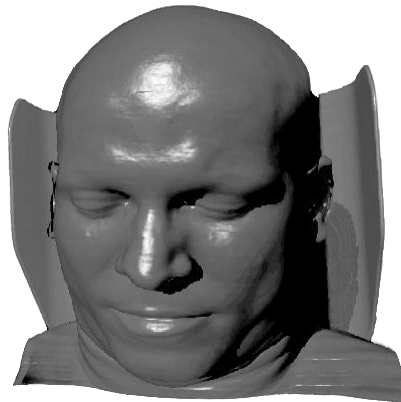


**Universidad de las Ciencias Informáticas  
Facultad 5**



# **MÓDULO DE ILUMINACIÓN PARA VISUALIZACIÓN DIRECTA DE VOLUMEN**

**Trabajo de diploma para optar por el título de  
Ingeniero en Ciencias Informáticas**



**Autor: Rubén Alcolea Núñez**

**Tutor: MSc. Osvaldo Pereira Barzaga  
Co-Tutor: Ing. Ernesto Carrasco De la Torre**

**Mayo 2011**

## Declaración de Autoría

Declaramos ser los únicos autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Rubén Alcolea Núñez

**Autor**

---

MSc Osvaldo Pereira Barzaga

**Tutor**

---

Ing Ernesto Carrasco De la Torre

**Co-tutor**

## Datos de Contacto

**Tutor:** MSc. Osvaldo Pereira Barzaga.

**Edad:** 26.

**Ciudadanía:** Cubano.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** MSc. en Informática Aplicada.

**Categoría Docente:** Instructor.

**E-mail:** opereira@uci.cu

Graduado de la UCI, con cuatro años de experiencia en el tema de la Gráfica Computacional y líder de un proyecto de Realidad Virtual en la Universidad de las Ciencias Informáticas.

**Co-tutor:** Ing. Ernesto Carrasco De la Torre.

**Edad:** 25.

**Ciudadanía:** Cubano.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Ing. en Ciencias Informáticas.

**Categoría Docente:** Instructor Recién Graduado.

**E-mail:** ecarrasco@uci.cu.

Graduado de la UCI, con dos años de experiencia en el tema de la Gráfica Computacional y profesor de un proyecto de Realidad Virtual en la Universidad de las Ciencias Informáticas.

## Dedicatoria

### **A mi madre:**

*Esa persona tan especial y maravillosa que ha sido mi ángel guardián y me ha guiado por el camino de la vida para que yo pueda estar aquí el día de hoy.*

### **A mi padre:**

*Ese fiel compañero y amigo de toda la vida que siempre me supo dar su cariño y me enseñó a caminar y a ser la persona que soy.*

### **A mi hermana:**

*Quién desde que nació hizo más feliz cada uno de mis días y a quien llevo siempre presente en mi corazón.*

### **A mis abuelos:**

*A mis abuelos Pedro, Lilia, Irma y Rubén. A ellos por el cariño que siempre me han brindado y porque siempre los llevo conmigo y son una fuente de inspiración para seguir adelante en los momentos más difíciles.*

## Agradecimientos

*Quisiera agradecerle a cada una de las personas que de una forma u otra han contribuido a mi formación intelectual y han hecho posible la conclusión de este trabajo.*

### **Especialmente a mis padres:**

*A mi madre por todo su amor y por cuidar de mí y estar conmigo en los momentos felices y en los momentos de tristeza cuando más la necesité dándome su apoyo y las fuerzas para levantarme y seguir adelante.*

*A mi padre por la educación y por todas las cosas que me enseñó, por regañarme cuando fue necesario aun cuando sabía que lo hacía por mi bien, por ser el amigo y el compañero que siempre necesité.*

### **A mi hermana:**

*A mi adorada hermana por todo el cariño que siempre me ha demostrado y por ser tan especial y estar siempre para mí.*

### **A mis profesores:**

*A mis profesores, que me han enseñado todo lo que sé desde que comencé a dar mis primeros pasos: los de la primaria y la secundaria, mis profesores de la vocacional y de la UCI. A todos ellos gracias por formarme como profesional. A mi tutor Osvaldo, por guiarme siempre en el trabajo y por ser mi ejemplo a seguir. Él me enseñó que con esfuerzo y sacrificio es posible alcanzar las metas que nos tracemos. A mi cotutor Ernesto, porque siempre estuvo presente en la investigación y fue un guía muy importante para la conclusión satisfactoria del trabajo.*

### **A mis tías:**

*A mis tías Cira, Alba y Marucha porque han sido otras madres que me han dado todo el cariño y el apoyo que alguien puede desear y me han alentado siempre por el camino del estudio y la superación personal.*

### **A mis primos:**

*A mis primos Yonien, Eliecer y a Yanoska mi otra prima, porque han estado conmigo durante estos cinco años y siempre se han preocupado por mis resultados y han sido muy atentos conmigo.*

**A mis compañeros y amigos:**

*A mis compañeros de laboratorio, a mis amistades del grupo de primer año y a los que hoy terminan y se gradúan conmigo. A todos gracias porque de una forma u otra me hicieron mejor persona y de todos aprendí siempre algo importante. A mis amigos Leonel y Guille, porque sin su ayuda no hubiera sido posible terminar este trabajo. Al resto del equipo de trabajo, porque cada uno me aportó algo útil en la elaboración de la tesis.*

**A la UCI:**

*A esta hermosa Universidad que me ha formado como profesional y me compromete más con este valioso proyecto que supera sus resultados cada día.*

**A Fidel:**

*A nuestro máximo líder y conductor de la Revolución Cubana, el Comandante Fidel, el principal creador de esta universidad que me ha dado la oportunidad de convertirme en un profesional listo para servir a la Revolución.*

**A la brigada médica cubana:**

*A todo el personal de la brigada médica de Santa Ana, en el Táchira, Venezuela. A todos ustedes porque me hicieron más humano, revolucionario y comprometido con la Revolución y me enseñaron que nuestra ayuda solidaria seguirá llegando a cada rincón del planeta donde alguien la necesite.*

*Quiero agradecer a todas aquellas personas que aunque no menciono por cuestiones de espacio siempre las tendré en cuenta porque aportaron su granito de arena para que yo pudiera estar aquí hoy. A todos ustedes gracias.*

## Resumen

La iluminación de escenas tridimensionales es un tema muy importante en los gráficos por computadoras. Lograr una adecuada iluminación, permite visualizar las escenas con un alto grado de realismo y resaltar la tridimensionalidad de los objetos, la cual en ocasiones es decisiva para comprender los fenómenos que se desean representar.

Este trabajo propone un modelo de iluminación local de datos volumétricos basado en Phong y un modelo de iluminación global (*Raytracing*), el cual permite la generación de sombras y aumentar el nivel de detalle de las visualizaciones.

Ambos modelos fueron implementados y probados sobre varios volúmenes de datos que se utilizan en la Visualización Médica. Los resultados fueron obtenidos fundamentalmente en la visualización de estructuras anatómicas, aunque también se probaron sobre otros modelos que se usan con propósitos académicos. Incluir la iluminación en la visualización de volumen posibilita mejorar la calidad visual y el sentido de tridimensionalidad de los objetos que se desea representar.

**Palabras clave:** Iluminación global, Iluminación local, Luces, Modelos de iluminación, Mapas de sombras, Sombras, Trazado de rayos.

# Índice de Contenido

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPITULO 1. FUNDAMENTACIÓN TEÓRICA .....</b>	<b>8</b>
1.1. Propiedades de la luz .....	8
1.2. Dualidad de la luz .....	10
1.2.1. <i>La luz como partícula</i> .....	11
1.2.2. <i>La luz como onda</i> .....	13
1.3. Visualización de volumen.....	15
1.3.1. <i>Visualización directa de volumen</i> .....	15
1.3.2. <i>Modelos ópticos</i> .....	16
1.3.3. <i>RayCasting</i> .....	17
1.4. Modelos de iluminación.....	17
1.4.1. <i>Modelos de iluminación local</i> .....	18
1.4.1.1. Modelo de Phong .....	19
1.4.1.2. Mapas de reflexión.....	23
1.4.1.3. Modelo de reflexión de Oren–Nayar.....	24
1.4.1.4. Modelo de Ward .....	26
1.4.1.5. Modelo de reflexión de Cook–Torrance.....	28
1.4.2. <i>Modelos de iluminación global</i> .....	29
1.4.2.1. Raytracing.....	29
1.4.2.2. Formación de sombras .....	31
1.4.3. <i>Modelos de sombras</i> .....	32
1.4.3.1. Rayos de sombras.....	32
1.4.3.2. Mapas de sombras .....	34
1.4.3.3. Mapas de sombras profundas .....	35
<b>CAPITULO 2. CARACTERÍSTICAS DEL SISTEMA. ....</b>	<b>39</b>
2.1. Iluminación en la visualización de volumen.....	39
2.2. Modelo de Phong.....	40
2.3. Rayos de Sombras .....	41
2.4. Estimación de gradientes .....	42
2.4.1. <i>Diferencias intermedias</i> .....	42
2.4.2. <i>Operador de Sobel</i> .....	43
2.4.3. <i>Diferencias centrales</i> .....	44
2.5. Implementación sobre GPU .....	45
2.6. Herramientas .....	46
2.7. Lenguajes .....	47
2.8. Metodología de desarrollo de software .....	48
<b>CAPITULO 3. DESCRIPCIÓN DE LA SOLUCIÓN.....</b>	<b>49</b>
3.1. Reglas del Negocio .....	49



---

3.2. Modelo de Dominio .....	49
3.3. Captura de Requisitos .....	51
3.3.1. <i>Requisitos Funcionales</i> .....	51
3.3.2. <i>Requisitos No Funcionales</i> .....	51
3.4. Modelo de Casos de Uso del Sistema .....	52
3.4.1. <i>Actores del Sistema</i> .....	52
3.4.2. <i>Diagrama de Casos de Uso del Sistema</i> .....	53
3.4.3. <i>Descripción de Casos de Uso del Sistema</i> .....	53
3.5. Diagrama de Clases del Diseño del Paquete de Iluminación .....	55
3.5.1. <i>Diagrama de Clases del Paquete de Iluminación</i> .....	56
3.6. Diagrama de Secuencia del Diseño .....	56
3.6.1. <i>Diagrama de Secuencia del Caso de Uso Gestionar Luz</i> .....	57
<b>CAPITULO 4. IMPLEMENTACIÓN Y VALIDACIÓN DE LOS RESULTADOS.....</b>	<b>59</b>
4.1. Implementación .....	59
4.1.1. <i>Diagrama de componentes</i> .....	59
4.2. Validación de los Resultados.....	60
4.2.1. <i>Datos de pruebas</i> .....	60
4.2.1.1. Caso de Prueba # 1 .....	60
4.2.1.2. Caso de prueba # 2 .....	61
4.2.1.3. Caso de prueba # 3 .....	62
4.2.1.4. Caso de prueba # 4 .....	62
<b>CONCLUSIONES .....</b>	<b>63</b>
<b>RECOMENDACIONES.....</b>	<b>64</b>
<b>BIBLIOGRAFÍA .....</b>	<b>65</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>67</b>

# Índice de Figuras

FIG. 1 EL HOMBRE CREÓ LA CÁMARA OSCURA PARA REFLEJAR EL MUNDO A SU ALREDEDOR. ....	1
FIG. 2 IMAGEN GENERADA UTILIZANDO LOS GRÁFICOS POR COMPUTADORAS. ....	3
FIG. 3 REFLEXIÓN DE LA LUZ. ....	8
FIG. 4 REFRACCIÓN DE LA LUZ. ....	9
FIG. 5 TRASMISIÓN DE LA LUZ. ....	9
FIG. 6 DIFUSIÓN DE LA LUZ. ....	10
FIG. 7 ABSORCIÓN DE LA LUZ. ....	10
FIG. 8 REBOTES DE UNA BOLA DE BILLAR. ....	12
FIG. 9 REFLEXIÓN DE LA LUZ EN UNA SUPERFICIE. ....	13
FIG. 10 PROPIEDADES DE LAS ONDAS ELECTROMAGNÉTICAS. ....	13
FIG. 11 EL ESPECTRO VISIBLE. ....	14
FIG. 12 LOS VÓXELES CONSTITUYEN EL VOLUMEN DE DATOS DESPUÉS DE SER DISCRETIZADO. ....	15
FIG. 13 ETAPAS DE LA VISUALIZACIÓN DIRECTA DE VOLUMEN. ....	18
FIG. 14 ILUMINACIÓN AMBIENTAL DE UN MODELO ANATÓMICO. ....	19
FIG. 15 PARÁMETROS DE LA REFLEXIÓN DIFUSA. ....	20
FIG. 16 PARÁMETROS DE LA REFLEXIÓN ESPECULAR. ....	21
FIG. 17 EL MODELO DE PHONG ESTÁ COMPUESTO POR TRES COMPONENTES: LA AMBIENTE, LA DIFUSA Y LA ESPECULAR. ....	23
FIG. 18 EJEMPLO DE MAPA DE UN CUBO. IMAGEN TOMADA DEL ARTÍCULO “REAL-TIME VOLUME GRAPHICS”. ....	24
FIG. 19 BLOQUE DE UN MOTOR CON MAPAS DE REFLEXIÓN DIFUSA (IZQUIERDA) Y MAPA DE REFLEXIÓN ESPECULAR (DERECHA). IMAGEN TOMADA DEL ARTÍCULO “REAL-TIME VOLUME GRAPHICS”. ....	24
FIG. 20 OBJETOS DIFUSOS ILUMINADOS: (IZQUIERDA) MODELO LAMBERTIANO, (DERECHA) MODELO DE OREN–NAYAR. ....	25
FIG. 21 ILUMINACIÓN CON EL MODELO ISOTRÓPICO DE WARD. ....	27
FIG. 22 ILUMINACIÓN CON EL MODELO ANISOTRÓPICO DE WARD. ....	27
FIG. 23 TRAZADO DE RAYOS RECURSIVO. ....	30
FIG. 24 ALGORITMO DE RAYOS DE SOMBRA UTILIZANDO RAYTRACING. ....	33
FIG. 25 EL MAPA DE SOMBRAS UTILIZA UNA TEXTURA PARA ALMACENAR LA INFORMACIÓN DE VISIBILIDAD DESDE LA FUENTE DE LUZ. ....	34
FIG. 26 IZQUIERDA: PERCENTAGE CLOSER FILTERING. DERECHA: MAPAS DE SOMBRAS NORMAL. ....	35
FIG. 27 LOS MAPAS DE SOMBRAS UTILIZAN UNA PILA DE TEXTURAS QUE ALMACENA UNA APROXIMACIÓN DE LA FUNCIÓN DE SOMBRA. ....	36
FIG. 28 IZQUIERDA: PELO SIN SOMBRAS. DERECHA: PELO CON MAPAS DE SOMBRAS PROFUNDAS. ....	36
FIG. 29 MODELO DE ILUMINACIÓN LOCAL EN LA VISUALIZACIÓN DE VOLUMEN. ....	40
FIG. 30 MODELOS DE ILUMINACIÓN LOCAL Y GLOBAL EN LA VISUALIZACIÓN DE VOLUMEN. ....	40
FIG. 31 EL MÉTODO DE DIFERENCIAS INTERMEDIAS CONSIDERA EL VALOR DEL VÓXEL $V_{000}$ PARA EL VALOR FINAL. ....	43
FIG. 32 MÉTODO DE DIFERENCIAS CENTRALES. LOS COLORES AZUL, ROJO Y VERDE IDENTIFICAN A LOS VÓXELES VECINOS. ....	44
FIG. 33 MODELO DE DOMINIO. ....	50
FIG. 40 DIAGRAMA DE CASOS DE USO DEL SISTEMA. ....	53
FIG. 35 RELACIÓN ENTRE EL PAQUETE DE ILUMINACIÓN IMPLEMENTADO Y LOS DEMÁS MÓDULOS DEL SISTEMA VISMEDIC. ....	56
FIG. 36 RELACIÓN ENTRE LAS CLASES DEL PAQUETE DE VISUALIZACIÓN Y EL MÓDULO DE ILUMINACIÓN PROPUESTO. ....	56
FIG. 37 DIAGRAMA DE SECUENCIA DEL ESCENARIO ADICIONAR LUZ. ....	57
FIG. 38 DIAGRAMA DE SECUENCIA DEL ESCENARIO MODIFICAR PROPIEDADES DE LA LUZ. ....	57
FIG. 39 DIAGRAMA DE SECUENCIA DEL ESCENARIO ELIMINAR LUZ. ....	58
FIG. 40 DIAGRAMA DE COMPONENTES. ....	59

# Índice de Tablas

TABLA 1 ACTOR DEL SISTEMA. ....	52
TABLA 2 DESCRIPCIÓN DEL CU GESTIONAR LUZ.....	53
TABLA 3 DESCRIPCIÓN DE LA SECCIÓN ADICIONAR LUZ DEL CU GESTIONAR LUZ. ....	54
TABLA 4 DESCRIPCIÓN DE LA SECCIÓN MODIFICAR PROPIEDADES DE LA LUZ DEL CU GESTIONAR LUZ. ....	54
TABLA 5 DESCRIPCIÓN DE LA SECCIÓN ELIMINAR LUZ DEL CU GESTIONAR LUZ. ....	55

## Introducción

La luz es uno de los fenómenos más curiosos de la naturaleza y es probablemente la primera experiencia que tienen los seres humanos al nacer. Esta se estudia desde hace millones de años y muchas fueron las teorías que surgieron para explicar su origen. Desde fechas tan tempranas como 582 (A.C) Pitágoras de Samos pensaba en la luz como antenas o tentáculos que tenían las personas en los ojos y que le permitían a las personas ver los objetos. Esto dio inicio a la teoría visual del rayo como forma de explicación de la luz en la época. Otros hombres de ciencia como Leonardo Da Vinci (1452–1519) también dieron sus aportes a la explicación de la luz y plantearon teorías basadas en el funcionamiento del ojo humano como una cámara oscura y la proyección de la imagen al cerebro [1].

Con el paso de los siglos, el hombre trató de buscar otras formas más rápidas de capturar el mundo a su alrededor que los dibujos y las pinturas. Estos eran los resultados más importantes que se habían logrado con este propósito. Es por eso que se inventaron algunos instrumentos para lograr representar la realidad de una forma más fácil. Uno de los primeros instrumentos fue la *cámara oscura*. Esta herramienta consistía inicialmente de un cuarto oscuro con una pequeña abertura y se usaba para observar eclipses solares y ayudar a los artistas en la comprensión de la perspectiva [1]. En la Fig. 1 se puede ver una imagen de las primeras cámaras oscuras que inventó el hombre.

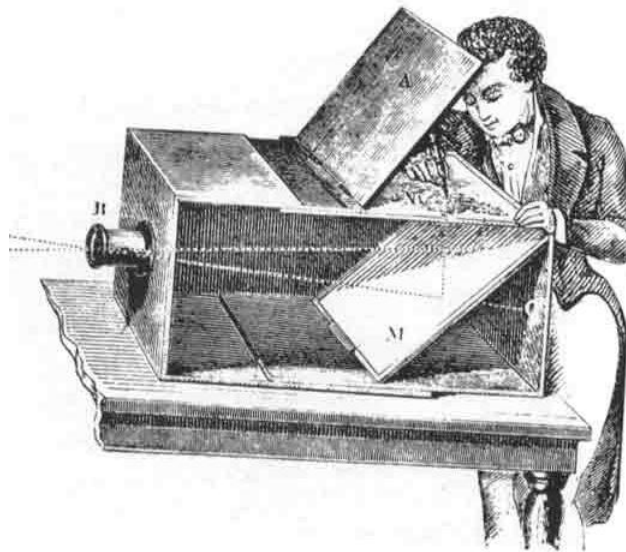


Fig. 1 El hombre creó la cámara oscura para reflejar el mundo a su alrededor.

La rápida evolución de la ciencia y la tecnología hizo posible el surgimiento de la fotografía, el cine, los ordenadores y con estos la aparición de los gráficos por computadoras y la realidad virtual. Cuando los gráficos por computadora se desarrollaron por primera vez, sus creadores trataban de imitar a los fotógrafos, convirtiéndose así la fotografía en la referencia primaria para las imágenes generadas por computadora [2], [3].

El objetivo principal de los programadores de gráficos por computadoras desde sus inicios ha sido simular o sintetizar en un monitor de computadora escenas naturales al igual que son capturadas por dispositivos fotográficos como las cámaras. Entre los primeros algoritmos diseñados para generar imágenes utilizando las computadoras están el *Raytracing* creado por Lee Westover y Turner Whitted y otros más complejos realizados por Eric Haines y Donald Greenberg que asombraron al mundo por la gran fidelidad fotográfica de las imágenes obtenidas con sus trabajos. Sin importar el algoritmo utilizado por estos y otros autores, su objetivo siempre fue que las imágenes generadas fueran lo más reales posibles [1].

