

Estructura y funcionamiento de un algoritmo de renderizado: Raytracing

Structure and how works of a rendering algorithm: Raytracing.

Luis Fernando Moncada Idárraga, Ramiro Andrés Barrios Valencia, Julián Montes Franco
Ingeniería de Sistemas y Computación, Universidad Tecnológica de Pereira
Pereira, Colombia

lfmoncada@utp.edu.com
ramiro@sirius.utp.edu.co
julianmontesf@sirius.utp.edu.co

Resumen— En éste artículo se hará una descripción del funcionamiento y la base matemática detrás de el algoritmo de renderizado llamado Raytracing o Trazador de rayos, además las ventajas y desventajas que presenta con respecto a otros algoritmos y cuáles métodos son los más utilizados para reducir éstas desventajas.

Palabras clave— algoritmos de renderizado, computación gráfica, estructura y funcionamiento, trazador de rayos.

Abstract— in this article will be described how works and the mathematical base of the rendering algorithm called Raytracing, as well as the advantages and disadvantages it has respect to other rendering algorithms and which methods are the most used to reduce this disadvantages.

Key Word— computer graphics, structure and how works, ray tracing, rendering algorithms.

I. INTRODUCCIÓN

En la actualidad los gráficos generados por computador se utilizan en una amplia variedad de campos de la sociedad, tanto en el ámbito profesional y académico como en el entretenimiento. Todo esto es gracias a una técnica conocida como renderizado, la cual consiste en generar una imagen a partir de un modelo de datos abstracto mediante un algoritmo que realiza todo el proceso.

Existe una variedad de algoritmos de renderizado que se diferencian tanto en complejidad como en la calidad de la imagen resultante y entre estos se encuentra el algoritmo conocido como *Raytracing* el cual presenta muy buenos resultados y cuya base conceptual es relativamente sencilla.

En este artículo se describirá cada uno de los conceptos mediante los cuales trabaja este algoritmo y posteriormente analizar algunas ventajas y desventajas que presenta con

respecto a otros algoritmos de renderizado y mencionando los métodos para superar dichas desventajas.

II. CONCEPTOS PRELIMINARES

Para comprender de forma precisa el funcionamiento del algoritmo es necesario tener claros los siguientes conceptos ya que son la base con la que trabaja este algoritmo. Estos conceptos son los siguientes:

1) Raytracing. Antes que nada hay que saber en qué consiste el algoritmo. El algoritmo de Raytracing consiste en simular de forma simplificada el comportamiento de los rayos de luz que inciden sobre un objeto y que son vistos por un observador. En la práctica el algoritmo traza los rayos desde el observador procesando las posibles intersecciones de ese rayo con los objetos y trazándose nuevamente otro rayo si es necesario para efectos de reflexión o refracción.

2) Escena. Una escena es una estructura de datos que contiene la descripción geométrica de todos los objetos que se encuentran dentro de la imagen y una cámara que representa el observador. Normalmente en una implementación la escena suele guardarse en un archivo para la persistencia de los datos.

3) Rayo. Consiste en un vector situado en un punto del espacio el cual se utiliza para simular un rayo de luz que parte desde el observador y atraviesa toda la escena.

4) Matriz. Para el caso del algoritmo, tenemos 3 tipos de matrices asociadas a las 3 transformaciones básicas. Una transformación es una forma de aplicar un cambio de coordenadas a un punto.

5) Observador. Es la cámara que captura la imagen deseada desde la escena, del cual dependen ciertos aspectos de visualización en función algunas de sus propiedades.

6) Plano de proyección. Es el plano en el cual se plasma la escena que se está renderizando. Es una proyección de la escena en tres dimensiones al plano en dos dimensiones de la pantalla.

