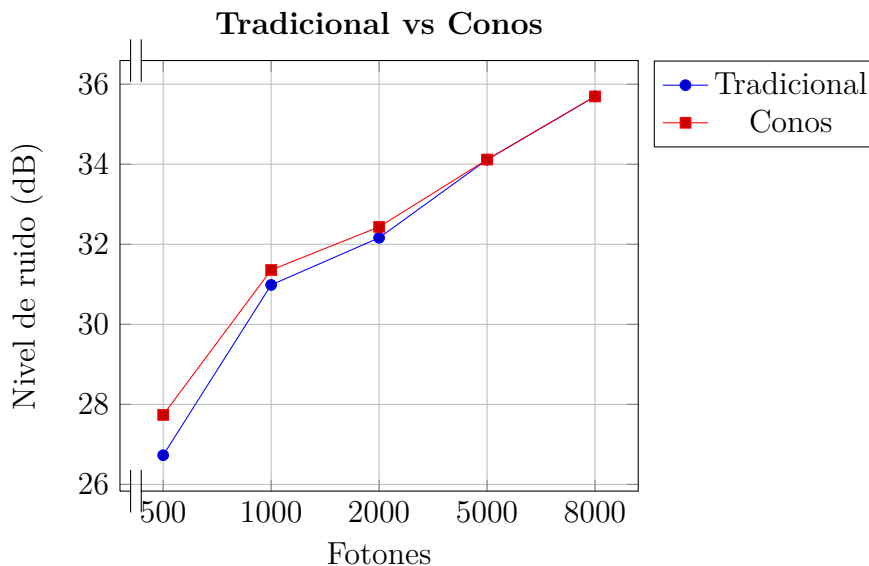
## CAPÍTULO 4. PROPUESTA Y ANÁLISIS DE RESULTADOS

posible los efectos visuales de la imagen base.

Al hacer una comparativa entre las distintas cantidades de fotones utilizados para renderizar las imágenes de la Figura ??, se puede apreciar en la Figura ?? que la diferencia entre las distintas cantidades de fotones y la imagen base disminuye al incrementar el número de fotones, por lo que la fidelidad aumenta, esto se muestra en la Figura ??.



(a) MSE del escenario CornellBox



(b) PSNR del escenario CornellBox

Figura 4.25: MSE y PSNR del escenario CornellBox.

En la Figura ?? se resalta la diferencia entre el render tradicional y el esquema de conos, utilizando la misma cantidad de fotones, se puede observar que en el esquema de conos el ruido es más difuminado, se hace una ampliación en las áreas

#### 4.4. MÉTRICAS

mas notables.