Los gráficos generados por computadoras como disciplina comenzó con un interés de investigación académica y se desarrolló para aplicaciones militares antes de convertirse en un producto ampliamente usado con fines comerciales. Las primeras aplicaciones de gráficos por computadoras eran muy técnicas y se concentraban en la visualización de problemas matemáticos que no podían observarse de otra forma y en la creación de herramientas para contribuir a mejorar la eficiencia de las industrias. Años más tarde la gráfica computacional revolucionó como fenómeno y cambió para siempre la forma de visualizar la información en publicaciones, en el cine y en la Web. Los gráficos por computadoras cambiaron la forma en que el mundo percibía las computadoras, dejando de ser curiosos instrumentos de laboratorio para convertirse en herramientas vitales de comunicación. La Fig. 2 muestra una imagen generada utilizando los modernos ordenadores que ya se han convertido en una herramienta de trabajo muy importante para el hombre de hoy.

Los gráficos por computadoras en la actualidad se han extendido a casi todas las ciencias que estudia el hombre. Se utiliza en la elaboración de herramientas para la enseñanza, en el diagnóstico y entrenamiento de médicos a través del uso de simuladores, en operaciones asistidas por computadora, en la realidad aumentada y más recientemente en la Visualización Científica.



**Fig. 2 Imagen generada utilizando los gráficos por computadoras.**

La medicina es una de las ciencias donde mayores resultados se han obtenido, utilizando las aplicaciones gráficas en bien de la humanidad para asistir los diagnósticos realizados por los especialistas sobre las patologías de los pacientes. Para que esto sea posible, es necesario combinar la información obtenida en las pruebas y exámenes realizados a los pacientes con las potencialidades que brindan las nuevas tecnologías de la información. Hoy en día gracias a la existencia de disciplinas como la Visualización Científica, esto deja de ser una quimera para convertirse en una maravillosa realidad que permite salvar miles de vidas.

La Visualización Científica consiste en transformar los datos científicos obtenidos a partir de técnicas médicas como tomografías computarizadas, resonancias magnéticas o ultrasonidos en imágenes que pueden ser estudiadas y analizadas de una forma más profunda y detallada por los especialistas. Las herramientas informáticas y los métodos de análisis y visualización de imágenes médicas digitales han sido determinantes en el desarrollo de aplicaciones médicas con esta finalidad. De esta forma, se hace posible el proceso de análisis, reconstrucción 3D y visualización de estas imágenes, lo que contribuye a una mejor calidad de los diagnósticos de los especialistas. La visualización de estas imágenes necesita una gran calidad debido a que las hipótesis y los resultados emitidos por los especialistas deben ser lo suficientemente confiables para hacer un uso seguro de estas modernas herramientas que les facilitan en gran medida el trabajo a los médicos.

En el mercado actual existen aplicaciones de visualización de imágenes médicas: algunas resultan ineficientes y otras vienen incorporadas a los costosos equipos que actualmente solo pueden ser adquiridos por países del primer mundo como Estados Unidos y otros países de

Europa con un alto desarrollo tecnológico. Es por eso que Cuba, como país en vía de desarrollo se ve obligada a buscar alternativas y estrategias para seguir desarrollando su sistema de salud que ya se ubica entre los primeros a nivel internacional.

Un ejemplo de esta lucha incesante por crear aplicaciones de visualización médica que puedan utilizarse sin tener que pagar los altos costos que demanda la adquisición de tan modernas tecnologías es el proyecto de Visualización Médica (Vismedic) perteneciente a la facultad 5 de la Universidad de las Ciencias Informáticas. El proyecto Vismedic realiza la visualización de modelos tridimensionales con características anatómicas obtenidos a partir de imágenes DICOM, formato estándar para el trabajo con imágenes médicas digitales. El proyecto está utilizando dos tipos de visualización: la directa y la indirecta. En la visualización directa el volumen de datos se visualiza directamente sin necesidad de utilizar representaciones intermedias previas a la visualización. En la visualización indirecta ocurre lo contrario, previo al proceso de visualización se genera una representación geométrica del modelo que se desea visualizar y posteriormente se realiza el proceso de visualización.

Actualmente el proyecto carece de un módulo de iluminación que incorpore sensación de profundidad y realismo óptico a los modelos tridimensionales que se están visualizando. La ausencia de estos elementos provoca falta de realismo y calidad en la visualización de los modelos tridimensionales del proyecto, por ejemplo al visualizar estructuras anatómicas. Minimizar el margen de error en la visualización de los modelos reconstruidos a partir de los datos obtenidos de los dispositivos médicos es una tarea de primer orden. Estos errores pueden influir en la toma de decisiones de los especialistas, por ejemplo, al analizar un tumor en una región delicada como el cerebro y puede ser decisivo a la hora de salvar una vida o provocar la muerte de un paciente.

Ante la **situación problémica** planteada anteriormente, se plantea como **problema de investigación**: ¿Cómo aumentar el realismo y la calidad de la visualización directa de datos volumétricos en el proyecto Vismedic? Como resultado del problema de investigación, se define como **objeto de estudio** los modelos de iluminación para visualización de volumen. Como **objetivo general** del trabajo se propone elaborar un módulo de iluminación para aumentar el realismo y la calidad de la visualización directa de volumen del proyecto Vismedic.

A partir del **objeto de estudio** definido se propone como **campo de acción** los modelos de iluminación para visualización directa de volumen. El presente trabajo defiende la siguiente **idea**: la implementación de los modelos de iluminación propuestos permitirá mejorar la calidad

y el realismo de los modelos tridimensionales del proyecto Vismedic. Para darle cumplimiento a los objetivos planteados se enumeraron una serie de **tareas investigativas** que se muestran a continuación:

1. Elaboración del marco teórico a partir del estado del arte actual del tema a investigar.
2. Selección de los modelos ópticos para visualización directa de volumen.
3. Selección de los modelos de iluminación local y global que más se ajusten a las necesidades del proyecto.
4. Definición de la influencia del hardware en la implementación de modelos de iluminación.
5. Implementación de un prototipo funcional con los modelos de iluminación propuestos.
6. Integración del módulo elaborado a la aplicación principal del proyecto.

En el proceso de investigación y elaboración de este trabajo se tendrá en cuenta la utilización de varios **métodos científicos de investigación** como:

- ✓ **Histórico–Lógico:** Método teórico utilizado para analizar la evolución y las tendencias actuales del proceso de iluminación en la visualización de volumen.
- ✓ **Analítico–Sintético:** Método teórico utilizado para extraer y analizar la información sobre los modelos de iluminación para la visualización directa de volumen.
- ✓ **Consulta de fuentes de información:** Método empírico utilizado para la consulta de las fuentes bibliográficas durante la investigación.
- ✓ **Pruebas:** Método empírico para validar los resultados obtenidos con la solución propuesta a partir de la investigación realizada.
- ✓ **Observación:** Método empírico utilizado para apreciar la influencia de los modelos de iluminación propuestos en el rendimiento de la aplicación y en la calidad de la visualización.

La investigación introduce elementos novedosos, pues los modelos de iluminación seleccionados se aplicarán a la visualización directa de volumen. Este tipo de visualización se ha convertido en los últimos tiempos en una de las más utilizadas en la Visualización Científica.



Entre las razones para su elección se encuentran la rapidez e interactividad que le permite al usuario y la calidad de los resultados obtenidos, siendo estos elementos muy importantes para este tipo de aplicaciones.

Además, como parte de la investigación es necesario el estudio de la luz como fenómeno de la naturaleza, para explicar aspectos de la vida diaria tales como la interacción de la luz con los objetos que nos rodean, la variación del proceso de interacción de la luz con los materiales que componen a los objetos y otros aspectos que permitirán dar una mejor solución al problema planteado.

Una vez concluido este estudio, el proyecto de Visualización Médica constará con un módulo que permitirá incluir múltiples fuentes de luces a las escenas y visualizar modelos tridimensionales con un mayor nivel de detalle. El trabajo se ha estructurado para una mejor organización en cuatro capítulos. A continuación se muestra una breve síntesis de cada capítulo:

### **Capítulo 1: Fundamentación Teórica.**

En este capítulo se explicará el estado del arte de los principales modelos y algoritmos que se utilizan en la visualización directa para calcular la iluminación. Se explicarán algunos conceptos que pueden ser importantes para determinar los modelos que más se ajustan a las necesidades del proyecto.

### **Capítulo 2: Características del Sistema.**

En este capítulo se explicarán los modelos de iluminación seleccionados para implementar la solución propuesta y se explicará con mayor profundidad las ventajas de la solución que se propone. Se introducirá como elemento adicional el trabajo con la tarjeta gráfica mediante el uso de los shader. Se explicará brevemente las metodologías, herramientas y lenguajes seleccionados para desarrollar la solución.

### **Capítulo 3: Descripción de la Solución.**

En este capítulo se definirán las reglas del negocio así como el modelo de dominio. Se expondrá la captura de requisitos. Se elaborará el Modelo de Casos de Uso que estará formado por los actores, el Diagrama de Casos de Uso y la descripción textual de cada Caso de Uso. Finalmente se elaborarán los diagramas de clases del diseño y los diagramas de secuencia del diseño.

#### **Capítulo 4: Implementación y Validación de los Resultados.**

En este capítulo se definirán los detalles más específicos de la implementación y se harán las pruebas para validar los resultados de la investigación realizada. Se modelará el diagrama de componentes para la implementación y se mostrarán los resultados de las pruebas realizadas al módulo elaborado para medir el rendimiento y la calidad de la imagen, validando así los resultados obtenidos.

#### **Glosario de Términos.**

Se elaboró un Glosario de Términos con el objetivo de facilitar la comprensión del lenguaje utilizado en la investigación.

# CAPITULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se abordarán las principales técnicas y algoritmos que son utilizados a nivel mundial para la iluminación en la visualización de volumen. Este estudio constituye el estado del arte existente actualmente de los modelos de iluminación y sombreado para la visualización de volumen. Se tratarán además los modelos ópticos utilizados en la visualización de volumen por su estrecha relación con los algoritmos de iluminación.

## 1.1. Propiedades de la luz

La luz al interactuar con la materia es capaz de manifestar diversos comportamientos en dependencia del tipo de interacción. Estos comportamientos tan diversos tienen su explicación en las propiedades de la luz. Estas propiedades son muy importantes porque permiten explicar de una forma más científica muchos de los fenómenos de la naturaleza y comprender a un nivel más profundo el proceso de iluminación. A continuación se mostrarán algunas propiedades útiles para una correcta simulación de la luz en los entornos virtuales:

**Reflexión:** La reflexión es la propiedad de la luz que se produce por el lanzamiento o el rebote de la luz cuando impacta alguna superficie en el medio por donde viaja [1]. Ver Fig. 3.

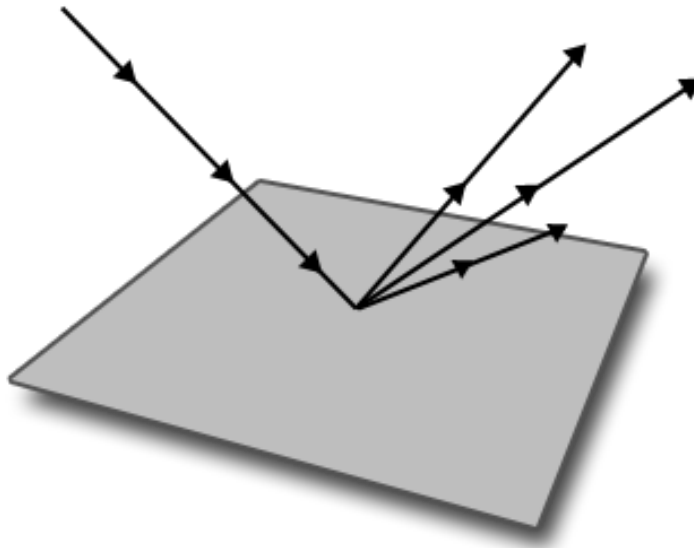


Fig. 3 Reflexión de la luz.

**Refracción:** La refracción es la inflexión o curva que experimenta la luz al cruzar de un medio a otro, por ejemplo al pasar de la atmósfera al agua. En este caso la luz se desvía de su trayectoria inicial y se aprecia una curva en la trayectoria dentro del nuevo medio [1]. Ver Fig. 4.

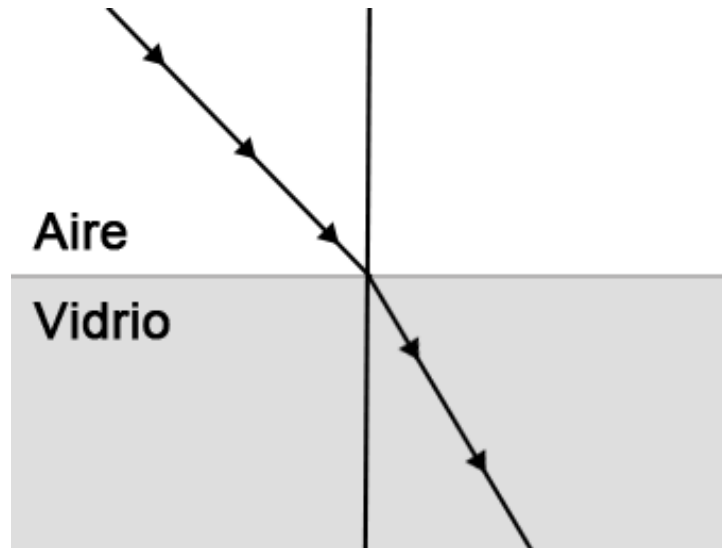


Fig. 4 Refracción de la luz.

**Trasmisión:** La trasmisión de la luz se produce como resultado del transporte o la conducción de la luz a través de un medio [1]. Ver Fig. 5.

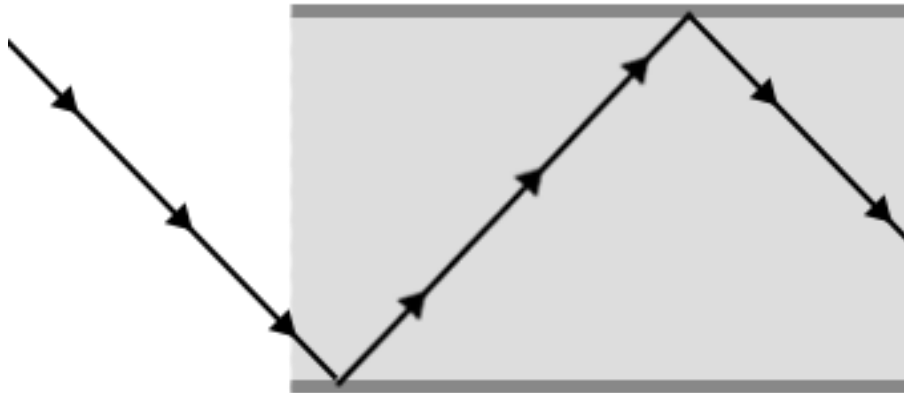
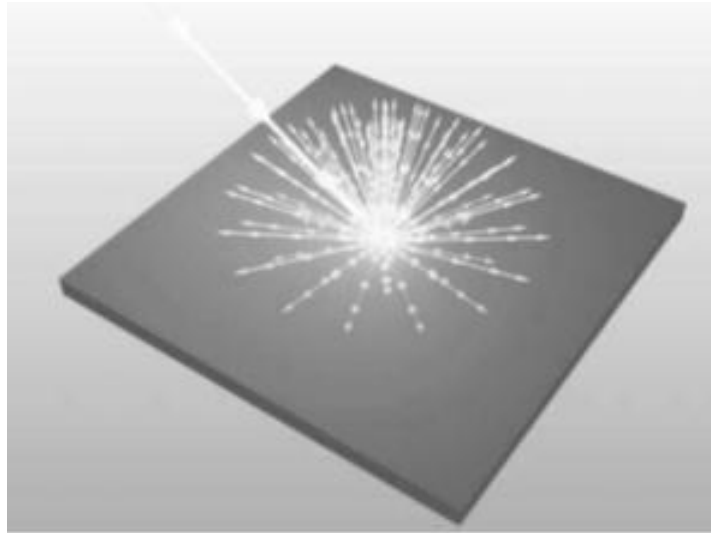


Fig. 5 Trasmisión de la luz.

**Esparcimiento:** El esparcimiento es la diseminación de la luz al interactuar con la materia o el medio. Es la reflexión múltiple de la luz en diferentes direcciones [1].

**Difusión:** La difusión es la dispersión uniforme de la luz por reflexión de una superficie. La difusión también se refiere a la trasmisión de la luz a través de un material translúcido [1]. Ver Fig. 6.



**Fig. 6 Difusión de la luz.**

**Absorción:** La absorción es la retención de la luz ya sea por la materia y algún medio, que impide la reflexión o la transmisión de la luz [1]. Ver Fig. 7.



**Fig. 7 Absorción de la luz.**

La interacción de la luz puede mostrar una o más propiedades simultáneamente. Por ejemplo, la luz puede reflejarse a la vez que se dispersa y es absorbida al interactuar con el medio. La interacción de la luz es un fenómeno complejo, pero puede calcularse debido al patrón que describe su comportamiento.

## 1.2. Dualidad de la luz

Los físicos de 1920 se enfrentaban con un problema interesante que todavía en la época los científicos no habían podido resolver. Newton planteaba su teoría de que la luz se componía de

partículas. Un poco más tarde Thomas Young y otros científicos demostraron que la luz se comportaba como una onda electromagnética [4].

Un par de años más tarde Albert Einstein demostró con diferentes experimentos que la luz se comportaba como partículas, demostrando así la existencia de los *fotones* [4]. Con este descubrimiento finalmente ganó el Premio Nobel de Física en 1921. Dos experimentos probados arrojaban dos resultados diferentes, por lo que los físicos se encontraron ante un dilema: la dualidad onda – partícula de la luz. Todo esto pudo ser muy interesante para los físicos, ¿pero qué significa para los programadores de gráficos? Este tema es muy importante porque los programadores diariamente utilizan ambas formas en su trabajo. Los efectos de reflexión y refracción se describen con mayor facilidad utilizando la teoría corpuscular de la luz, mientras que el color se describe fundamentalmente en términos de ondas. Para poder comprender de una forma más profunda los efectos de la iluminación se necesita de ambas, de ahí su importancia para el estudio de la luz.

### **1.2.1. La luz como partícula**

La luz está compuesta de pequeñas partículas denominadas fotones. Estas partículas no son partículas físicas como en el caso del polvo, los átomos o los electrones. Se puede pensar en ellas como pequeños paquetes de energía que viajan a velocidad constante y no tienen una masa real [4]. Con estas consideraciones, se puede establecer una comparación con un ejemplo de la vida real que muestra perfectamente la naturaleza corpuscular de la luz.

Los fotones se comportan muy similares a las bolas del conocido juego de billar que disfrutan muchas personas. La luz se refleja en las superficies al igual que una bola de billar rebota sobre los lados de la mesa.

La bola de billar rebota sobre la superficie con un ángulo igual pero de dirección opuesta a la dirección del ángulo de entrada. Cada vez que la bola golpea un lado de la mesa, pierde una pequeña cantidad de energía en la colisión. Una pequeña cantidad de energía se pierde también a través de la fricción entre la bola y la superficie, transformándose esta energía en calor y sonido. La bola continuará moviéndose y golpeando las superficies de la mesa hasta que se agote toda la energía cinética, llegando finalmente al reposo. En la Fig. 8 se observa el proceso explicado para la bola de billar.

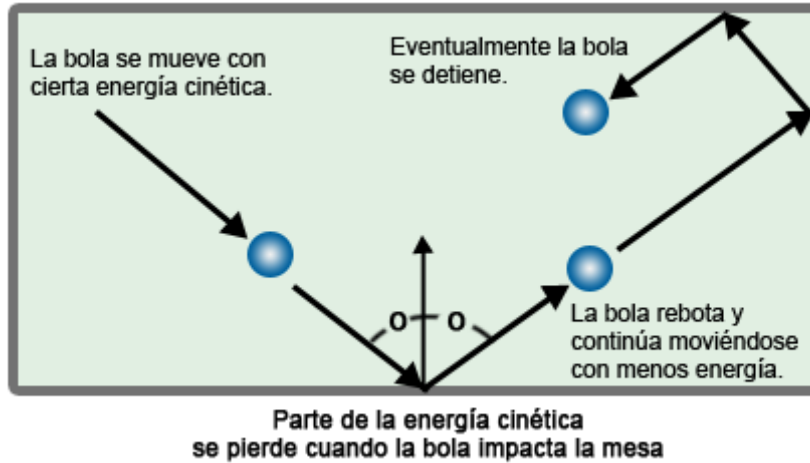


Fig. 8 Rebotes de una bola de billar.