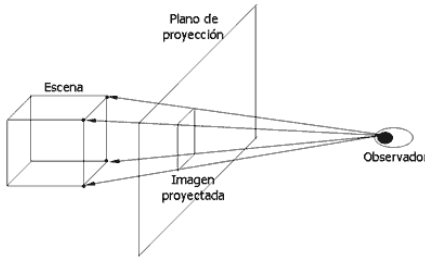


Figura 1. Plano de proyección.

III. TRANSFORMACIONES BÁSICAS

A menudo es necesario aplicar cambios a la posición, al tamaño o a la orientación de un objeto y se requiere hacerlo indicando los parámetros exactos que se quieren cambiar, a esto se le denomina transformaciones y son la traslación, el escalado y la rotación las más básicas y que se utilizan en todo momento mediante el uso de matrices. Al aplicar transformaciones a un punto se utiliza una matriz de una dimensión superior, es decir, para aplicar una transformación a un punto en 2 dimensiones se utiliza una matriz de 3x3 y para un punto de 3 dimensiones se utiliza una matriz 4x4, al efectuar las operaciones necesarias simplemente se descarta la dimensión sobrante. A continuación detallamos cada una de las transformaciones:

1) Traslación. Ésta transformación, como su nombre lo indica, consiste en mover un punto una distancia determinada. El vector T indica la dirección de la traslación y su magnitud la distancia deseada.

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+t_x \\ y+t_y \\ z+t_z \\ 1 \end{bmatrix}$$

Traslación 3D

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+t_x \\ y+t_y \\ 1 \end{bmatrix}$$

Traslación 2D

Figura 2. Matriz de traslación.

Esto se logra ya que al realizar esta operación entre las matrices se suman los componentes de T en cada eje se suman a las coordenadas del punto obteniendo así la traslación.

2) Escalado. Se utiliza para agrandar o encoger un objeto en un factor S determinado aplicándola a cada uno de sus puntos.

$$\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x*S_x \\ y*S_y \\ z*S_z \\ 1 \end{bmatrix}$$

Escalado 3D

$$\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x*S_x \\ y*S_y \\ 1 \end{bmatrix}$$

Escalado 2D

Figura 3. Matriz de escalado.

En este caso el resultado se logra de forma similar a la traslación, al aplicar el escalado a cada punto sus coordenadas se multiplican por el factor de escalado deseado logrando así una figura más grande o más pequeña dependiendo de dicho factor.

3) Rotación. La matriz de rotación opera aplicando un desplazamiento a un punto con respecto a un ángulo β determinado. Para las rotaciones en 2 dimensiones no hace falta especificar el eje puesto que sólo hay un eje de rotación, pero para el caso de 3 dimensiones sí es necesario especificarlo y se hace de la siguiente manera:

$$\begin{bmatrix} \cos\beta & -\sin\beta & 0 & 0 \\ \sin\beta & \cos\beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotación respecto Z

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta & 0 \\ 0 & \sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotación respecto X

$$\begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotación respecto Y

Figura 4. Matriz de rotación.

Para el caso de la rotación de una figura sin que se desee cambiar su posición es necesario hacer una traslación de ésta al origen (0, 0, 0), aplicar la rotación y posteriormente trasladarla a la posición anterior, de lo contrario la figura simplemente se desplazará alrededor del eje especificado en lugar de rotar sobre sí misma.

Cabe notar que al trabajar con matrices es posible aplicar varias transformaciones consecutivas concatenándolas en una multiplicación, en otras palabras, si se desea aplicar una traslación T1, luego una rotación R1 y posteriormente otra traslación T2, se tendrá lo siguiente:

$$M = T1 * R1 * T2$$

Siendo M la matriz de la transformación resultante a aplicar al punto deseado.