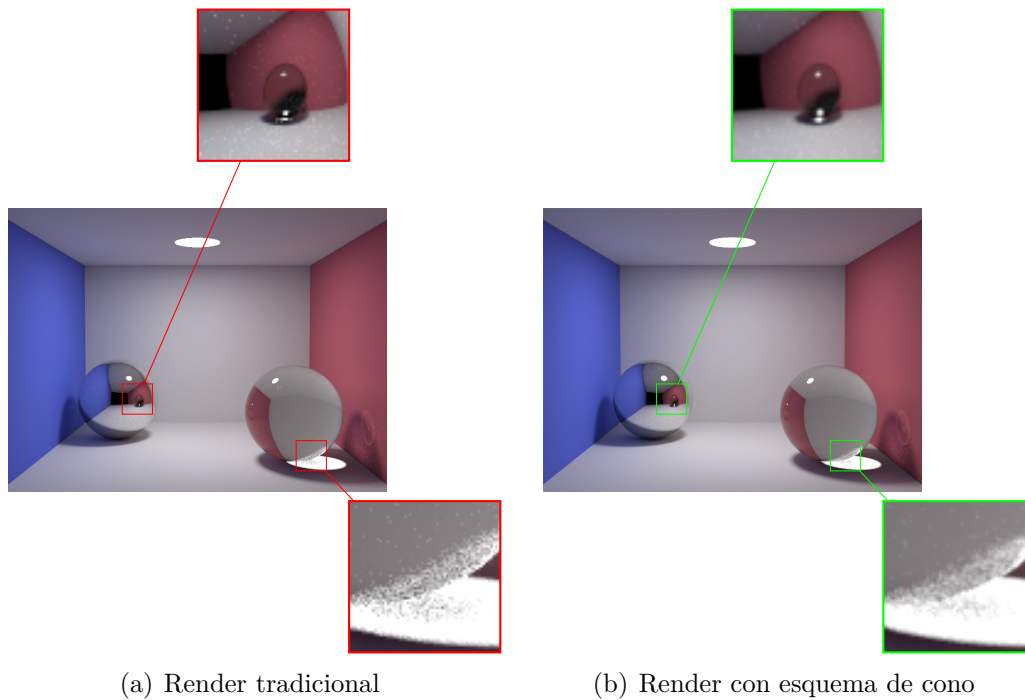


Figura 4.26: Render tradicional vs render por conos

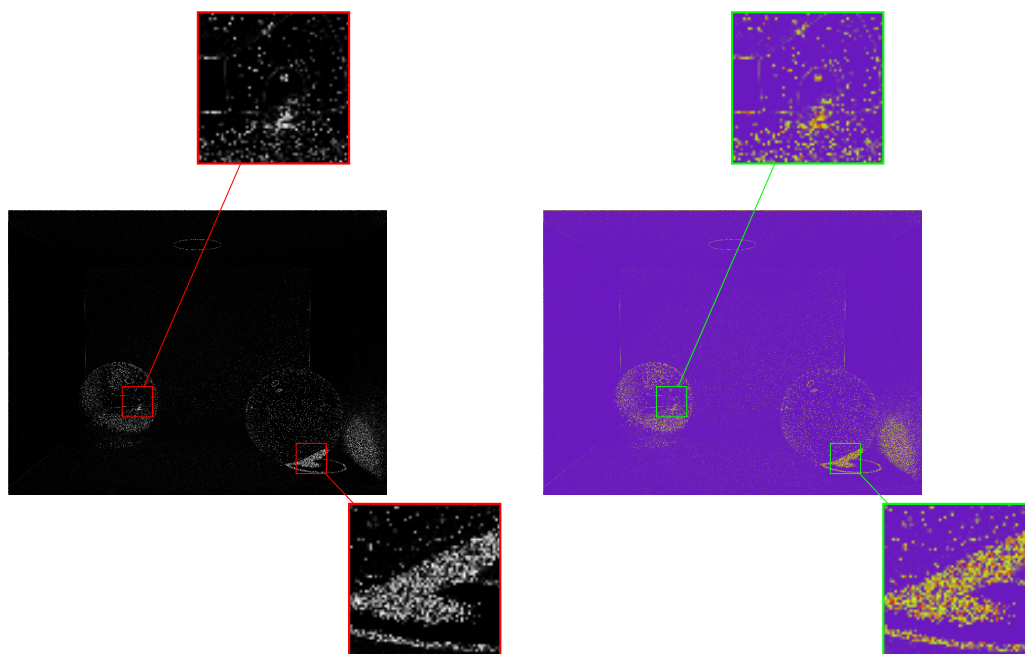


Figura 4.27: Se muestra la diferencia de las intensidades de los píxeles coincidentes.

El algoritmo de trazado de rayos tradicional define la iluminación para cada pixel, lo

## CAPÍTULO 4. PROPUESTA Y ANÁLISIS DE RESULTADOS

que puntos de iluminación quedan bien definidos y delimitados, de la misma forma resulta cuando un rayo no converge en un pixel de la imagen, quedando un efecto de *sal y pimienta*, con el enfoque de conos se logra reducir esta característica, permitiendo iluminar un conjunto de píxeles en menor tiempo.

Cuando la iluminación del pixel converge con la tonalidad de la imagen base, la diferencia entre estas dos es igual a cero ( ver Figura ?? ), lo que se espera es que los píxeles de la imagen por conos alcancen los tonos de la imagen base en menor tiempo en comparación con el algoritmo de renderizado tradicional.

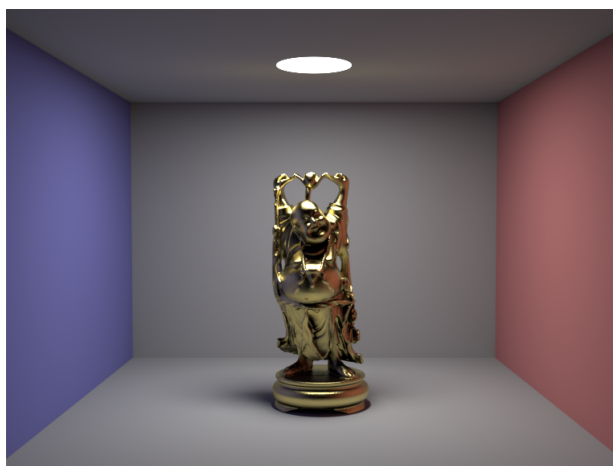
Renderizar el escenario con una menor cantidad de fotones se puede apreciar la diferencia entre el renderizado tradicional y el renderizado por conos, en la cual el ruido disminuye, estas diferencias se muestran como resultado en falso color [51] (ver Figura ??), en la cual la mayor diferencia se han resaltado en color rojo, mientras que la convergencia de valores tienden al color morado.