La reflexión de la luz es bastante similar, aunque la transferencia de energía varía un poco. Cada rayo de luz puede verse como un flujo compuesto por muchos fotones [4]. Cada vez que este flujo golpea un objeto, un porcentaje de estos fotones se absorbe y se transforma en calor. La cantidad de fotones absorbidos depende de las propiedades del material que compone el objeto [4]. Los fotones restantes son reflejados sin disminuir su velocidad, pero ahora en menor cantidad y con menor energía. El flujo de fotones continuará su movimiento por el espacio hasta que toda la energía haya sido absorbida por los objetos del entorno. La Fig. 9 muestra el mecanismo de transferencia de la luz en el espacio y a diferencia de la bola de billar, no llega a un estado de reposo, simplemente llega a un estado en el que todos los fotones han sido absorbidos.

Estas situaciones aunque pudieran parecer un poco diferentes, si son analizadas desde un nivel más general, ambas situaciones coinciden. Los dos ejemplos son sistemas que van transfiriendo energía al medio circundante hasta que esta energía se agota. En el caso de la bola de billar, mientras más suave sea la superficie de la mesa, menos energía se perderá en cada impacto y la bola viajará por más tiempo. En el caso de la luz, mientras más reflectante sea la superficie, menos serán los fotones absorbidos en cada reflexión y la luz viajará más lejos.

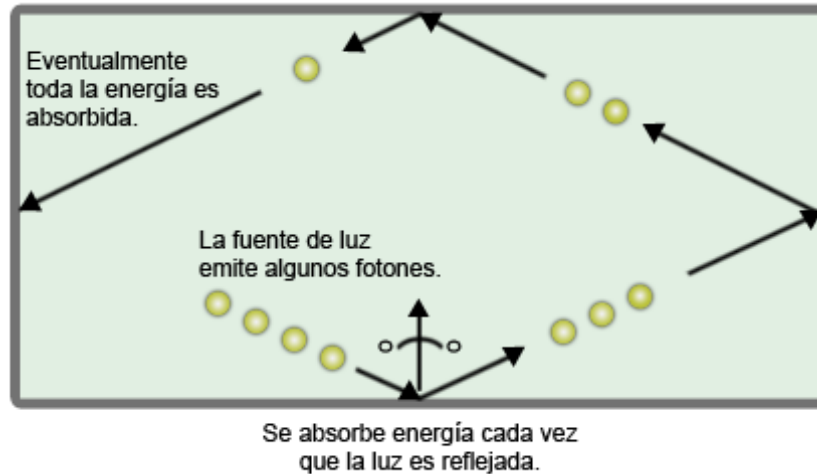


Fig. 9 Reflexión de la luz en una superficie.

### 1.2.2. La luz como onda

La luz también tiene todas las propiedades de una onda electromagnética [4]. Por esta razón se comporta igual que algunas formas electromagnéticas como las ondas de radio y los rayos x. Las propiedades básicas se muestran en la Fig. 10.

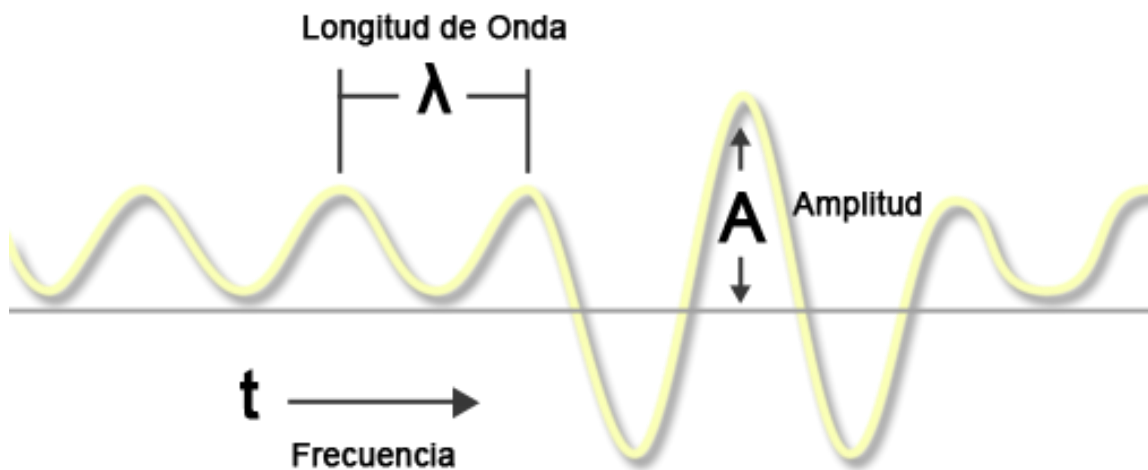


Fig. 10 Propiedades de las ondas electromagnéticas.

La *amplitud* de la onda se corresponde con la intensidad de la luz [4]. La *longitud de onda* es la distancia que cubre una sola onda. Esta distancia generalmente es muy corta para las ondas de luz y se mide en nanómetros ( $1 \text{ nm} = 10^{-9} \text{ m}$ ). Para representarla se utiliza la letra griega  $\lambda$ . La *frecuencia* es el número de ondas que pasan un punto dado en el espacio en un segundo y se mide en Hertz o ciclos por segundo. Esta se denota por la letra griega  $\gamma$ . Entre la longitud de la onda y la frecuencia existe la siguiente relación:



$$f = \frac{c}{\lambda} \quad (1)$$

donde:

$f$ : Frecuencia.

$c$ : Velocidad de la luz = 100 000 km/s.

$\lambda$ : Representa la longitud de onda.

El color de la luz visible depende de la longitud de onda y una fuente de luz puede emitir ondas de diferentes longitudes de ondas. Las ondas electromagnéticas se clasifican de acuerdo a sus rangos de longitud de onda [4]. Existen varias clases de ondas electromagnéticas por encima y por debajo del rango visible. La luz visible está entre los 400 nm y los 700 nm y se muestra en la Fig. 11. Los extremos del espectro visible son la radiación infrarroja y la radiación ultravioleta. La radiación infrarroja se caracteriza por el calor y está ubicada en el extremo rojo del espectro visible. La radiación ultravioleta se ubica en el otro extremo: el extremo violeta del espectro visible. Esta radiación es considerada peligrosa para la piel y los ojos de los seres humanos.

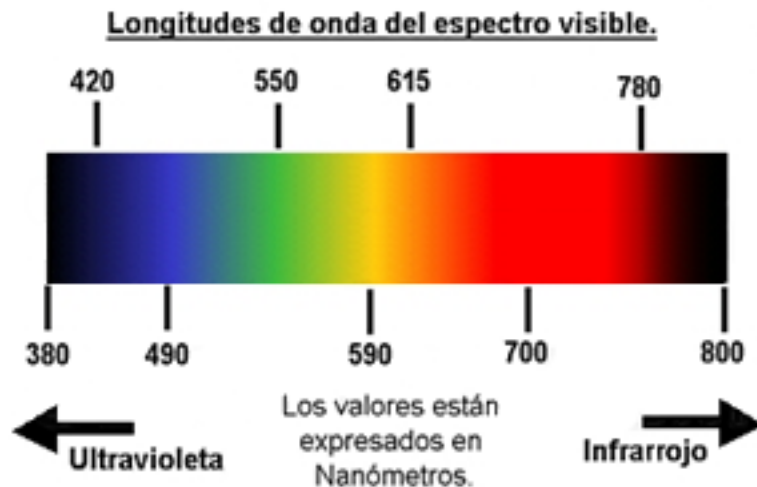


Fig. 11 El espectro visible.

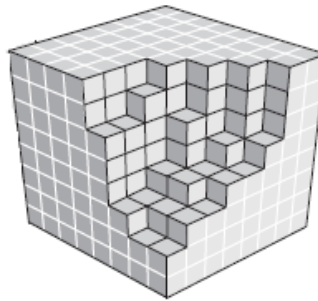
Algunos materiales absorben solo la luz de ciertas longitudes y no otras [4]. Por ejemplo, un objeto rojo al ser iluminado por una fuente de luz blanca, refleja solo la porción de luz roja y las otras longitudes de onda son absorbidas. La mayoría de las ecuaciones en los gráficos por computadoras consideran el color como una simple propiedad y deciden ignorar su relación con la longitud de onda. Sin embargo, para algunos modelos de iluminación la longitud de onda juega un papel clave para crear efectos de interferencia entre ondas de una longitud de onda dada.

### 1.3. Visualización de volumen

En la mayoría de las aplicaciones actuales de modelación, los objetos 3D son creados usando representaciones de superficies como mallas poligonales. Las propiedades visibles de las superficies como la rugosidad y el color son modeladas mediante algoritmos de iluminación. Esta técnica se evalúa solo en los puntos de la superficie por lo que no tiene en cuenta el comportamiento de la luz en el interior de los objetos.

A diferencia de las representaciones de superficies, la representación de volumen [5] incluye técnicas para generar imágenes a partir de datos escalares tridimensionales como los que se obtienen mediante tomografías computarizadas (CT) o resonancias magnéticas (MRI).

Los datos del volumen pueden ser vistos como un arreglo tridimensional de elementos cúbicos denominados *vóxeles* [6] (Ver Fig. 12) y que representan una unidad del espacio. Estos datos son ideales para describir objetos difusos como fluidos, gases y otros fenómenos como niebla y fuego.



**Fig. 12** Los vóxeles constituyen el volumen de datos después de ser discretizado.

Existen dos técnicas fundamentales en la visualización de volumen: la directa y la indirecta. La visualización indirecta genera una representación intermedia del volumen de datos que posteriormente es visualizada. Uno de los algoritmos utilizados en este tipo de visualización es el *Marching Cubes* [8]. Como la representación que se visualiza no es el volumen de datos original, recibe el nombre de indirecta. La visualización directa por otro lado, visualiza directamente el volumen de datos sin la necesidad de generar una representación intermedia de los datos.

#### 1.3.1. Visualización directa de volumen

La visualización directa de volumen o abreviadamente visualización de volumen, como su nombre lo indica, representa los datos volumétricos directamente sin generar una representación intermedia. La complejidad de este tipo de algoritmos está determinada por el número de vóxeles del volumen y por el número de píxeles del *viewport*.

Los métodos de visualización directa representan los datos de los vóxeles al evaluar un modelo óptico que describe como el volumen emite, refleja, dispersa, absorbe y ocluye la luz [7]. El valor escalar se mapea a cantidades físicas que describen la interacción de la luz en el punto correspondiente al espacio 3D. Este mapeo recibe el nombre de clasificación y generalmente se hace utilizando las funciones de transferencias [8]. Estas cantidades físicas son utilizadas en la síntesis de imágenes.

Durante la síntesis de la imagen, la propagación de la luz se calcula integrando los efectos de interacción de la luz a lo largo del rayo de visualización, basado en el modelo óptico. Esta integral se conoce como *integral de visualización de volumen* [8].

### **1.3.2. Modelos ópticos**

La mayoría de los algoritmos de visualización de volumen consideran el volumen como una distribución de luz que emite partículas de una cierta densidad. Estas densidades permiten la composición de la imagen a lo largo del rayo que atraviesa el volumen. Este comportamiento está determinado por un modelo óptico basado en física. Los modelos ópticos más importantes para visualización de volumen los describe Nelson Max en uno de sus trabajos [7], por lo que solo se mencionarán brevemente algunos modelos:

- ✓ **Solo absorción:** El volumen consiste de partículas frías y perfectamente negras que absorben toda la luz que incide sobre ellas. No emite ni dispersa la luz.
- ✓ **Solo emisión:** El volumen consiste de partículas que solo emiten luz, pero no absorben la luz porque el grado de absorción es insignificante.
- ✓ **Absorción más emisión:** Este modelo es el más común en la visualización de volumen. Las partículas emiten luz y ocluyen, es decir, absorben la luz incidente, pero no hay dispersión ni iluminación indirecta.
- ✓ **Dispersión y sombreado:** Este modelo incluye la dispersión de la iluminación en el exterior de los vóxeles.
- ✓ **Múltiple dispersión:** Este es uno de los modelos más sofisticados y soporta luz incidente que ya ha sido dispersada por múltiples partículas.

El modelo que se utilizará en este trabajo es el que considera que las partículas simultáneamente emiten y absorben la luz.

### 1.3.3. RayCasting

*RayCasting* [5] es un algoritmo de visualización de volumen que utiliza directamente la evaluación de la integral para visualización de volumen. Para cada píxel de la imagen se lanza un rayo dentro de la escena. A lo largo de cada rayo se toman muestras del volumen en diferentes intervalos de igual tamaño y generalmente se usa interpolación como filtro de reconstrucción. Es decir, para cada ubicación de la muestra, se les da valores a los ocho vóxeles vecinos de acuerdo a la distancia al vóxel actual para el cual el valor es necesario. Después de obtener el valor para la muestra, este valor es mapeado a propiedades ópticas a través de una tabla de búsqueda. Esta obtiene como resultado una cuádrupla RGBA que incorpora los coeficientes correspondientes de emisión y absorción [7] para esta ubicación.

### 1.4. Modelos de iluminación

La iluminación dentro del área de los gráficos por computadoras consiste en el proceso que permite calcular la intensidad de la luz en cada uno de los puntos de una superficie o escena [9], basados en la posición, orientación, características de la superficie y fuentes de luz incidentes. Para obtener esta intensidad luminosa se utilizan los denominados modelos de iluminación.

La iluminación es uno de los pasos fundamentales en la búsqueda de realismo en una escena. Esta se puede conseguir ubicando varias fuentes de luz que interactúan sobre los objetos presentes en la escena, teniendo en cuenta las propiedades de los materiales de que estén constituidos. La luz es importante en la visualización científica, la forma en que las estructuras reflejan la luz da una idea más cercana a la realidad y permite comprender de una forma más acertada el fenómeno que se está estudiando.

El problema de iluminar una escena básicamente consiste en:

1. Modelar las fuentes de luz en una escena: ubicación, orientación, coeficientes de atenuación.
2. Construir un modelo de reflexión que trate con las interacciones entre los materiales de los objetos y las fuentes de luz.

La iluminación en la visualización directa de volumen se calcula en la cuarta etapa del pipeline [10]. Ver Fig. 13. La forma en que las superficies de los objetos son sombreadas, la forma que ellos representan y la forma en que interactúan con la luz se debe a las propiedades de los materiales de los objetos. Es por eso que para reflejar la realidad, estos deben interactuar con la luz. Esta interacción luz-material requiere el desarrollo de los modelos de iluminación. Estos

modelos son reglas simplificadas de la interacción luz-objeto y simulan el comportamiento de los materiales iluminados al ser observados en el mundo real [1].

Un modelo de iluminación es una forma de hacer que los objetos dibujados en una escena parezcan reales sin necesidad de utilizar cálculos complejos. Los modelos de iluminación generalizan e idealizan el comportamiento de la interacción entre la luz y los objetos.

Algunos modelos de iluminación simulan la interacción entre la luz y los objetos, pero solo tienen en cuenta la luz que incide directamente en estos. Otros modelos de iluminación simulan la transferencia de luz indirecta en una escena y tienen en cuenta algunas propiedades más complejas de la luz como son: la reflexión, la refracción y las sombras. Estos conceptos se denominan iluminación local e iluminación global respectivamente.

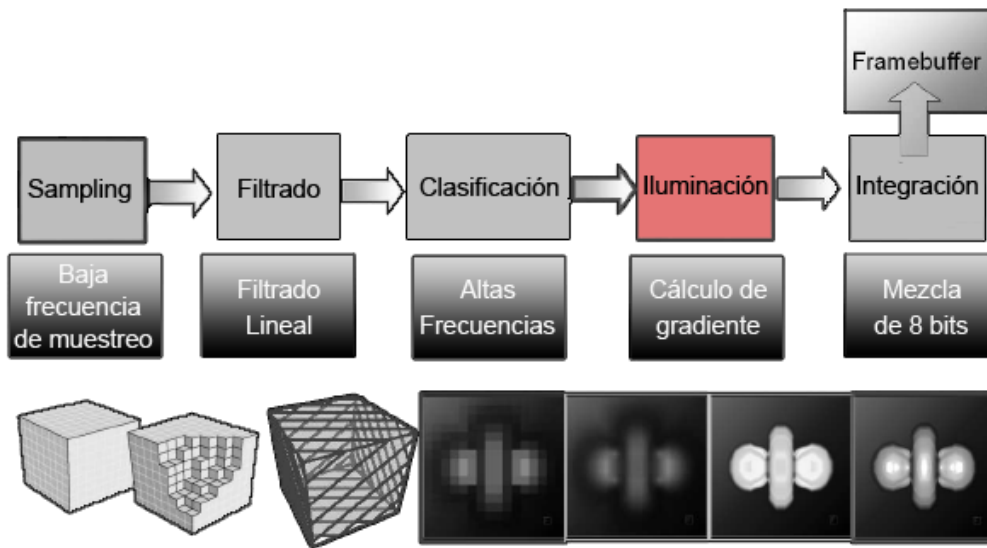


Fig. 13 Etapas de la visualización directa de volumen.

#### 1.4.1. Modelos de iluminación local

Los modelos de iluminación local permiten la aproximación de la intensidad de la luz reflejada desde un punto de la superficie de un objeto. Esta intensidad es evaluada como una función de la orientación local de la superficie respecto a la posición de la fuente de luz y algunas propiedades de los materiales. En comparación con los modelos de iluminación global, no tienen en cuenta los efectos de la luz indirecta o de las sombras. Los modelos de iluminación local son simples, fáciles de evaluar y no requieren los altos costos computacionales de la iluminación global.

### 1.4.1.1. Modelo de Phong

El modelo de iluminación de Phong [1975] combina tres componentes en su fórmula: luz ambiente, reflexión difusa y reflexión especular [11], [12], [13], [14]. Ver ecuación 2.

$$I_{total} = I_{amb} + c_{att} \sum_{i=0}^{n-1} (I_{diff,i} + I_{spec,i}) \quad (2)$$

#### Componente ambiental

Los cuerpos en la naturaleza aunque no estén expuestos de manera directa a una fuente de luz son visibles, aún si los objetos circundantes no están iluminados. Esto se debe a un nivel general de brillantez para toda la escena que se conoce como iluminación ambiental. La componente ambiental describe la luminosidad del entorno dentro de la escena. De este modo, todos los objetos de una escena recibirán la misma cantidad de luz ambiental y reflejarán tanta luz como el material del que estén constituidos permita.

La luz ambiente no presenta características de espacio ni de dirección. La cantidad de luz incidente para cada objeto es constante para todas las superficies y en todas las direcciones. Este comportamiento es modelado mediante una constante de reflexión ambiental  $K_a$ , que da una medida de la reflectividad de la superficie y que está en el rango entre cero y uno [0,1]. La expresión de la intensidad de la luz para esta componente es la siguiente:

$$I = I_a K_a \quad (3)$$

donde:

**I<sub>a</sub>**: Intensidad de la luz ambiente y es constante para todos los objetos de la escena.

**K<sub>a</sub>**: Coeficiente de reflexión ambiental. Propiedad material y no física.

El principal problema de esta componente es que ilumina todo el objeto uniformemente. Este resultado se puede observar en la Fig. 14.



Fig. 14 Iluminación ambiental de un modelo anatómico.

### Componente difusa

La reflexión difusa se refiere a la luz que es reflejada con igual intensidad en todas las direcciones y que se conoce también como *Reflexión Lambertiana* [8]. El brillo de la superficie es independiente de la dirección del observador y depende solo del ángulo de incidencia  $\theta$ , entre el vector  $L$  de la fuente de luz y el vector  $N$  que es el vector normal de la superficie. Ver Fig. 15.

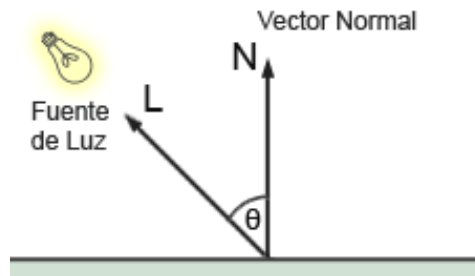


Fig. 15 Parámetros de la reflexión difusa.

La cantidad fraccional de luz incidente que se refleja de manera difusa se puede establecer para cada superficie con el parámetro  $K_d$ . Este es el coeficiente de reflexión difuso y se le asigna un valor constante entre cero y uno  $[0,1]$  de acuerdo a las propiedades de reflexión que se desea dar a la superficie. Un valor próximo a 1 indicará una superficie muy reflectante, mientras que para simular una superficie que absorba la mayor parte de la luz incidente se establecerá un valor cercano a 0.