IV. PROPIEDADES DEL OBSERVADOR

Éste es el componente más importante del Trazador de Rayos porque de éste depende el resultado de la imagen final ya que se encarga de varios aspectos que influyen en la visualización de la imagen. Estos aspectos son:

- Tipo de proyección: indica de qué forma la cámara visualiza la escena. Existen dos tipos:
 1. Perspectiva: en este tipo de proyección el observador se mantiene fijo en un punto y a partir de éste se trazan los rayos hacia la escena por lo tanto cada rayo será diferente durante el proceso del algoritmo.
 2. Paralela: este tipo de proyección consiste en trazar los rayos paralelos desde un observador situado en un punto muy lejano (infinito) hacia el plano de proyección. En este caso el rayo que se traza siempre será el mismo mientras que el observador cambiará de posición constantemente.

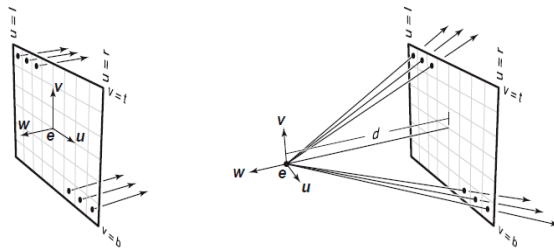


Figura 5. Proyección en paralelo (izquierda) y proyección en perspectiva (derecha).

- Vector ViewUP (VUP): este vector determina cuál es la dirección hacia la que el observador considera ‘arriba’.
- Near, Far: indican la distancia mínima y la distancia máxima de renderizado respectivamente, es decir, todos aquellos objetos de la escena que se encuentren a una distancia del observador menor que la indicada por ‘Near’ y/o mayor que la indicada por ‘Far’ no serán renderizados en la imagen resultante.

V. EL ALGORITMO

Luego de observar los anteriores aspectos los cuales son de suma importancia para la estructura del algoritmo vamos a realizar una descripción más detallada de la forma como trabaja el Raytracing. En primer lugar observemos una implementación en pseudo-código para conocer la estructura del algoritmo.

```

for each pixel do
  compute viewing ray
  if (ray hits an object with  $t \in [0, \infty)$ ) then
    Compute  $\mathbf{n}$ 
    Evaluate shading model and set pixel to that color
  else
    set pixel color to background color
  
```

Esta implementación es bastante sencilla y simplificada pero ilustra claramente el funcionamiento general del algoritmo. Ahora, explicamos con más detalles el procedimiento.

Para cada pixel de la imagen se calcula un rayo que parte del observador y pasa por el punto que corresponde a ese pixel. Si el rayo se interseca con algún objeto, se calcula el vector \mathbf{n} normal a la superficie del objeto y se evalúa el modelo de iluminación para dar el color al pixel. En caso de que el rayo no se interseca con ningún objeto, se le asigna el color de fondo al pixel actual.

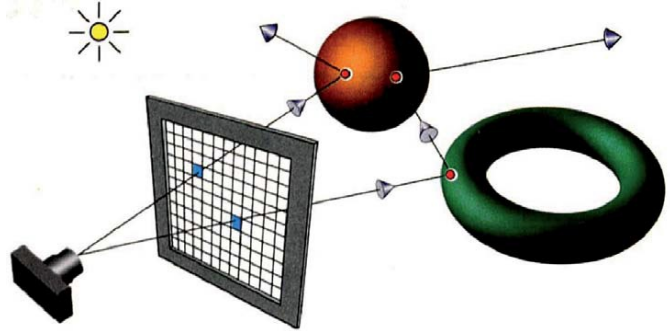


Figura 6. Diagrama que ilustra el funcionamiento de un Raytracer.

La característica principal de éste algoritmo es que es un algoritmo recursivo, es decir, en el momento en que el rayo se interseca con un objeto dependiendo de las propiedades materiales de este se puede producir reflexión o refracción lo que ocasiona que se lance otro rayo de forma recursiva para determinar esos efectos. En la práctica se establece un número máximo de nivel de recursión para evitar ciclos infinitos o consumir demasiados recursos de cómputo sin ser necesario.