Una vez que la propuesta converge en tiempos adecuados ( con menor tiempo posible) se realizaron pruebas sobre los modelos propuestos por la comunidad ( ver Figura ?? y Figura ?? ), utilizando diferentes texturas que tienen un grado de dificultad.

#### 4.4. MÉTRICAS



(a) Material de cristal



(b) Material metal oro con superficie no lambertiana



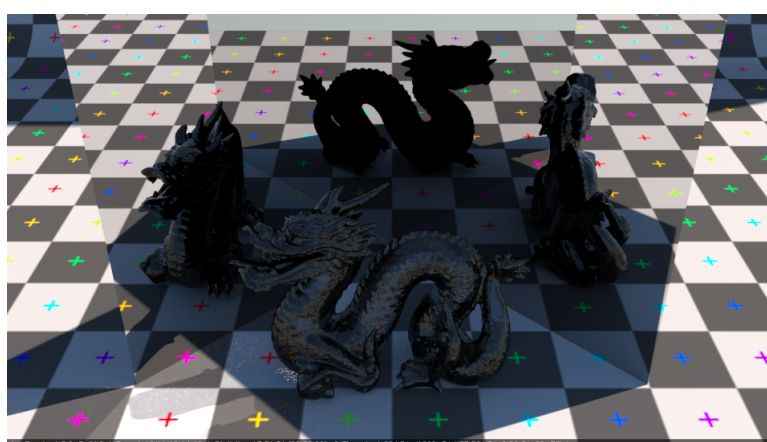
(c) Material metal oro con superficie lambertiana

Figura 4.28: Modelo Buddha con distintos materiales.

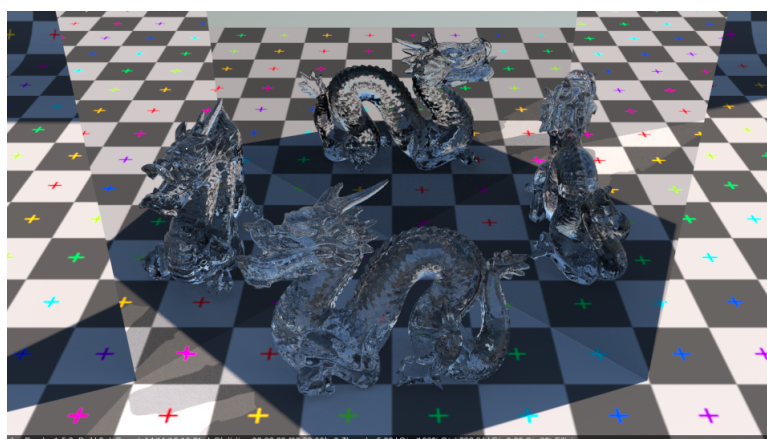
## CAPÍTULO 4. PROPUESTA Y ANÁLISIS DE RESULTADOS



(a) Segunda recursion interacción entre la luz y los objetos.



(b) Tercera recursion interacción de la luz.



(c) Cuarta recursion de la luz.

Figura 4.29: Dragón frente al espejo.

El modelo dragón tiene un efecto de material cristal, el cual esta frente a un espejo, lo que se puede apreciar en estas imágenes es el efecto de recursion el cual incrementa el realismo visual.

## 4.5. CONCLUSIONES

### 4.5 CONCLUSIONES

---

Después de implementar un trazador de rayos de forma cónica podemos ver algunas características que los métodos tradicionales no logran fácilmente, esto se ve específicamente en efectos físicos de la luz, como lo son las sombras, reflejos, transparencias y las denominadas cáusticas.

Comparando las técnicas tradicionales con respecto al tiempo de renderizado podemos decir que las igualamos o sobrepasamos con segundos, esto pudiera verse como una desventaja a la vista del tiempo de obtención de la imagen, sin embargo la técnica propuesta logra casi en el mismo tiempo obtener fenómenos cáusticos, que los métodos tradicionales les toma el doble de tiempo en este caso será decisión del usuario, decidir qué imagen le resuelve mejor el problema.