Un reflector difuso perfecto esparce la luz que refleja de igual forma en todas las direcciones. Es por eso que todos los observadores sin importar la dirección donde estén ubicados verán el mismo resultado. Sin embargo, la cantidad de luz reflejada depende del material, dado que parte de la luz es absorbida y de la posición de la fuente de luz relativa al objeto que se desea iluminar. Al adicionar el término ambiente y el término difuso, el modelo que se obtiene es el siguiente:

$$I = I_a K_a + I_l K_d \cos\theta \quad (4)$$

donde:

$I_l$ : Intensidad de la fuente de luz

$K_d$ : Coeficiente de reflexión difuso

$\theta$ : Ángulo entre la normal y el vector de la luz

$$0 \leq \theta \leq \frac{\pi}{2}, \quad K_d + K_a < 1$$

Como  $N$  y  $L$  son vectores unitarios, la expresión  $\cos\theta$  se puede sustituir por el producto vectorial entre  $N$  y  $L$  ( $N \cdot L$ ) obteniendo una nueva expresión:

$$I = I_a K_a + I_l K_d (N \cdot L) \quad (5)$$

### Componente especular

Cuando se observa una superficie brillante iluminada, como el metal pulido, se puede apreciar un punto de luz o una mancha brillante cuando se observa desde algunas direcciones del observador. Este fenómeno se conoce como reflexión especular y ocurre sobre un estrecho rango de ángulos. En estos casos, el color de la luz que se refleja coincide con el color de la luz y no con el de la superficie. En la Fig. 16 se puede observar un esquema que representa el fenómeno.

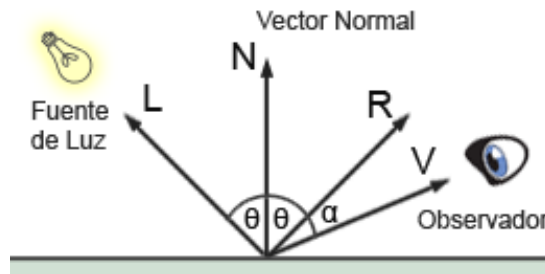


Fig. 16 Parámetros de la reflexión especular.

En la figura se tiene una superficie y su vector normal  $N$  en el punto a iluminar;  $L$  es el vector dirigido hacia la fuente de luz;  $R$  se obtiene mediante la reflexión del vector  $L$  sobre el vector  $N$  y  $V$  es el vector dirigido hacia la posición del observador. El ángulo  $\alpha$  es el ángulo que se forma entre los vectores  $V$  y  $R$ .

Para un reflector ideal –un espejo perfecto, la luz incidente se refleja solo en la dirección de reflexión especular  $R$ . En este caso solo se observa la luz reflejada cuando coinciden los vectores  $V$  y  $R$  ( $\alpha = 0$ ).

Debido a las complejas características físicas de la luz especular, se utiliza un modelo empírico que toma la función  $f(\alpha) = \cos^n(\alpha)$ , donde  $n$  depende de las propiedades del objeto. En general, se usan valores grandes de  $n$  para metales y superficies brillantes y valores pequeños de  $n$  para superficies no metálicas como el papel. Para esta componente al igual que en las anteriores, existe un coeficiente de reflexión especular  $K_s$  que regula la cantidad de luz que se refleja especularmente. El modelo de la luz especular quedaría de esta forma:



$$I = I_l K_s \cos^n(\alpha) \quad (6)$$

donde:

$I_l$ : Intensidad de la fuente de luz

$K_s$ : Coeficiente de reflexión especular

$\alpha$ : Ángulo entre el vector de reflexión y el vector del observador.

Como R y V son vectores unitarios, la expresión  $\cos(\alpha)$  puede ser sustituido por el producto vectorial de R y V ( $R \cdot V$ ) y al integrarlo al modelo anterior, se obtiene finalmente el modelo de Phong como la suma lineal de todas las componentes. Este modelo es correcto debido a la propiedad aditiva de la energía de la luz:

$$I = I_a K_a + I_l K_d (N \cdot L) + I_l K_s (R \cdot V)^n \quad (7)$$

Sin embargo, dos objetos a diferentes distancias pero con la misma orientación hacia la fuente de luz, muestran la misma intensidad. La intensidad de la luz decrece inversamente con el cuadrado de la distancia desde el punto iluminado hasta la fuente de luz. Es por eso que para corregir estas variaciones de intensidad, se introduce un factor de atenuación luminosa  $f_{att}$ , que es igual a la inversa del cuadrado de la distancia desde la fuente de luz hasta el objeto que se desea iluminar. Para ello se utiliza un polinomio de segundo grado cuyos coeficientes se ajustan empíricamente:

$$f_{att} = \min \left[ \left( \frac{1}{c_1 + c_2 \cdot d_L + c_3 \cdot d_L^2} \right), 1 \right] \quad (8)$$

donde:

$c_1, c_2, c_3$ : Coeficientes de atenuación luminosa

$d_L$ : Distancia hacia la fuente puntual

$f_{att}$ : Función de atenuación.

Al introducir el factor de atenuación para las componentes difusa y especular y extendiendo el modelo para múltiples luces en la escena, el modelo de Phong final es:

$$I = I \cdot K_a + \frac{I_l}{f_{att}} \sum_{i=0}^{n-1} (K_d(N \cdot L), i + K_s(R \cdot V^n), i) \quad (9)$$

En la Fig. 17 se observa el resultado final de aplicar el modelo de Phong sobre un modelo anatómico con cada una de sus componentes individuales y como contribuye cada componente al modelo final.



Fig. 17 El modelo de Phong está compuesto por tres componentes: la ambiente, la difusa y la especular.

#### 1.4.1.2. Mapas de reflexión

Muchas veces el cálculo de la iluminación es muy costoso para realizarlo en tiempo real. En estos casos existen técnicas de iluminación alternativas tales como los mapas de reflexión, que permiten obtener resultados aceptables en estos casos.

Esta idea tiene sus orígenes en los juegos de computadoras y es un método que precalcula la iluminación en escenarios complejos [10]. La mayor utilidad de esta técnica proviene de la capacidad de aplicar iluminación local con un número arbitrario de fuentes de luces y diferentes parámetros de iluminación a un bajo costo computacional. Un mapa de reflexión almacena la iluminación incidente desde todas las direcciones en un punto en el espacio.

Al igual que los términos especular y difuso del modelo de Phong, los mapas de reflexión pueden ser ejecutados con mapas difusos o mapas de entornos reflectivos [10]. Los índices en un mapa de reflexión difuso son calculados directamente a partir del vector normal y del vector del observador del entorno. En estos casos se asume que el objeto iluminado es pequeño respecto al entorno que lo contiene.

Una parametrización especial de la dirección normal se utiliza para construir un mapa de un cubo, como se muestra en la Fig. 18. En este caso, la escena se proyecta sobre las seis caras de un cubo.



**Fig. 18 Ejemplo de mapa de un cubo. Imagen tomada del artículo “Real-Time Volume Graphics”.**

Los mapas cúbicos son populares porque se pueden construir fácilmente a partir de sistemas de render y de técnicas como la fotografía. Los mapas de reflexión son generados en coordenadas del espacio mundo, es por eso que para una correcta aplicación del mapa, este requiere tener en cuenta la transformación local representada por la matriz de modelado actual. Para los mapas reflectivos la dirección del observador debe ser analizada a la hora de renderizar imágenes. Esto se puede observar en la Fig. 19.



**Fig. 19 Bloque de un motor con mapas de reflexión difusa (izquierda) y mapa de reflexión especular (derecha). Imagen tomada del artículo “Real-Time Volume Graphics”.**

#### 1.4.1.3. Modelo de reflexión de Oren–Nayar

Todos los materiales difusos no se comportan igual que los materiales descritos por los modelos Lambertianos. Estos materiales no reflejan la luz con igual intensidad en todas las direcciones. Es por eso que para poder obtener resultados correctos para estos materiales se utilizan otros modelos de iluminación. Un ejemplo es el modelo de reflexión de Oren–Nayar [15].

Algunas superficies rugosas como la arcilla y algunos tipos de ropa, tienen un alto grado de retrorreflexión (reflejan la luz en la dirección de la fuente de luz). Este efecto provoca una apariencia más plana que los modelos Lambertianos y se puede apreciar fundamentalmente cuando la dirección del observador coincide con la posición del vector de la luz. La Fig. 20 muestra las diferencias entre el modelo Lambertiano y el modelo de Oren–Nayar al iluminar un objeto.



**Fig. 20** Objetos difusos iluminados: (Izquierda) Modelo Lambertiano, (Derecha) Modelo de Oren–Nayar.

Como se puede observar en la figura, el modelo de la derecha está iluminado de una manera más uniforme con el modelo de Oren–Nayar que el modelo de la izquierda con el modelo Lambertiano. Esto se corresponde con la naturaleza de los materiales como la arcilla, elemento muy utilizado en la alfarería. El modelo de Oren y Nayar considera que las superficies rugosas están compuestas por facetas más pequeñas que el área de un píxel de la escena. Es decir, cada píxel contiene un número grande de facetas orientadas en direcciones diferentes.

Mientras que una faceta puede comportarse como Lambertiana, el conjunto de una cantidad mayor no se comporta de igual manera. Para este tipo de materiales, las facetas tienden a reflejar una mayor cantidad de luz en la dirección de la fuente de luz, determinando este comportamiento la rugosidad del material.

La técnica utilizada por los autores para modelar la rugosidad de los materiales es describir las direcciones de las facetas como una distribución aleatoria con un valor medio de 0. Luego, la distribución estándar de las direcciones de las facetas se puede utilizar como una medida de la rugosidad. Mientras mayor sea la desviación estándar, mayor será la variación de las direcciones de las facetas y más rugosa será la superficie. Por esta razón, la rugosidad se representa por la letra griega para desviación estándar, sigma ( $\sigma$ ). A continuación se resumen las ecuaciones necesarias para el modelo de Oren–Nayar:

$$I_0 = D * (N \cdot L) * (A + B * \sin(\alpha) * \tan(\beta) * \text{MAX}(0, \cos(C))) * I_1 \quad (10)$$

$$A = 1 - 0.5 \frac{\sigma^2}{\sigma^2 + 0.33}$$

$$B = 0.45 \frac{\sigma^2}{\sigma^2 + 0.09}$$

$$\alpha = \text{MAX}(\text{acos}(N \cdot L), \text{acos}(N \cdot V))$$

$$\beta = \text{MIN}(\text{acos}(N \cdot L), \text{acos}(N \cdot V))$$

$$C = \text{ángulo entre el vector de la luz} \\ \text{y el vector del observador}$$

Cada uno de los términos es relativamente fácil de calcular. El más complejo pudiera ser C, pues es necesario utilizar los vectores de la luz y del observador y el cálculo pudiera no ser tan trivial como calcular el producto de dos vectores.

El efecto resultante puede ser de mucha calidad para algunas situaciones, principalmente en los casos donde las direcciones del observador y de la luz son muy similares. Sin embargo, este modelo es más costoso que el cálculo de la simple iluminación difusa. Es por eso que se debe de tener cuidado de no usarlo cuando el efecto no sea tan notable.

#### 1.4.1.4. Modelo de Ward

En la naturaleza se pueden encontrar materiales con características muy diversas. Esta diferencia se manifiesta al iluminar un objeto utilizando diferentes modelos de iluminación. Los resultados con el modelo de Phong en ocasiones parecen “plásticos” cuando se iluminan objetos brillantes, siendo necesario utilizar otros modelos para materiales como los metales para poder obtener resultados más correctos y de mejor calidad.

Uno de estos modelos fue desarrollado por Ward [4] y tiene algunas ventajas sobre otros modelos de iluminación. La primera es que es físicamente más correcto que el modelo de Phong. La segunda es que puede ser utilizado en las formas isotrópicas (sus propiedades reflectivas no dependen del ángulo del observador respecto a la normal) y anisotrópicas (sus propiedades reflectivas dependen del ángulo del observador respecto a la normal).

En ambas formas, la componente difusa es un valor constante. La componente especular es un poco más compleja que en el modelo de Phong. La ecuación 11 resume los parámetros para la forma isotrópica. La iluminación especular se regula por un factor de rugosidad que es constante sobre la superficie y se denota por la letra griega sigma, ( $\sigma$ ).

$$I_0 = \left( D + S * \frac{e^{-\frac{\tan^2 \gamma}{\sigma^2}}}{2\pi\sigma^2\sqrt{(N \cdot L)(N \cdot V)}} \right) I_i(N \cdot L) \quad (11)$$

$$\gamma = \text{acos}(N \cdot H)$$

En la forma anisotrópica, el factor de rugosidad se define en dos direcciones ortogonales en la superficie. El resultado es un efecto anisotrópico en el brillo especular cuando la dirección del observador cambia. La ecuación 12 muestra la forma de calcular la forma anisotrópica.

$$I_0 = \left( D + S * \frac{e^{-\tan^2 \gamma \left( \frac{\cos^2 \phi}{\sigma_x^2} + \frac{\cos^2 \phi}{\sigma_y^2} \right)}}{2\pi\sigma_x\sigma_y\sqrt{(N \cdot L)(N \cdot V)}} \right) I_i(N \cdot L) \quad (12)$$

$$\gamma = \text{acos}(N \cdot H)$$

$\phi$  = ángulo del vector de la luz sobre el plano tangente

El resultado del modelo isotrópico se muestra en la Fig. 21 y como se puede observar es muy similar al modelo de Phong, aunque se pudieran notar sutiles diferencias entre ambos.



**Fig. 21 Iluminación con el modelo isotrópico de Ward.**

La Fig. 22 muestra dos variaciones del modelo anisotrópico. En el modelo de la izquierda la dirección de anisotropía es más horizontal, en el modelo de la derecha es más vertical.



**Fig. 22 Iluminación con el modelo anisotrópico de Ward.**

El ángulo  $\phi$  incluido en la ecuación 12 explica la dirección de anisotropía de algunas superficies anisotrópicas. Cuando el vector de la luz se proyecta sobre el plano tangente, el ángulo resultante entre el vector resultante proyectado y la dirección de anisotropía determina la cantidad de reflexión especular. Este modelo se utiliza con frecuencia cuando se representan objetos brillantes o metálicos en las escenas y donde otros modelos no muestran resultados reales.

#### 1.4.1.5. Modelo de reflexión de Cook–Torrance

El modelo de reflexión de Cook–Torrance está basado como su nombre lo indica en los trabajos de Torrance y Sparrow [4] y tuvo como objetivo la creación de un modelo de iluminación para las superficies metálicas. Posteriormente, este trabajo fue reformulado para los Gráficos por Computadoras con la ayuda de Cook y en alguna forma por Blinn, Beckman y otros que también dieron su aporte en este sentido.

Este modelo refleja una superficie como un conjunto formado por una gran cantidad de pequeñas microfacetas que reflejan la luz en direcciones diferentes. Este modelo también considera que algunas microfacetas bloquean la luz incidente sobre sus microfacetas adyacentes (sombreado), mientras que otras bloquean la luz emitida por las microfacetas adyacentes (enmascaramiento). Este fenómeno afecta la cantidad de reflexión difusa y especular que se refleja hacia el observador. Este modelo incorpora además un término de Fresnel para controlar la cantidad de luz reflejada en diferentes ángulos de incidencia. La ecuación 13 muestra la expresión final para el cálculo de la iluminación utilizando este modelo:

$$I_0 = I_i * \left( \frac{FDG}{(N \cdot L)(N \cdot V)} \right) \quad (13)$$

F representa el término de Fresnel y determina la cantidad de reflexión que emite cada microfaceta así como la apariencia de los objetos metálicos. Este término puede ser longitud de onda o un canal de color, permitiendo efectos de mayor calidad. En algunos casos es más fácil utilizar datos almacenados en texturas que formular una ecuación que reproduzca el efecto correcto. El término F se muestra en la ecuación 14:

$$F = \frac{(g - c)^2}{2(g + c)^2} \left( 1 + \frac{(c(g + c) - 1)^2}{(c(g + c) + 1)^2} \right) \quad (14)$$

$$c = (V \cdot H)$$

$$g = \sqrt{\eta^2 + c^2} - 1$$

D representa el término de la rugosidad de la superficie como la distribución de las pendientes de las microfacetas. Hay un par de fórmulas para calcular este término. Una de ellas es la función de distribución de Beckmann que se representan en la ecuación 15:

$$D = \frac{e^{-\left(\frac{\tan \alpha}{m}\right)^2}}{m^2 \cos^4 \alpha} \quad (15)$$

*m es la pendiente promedio de las microfacetas*

$$\alpha = \text{acos}(N \cdot H)$$

Una función de distribución más simple basada en una distribución Gaussiana fue sugerida por Blinn [4]. Esta forma es más rápida de calcular y se muestra en la ecuación 16:

$$D = ce^{-\left(\frac{\alpha}{m}\right)^2} \quad (16)$$

*c es una constante seleccionada por el usuario*

G representa el término geométrico y tiene en cuenta los efectos de sombreado y enmascaramiento entre las microfacetas y determina la cantidad de reflexión especular en una dirección dada. El término G a diferencia de los dos anteriores, no tiene ningún parámetro que dependa de las propiedades físicas de los materiales. La ecuación 17 representa el término geométrico:

$$G = \min \left\{ 1, \frac{2(N \cdot H)(N \cdot V)}{V \cdot H}, \frac{2(N \cdot H)(N \cdot L)}{V \cdot H} \right\} \quad (17)$$

#### **1.4.2. Modelos de iluminación global**

Los modelos de iluminación local permiten obtener buenos resultados, pero estos no tienen el grado máximo de realismo y no analizan la influencia de la luz en un punto después de la reflexión ni otros aspectos importantes como la generación de sombras. Es por eso que se utilizan los modelos de iluminación global, que si tienen en cuenta estos fenómenos y aumentan considerablemente la calidad de la iluminación de la escena. A continuación se explicarán algunos de estos modelos.

##### **1.4.2.1. Raytracing**

Este modelo es muy utilizado por la flexibilidad de su principio de funcionamiento y sirve como base para muchos algoritmos entre los que podemos mencionar el *RayCasting*. Este método consiste en trazar rayos a través de las trayectorias de reflexión y refracción entre los objetos de la escena [9]. Cada rayo se traza desde el observador, pasando a través de un píxel de la pantalla, hasta encontrar el primer objeto de la escena. Si el rayo no corta ningún objeto, entonces se pinta el píxel con el color del fondo.



Si el rayo trazado intercepta un objeto A, se calcula algún modelo de iluminación local para determinar el color, y se calcula además un rayo hacia la fuente de luz para determinar si otros objetos de la escena le hacen sombra al punto de la escena que se está iluminando. Si los objetos de la escena tienen además un coeficiente de reflexión especular o de refracción, entonces es necesario computar un factor proveniente de los rayos idealmente reflejados y refractados. Este método se repite recursivamente para calcular la intensidad de los rayos, para esto se trazan nuevos rayos que siguen el mismo procedimiento y que contribuyen al color del píxel. Ver Fig. 23.

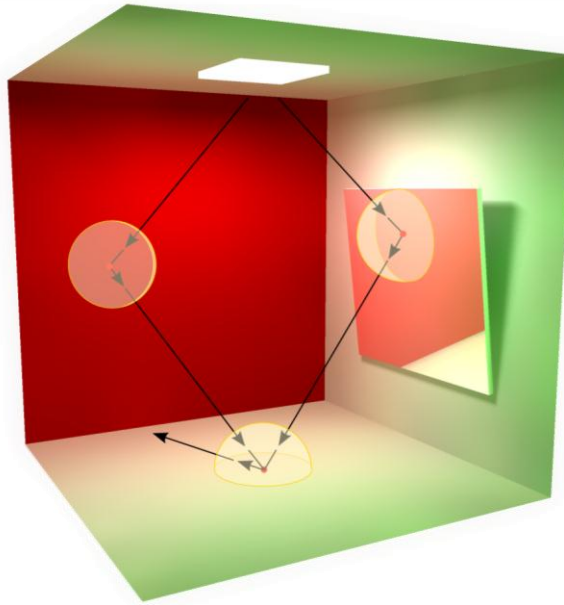


Fig. 23 Trazado de rayos recursivo.

La ecuación del trazado de rayos se puede obtener al incluir el factor reflejado y el factor refractado, obteniéndose la ecuación 18:

$$I = I_a K_a + I_l K_d (N \cdot L) + I_l K_s (R \cdot V)^n + I_r K_s + I_t K_t \quad (18)$$

donde:

$I_r$ : Intensidad del rayo reflejado

$K_s$ : Coeficiente de reflexión especular.

$I_t$ : Intensidad del rayo transmitido refractado.

$K_t$ : Coeficiente de transmisión

La principal ventaja de este método es que calcula correctamente el transporte especular y transparente, aunque no está exento de problemas de precisión numérica y como la mayoría de los modelos de iluminación global requiere un alto costo computacional.

#### **1.4.2.2. Formación de sombras**

La luz es capaz de interactuar con la mayoría de los objetos que encontramos en la naturaleza y como resultado generalmente se refracta, se trasmite o se refleja de acuerdo al material del objeto con el que interactúe. La luz puede pasar a través de un objeto o ser reflejada por este. Sin embargo, cuando un objeto impide el paso de la luz y no le permite atravesar el objeto parcial o totalmente, se genera una *sombra*.

Una sombra se define como el área en la cual existe una ausencia parcial o total de iluminación como resultado de la obstrucción entre la fuente de luz y el área que está siendo iluminada [1]. La sombra cambia de acuerdo a la iluminación y se caracteriza por la no uniformidad en cuanto a su forma y calidad. Esta variación de la forma permite su clasificación en *umbra* y *penumbra* [1].