VI. CÁLCULO DE INTERSECCIONES

El cálculo de intersecciones es un proceso muy importante en el desarrollo de este algoritmo puesto que toda la mecánica de decisión y de aplicación de los modelos de iluminación se basa en las intersecciones. En este capítulo vamos a detallar de una manera más formal el cálculo de las intersecciones para dos de las figuras más comunes a modo de ilustración, ya que en general el cálculo de intersecciones con figuras más complejas requiere de una profundización en algunos conceptos matemáticos relacionados con este propósito.

En primer lugar vamos a definir una representación matemática de lo que es un rayo. Un rayo es simplemente un punto de origen y una dirección de propagación, por lo tanto se puede representar como una línea 3D paramétrica que parte desde el observador hacia el plano de la imagen.

$$\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{V}$$

Donde \mathbf{p}_0 es la posición del observador, \mathbf{V} es el vector que va desde \mathbf{p}_0 hacia el pixel correspondiente en la imagen e indica la dirección del rayo y t es el parámetro de la ecuación, por lo que dado un valor de t podremos obtener un punto $\mathbf{p}(t)$. La forma como se define la dirección del rayo depende del tipo de proyección del observador.

Ahora pasamos a calcular la intersección de éste rayo con dos de las figuras más comunes encontradas en una escena: la esfera y el polígono.

1) Intersección Rayo-Esfera. Dada la ecuación que representa el rayo y dada una esfera con centro en el punto $c = (x_c, y_c, z_c)$ y radio R representada por la ecuación

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 - R^2 = 0.$$

Podemos escribir la ecuación de la esfera en forma vectorial:

$$(\mathbf{p} - \mathbf{c}) \cdot (\mathbf{p} - \mathbf{c}) - R^2 = 0.$$

Donde cualquier punto \mathbf{p} que satisfaga la ecuación pertenece a la superficie de la esfera. Ahora si reemplazamos la ecuación del rayo para un punto en esta ecuación, obtenemos una ecuación en términos de t la cual se satisface por los valores de t que indican puntos pertenecientes a la esfera:

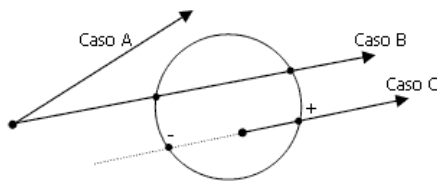
$$(\mathbf{p}_0 + t\mathbf{V} - \mathbf{c}) \cdot (\mathbf{p}_0 + t\mathbf{V} - \mathbf{c}) - R^2 = 0.$$

$$(\mathbf{V} \cdot \mathbf{V})t^2 + 2\mathbf{V} \cdot (\mathbf{p}_0 - \mathbf{c})t + (\mathbf{p}_0 - \mathbf{c}) \cdot (\mathbf{p}_0 - \mathbf{c}) - R^2 = 0.$$

La cual tiene la forma de una ecuación cuadrática clásica en la que la variable t es la única desconocida y su solución se puede hallar de la siguiente forma:

$$t = \frac{-\mathbf{V} \cdot (\mathbf{p}_0 - \mathbf{c}) \pm \sqrt{(\mathbf{V} \cdot (\mathbf{p}_0 - \mathbf{c}))^2 - (\mathbf{V} \cdot \mathbf{V})((\mathbf{p}_0 - \mathbf{c}) \cdot (\mathbf{p}_0 - \mathbf{c}) - R^2)}}{(\mathbf{V} \cdot \mathbf{V})}$$

Esto indica que la ecuación puede o no tener solución y en caso de no tenerla significa que el rayo no interseca con la esfera. En caso de tener solución la correcta es la menor no negativa, ya que los valores negativos para t expresan puntos que se encuentran detrás del observador.



Caso A: El rayo no intersecciona la esfera, la ecuación no tiene solución ($B^2 - 4AC < 0$).
Caso B: Hay dos soluciones positivas, nos quedamos la menor, la más cercana al origen del rayo.
Caso C: El origen del rayo está en el interior de la esfera, hay dos soluciones, una negativa y otra positiva. La correcta es la positiva.