Se renderizaron objetos tridimensionales como lo es el Buddha y el dragón con materiales difíciles de procesar como lo es el oro y el cristal, logrando la obtención de las imágenes con la mayoría de las cáusticas que los métodos tradicionales no obtienen.

Hablando de renderizado en tiempo real, este solamente se aborda en la división de la estructura en voxeles, dejando al CPU y GPU's tomar la decisión de paralelizar los cálculos. En este caso dejamos la implementación de las estructuras paralelizables como trabajo futuro.

El uso del error medio cuadrático como comparativa entre las imágenes obtenidas nos permite validar la obtención de características adicionales sobre los métodos tradicionales, sin embargo otro tipo de validación se hace necesaria para afirmar que las imágenes obtenidas se asemejan a la realidad.

La posibilidad de representar a la luz por medio de algoritmos ha permitido la creación de imágenes foto-realistas gracias a las propiedades de la luz sobre los objetos. La utilización de fenómenos ópticos nos permite observar comportamientos observados en ambientes naturales que hemos simulado en los ambientes virtuales.

El renderizado basado en conos permite renderizar modelos tridimensionales obteniendo cáusticas que los métodos tradicionales no logran fácilmente o requieren de un tiempo considerable.

La estructura utilizada permite optimizar el uso de la memoria logrando tratar objetos tridimensionales con un volumen considerable de polígonos que la conforman.

En nuestro caso utilizamos materiales basados en su física, lo que nos permitió reducir el tiempo con un mayor número de detalle en los objetos renderizados.

Se utilizaron los modelos propuestos por la comunidad de gráficos por computadora, los cuales debido a las distintas cantidades de polígonos que los componen los hacen ideales para medir el desempeño tanto de memoria, como de tiempo.

## 4.6 TRABAJO FUTURO

---

Es necesario generar una librería que contenga texturas que puedan ser fácilmente utilizadas en los ambientes virtuales, sobre todo aquellos denominados dinámicos, por ejemplo los video juegos.

La caracterización de materiales requiere un mayor estudio para poder obtener materiales semi-translucidos como son pantallas luminosas que dejan pasar parcialmente la luz.

Sí bien la tecnología de hoy día cuenta con procesadores constituidos por varias unidades de procesamiento que permiten aumentar la velocidad en los cálculos, sobre todo hablando de raytracing, es necesario que las estructuras de datos puedan ser paralelizables bajo paradigmas como CUDA, MPI y Clustering.

## APÉNDICE A

### ESTRUCTURA DE DATOS KD-TREE

---

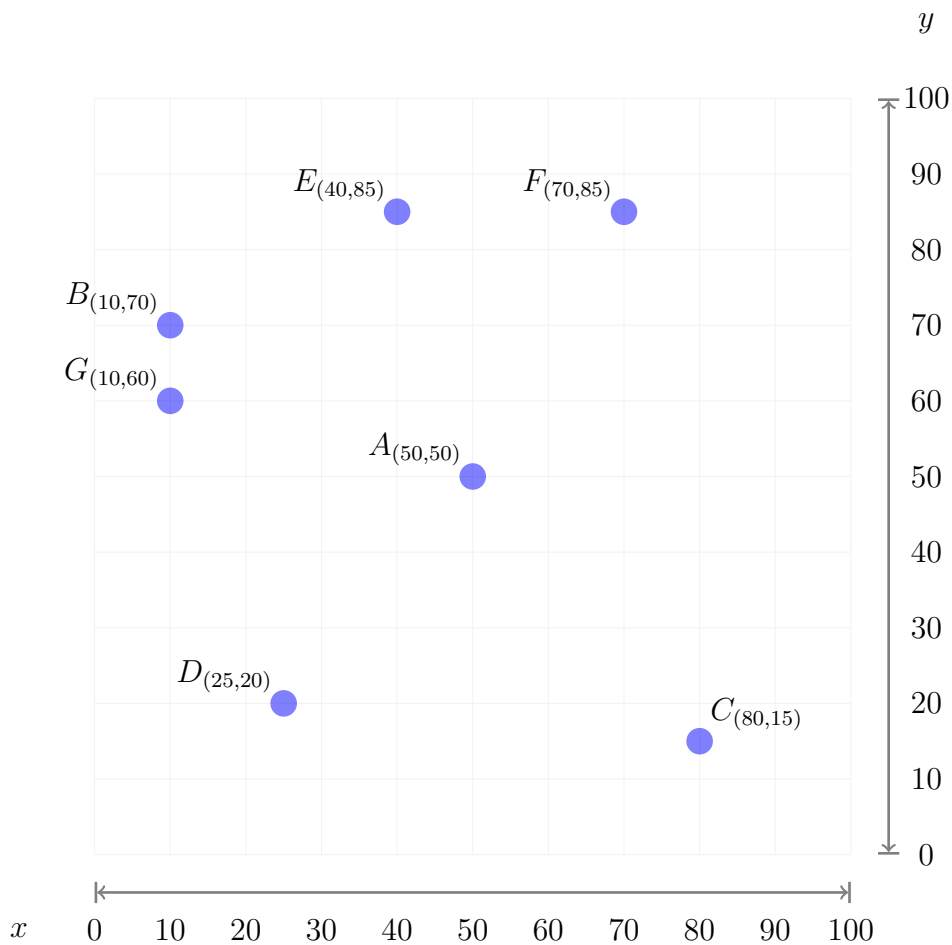
La estructura de datos Kd-tree divide el espacio en dos por cada dimension que constituya la k-tupla, aquí se describe un ejemplo. Cada nodo se encuentra en un espacio  $\mathbb{R}^k$ , por lo que un nodo  $P$  está constituido por  $k - 1$  valores de forma que se tiene un nodo  $P(v_0, v_1, \dots, v_{k-1})$ .