La penumbra es el área de la sombra que está parcialmente iluminada y ocluida. Esta área tiene tonos más claros y suaves que el área central de la sombra, que es un poco más oscura. La umbra es el área de la sombra que está totalmente ocluida y que no tiene iluminación. Generalmente es más oscura en el centro y tiene cambios graduales de tonalidad que se mezclan con los de la penumbra.

La presencia de la umbra y la penumbra permiten definir dos grandes grupos para clasificar las sombras: las *sombras suaves* y las *sombras duras* [16]. Las sombras suaves son mucho más reales que las sombras duras y el grado de suavidad varía según la distancia entre la fuente de luz, el objeto ocluidor y el objeto receptor de la sombra. Las sombras suaves se logran obtener cuando se calcula las zonas de umbra y penumbra y el resultado es una imagen de mayor nivel de detalle y más cercana a la realidad. Las sombras duras se manifiestan como un efecto binario, cada píxel está sombreado o no. Este tipo de sombras es muy difícil encontrarlas en la vida real debido a la escasa existencia de fuentes de luces puntuales, que son las que generan este tipo de sombras y por esta razón las sombras duras brindan una sensación poco realista en la escena.

En los gráficos por computadoras es importante la formación de sombras porque su uso posibilita la obtención de visualizaciones muy reales y con buena calidad. El sistema visual humano utiliza los detalles de las sombras para caracterizar los materiales que componen los objetos, para

determinar si el objeto es de composición dura, para conocer la ubicación de la fuente de luz, para dar sensación de profundidad y para la relación entre los objetos del entorno. La simulación realista de sombra influye en la forma en que el observador percibe la escena.

La generación de sombras se ubica dentro de los modelos de iluminación global porque analiza el comportamiento indirecto de la luz para generar de forma correcta sombra en las escenas. Para cumplir este propósito se usan los modelos de sombras. Estos modelos son muy diversos y se aplican en dependencia de las características de la escena. En la siguiente sección se analizarán estos modelos así como sus ventajas y desventajas para la generación de sombras en los entornos virtuales.

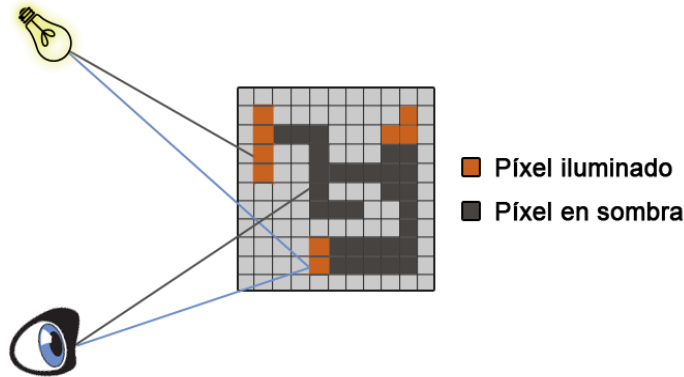
### **1.4.3. Modelos de sombras**

Muchos son los modelos que han sido creados para generar sombras. Cada uno tiene sus características, ventajas y desventajas que facilitan su elección en dependencia del efecto final que se desea representar en la escena. A continuación se explicarán algunos de los modelos que más se utilizan en la visualización directa de volumen.

#### **1.4.3.1. Rayos de sombras**

El algoritmo de rayos de sombra es una forma de integrar sombras utilizando la técnica de *Raytracing*. Este algoritmo se puede aplicar en algoritmos de visualización directa como el *RayCasting*. La esencia del algoritmo funciona de la siguiente forma: Para cada muestra del rayo que se traza desde el observador y que atraviesa el volumen de datos, se traza otro rayo hasta la fuente de luz para determinar si la muestra actual está en sombra. Si el rayo trazado hasta la fuente de luz intercepta algún píxel en la escena, quiere decir que algún objeto está ocluyendo el píxel analizado y por lo tanto este píxel estará sombreado. Por el contrario, si el rayo trazado hasta la fuente de luz no intercepta ningún píxel, entonces la luz incide directamente sobre el píxel y este se encontrará iluminado. El algoritmo de trazado de rayos para la generación de sombras se puede observar en la Fig. 24.

De este algoritmo se pueden implementar diferentes variantes en dependencia de la sombra que se desea obtener. En caso de que los objetos que ocluyan la fuente de luz sean opacos solamente, es decir que no tienen transparencia, el rayo trazado se va analizando hasta que se tome una muestra donde la opacidad acumulada supere el umbral definido para la sombra.



**Fig. 24 Algoritmo de Rayos de Sombra utilizando Raytracing.**

Si encontramos una muestra que cumple esta condición mientras se muestrea el rayo hacia la luz, entonces se termina antes el rayo y la muestra inicial estará sombreada. La desventaja de este método es que el efecto de la sombra se convierte en un resultado binario, es decir, la muestra inicial está sombreada o no. Para obtener resultados diferentes se puede modificar el valor del umbral utilizando alguna interfaz que le permita al usuario modificar este valor y obtener resultados visualmente diferentes.

Para incluir objetos oclusores semitransparentes el algoritmo incorpora las intensidades de las muestras tomadas en el rayo de sombra. Al tomar cada muestra del rayo, se va acumulando como intensidad el valor de transparencia de la muestra como grado de absorción de la luz. La desventaja de este método es que al no haber terminación temprana del rayo como en el primer caso, esta variante tiene un alto costo computacional y un mayor impacto en el rendimiento de la aplicación.

Para ambos algoritmos de sombra, un rayo es muestreado para cada muestra del rayo que atraviesa el volumen. Es por eso que el tiempo de ejecución es proporcional a la resolución de la pantalla y al tamaño de la muestra tomada en cada caso. Para aquellas escenas donde la calidad de la imagen deseada es alta, esta técnica puede impedir la interacción en tiempo real de la aplicación.

Existen técnicas que permiten optimizar este tipo de algoritmos. Una solución para optimizar la técnica de oclusores opacos es combinarla con la funcionalidad de render hacia una textura 3D con el objetivo de guardar el resultado del rayo de sombra. Así solo es necesario recalcular la sombra cuando cambia la fuente de luz o el objeto que ocluye la fuente de luz.

### 1.4.3.2. Mapas de sombras

La operación básica en la generación de sombras consiste en identificar las partes de la escena que no son iluminadas por la fuente de luz. Una forma equivalente de hacer esto es determinar las partes que son visibles por la luz.

El mapa de sombra es una técnica de sombreado de imágenes desarrollada por Lance Williams en 1978 [17]. Esta se puede implementar utilizando el hardware porque hace uso de funcionalidades de hardware existentes como los buffers de textura y de profundidad.

La primera fase del algoritmo para el cálculo de sombras inicia calculando una vista de la escena desde el punto de vista de la fuente de luz. Los valores de z de la imagen son almacenados en un buffer de profundidad y es lo que se conoce como *mapa de sombras* [16]. Ver Fig. 25. Posteriormente este mapa es usado para dibujar la escena desde el punto de vista del observador en un proceso de dos pasadas de render. Los pasos del algoritmo son los siguientes:

- 1) La técnica estándar del buffer de profundidad para eliminar las superficies ocultas.
- 2) Para cada píxel de la escena, se conoce la posición geométrica del objeto que se observa en este píxel. Si la distancia entre este objeto y la luz es mayor que la distancia almacenada en el mapa de sombras, el objeto está en sombra. De lo contrario se encuentra iluminado.
- 3) El color de los objetos es modulado de acuerdo a si estos están en sombra o no.

Muchas veces el acceso a los mapas de texturas consiste en filtrar valores sobre alguna región de la textura. El resultado de esta consulta es binario producto de la comparación y la sombra resultante se convierte en una sombra dura debido a que cada píxel está en sombra o está iluminado. La solución para este problema consiste en invertir el orden de la operación: primero se compara y luego se filtra para obtener bordes de sombras suaves. Esta técnica se conoce como *percentage closer filtering* [17].

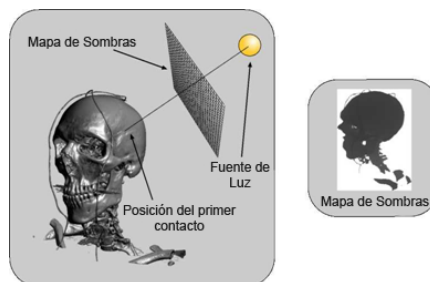


Fig. 25 El mapa de sombras utiliza una textura para almacenar la información de visibilidad desde la fuente de luz.

El resultado de esta técnica se calcula como el promedio de cuatro muestras obtenidas del mapa de profundidad. Por esta razón es necesario muestrear cuatro veces, calcular el valor promedio de las cuatro muestras y luego ponderar el resultado. Este es un filtrado bilineal ponderado en hardware. El resultado obtenido con este método brinda una medida de la cantidad de sombra en un fragmento de píxel dado y permite suavizar las sombras de las escenas, obteniéndose un resultado más real y con mejor calidad. Las diferencias entre las dos técnicas antes mencionadas se pueden ver en la Fig. 26.



**Fig. 26 Izquierda: Percentage closer filtering. Derecha: Mapas de sombras normal.**

Los mapas de sombras tienen algunas ventajas entre las que podemos mencionar las siguientes:

- ✓ Puede implementarse utilizando las modernas tarjetas gráficas.
- ✓ Crear el mapa de sombras es una operación relativamente rápida a pesar de que depende del número y la complejidad de los objetos que ocluyen la luz.
- ✓ Manipula correctamente el auto-sombreado.

Los mapas de sombras tienen algunas desventajas que es necesario tener en cuenta:

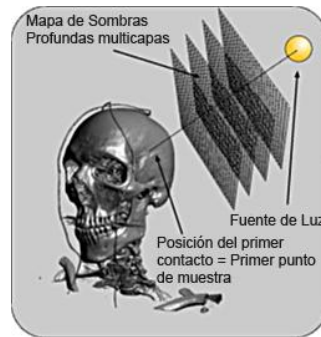
- ✓ Tiene problemas de sampling y de aliasing.
- ✓ No es capaz de manipular fuentes de luces omnidireccionales.
- ✓ Requiere al menos dos pasadas de render una desde la fuente de luz y otra desde el observador.

#### **1.4.3.3. Mapas de sombras profundas**

Los mapas de sombras permiten obtener sombras eficientes basadas en fragment shader, sin embargo no soportan los objetos oclusores semitransparentes que aparecen con frecuencia en la

visualización de volumen. Para lograr este objetivo se utilizan en ocasiones los mapas de opacidad de sombras. Estos funcionan como pilas de mapas de sombras que almacenan los valores de transparencia en vez de los valores de profundidad en cada mapa de sombras [18].

Una representación más compacta para el uso de ocluidores semitransparentes son los mapas de sombras profundas [19]. La estructura de datos consiste de una pila de texturas o capas que a diferencia de los mapas de opacidad almacenan una aproximación de la función de sombra. Ver Fig. 27.



**Fig. 27 Los mapas de sombras utilizan una pila de texturas que almacena una aproximación de la función de sombra.**

Los mapas de sombras profundas fueron propuestos por Lokovic y Veach en el año 2000. Esta técnica produce sombras eficientes y de alta calidad para primitivas como pelo, piel y humo. A diferencia de los mapas de sombras que almacenan un único valor de profundidad por cada píxel, los mapas de sombras profundas almacenan una distribución de profundidades por cada píxel. Esto posibilita que la técnica de percentage closer filtering pueda aplicarse en etapa de preprocesamiento. Como consecuencia cada consulta requiere una cantidad de trabajo constante e independiente del tamaño del filtro. En la Fig. 28 se observa la calidad de los resultados al utilizar los mapas de sombras profundas para renderizar pelo.



**Fig. 28 Izquierda: Pelo sin sombras. Derecha: Pelo con mapas de sombras profundas.**

Los mapas de sombras profundas son arreglos rectangulares de píxeles en los cuales cada píxel almacena una *función de visibilidad*. Una función de visibilidad se define considerando un rayo de luz con origen en la cámara y que pasa a través de un píxel dado. El valor de la función en una profundidad dada es una fracción de la energía del rayo inicial que penetra hasta esa profundidad.

Cada función inicia con un valor de 1 y va disminuyendo hasta 0 a medida que van apareciendo objetos a diferentes profundidades. La fracción de la luz que penetra hasta cierta profundidad se denomina *transmitancia* [17] y se representa por  $\tau(x, y, z)$ , donde  $\tau$  representa la función de transmitancia en el punto  $(x,y)$  como función de  $z$ .

La función de visibilidad para cada píxel se obtiene filtrando las funciones de transmitancia adyacentes y muestreando nuevamente el píxel central. Tomando una profundidad arbitraria  $z$ , la transmitancia para cada punto es dado por  $\tau$  y la función de visibilidad  $V_{i,j}$  puede obtenerse al filtrar estos valores:

$$V_{i,j}(z) = \int_{-r}^r \int_{-r}^r f(s, t) \tau\left(i + \frac{1}{2} - s, j + \frac{1}{2} - t, z\right) ds dt \quad (19)$$

Donde  $\left(i + \frac{1}{2} - s, j + \frac{1}{2} - t, z\right)$  es el centro del píxel examinado,  $f$  es el filtro del píxel y  $r$  es el radio del filtro.

Los mapas de sombras profundas fueron aplicados a la visualización directa de volumen por primera vez por Markus Hadwiger en su trabajo para la implementación de mapas de sombras profundas en la GPU [20]. La técnica original del mapa de sombras profundas almacena toda la intensidad de la luz en cada una de las capas. En la visualización directa de volumen aporta grandes ventajas guardar el nivel de absorción dada por el valor de transparencia. Así para cada rayo de sombra se analiza este valor alfa y se aproxima utilizando funciones lineales [21].

La aproximación se hace de la siguiente forma. Primeramente se calcula el primer punto que intercepta cada rayo de sombra al igual que con los mapas de sombras descritos anteriormente. Luego se guardan la distancia hasta la fuente de luz y el valor de transparencia para esta posición en la primera capa del mapa. En este primer punto, generalmente el valor de transparencia es 0 cuando el umbral de sombra se define apropiadamente. Luego, iniciando en este punto se recorre cada rayo de sombra y se chequea iterativamente que cada muestra encontrada se pueda aproximar con una función lineal. En caso de que la aproximación no sea



suficiente, la distancia de la muestra anterior a la fuente de luz así como el valor de transparencia acumulada en la muestra anterior son los valores que se almacenan en la siguiente capa del mapa. Esto se repite hasta que todas las capas del mapa hayan sido creadas. En la práctica con 8 capas para un mapa es suficiente para crear escenas de alta complejidad y realismo.

Para generar estas capas de una forma eficiente se pueden utilizar las tarjetas gráficas, pues es posible calcular 8 capas del mapa en una sola pasada de render y hacer un uso más eficiente de la memoria para el almacenamiento de los datos. Como resultado, la distancia hasta la fuente de luz y el valor de transparencia pueden ser almacenados en dos canales sucesivos de una textura RGBA, es decir, R y G así como B y A. De este modo dos capas se pueden representar usando una sola textura RGBA [21].

Los mapas de sombras profundas tienen varias ventajas. Al tener valores pre-filtrados, permiten consultas de sombras más rápidas y con menor consumo de memoria que los tradicionales mapas de sombras de similar calidad. Además soportan sombras de superficies parcialmente transparentes y objetos volumétricos como niebla. También permiten manipular casos importantes de desenfoques de movimiento sin costo adicional.

La desventaja fundamental de los mapas de sombras profundas es que son significativamente más costosos de calcular que los mapas de sombras regulares de la misma resolución porque es necesario tomar muchas más muestras por cada píxel analizado.

## CAPITULO 2. CARACTERÍSTICAS DEL SISTEMA.

En este capítulo se expondrán los modelos seleccionados como solución al problema de la iluminación de un modelo tridimensional reconstruido a partir de *RayCasting*. Partiendo de la explicación detallada de cada una de las etapas del cálculo de las variables que conforman los modelos implementados, se procederá a una integración de ambos modelos implementados. Posteriormente se explicará el uso del vector gradiente como una aproximación del vector normal para los datos volumétricos en el proceso de iluminación y finalmente se brindarán algunos detalles de implementación que involucran el uso de la tarjeta gráfica en el desarrollo de la solución final.

### 2.1. Iluminación en la visualización de volumen

La visualización de volumen se compone de cinco etapas [10]. Esta representa los datos de los vóxeles evaluándolos en el modelo óptico. Estos datos son mapeados a cantidades físicas y clasificados en la tercera etapa mediante la utilización de las funciones de transferencia y posteriormente en la cuarta etapa se calcula la iluminación. Finalmente en la última etapa se realiza la composición y se obtiene el valor final del píxel en la escena.

Para calcular la iluminación dentro del volumen de datos se seleccionaron dos modelos de iluminación: el modelo de iluminación local de Phong y el modelo de iluminación global de Rayos de Sombras.

En el *Raytracing* la combinación de ambos modelos permite calcular la iluminación y las sombras en un solo proceso: Por cada rayo que atraviesa el volumen de datos se toman muestras del volumen en diferentes intervalos de igual tamaño. Después de obtener el valor de la muestra, este valor es mapeado a propiedades ópticas en una tabla de búsqueda que proporciona la función de transferencia y que devuelve el color que será utilizado como material para el volumen. El color resultante incorpora los coeficientes correspondientes de emisión y absorción para evaluar el modelo óptico y finalmente realizar el proceso de composición. Este proceso se puede observar en la Fig. 29.

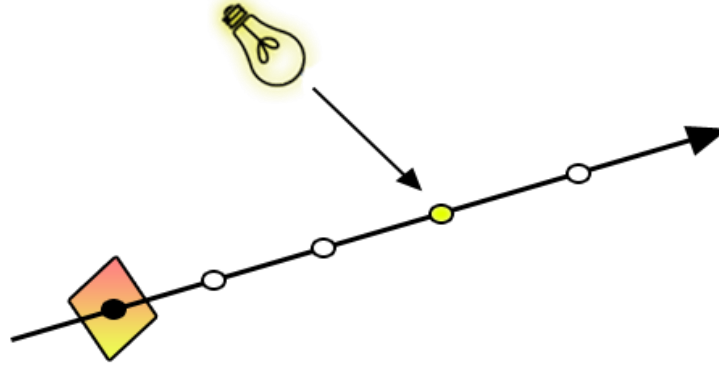


Fig. 29 Modelo de iluminación local en la visualización de volumen.

Cuando el rayo trazado intercepta algún objeto A, entonces se calcula el modelo de iluminación local de Phong (Ver epígrafe 1.4.1.1) utilizando el color de la función de transferencia como color del material y el vector gradiente como una aproximación del vector normal del vóxel. Luego se calcula otro rayo hasta la fuente de luz para determinar si algún objeto provoca sombra sobre el píxel analizado. Ver Fig. 30.

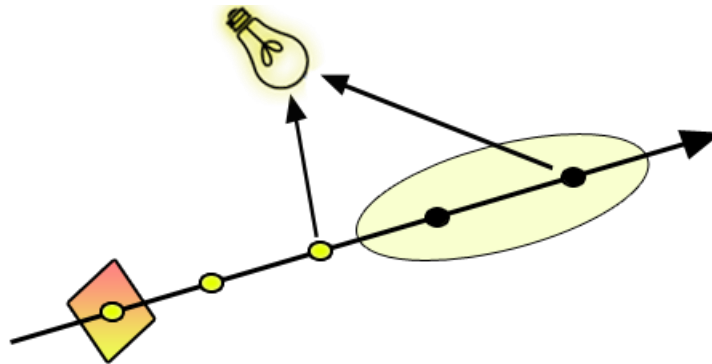


Fig. 30 Modelos de iluminación local y global en la visualización de volumen.

## 2.2. Modelo de Phong

El modelo de iluminación local seleccionado para la iluminación del volumen es el modelo de Phong. Este modelo permite obtener buenos resultados en la visualización de modelos anatómicos y permite su uso en la iluminación de otros datos como bloques de motores y bonsáis se utilizan en la Visualización Científica. Existen otros modelos como el de Oren–Nayar que son más útiles para materiales difusos o el modelo de Cook–Torrance diseñado para superficies metálicas. Otro modelo estudiado fue el mapa de reflexión. Este modelo es muy útil para escenas estáticas y pequeñas en cuanto al número de objetos en la escena, teniendo como mayor ventaja el precálculo de la iluminación de la escena y evitando la repetición de

cálculos para obtener aplicaciones en tiempo real. Este modelo no es muy útil para la visualización de volumen porque esta incluye un análisis detallado de los modelos visualizados y por esta razón las escenas deben tener un alto grado de dinamismo; este elemento impide aprovechar las ventajas de este modelo. Los modelos antes mencionados no permiten obtener resultados tan reales ni con la calidad que necesita la visualización de volumen.