El nodo  $Q$  tendrá dos hijos; hijos mayores, en el cual se almacenan los datos que tienen un mayor valor que el nodo actual  $P$ , denominándose  $HISON(P)$ , por el contrario, si el valor es menor que el nodo  $Q$ , será  $LOSON(P)$ , el discriminante  $DISC(P)$ , el cual solo toma en cuenta la dimension en la que se esta operando para realizar las comparaciones, descartando de ésta forma las demás, así solo se tiene que realizar una comparación por dimension.

El árbol al estar vacío toma el primer dato entrante como nodo raíz, los siguientes datos  $P$  se comparan con el nodo del árbol  $Q$ , el nodo a consultar es parte del árbol. El nodo divide el espacio  $k_0$  en dos, el siguiente dato  $P$  se comparara en esta dimension y solo puede ser mayor o menor, por lo que si es mayor se identifica como hijo mayor de  $P$  ( $HISON(P)$ ), en caso contrario se identifica como hijo menor de  $P$  ( $LOSON(P)$ ).

Entonces tenemos que  $Q_{k_0} > P_{k_0}$ , si es mayor  $HISON(P)$ , si es menor  $LOSON(P)$ , el proceso se repite, a medida que se desciende en el árbol la dimensionalidad a comparar es  $(i + 1) \bmod k$ , por lo que con cada nivel se compara  $K_{n-1}$ .

Se tiene un conjunto de puntos :  $A_{(50,50)}$ ,  $B_{(10,70)}$ ,  $C_{(80,15)}$ ,  $D_{(25,20)}$ ,  $E_{(40,85)}$ ,  $F_{(70,85)}$  y  $G_{(10,60)}$ .



(a)

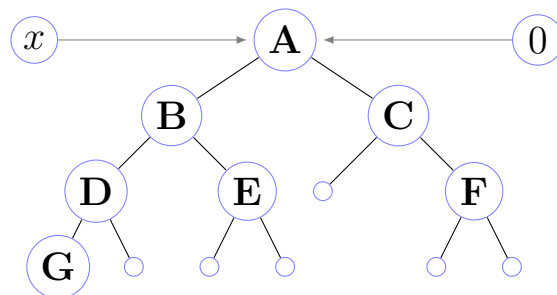
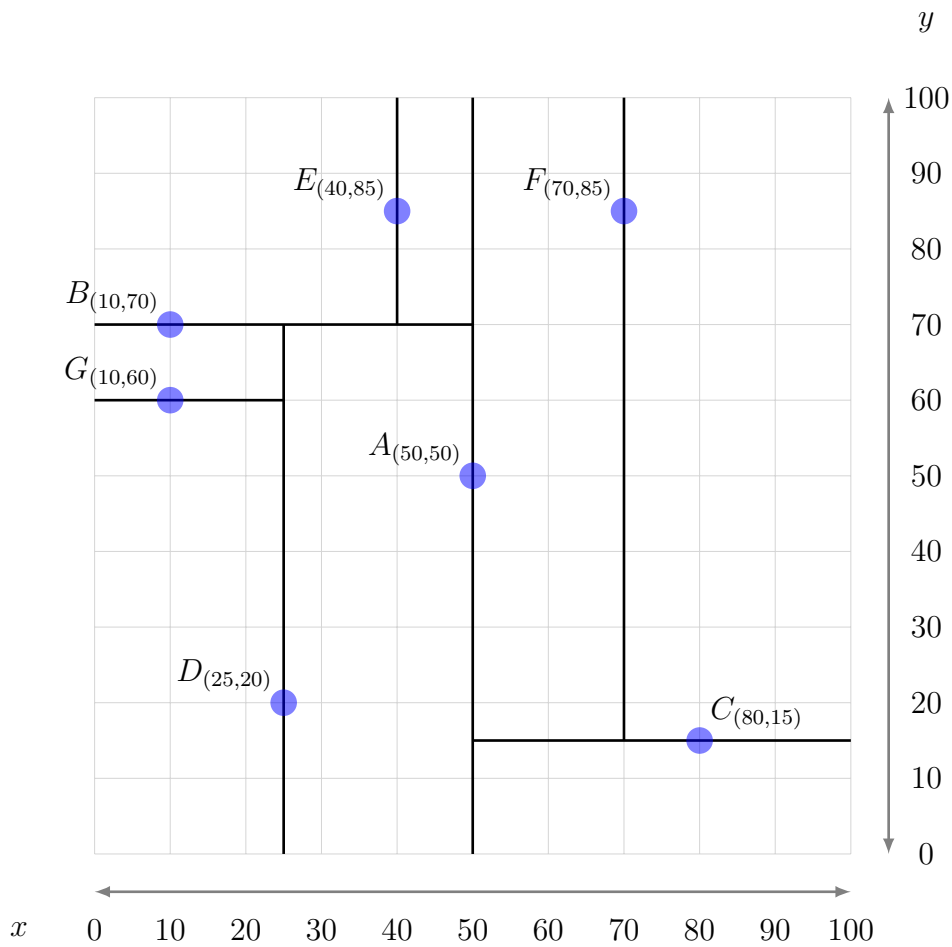
$A_{(50,50)}$  es primer dato a graficar, se toma como la raíz del árbol, como  $A$  esta constituido por dos dimensiones  $x$  y  $y$ , estas se identificaran como  $k_0$  y  $k_1$  respectivamente, por fines de claridad se denotará como  $A_{(k_0,k_1)}$ .

Nodo	$k_0$	$k_1$	$k_0$	Nodo
A	$A_{k_0}$			raíz
B	$A_{k_0} B_{k_0}$			$A : LOSON(B)$
C	$A_{k_0} C_{k_0}$			$A : HISON(C)$
D	$A_{k_0} D_{k_0}$	$B_{k_1} D_{k_1}$		$B : LOSON(D)$
E	$A_{k_0} E_{k_0}$	$B_{k_1} E_{k_1}$		$B : HISON(E)$
F	$A_{k_0} F_{k_0}$	$C_{k_1} F_{k_1}$		$C : HISON(F)$
G	$A_{k_0} G_{k_0}$	$B_{k_1} G_{k_1}$	$D_{k_0} G_{k_0}$	$D : LOSON(G)$

Al utilizar los valores de cada dato, se obtiene el siguiente cuadro:

Nodo	$k_0$	$k_1$	$k_0$	Nodo
A	$A_{50}$			raíz
B	$A_{50} B_{10}$			$A : LOSON(B)$
C	$A_{50} C_{80}$			$A : HISON(C)$
D	$A_{50} D_{25}$	$B_{70} D_{25}$		$B : LOSON(D)$
E	$A_{50} E_{40}$	$B_{70} E_{85}$		$B : HISON(E)$
F	$A_{50} F_{70}$	$C_{15} F_{85}$		$C : HISON(F)$
G	$A_{50} G_{10}$	$B_{70} G_{60}$	$D_{25} G_{10}$	$D : LOSON(G)$

La ubicación de los puntos después procesarlos.