El modelo de Phong también permite adaptar la iluminación clásica de las superficies poligonales a la iluminación de volumen mediante la utilización de gradientes. El modelo de Phong analiza las propiedades de los materiales que ilumina, permitiendo obtener resultados más reales que otros modelos. Otros elementos importantes que lo hacen un buen candidato para la solución es que es un modelo simple, fácil de evaluar y que no requiere un alto costo computacional. Es por eso que su utilización posibilita mantener la interactividad en tiempo real con la aplicación una vez que se incorpora el modelo de iluminación a la visualización de volumen. Este elemento es muy importante porque a pesar de incrementar los cálculos y mejorar la calidad de la visualización, mantiene un buen rendimiento para la aplicación.

### **2.3. Rayos de Sombras**

El modelo de iluminación global seleccionado para la generación de sombras fue el de Rayos de Sombra. Este pertenece al grupo de técnicas de Raytracing. Este modelo es muy utilizado por la flexibilidad de su principio de funcionamiento y porque su integración con el algoritmo *RayCasting* muestra resultados visuales muy reales a diferencia de otros algoritmos. Otros elementos que influyeron en la selección de este modelo son algunas desventajas que presentan los otros modelos analizados.

El modelo de mapas de sombras es un modelo muy bueno y permite obtener aplicaciones en tiempo real si es capaz de aprovechar al máximo el precálculo de la sombras durante la visualización. Para la visualización de volumen, las escenas que se requieren son muy dinámicas y por esta razón los cambios de cámara y de la fuente de luz hacen necesario recalcular constantemente la sombra de la escena y de esta forma no es posible aprovechar su mayor ventaja.

El modelo de mapas de sombras profundas por otra parte utiliza mayor cantidad de memoria para el cálculo de las sombras porque almacena esta información en varias texturas, aumentando el consumo de recursos. Además durante la generación de sombras para estructuras muy finas introduce artefactos que disminuyen el realismo de la visualización. Es por eso que se seleccionó el modelo de Rayos de Sombras como solución. Este permitió

alcanzar un equilibrio entre los resultados visuales alcanzados y el consumo de recursos. Otro resultado importante es que permite generar ambos tipos de sombras: sombras duras y sombras suaves semitransparentes (Ver epígrafe 1.4.3.1).

## 2.4. Estimación de gradientes

El modelo de Phong utiliza el vector normal para describir la forma de un objeto y se usa fundamentalmente en la iluminación de superficies poligonales. Al incluir el modelo de Phong en la visualización directa de volumen, este vector normal no existe porque los datos volumétricos no tienen superficies definidas. Por esta razón necesitamos construir una normal sustituta para estos casos.

Una técnica estándar que se emplea para sustituir la normal en la visualización de volumen es el valor del vector gradiente de los vóxeles como normales, aun cuando no exista una superficie presente [10]. El uso del gradiente es motivado porque asegura la suavidad de la intensidad de los vóxeles en el área analizada y garantiza que no haya grandes diferencias en la intensidad de los vóxeles. Este permite adaptar la iluminación de superficies poligonales para iluminar datos volumétricos. El vector gradiente es la derivada de primer orden de la función escalar  $f(x, y, z)$  y se representa con la siguiente ecuación:

$$\nabla f = (f_x, f_y, f_z) = \left( \frac{\delta}{\delta x} f, \frac{\delta}{\delta y} f, \frac{\delta}{\delta z} f \right) \quad (20)$$

Esta ecuación se puede calcular usando las derivadas parciales en las direcciones x, “y” y z respectivamente. La magnitud escalar del gradiente brinda una medida de la variación local de la intensidad de la luz cuantitativamente. Esta se puede calcular como el valor absoluto del vector:

$$\|f\| = \sqrt{f_x^2 + f_y^2 + f_z^2} \quad (21)$$

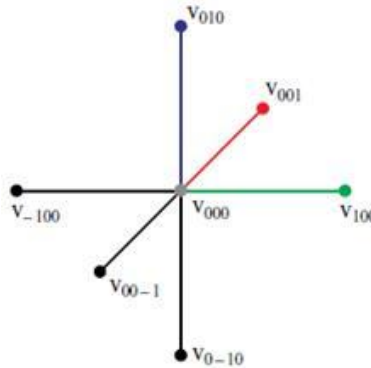
Para propósitos de iluminación solo la dirección del vector gradiente es necesaria. Existen muchas técnicas que se han desarrollado para el cálculo del gradiente. A continuación se analizarán algunas de las técnicas más usadas para la estimación de gradientes y se determinará el método utilizado como solución.

### 2.4.1. Diferencias intermedias

El cálculo de gradiente usando las diferencias intermedias considera una vecindad limitada del vóxel actual. Consiste de las intensidades de los vóxeles adyacentes con un índice

incrementado (diferencias delanteras) y el vóxel actual. El resultado se puede observar en la Fig. 31. La ecuación de este método se formaliza como se muestra la ecuación 22:

$$D_{000} = \begin{pmatrix} D_{000,x} \\ D_{000,y} \\ D_{000,z} \end{pmatrix} = \begin{pmatrix} V_{100} - V_{000} \\ V_{010} - V_{000} \\ V_{001} - V_{000} \end{pmatrix} \quad (22)$$



**Fig. 31 El método de diferencias intermedias considera el valor del vóxel  $V_{000}$  para el valor final.**

Como una variante de los vóxeles adyacentes siguientes utilizados en este método, se puede considerar el uso de los vóxeles adyacentes anteriores. Estos se pueden obtener al decrementar el índice del vóxel a seleccionar. En dependencia del índice utilizado será el nombre que recibe el método: diferencias delanteras cuando se incrementa el índice y diferencias traseras al decrementar el índice.

Este método se puede definir como el kernel  $[-1,1]$  en dependencia de la operación que se le aplique al índice del vóxel seleccionado para calcular las componentes del gradiente. Los resultados de este método no son tan reales como es el caso de las diferencias centrales y el Operador de Sobel. Su principal ventaja es la rapidez porque por cada vóxel se analiza una vecindad limitada de seis vóxeles. Esto trae como consecuencia que calcular el gradiente sea una operación muy rápida.

### **2.4.2. Operador de Sobel**

De todos los métodos utilizados para la aproximación de gradientes, este es quizás el más complejo y el de mayor suavidad en su resultado visual. Este método fue acreditado a Sobel en 1969 [8].

Este método considera el valor de la intensidad del vóxel actual y el de sus 26 vecinos inmediatos. La ecuación resultante para cada vóxel es una ecuación bastante compleja, por este motivo solo se mostrarán tres iteraciones para cada componente del gradiente, usando las

ecuaciones 23 hasta la 25 para representar la componente x, la “y” y la z respectivamente. La primera matriz es para la iteración anterior, la segunda para la iteración actual y la tercera para la siguiente iteración.

$$\begin{bmatrix} -1 & -3 & -1 \\ -3 & -6 & -3 \\ -1 & -3 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 3 & 1 \\ 3 & 6 & 3 \\ 1 & 3 & 1 \end{bmatrix} \quad (23)$$

$$\begin{bmatrix} 1 & 3 & 1 \\ 0 & 0 & 0 \\ -1 & -3 & 1 \end{bmatrix}, \begin{bmatrix} 3 & 6 & 3 \\ 0 & 0 & 0 \\ -3 & -6 & -3 \end{bmatrix}, \begin{bmatrix} 1 & 3 & 1 \\ 0 & 0 & 0 \\ -1 & -3 & -1 \end{bmatrix} \quad (24)$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -3 & 0 & 3 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -3 & 0 & 3 \\ -6 & 0 & 6 \\ -3 & 0 & 3 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 1 \\ -3 & 0 & 3 \\ -1 & 0 & 1 \end{bmatrix} \quad (25)$$

El método de Sobel tiene un relativo alto costo computacional, es por eso que en ocasiones este operador se aplica en una fase de preprocesamiento. Esto tiene la desventaja de no tener en cuenta el efecto de la función de transferencia en el valor final del gradiente. Es por eso que los métodos diferencias centrales y diferencias intermedias son preferidos cuando el rendimiento del render y de la calidad visual es muy importante.

### 2.4.3. Diferencias centrales

El cálculo aproximado del gradiente usando este método fue propuesto por Höhne y Bernstein en 1986 [8]. Este aproxima el gradiente en el vóxel  $V_{000}$  calculando la diferencia de intensidad en su vecindad cercana. Este considera los seis vóxeles vecinos, excepto el propio vóxel actual ( $V_{000}$ ). Ver Fig. 32. La ecuación de este método se formaliza en la ecuación 26:

$$D_{000} = \begin{pmatrix} D_{000,x} \\ D_{000,y} \\ D_{000,z} \end{pmatrix} = \begin{pmatrix} V_{100} - V_{-100} \\ V_{010} - V_{0-10} \\ V_{001} - V_{00-1} \end{pmatrix} \quad (26)$$

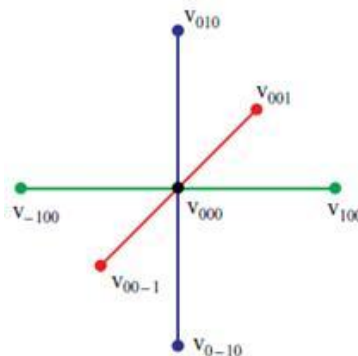


Fig. 32 Método de diferencias centrales. Los colores azul, rojo y verde identifican a los vóxeles vecinos.

Para la utilización del vector gradiente como normal, el vector con las diferencias calculadas debe ser normalizado:  $\vec{N} = \nabla V_{ijk} = ||D_{000}||$ . El operador para calcular las diferencias centrales de un vóxel se puede definir como un filtro de paso alto que se puede expresar como un kernel [-1,0,1]. Este filtro se aplica para cada componente del vector gradiente dentro del volumen.

La elección de un método o de otro no es un tema que se debe decidir a la ligera. Esta constituye una decisión que tiene que analizar el equilibrio entre la calidad de la imagen final que se desea obtener y el costo computacional que trae consigo cada método.

El operador de Sobel permite obtener los resultados de mejor calidad visual porque el cálculo de los gradientes analiza una mayor vecindad. Por otro lado, las diferencias intermedias muestran una mayor sensibilidad al ruido de la imagen y a los artefactos de interpolación.

El operador de Sobel permite una mejor reconstrucción de estructuras que los otros métodos, aunque en ocasiones puede eliminar estructuras finas y pequeñas y hace un mayor consumo de los recursos computacionales. En estos casos las diferencias centrales e intermedias son más apropiadas para la aproximación de la normal. Las diferencias centrales ofrecen soluciones con buena calidad visual a un costo modesto. Las diferencias intermedias muestran los resultados de menor calidad al mismo costo computacional que las diferencias centrales, a pesar de ser un poco más fiable a los datos originales. Considerando todas estas ventajas y desventajas, se seleccionó el método de las diferencias centrales como solución. Este se ha convertido en el método estándar para el cálculo de las normales basadas en gradientes para la visualización de volumen [8], tanto en la visualización directa como en la visualización indirecta.

## 2.5. Implementación sobre GPU

El uso de la tarjeta gráfica tiene una gran ventaja por la rapidez de los cálculos sobre el CPU y por la capacidad del trabajo en paralelo y por esta razón se utilizó para realizar el cálculo de los modelos de iluminación. Para utilizar la tarjeta gráfica es necesario conocer el pipeline gráfico del hardware [10] que es donde se especifica el orden de las operaciones al ejecutarse en la tarjeta.

El hardware gráfico permite programar secuencias de códigos capaces de ejecutarse en dos módulos fundamentales: el vertex shader, donde se realizan las operaciones relacionadas con el procesamiento de los vértices y el fragment shader, donde se realizan operaciones como el cálculo del color, el proceso de texturizado y el cálculo de la iluminación por píxel [10], [22].



Como lenguaje para la implementación se utilizó GLSL. Este es un lenguaje de alto nivel basado en C que tiene gran compatibilidad con OpenGL. Además tiene una sintaxis muy similar a la del lenguaje C++ que facilita el desarrollo y el aprendizaje del mismo para el desarrollo de programación de shader [22].

El cálculo de la iluminación se puede realizar tanto en el vertex como en el fragment shader. En la solución propuesta el cálculo se realizó en el fragment shader porque los datos visualizados carecen de geometría y el volumen de datos no tiene vértices definidos. Por esta razón el vertex shader solo se utilizó para el cálculo de las coordenadas que se utilizaron en el proceso de texturizado.

En el fragment shader se implementó el cálculo de la iluminación en cada píxel de la imagen. Para cada píxel se calculó cada componente del modelo de Phong y se adicionó el color resultante al proceso de composición final del *RayCasting*. De esta forma el resultado que se obtiene es más real y exacto al no realizar interpolación para valores intermedios. Esto provoca un mayor costo computacional a cambio del aumento de la calidad de la imagen final.

Otro cálculo importante realizado en el fragment shader fue el cálculo de la sombra con el algoritmo de rayos de sombra. Esta técnica permite incluir sombras a la escena y obtener un resultado más real y con mayor sensación de profundidad. Esto posibilita mejorar la visualización de estructuras finas y pequeñas y hacer un análisis más detallado del modelo presente en la escena.

## **2.6. Herramientas**

Para implementar la solución propuesta se utilizaron algunas herramientas que se describirán en este epígrafe para tener una idea más profunda sobre la implementación realizada.

Como herramienta de modelado se utilizó Visual Paradigm. Esta es una herramienta que utiliza UML como lenguaje de modelado. Es considerada una herramienta muy completa, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Ofrece además la capacidad de ingeniería directa (versión profesional) e inversa. Es un producto de probada calidad que soporta varios idiomas y con licencia gratuita y comercial. La herramienta está diseñada para una amplia gama de usuarios interesados en construir sistemas de software fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios.

Como entorno de desarrollo integrado se utilizó *Microsoft Visual Studio.NET 2008*. Este entorno de trabajo ofrece a los desarrolladores un conjunto unificado, orientado a objetos, jerárquico y extensible de bibliotecas de clases (API). Este cuenta además con herramientas como el Visual Assist que permiten el completamiento de código y contribuyen a la codificación de la aplicación. Otras hacen posible comprobar el rendimiento de la aplicación basados en la memoria y el uso del CPU.

Para el diseño y programación de la GUI de la aplicación se utilizó el framework Qt. Este es un framework de alta compatibilidad, tiene características que lo hacen muy versátil y el código C++ que utiliza tiene un alto rendimiento. El código fuente está disponible y así como una muy buena documentación que hace de este una opción muy buena para los desarrolladores. Tiene una arquitectura muy flexible que permite diseñar aplicaciones sin mucho esfuerzo y con una gran calidad. Tiene un apoyo técnico de alta calidad y sigue el principio de reutilizar el código para crear más y hacer despliegues sin importar el lugar.

## **2.7. Lenguajes**

En este epígrafe se abordarán los principales lenguajes que se utilizaron como parte de la solución propuesta. Estos incluyen el lenguaje principal de desarrollo, el lenguaje para la programación de shader y el lenguaje para modelar el análisis y diseño de la aplicación.

El lenguaje de programación seleccionado fue C++. Este es uno de los lenguajes más potentes para desarrollar aplicaciones porque permite programar a un alto nivel y tiene mecanismos muy buenos como la herencia y el polimorfismo que le brindan al desarrollador gran flexibilidad para diseñar. Al ser un lenguaje orientado a objetos permite encapsular los datos y los métodos en clases, esto posibilita obtener un código más seguro y con una mejor organización. Otra ventaja muy importante que se tuvo en cuenta es que este tipo de aplicaciones manejan grandes volúmenes de datos y necesitan interacción en tiempo real y no todos los lenguajes posibilitan esto: en el caso de C++ es un candidato ideal por su gran rapidez.

El lenguaje utilizado para la programación de shader fue GLSL. GLSL es el acrónimo de OpenGL Shading Language y es un lenguaje de alto nivel basado en C. Fue creado por OpenGL ARB para aportar a los desarrolladores un control directo sobre el pipeline gráfico sin la necesidad de usar lenguaje ensamblador o lenguajes específicos de hardware. Este lenguaje es multiplataforma y compatible con múltiples sistemas operativos como Linux, Macintosh y Windows. Tiene la capacidad de escribir los shader para utilizarse en cualquier gráfico del vendedor de la tarjeta de video. El lenguaje tiene una sintaxis muy similar a la del lenguaje

C++, esto facilita el desarrollo y el aprendizaje a los que se inician en el desarrollo de programación de shader.

El lenguaje seleccionado para modelar el análisis y diseño de la aplicación fue UML. Este permite verificar y validar el modelo realizado. Es un lenguaje independiente de la plataforma y constituye un estándar mundial en el modelado. Este brinda además la posibilidad de modelar sistemas utilizando técnicas orientadas a objetos (OO) así como la documentación de todos los artefactos de desarrollo. Puede conectarse con los lenguajes de programación y posibilita la ingeniería inversa.

## **2.8. Metodología de desarrollo de software**

Como metodología de desarrollo de software fue seleccionado el Proceso Unificado para Desarrollo de Software (RUP). Este se basa en las mejores prácticas que se han probado en este campo y es el desarrollo más general de los existentes actualmente [23].

Este proceso es el resultado del esfuerzo de las tres últimas décadas de desarrollo de software y de la experiencia de sus creadores Ivar Jacobson, Grady Booch y James Rumbaugh. Utiliza como único lenguaje de modelado UML, este puede ser utilizado para la construcción de todos los modelos necesarios durante el desarrollo de la aplicación [24]. Otras de las ventajas que tiene RUP son que está guiada por casos de uso, centrada en la arquitectura y se desarrolla como un proceso iterativo e incremental [25].

## CAPITULO 3. DESCRIPCIÓN DE LA SOLUCIÓN

Este capítulo tendrá como tema fundamental el análisis y diseño de la aplicación. Primeramente se definirán algunas reglas del negocio y el modelo de dominio a utilizar. Luego se hará la captura de requisitos, que será utilizada posteriormente en la elaboración del Modelo de Casos de Uso. Se definirán los actores del sistema y se especificará cada caso de uso obtenido a partir de la captura de requisitos. Se modelarán diferentes diagramas de secuencia del diseño con el objetivo de proporcionar una comprensión más profunda del sistema que se desea desarrollar.

### 3.1. Reglas del Negocio

1. Las imágenes médicas que se deseen visualizar deben estar en el formato \*.dcm o en el \*.raw.
2. Las imágenes con el formato \*.dcm deben tener la estructura original, las que no cumplan con este requisito no podrán ser visualizadas de forma correcta.
3. Las imágenes con el formato \*.raw deben tener las dimensiones del estudio sin modificaciones. En caso de errores en las dimensiones originales, la visualización no será siempre correcta.

### 3.2. Modelo de Dominio

El modelo de dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema y conceptos del mundo real en vez de componentes de software.

El entorno donde está presente el problema describir de la siguiente forma: un médico le orienta a un paciente una serie de estudios médicos que generan un conjunto de imágenes, las cuales son generadas por las tomografías computarizadas (TAC) o resonancias magnéticas (MRI); donde a partir del análisis de las mismas el médico emite un diagnóstico. En la Fig. 33 se muestra la descripción del negocio mediante un modelo de dominio. A continuación se describen los conceptos que forman parte del modelo de negocio:

El **Médico** es la persona capacitada en el manejo y conocimiento de los métodos de Imagenología para emitir un diagnóstico.

Un **Estudio** es el examen orientado por el médico al paciente para obtener una patología y emitir un diagnóstico.

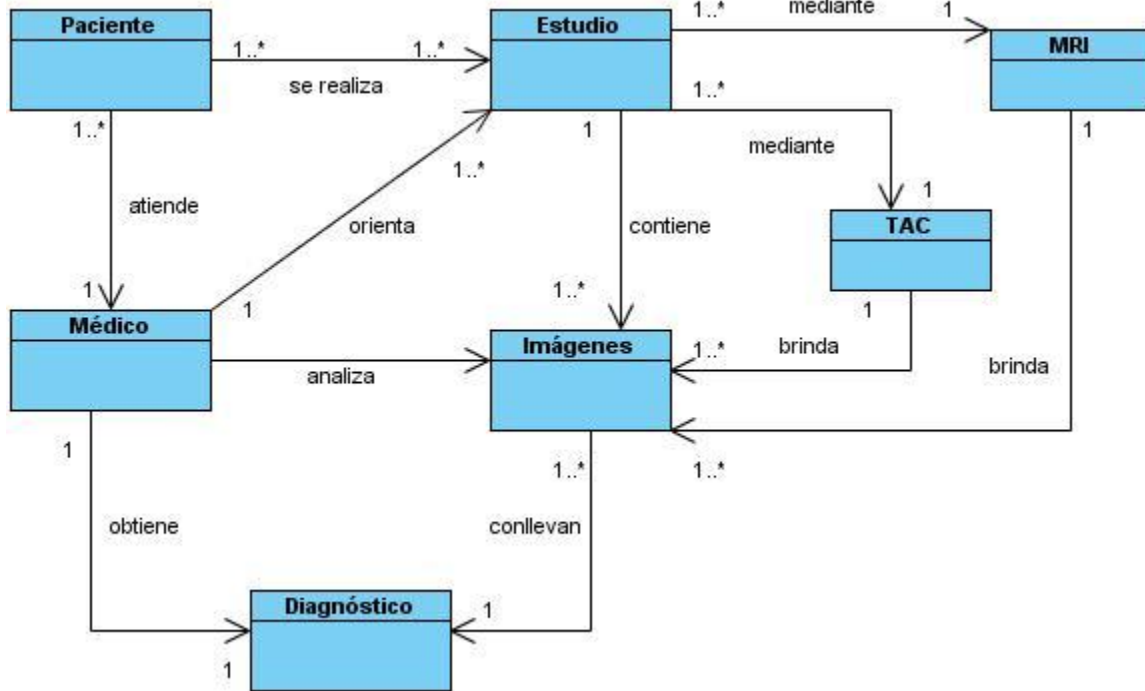


Fig. 33 Modelo de Dominio.

Las **Imágenes** constituyen el resultado del estudio orientado por el médico al paciente a través del TAC y MRI.

El **Paciente** es aquel que recibe los servicios de un médico u otro profesional de la salud, sometiéndose a un examen.

Una **MRI** (Resonancia Magnética) es una prueba que realiza una fotografía de los órganos y tejidos internos sin exponer el paciente a radiaciones. Utiliza un campo magnético poderoso y un tipo de frecuencia de radio que realiza imágenes computarizadas.

Una **TAC** (Tomografía Axial Computarizada) es un procedimiento de diagnóstico médico que utiliza rayos X con un sistema informático que procesa las imágenes y permite obtener imágenes radiográficas en secciones progresivas de la zona del organismo estudiada, y si es necesario, imágenes tridimensionales de los órganos o estructuras orgánicas.

El **Diagnóstico** es el resultado que da el médico luego de realizar un estudio a un paciente.

### 3.3. Captura de Requisitos

La captura de requisitos es de vital importancia en todo sistema. Esta disciplina explica cómo obtener las solicitudes de los interesados y transformarlas en un conjunto de productos de trabajo de los requisitos que cubran el ámbito del sistema que va a crearse y proporcionen requisitos detallados sobre lo que el sistema debe hacer [25]. Estos pueden ser requisitos funcionales o no funcionales. Los requisitos funcionales son los que especifican las funcionalidades que debe de tener el sistema, mientras que los requisitos no funcionales especifican las características que desea el cliente que tenga el sistema. Los siguientes epígrafes tienen como objetivo identificar los requisitos funcionales y no funcionales utilizados en la solución propuesta.

#### 3.3.1. Requisitos Funcionales

Los *requisitos funcionales* representan las funcionalidades del sistema. Estos son transformados en casos de uso y modelados posteriormente. Los siguientes requisitos responden a las funcionalidades que el sistema debe tener una vez concluida su implementación.

RF1. Gestionar luz.

- RF1.1 Adicionar luz
- RF1.2 Modificar propiedades de la luz
- RF1.3 Eliminar luz

#### 3.3.2. Requisitos No Funcionales

Los *requisitos no funcionales* representan aquellos atributos que debe tener el sistema, pero que no son una funcionalidad específica. Por ejemplo requisitos de facilidad de uso, fiabilidad, eficiencia, portabilidad. Los requisitos no funcionales que debe de tener el módulo elaborado son los siguientes:

1. Software

Se debe de tener instaladas las bibliotecas de OpenGL y Glew para los shader.

2. Hardware

El modelo de microprocesador será Intel Pentium IV a 3.0 GHz o superior.

La memoria RAM será de 1GB.

La tarjeta gráfica usada será NVidia GeForce 9800 GT de 512 MB o superior.

3. Seguridad

El sistema debe ser seguro frente a posibles ataques que se produzcan en la introducción de los datos de las imágenes y debe realizar un correcto tratamiento de errores para evitar el colapso de la aplicación al ocurrir algún error.

4. Interfaz Externa

La interfaz de usuario debe ser sencilla y amigable para permitir al usuario una rápida y cómoda interacción con las funcionalidades del módulo.

5. Soporte

Se brindará soporte para los sistemas operativos Windows XP, Windows 7 y Ubuntu.

6. Diseño e Implementación

Se empleará como lenguaje de programación C++ y el framework Qt para el diseño de las interfaces gráficas.

**3.4. Modelo de Casos de Uso del Sistema**

En este epígrafe se identificarán los actores del sistema que se desea desarrollar así como los casos de uso. Además se hará la descripción textual de los casos de uso del sistema, permitiendo así una comprensión más precisa de la lógica del funcionamiento del módulo.

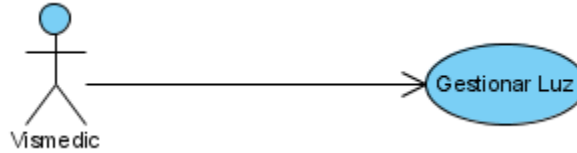
**3.4.1. Actores del Sistema**

Los actores del sistema son entidades externas al sistema que guardan una relación con este y que le demandan una o más funcionalidades. Esto incluye a los operadores humanos, pero también incluye a todos los sistemas externos. En este caso particular, quien hará uso de las funcionalidades del módulo de iluminación será la aplicación del proyecto Vismedic. Este será el actor del sistema y se llamará Vismedic.

Actores	Justificación
Vismedic	Se encarga de gestionar las luces de la escena para poder iluminar el volumen de datos y aumentar el realismo y la calidad de la visualización.

Tabla 1 Actor del Sistema.

**3.4.2. Diagrama de Casos de Uso del Sistema**



**Fig. 34 Diagrama de Casos de Uso del Sistema.**

**3.4.3. Descripción de Casos de Uso del Sistema**

Cada caso de uso tiene una descripción de las funcionalidades que realiza. Las tablas siguientes muestran los flujos operacionales de cada caso de uso.

<b>Caso de Uso:</b>	Gestionar Luz
<b>Actores:</b>	Médico Especialista
<b>Propósito:</b>	El caso de uso gestiona las luces de la escena. El usuario puede adicionar luces, modificarlas o eliminar las luces que desee.
<b>Resumen:</b>	El caso de uso se inicia al seleccionar la opción de iluminación. Luego se selecciona la opción que desea realizar: adicionar una nueva luz, modificar una existente o eliminar una luz.
<b>Referencia:</b>	RF1.1, RF1.2, RF1.3.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción adicionar, modificar o eliminar.	1.1 Ejecuta una de las siguientes acciones: a) Adicionar una luz. Consultar la sección Adicionar Luz. b) Modificar una luz. Consultar la sección Modificar Luz. c) Eliminar una luz. Consultar la sección Eliminar luz. 1.2 Actualiza la visualización de la escena.
<b>Postcondiciones:</b>	Se gestionó una luz de la escena.
<b>Prioridad:</b>	Crítica.

**Tabla 2 Descripción del CU Gestionar Luz.**



<b>Sección:</b>	Adicionar luz
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción adicionar.	1.1 Muestra en el cuadro de diálogo de luces la nueva luz adicionada y sus propiedades.
<b>Postcondiciones:</b>	Se adicionó una nueva luz en la escena.
<b>Prioridad:</b>	Crítica.

Tabla 3 Descripción de la sección Adicionar luz del CU Gestionar Luz.

<b>Sección:</b>	Modificar propiedades de la luz
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la luz que desea modificar.	1.1 Muestra las propiedades de la luz seleccionada por el usuario.
2. Modifica las propiedades que desee: el estado de la luz, el estado de la sombra, la posición de la luz o sus componentes ambiente, difusa y especular.	2.1 Actualiza las propiedades de la luz seleccionada por el usuario.
<b>Postcondiciones:</b>	Se modificó una luz de la escena.
<b>Prioridad:</b>	Crítica.

Tabla 4 Descripción de la sección Modificar propiedades de la luz del CU Gestionar Luz.

<b>Sección:</b>	Eliminar luz
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona en el cuadro de diálogo la luz que desea eliminar.	1.1 Muestra las propiedades de la luz seleccionada.
2. Presiona la opción eliminar.	2.1 Elimina la luz del cuadro de diálogo de luces de la escena. Si la luz eliminada era la única luz en la escena, se deshabilita la opción de eliminar hasta que se adicione una nueva luz.
<b>Postcondiciones:</b>	Se eliminó una luz de la escena.
<b>Prioridad:</b>	Crítica.

**Tabla 5 Descripción de la sección Eliminar luz del CU Gestionar Luz.**

### 3.5. Diagrama de Clases del Diseño del Paquete de Iluminación

El diagrama de clases del diseño de Vismedic fue concebido por paquetes con el objetivo de lograr una mayor organización e independencia para los módulos fundamentales de la aplicación final. Esto posibilita mantener una buena claridad y una mejor comprensión. Para ello se desarrollaron tres paquetes: el de Visualización, el de Función de Transferencia y el de Iluminación.

El paquete de Visualización contiene todas las clases y los parámetros necesarios para lograr la visualización del volumen de datos. Este será clasificado posteriormente utilizando la función de transferencia y finalmente los colores asignados serán los colores difusos utilizados por el paquete de iluminación. En la Fig. 35 se muestra la estructura organizativa del diagrama de paquetes y las relaciones que existen entre el paquete de Iluminación y los otros paquetes de Vismedic.

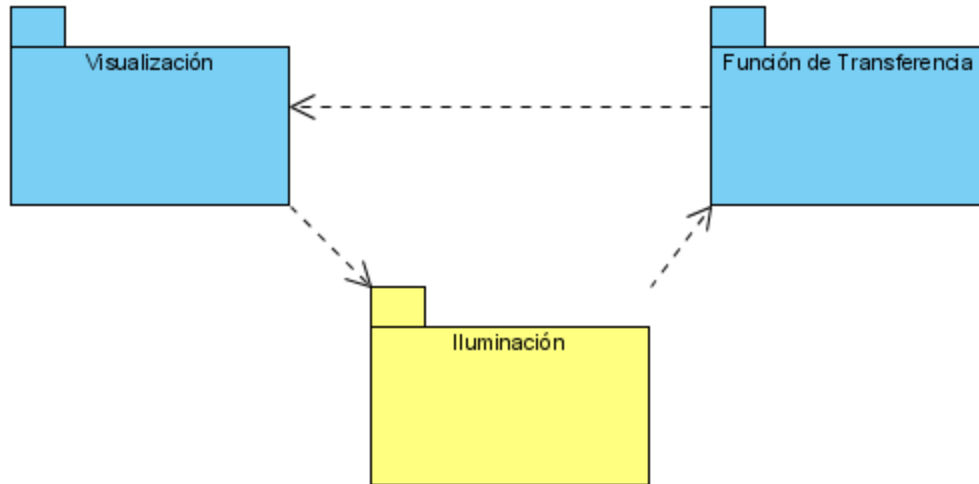


Fig. 35 Relación entre el paquete de iluminación implementado y los demás módulos del sistema Vismedic.

### 3.5.1. Diagrama de Clases del Paquete de Iluminación

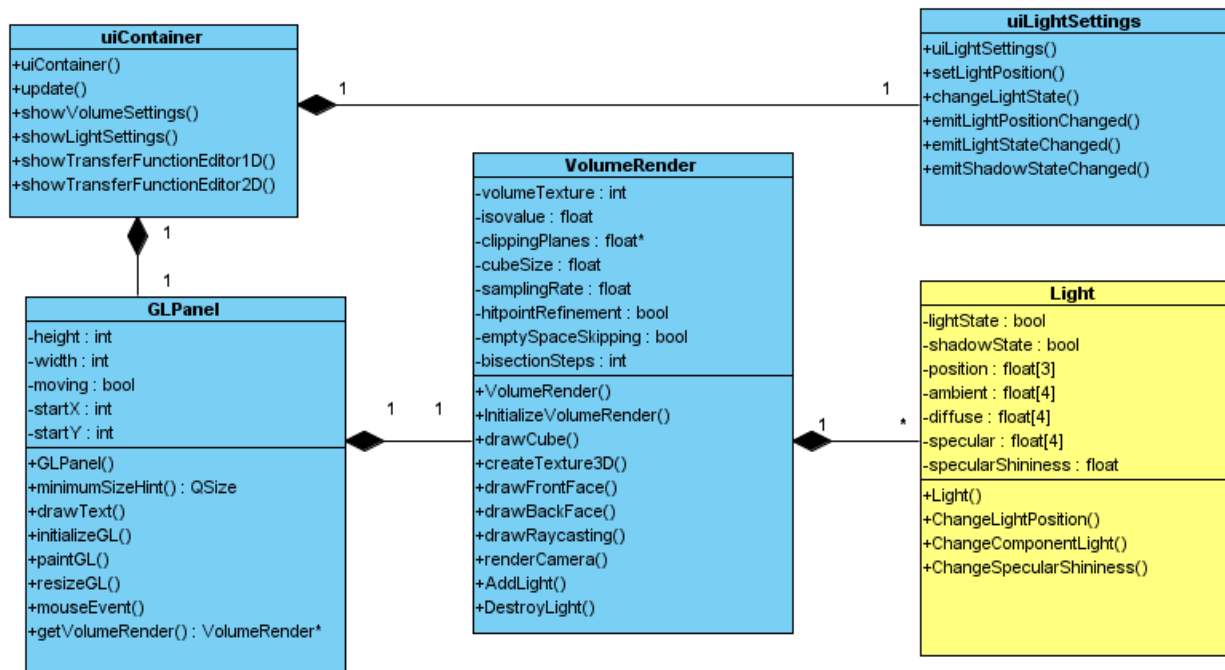


Fig. 36 Relación entre las clases del paquete de Visualización y el módulo de iluminación propuesto.

### 3.6. Diagrama de Secuencia del Diseño

A continuación se representarán los diagramas de secuencia del diseño para tener una idea más general sobre el flujo que se realiza entre las clases del diseño y que posibilita comprender mejor el módulo elaborado en términos de implementación.

3.6.1. Diagrama de Secuencia del Caso de Uso Gestionar Luz

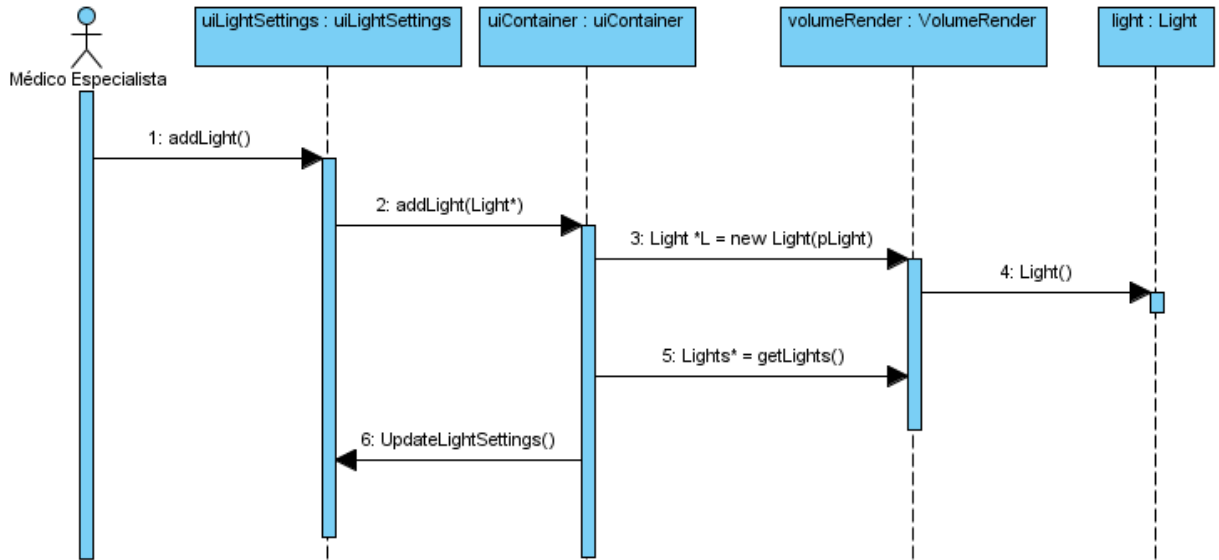


Fig. 37 Diagrama de Secuencia del escenario Adicionar luz.

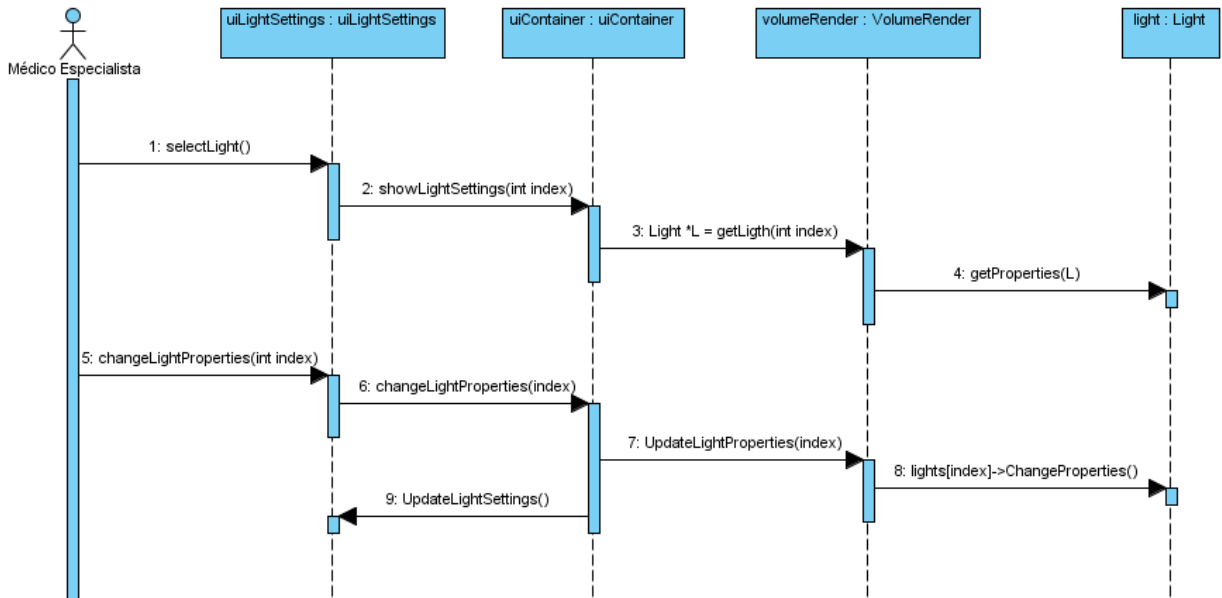


Fig. 38 Diagrama de Secuencia del escenario Modificar propiedades de la luz.

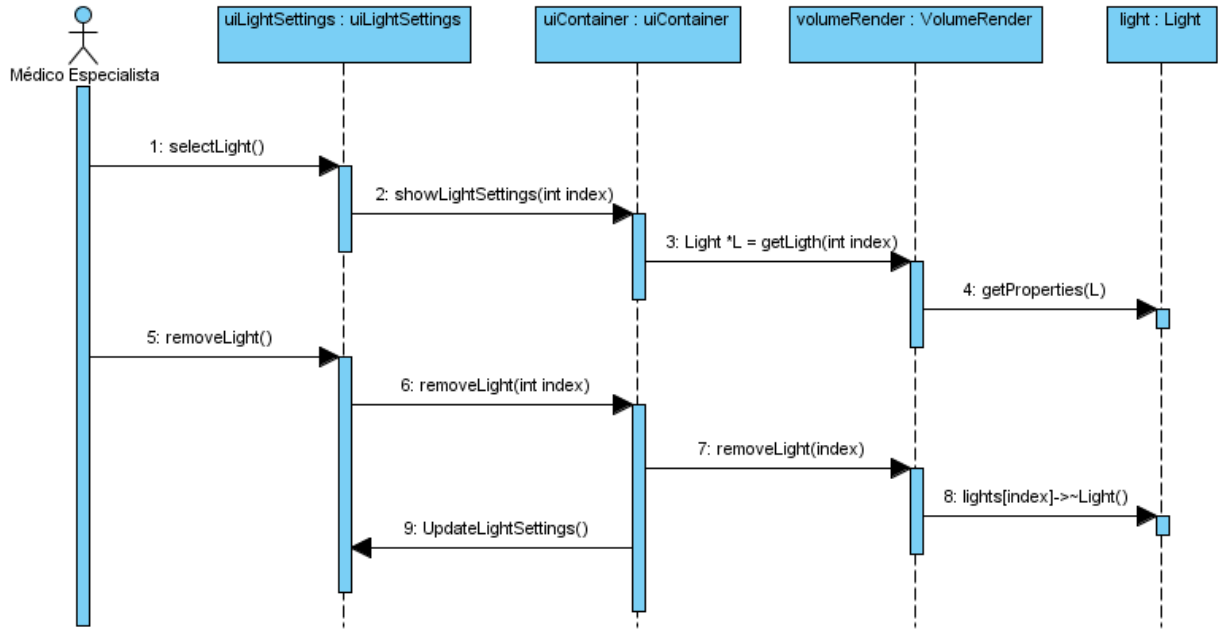


Fig. 39 Diagrama de Secuencia del escenario Eliminar luz.