## BIBLIOGRAFÍA

- [1] A. Appel, “Some techniques for shading machine renderings of solids,” in *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference, AFIPS '68 (Spring)*, (New York, NY, USA), pp. 37–45, ACM, 1968.
- [2] T. Whitted, “An improved illumination model for shaded display,” in *ACM Siggraph 2005 Courses*, p. 4, ACM, 2005.
- [3] T. L. Kay and J. T. Kajiya, “Ray tracing complex scenes,” in *ACM SIGGRAPH computer graphics*, vol. 20, pp. 269–278, ACM, 1986.
- [4] M. a. B. Donald Hearn, *Graficos Por Computadora con OpenGL*. Prentice Hall, 3a. ed. ed., 2004.
- [5] J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips, *Introduction to computer graphics*, vol. 55. Addison-Wesley Reading, 1994.
- [6] P. HELLARD, “Avatar. driving the avatar characters. weta digital’s jeff unay talks about the artistic team behind the faces,” *CGSociety, Society of Digital Artists: Production Focus* (<http://features.cgsociety.org/story-custom.php>), 2010.
- [7] A. S. Glassner, *An introduction to ray tracing*. Morgan Kaufmann, 1989.
- [8] I. Wald, W. R. Mark, J. Günther, S. Boulos, T. Ize, W. Hunt, S. G. Parker, and P. Shirley, “State of the art in ray tracing animated scenes,” in *Computer Graphics Forum*, vol. 28, pp. 1691–1722, Wiley Online Library, 2009.
- [9] U. O. Pinto, *Surface Mesh Characterization of Dynamic Scenes for Ray Tracing in Real-time*. PhD thesis, Cinvestav del IPN Unidad Guadalajara, 2014.
- [10] M. Wand, M. Fischer, I. Peter, F. Meyer auf der Heide, and W. Straßer, “The randomized z-buffer algorithm: Interactive rendering of highly complex scenes,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 361–370, ACM, 2001.
- [11] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” *Commun. ACM*, vol. 18, pp. 509–517, Sept. 1975.

- [12] H. Samet, *Foundations of multidimensional and metric data structures*. Morgan Kaufmann, 2006.
- [13] D. Gordon and S. Chen, “Front-to-back display of bsp trees,” *IEEE computer Graphics and Applications*, vol. 11, no. 5, pp. 79–85, 1991.
- [14] G. Zachmann and E. Langetepe, “Geometric data structures for computer graphics,” in *Proceedings of Eurographics 2002 Tutorials*, The Eurographics Association, Sept. 2002. Tutorial.
- [15] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf, “Computational geometry,” in *Computational geometry*, pp. 1–17, Springer, 2000.
- [16] J. U. Alfred V. Aho, John E. Hopcroft, *Estructuras de Datos y Algoritmos*. 1988.
- [17] R. Finkel and J. Bentley, “Quad trees a data structure for retrieval on composite keys,” vol. 4, no. 1, 1974.
- [18] H. Samet, *The Design and Analisis of Spatial Data Structures*. Addison-Wesley Publishing Company, Inc., 1994.
- [19] H. Eberhardt, V. Klumpp, and U. D. Hanebeck, “Density trees for efficient nonlinear state estimation,” in *Information Fusion (FUSION), 2010 13th Conference on*, pp. 1–8, IEEE, 2010.
- [20] H. J. Haverkort, “Introduction to bounding volume hierarchies,” *Part of the PhD thesis, Utrecht University*, 2004.
- [21] K. A. Andersen and C. Bay, “A survey of algorithms for construction of optimal heterogeneous bounding volume hierarchies,” *Saatavilla pdf-muodossa <http://image.diku.dk/projects/media/christian.bay.kasper.andersen.B>*, vol. 6, 2006.
- [22] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, “Fast proximity queries with swept sphere volumes,” tech. rep., Technical Report TR99-018, Department of Computer Science, University of North Carolina, 1999.
- [23] L. Chen, J. Dai, J. Feng, B. Fu, and H. Pan, “Hybrid bounding volume hierarchy tree (hbvht) algorithm for collision detection of articulated model robot,” *Information Technology Journal*, vol. 13, no. 10, p. 1723, 2014.
- [24] J. D. MacDonald and K. S. Booth, “Heuristics for ray tracing using space subdivision,” *The Visual Computer*, vol. 6, no. 3, pp. 153–166, 1990.
- [25] D. Malacara, *Handbook of optical engineering*. CRC Press, 2001.
- [26] F. L. Pedrotti and L. S. Pedrotti, “Introduction to optics,” *Prentice-Hall, New Jersey*, 1993.