# CAPITULO 4. IMPLEMENTACIÓN Y VALIDACIÓN DE LOS RESULTADOS.

En este capítulo se abordarán los temas de implementación del módulo utilizando el trabajo realizado en los capítulos anteriores. Se modelará el diagrama de componentes utilizado en el módulo y se harán algunas pruebas para validar los resultados y medir aspectos como el rendimiento y la calidad de las imágenes obtenidas con la implementación del módulo.

## 4.1. Implementación

En esta etapa se realiza la implementación de las clases y objetos en ficheros fuente, binarios y ejecutables. Como resultado se obtiene un sistema ejecutable que incluye todas las funcionalidades propuestas en la captura de requisitos funcionales. La estructura de todos estos modelos forma el modelo de implementación.

### 4.1.1. Diagrama de componentes

Las clases que se obtienen en el diseño se hacen físicas mediante componentes. Los componentes representan módulos de software (código fuente, código binario, ejecutables, DLL) con una interfaz bien definida. Para organizar mejor las dependencias entre los componentes se elabora el diagrama de componentes. El diagrama de componentes incluye fundamentalmente componentes de código fuente y ejecutables. En la Fig. 40 se muestra el diagrama de componentes correspondiente al módulo elaborado.

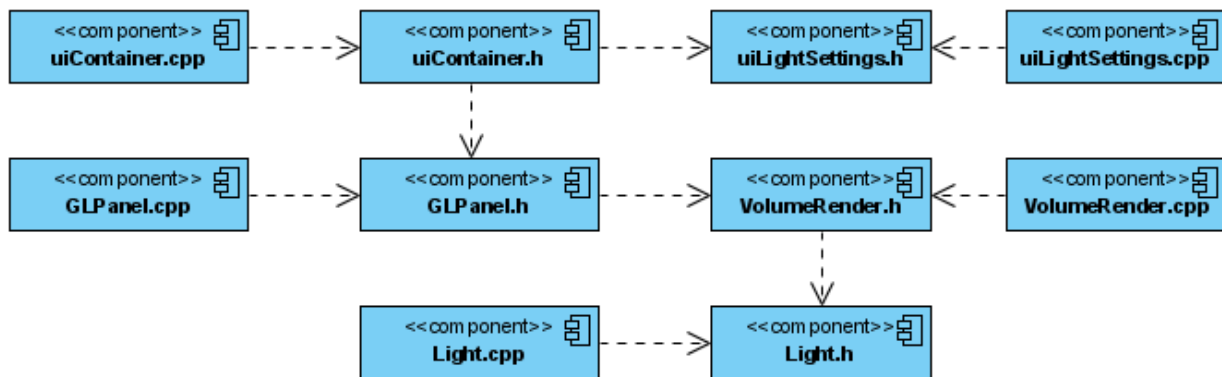


Fig. 40 Diagrama de Componentes.




## 4.2. Validación de los Resultados

La validación de la implementación de la solución propuesta se desarrolló en una computadora con un procesador Intel Core2 Quad Q6600 a una frecuencia de 2.4 GHz, 1 GB de memoria RAM y una tarjeta gráfica NVidia 9800 GT con 512 MB de RAM para video. Mediante la misma se pretendía evaluar la eficiencia y la calidad del módulo implementado y compararlos con los obtenidos por otras investigaciones internacionales.

### 4.2.1. Datos de pruebas


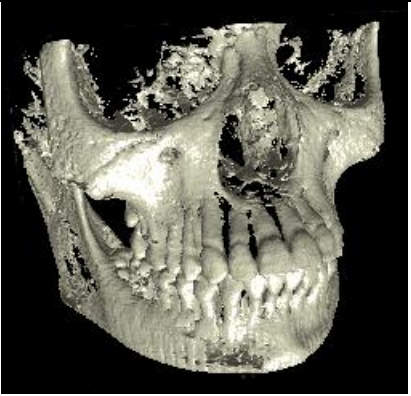
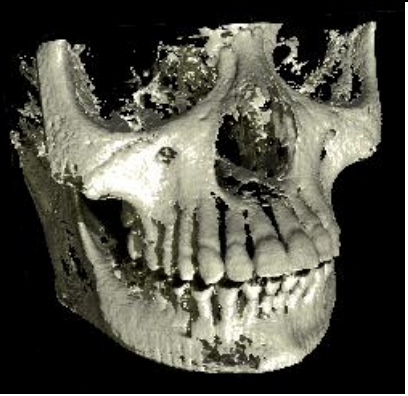
Los datos de pruebas del módulo de iluminación fueron seleccionados teniendo en cuenta dos criterios fundamentales. El primero es la necesidad que contar con un módulo de iluminación para la reconstrucción de modelos anatómicos 3D, por lo que se seleccionaron para algunos casos de pruebas imágenes de las modalidades de adquisición de datos médicos TAC y RM. El segundo fue la necesidad de comparar la robustez de los algoritmos implementados para cualquier fuente de adquisición de datos de naturaleza volumétrica; por lo que se escogen bloques de datos certificados internacionalmente para probar estos algoritmos.

#### 4.2.1.1. Caso de Prueba # 1

Identificador	Modalidad	Dimensión	Visualización sin iluminación	Visualización con iluminación	Visualización con iluminación y sombras
Dataset de Bonal	TAC	512x512x346	53	49	25
			sin iluminación	con iluminación	con iluminación y sombras
					

Como se aprecia en la tabla el cálculo de la iluminación influye muy poco en el rendimiento porque el método utilizado para el cálculo de gradiente es bastante rápido. En el caso de las sombras es diferente. La cantidad de cálculos que se realizan en el segundo rayo para cada muestra del volumen es mucho mayor que al calcular la iluminación. Esto se puede apreciar en una disminución notable del rendimiento al calcular las sombras para los modelos seleccionados.

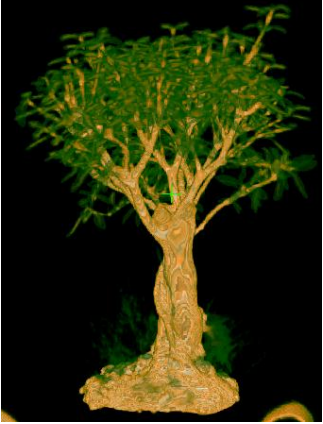
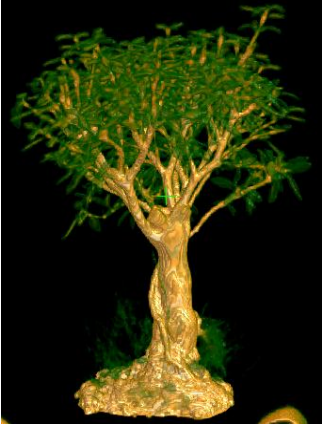
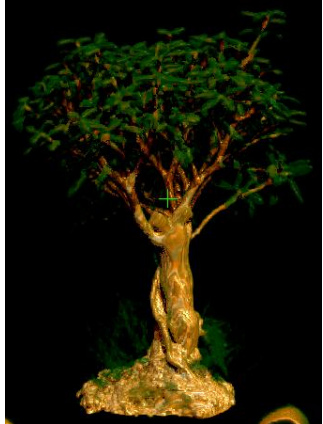
**4.2.1.2. Caso de prueba # 2**

Identificador	Modalidad	Dimensión	Visualización sin iluminación	Visualización con iluminación	Visualización con iluminación y sombras
Dataset del Cráneo	TAC	256x256x256	85	85	33
sin iluminación		con iluminación		con iluminación y sombras	
					


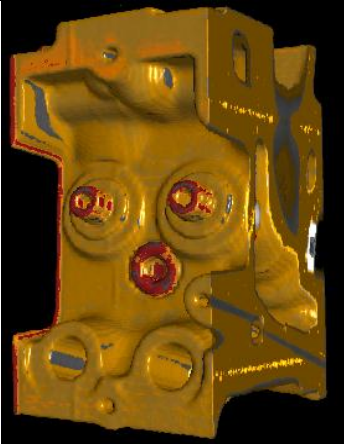

Durante las pruebas se decidió probar el módulo sobre algunos modelos que aunque no son anatómicos si son de los más utilizados a nivel mundial en la visualización de volumen. Estas pruebas demuestran que el módulo es flexible para ser utilizado en la visualización de datos volumétricos de forma general. Al no estar limitado a datos anatómicos es posible utilizarlo en la iluminación de otros modelos volumétricos y en un futuro pudiera pensarse en extender la visualización a otras áreas como por ejemplo, el petróleo. A continuación se muestran los resultados obtenidos sobre estos modelos.



**4.2.1.3. Caso de prueba # 3**

Identificador	Modalidad	Dimensión	Visualización sin iluminación	Visualización con iluminación	Visualización con iluminación y sombras
Dataset del Bonsai	TAC	256x256x256			
sin iluminación		con iluminación		con iluminación y sombras	
					

**4.2.1.4. Caso de prueba # 4**

Identificador	Modalidad	Dimensión	Visualización sin iluminación	Visualización con iluminación	Visualización con iluminación y sombras
Dataset del Engine	TAC	256x256x128			
sin iluminación		con iluminación		con iluminación y sombras	
					

## Conclusiones

Con la realización de este trabajo se elaboró un módulo de iluminación para la visualización de datos volumétricos que permite incorporar fuentes de luces externas a las escenas virtuales, dotando a la aplicación del proyecto de una mayor sensación de profundidad. Se le incorporó la generación de sombras para aumentar la percepción tridimensional de los modelos que se visualizan en el proyecto.

Los modelos de iluminación implementados aumentaron el realismo y la calidad de la visualización de los modelos anatómicos 3D visualizados en el sistema Vismedic. Se comprobó que el modelo de iluminación global rayos de sombras brinda mejores resultados visuales que el modelo de iluminación local de Phong aunque requiere de un mayor costo computacional y su complejidad de implementación es superior. Con la combinación de ambos modelos se obtuvieron los mejores resultados visuales.

Se adaptó el modelo de iluminación de superficies de Phong mediante el cálculo del vector gradiente utilizando la técnica de diferencias centrales. Esto permitió iluminar modelos volumétricos que no se obtienen a partir de la generación de una geometría.

De las técnicas utilizadas para calcular el vector gradiente, las que mejores resultados visuales mostraron fueron las técnicas de diferencias centrales y Sobel. Sin embargo, el método de diferencias centrales mostró resultados visuales aceptables con un menor costo computacional.

## Recomendaciones

Los modelos de iluminación propuestos como solución se pueden mejorar y continuar en el futuro a través de otras investigaciones que pueden buscar otros modelos que no se trataron aquí y que no están dentro de los resultados obtenidos con este trabajo. Es por eso que a continuación se mencionan algunas recomendaciones para los trabajos futuros:

- ✓ Combinar el modelo de iluminación global implementado con técnicas de render a texturas 3D para hacer más eficiente el rendimiento de la aplicación.
- ✓ Implementar el método de mapeo de sombras como una alternativa en la búsqueda de tiempo real para modelos tridimensionales de mayor resolución que los obtenidos hasta el momento.
- ✓ Utilizar otros métodos para estimar gradientes como el método de Sobel para comparar la suavidad de los resultados de la visualización obtenida aunque esto implica un mayor costo computacional.

## Bibliografía

1. Gallardo, Arnold. 3D Lighting: History, Concepts and Techniques. Rockland, Massachusetts : Charles River Media, 2001.
2. Paredes, Laura Rojas. Cámara Oscura. [En línea] [Citado el: 15 de Febrero de 2011.] [http://redescolar.ilce.edu.mx/educontinua/arte/luces\\_de\\_la\\_ciudad/Memorias/fotografia/camaraos.htm](http://redescolar.ilce.edu.mx/educontinua/arte/luces_de_la_ciudad/Memorias/fotografia/camaraos.htm).
3. Paredes, Laura Rojas. La Fotografía. [En línea] [Citado el: 15 de Febrero de 2011.] [http://redescolar.ilce.edu.mx/educontinua/arte/luces\\_de\\_la\\_ciudad/Memorias/fotografia/fotografia.htm](http://redescolar.ilce.edu.mx/educontinua/arte/luces_de_la_ciudad/Memorias/fotografia/fotografia.htm).
4. Dempski, Kelly y Viale, Emmanuel. Advanced Lighting and Materials with Shaders. Texas : Wordware Publishing, 2005. 1-55622-292-0.
5. Levoy, Marc. Display of Surfaces from Volume Data. Carolina del Norte : s.n., 1988.
6. Kaufman, Arie. Voxels as a Computational Representation of Geometry. s.l. : The Computational Representation of Geometry, 1994. Siggraph 94, Course Notes.
7. Max, Nelson. Optical Models for Direct Volume Rendering. California : IEEE Transactions on Visualization and Computer Graphics, 1995. Vol 1, No 2.
8. Preim, Bernhard y Bartz, Dirk. Visualization in Medicine. Theory, Algorithms and Applications. Oxford : Morgan Kaufmann Publishers, 2007. 978-0-12-370596-9.
9. Céspedes, Yirka y Yoander, Cabrera. Métodos Realistas de Iluminación para Juegos 3D. Ciudad de la Habana : s.n., 2007.
10. Engel, Klaus; Hadwiger, Markus; Kniss, Joe y Rezk-Salama, Christof. Real-Time Volume Graphics. s.l. : Evento de Eurographics, 2006.
11. Phong, Bui Tuong. Illumination for Computer Generated Pictures. Universidad de Utah : Editor W. Newman, Graphics and Image Processing, 1975.
12. Vidal Valenzuela, Javier. Taller de Computación Gráfica. Capítulo IV. Iluminación y Sombreado. [En línea] [Citado el: 13 de Octubre de 2010.] [http://asignaturas.inf.udec.cl/cgraf/public\\_html/iluminacion.pdf](http://asignaturas.inf.udec.cl/cgraf/public_html/iluminacion.pdf).
13. Iglesias, Andrés. COMPUTER-AIDED GEOMETRIC DESIGN AND COMPUTER GRAPHICS: ILLUMINATION MODELS. [En línea] 2001. [Citado el: 14 de Febrero de 2011.] <http://personales.unican.es/iglesias/inginf/Illumination.pdf>.
14. Weitzenfeld, Alfredo. Iluminación y Sombreado. [En línea] 2001. [Citado el: 15 de Febrero de 2011.] <http://cannes.itam.mx/Alfredo/Espaniol/Cursos/Grafica/Sombreado.pdf>.
15. Oren, Michal y Nayar, Shree K. Generalization of the Lambertian Model and Implications for Machine Vision. Nueva York : International Journal of Computer Vision, 1994. Vol 14:3.
16. Hasenfratz, J; Lapiere, M; Holzschuch, N y Sillion, F. A Survey of Real-Time Soft Shadows Algorithms. Oxford : Eurographics Association and Blackwell Publishing Ltd, 2003.

17. Da Cruz, Daniela. Shadow maps: A Survey. Universidade do Minho, CCTC, Braga, Portugal : s.n., 2008.
18. Kim, Tae-Yong y Neumann, Ulrich. Opacity Shadow Maps. California : Eurographics Rendering Workshop, 2001.
19. Lokovic, Tom y Veach, Eric. Deep shadow maps. Nueva York, Estados Unidos : ACM Press/Addison-Wesley Publishing Co, 2000. Evento SIGGRAPH'00: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques.
20. Hadwiger, Markus; Kratz, Andrea; Sigg, Christian y Bühler, Katja. GPU-Accelerated Deep Shadow Maps for Direct Volume Rendering. Nueva York : In Graphics Hardware '06: Proceedings of the 21st ACM SIGGRAPH/Eu-ACM Press, 2006.
21. Hadwiger, Markus; Ljung, Patric; Rezk Salama, Christof y Ropinski, Timo. Raycasting, Advanced Illumination Techniques for GPU-Based Volume. s.l. : Evento ACM SIGGRAPH, 2009.
22. GLSL Tutorial. [En línea] [Citado el: 15 de Febrero de 2011.] <http://www.lighthouse3d.com/opengl/glsl>.
23. Flores, Miriam Milagro Díaz. RUP vs XP. [En línea] [Citado el: 4 de Febrero de 2011.] [http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP vs. XP.pdf](http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf).
24. Electrónica, Laboratorio III de. Proceso Unificado para Desarrollo de Software. [En línea] [Citado el: 15 de Febrero de 2011.] <http://www.geocities.ws/gustsucc/Archivos/AnotacionesRUP.pdf>.
25. Ayuda del Rational Unified Process.

# Glosario de Términos

## D

**Diagnóstico:** Etimológicamente el concepto diagnóstico proviene del griego, tiene dos raíces, día- que es a través de, por. y gignoskein que es conocer, así etimológicamente diagnóstico significa conocer a través de. El concepto de este significado (imagen que representamos en la mente) es la identificación de la naturaleza o esencia de una situación o problema y de la causa posible o probable del mismo, es el análisis de la naturaleza de algo.

**DICOM:** En Inglés (Digital Imaging and Communication in Medicine) es el estándar reconocido mundialmente para el intercambio de imágenes médicas, pensado para el manejo, almacenamiento, impresión y transmisión de imágenes médicas. Incluye la definición de un formato de fichero y de un protocolo de comunicación de red. El protocolo de comunicación es un protocolo de aplicación que usa TCP/IP para la comunicación entre sistemas. Los ficheros DICOM pueden intercambiarse entre dos entidades que tengan capacidad de recibir imágenes y datos de pacientes en formato DICOM.

## F

**Fragment Shader:** Programa que calcula el color de los píxeles individuales. Controla cómo las texturas son aplicadas a los fragmentos.

**Frame buffer:** Memoria usada para retener uno o más frames para su posterior uso.

**Fotón:** Cada una de las partículas que constituyen la luz y, en general, la radiación electromagnética en aquellos fenómenos en que se manifiesta su naturaleza corpuscular.

## G

**GPU:** Unidad de procesamiento gráfico.

**Gradiente:** Denota una dirección en el espacio según la cual se aprecia una variación de una determinada propiedad o magnitud física.

## H

**Hardware:** Componentes físicos de una computadora o de una red (a diferencia de los programas o elementos lógicos que los hacen funcionar).

**I**

**Imagen:** Figura, representación, semejanza y apariencia de algo. En computación es formada por la unión de MxN píxeles (imagen 2D) o vóxeles (imagen 3D).

**Imagenología:** Comprende la realización de todo tipo de exámenes diagnósticos y terapéuticos en los cuales se utilizan equipos que reproducen imágenes del organismo. Los siete servicios de Imagenología son Ecotomografía, Imagenología Mamaria, Medicina Nuclear, Radiología, Rayos Infantil, Resonancia Magnética y Tomografía Computada o Scanner.

**M**

**Módulo:** Pieza o conjunto unitario de piezas que se repiten en una construcción de cualquier tipo, para hacerla más fácil, regular y económica.

**N**

**Normal:** Vector tridimensional, perpendicular a la cara de un objeto, determinando la dirección en que ella apunta.

**O**

**Oclusión:** Desaparición total o parcial de objetos en una escena.

**P**

**Pipeline:** División de un proceso en etapas, asignando a cada una de ellas distintos recursos.

**Píxel:** Abreviatura de "picture element". Es la mínima unidad de información dentro de una imagen bidimensional.

**S**

**Shader:** Conjunto de instrucciones capaces de ser ejecutadas por un procesador gráfico.

**Sombra:** Área en la cual existe una ausencia parcial o total de iluminación como resultado de la obstrucción entre la fuente de luz y el área que está siendo iluminada.

**T**

**Tarjeta gráfica:** Es una tarjeta de circuito impreso encargada de transformar las señales eléctricas que llegan desde el microprocesador en información comprensible y representable por la pantalla del ordenador.

**Textura:** Imagen que pueden ser mapeada en cualquiera de las primitivas gráficas para adicionar detalles a la escena.

## V

**Vertex shader:** Función del procesador gráfico que manipula los valores de un vértice en un plano 3D mediante operaciones matemáticas sobre un objeto. Estas variaciones pueden ser diferencias en el color, en las coordenadas de la textura, en la orientación en el espacio o en el tamaño del punto. Permite el control de las transformaciones sobre los vértices.

**Vértices:** Son puntos en el espacio 3D que definen primitivas gráficas tales como triángulos, polígonos y rectángulos, usados para construir la geometría de la escena que será dibujada.

**Vóxel:** Palabra que proviene de la contracción del término en inglés "volumetric píxel". Es la unidad cúbica que compone un objeto tridimensional. Constituye la unidad mínima procesable de una matriz tridimensional y es, por tanto, el equivalente del píxel (o píxel) en un objeto 2D.

**Viewport:** Rectángulo definido en el sistema de referencia de la pantalla, cuyo objetivo es seleccionar que área del mundo se desea ver en un sub-área de la pantalla.