- [27] E. Heitz, “Understanding the masking-shadowing function in microfacet-based brdfs,” *Journal of Computer Graphics Techniques (JCGT)*, vol. 3, pp. 32–91, June 2014.
- [28] K. E. Torrance and E. M. Sparrow, “Theory for off-specular reflection from roughened surfaces,” *JOSA*, vol. 57, no. 9, pp. 1105–1112, 1967.
- [29] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile, “Modeling the interaction of light between diffuse surfaces,” in *ACM SIGGRAPH Computer Graphics*, vol. 18, pp. 213–222, ACM, 1984.
- [30] M. Oren and S. K. Nayar, “Generalization of lambert’s reflectance model,” in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 239–246, ACM, 1994.
- [31] J. F. Blinn, “Models of light reflection for computer synthesized pictures,” in *Seminal graphics*, pp. 103–109, ACM, 1998.
- [32] T. Ritschel, C. Dachsbacher, T. Grosch, and J. Kautz, “The state of the art in interactive global illumination,” in *Computer Graphics Forum*, vol. 31, pp. 160–188, Wiley Online Library, 2012.
- [33] S. D. Roth, “Ray casting for modeling solids,” *Computer graphics and image processing*, vol. 18, no. 2, pp. 109–144, 1982.
- [34] T. Akenine-Moller, E. Haines, and N. Hoffman, *Real-time rendering*. AK, 2002.
- [35] J.-M. Hasenfratz, M. Lapierre, N. Holzschuch, and F. Sillion, “A survey of real-time soft shadows algorithms,” in *Computer Graphics Forum*, vol. 22, pp. 753–774, Wiley Online Library, 2003.
- [36] A. Woo, P. Poulin, and A. Fournier, “A survey of shadow algorithms,” *Computer Graphics and Applications, IEEE*, vol. 10, no. 6, pp. 13–32, 1990.
- [37] H. Gouraud, “Continuous shading of curved surfaces,” *IEEE transactions on computers*, vol. 100, no. 6, pp. 623–629, 1971.
- [38] R. B. Lotto, S. M. Williams, and D. Purves, “Mach bands as empirically derived associations,” *Proceedings of the National Academy of Sciences*, vol. 96, no. 9, pp. 5245–5250, 1999.
- [39] B. T. Phong, “Illumination for computer generated pictures,” *Commun. ACM*, vol. 18, pp. 311–317, June 1975.
- [40] M. F. Cohen and J. R. Wallace, *Radiosity and realistic image synthesis*. Elsevier, 2012.
- [41] H. W. Jensen, “Global illumination using photon maps,” in *Rendering Techniques’ 96*, pp. 21–30, Springer, 1996.
- [42] J. T. Kajiya, “The rendering equation,” in *ACM Siggraph Computer Graphics*, vol. 20, pp. 143–150, ACM, 1986.

- [43] M. Pharr and G. Humphreys, *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2010.
- [44] I. M. Sobol, “Método de montecarlo,” tech. rep., 1976.
- [45] M. H. Kalos and P. A. Whitlock, *Monte carlo methods*, vol. 1. John Wiley & Sons, 2008.
- [46] C. Crassin, F. Neyret, M. Sainz, S. Green, and E. Eisemann, “Interactive indirect illumination using voxel cone tracing,” in *Computer Graphics Forum*, vol. 30, pp. 1921–1930, Wiley Online Library, 2011.
- [47] J. Amanatides, “Ray tracing with cones,” in *ACM SIGGRAPH Computer Graphics*, vol. 18, pp. 129–135, ACM, 1984.
- [48] P. Laurila, “Geometry culling in 3d engines,” *Contribution culling, PVS-based arbitrary geometry occlusion culling, cell-based occlusion culling, back-face culling, view frustum culling, algorithms*, 2000.
- [49] M. Sugihara, R. Rauwendaal, and M. Salvi, “Layered reflective shadow maps for voxel-based indirect illumination.,” in *High Performance Graphics*, pp. 117–125, 2014.
- [50] R. R. Marie, “Data hiding in digital images using cryptography and steganography techniques (cryptsteg).,” *World of Computer Science & Information Technology Journal*, vol. 4, no. 4, 2014.
- [51] M. I. M. Ortiz, “Apuntes del curso: Procesamiento digital de imagenes,” tech. rep., Facultad de Ciencias de la Computación, BUAP., 2013.