Figura 7. Posibles casos en intersección rayo-esfera.

2) Intersección Rayo-Polígono. El polígono es una de las figuras más utilizadas en la descripción de escenas 3D. El cálculo de su intersección es algo más complejo que la de la

esfera pero no tanto como lo es la de otras figuras. Partimos de que un polígono es una figura plana con n vértices, \mathbf{p}_1 hasta \mathbf{p}_n , por lo tanto existe un plano que lo contiene y puede definirse con una ecuación vectorial, esto es:

$$(\mathbf{p} - \mathbf{p}_1) \cdot \mathbf{n} = 0.$$

Donde \mathbf{n} es el vector normal al plano, \mathbf{p} es un punto dentro del polígono y \mathbf{p}_1 uno de sus vértices. En ésta ecuación podemos reemplazar la ecuación del plano, $\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{V}$, de la siguiente forma:

$$(\mathbf{p}_0 + t\mathbf{V} - \mathbf{p}_1) \cdot \mathbf{n} = 0.$$

Despejando t de la ecuación, tenemos:

$$t = \frac{(\mathbf{p}_1 - \mathbf{p}_0) \cdot \mathbf{n}}{\mathbf{d} \cdot \mathbf{n}}$$

Lo cual nos permite calcular el punto \mathbf{p} . Si el punto \mathbf{p} está dentro del polígono entonces hay intersección con el rayo. La pregunta está en ¿Cómo saber si el punto pertenece al polígono? Existen varios métodos para encontrar esa respuesta, pero uno de los más sencillos sin mucho costo computacional consiste en proyectar el polígono y el punto \mathbf{p} hacia alguno de los planos xy , yz ó xz para el cuál se maximice el área obtenida en la proyección. Luego se traza un rayo 2D desde el punto paralelo a uno de los ejes y si el número de intersecciones con el perímetro del polígono es impar el punto está dentro del éste de lo contrario si es par, el punto se encuentra fuera del polígono.

VII. MODELOS DE ILUMINACIÓN

Una vez que se ha determinado la superficie visible para el pixel el color que se le asigna a este es calculado en base al modelo de iluminación utilizado. A continuación describiremos dos de los modelos de iluminación más básicos utilizados actualmente. El objetivo de un modelo de iluminación es el de capturar el proceso de reflexión de la luz, mediante el cual una superficie es iluminada por fuentes de luz y refleja parte de la luz hacia la cámara.

Las variables que interactúan en un modelo de iluminación son la dirección de la luz \mathbf{l} , que es un vector unitario apuntando hacia la fuente de luz; la dirección de observación \mathbf{v} , que es un vector unitario que apunta al observador; la normal a la superficie \mathbf{n} , que es un vector unitario perpendicular a la superficie en el punto donde tiene lugar la reflexión; y por último la propiedades de la superficie como el color, el brillo y otras propiedades dependiendo del modelo a aplicar. Veamos entonces los dos modelos de iluminación básicos:

1) Iluminación Lambert. Éste modelo de iluminación consiste en que la cantidad de iluminación en un punto de una superficie depende del ángulo de incidencia de la luz y es proporcional al coseno del ángulo θ formado entre la normal de la superficie y la fuente de luz, así:

$$L = k_d I \max(0, \mathbf{n} \cdot \mathbf{l})$$

Donde L será el valor de color para el pixel; k_d es la *constante de reflexión difusa*; e I es la intensidad de la fuente de luz. Dado que tanto \mathbf{n} como \mathbf{l} son vectores unitarios se puede utilizar de forma conveniente la notación $\mathbf{n} \cdot \mathbf{l}$ como $\cos \theta$. Si el resultado de esta operación da como resultado valores negativos se tomará como resultado el valor de 0. Este modelo se aplica a cada canal de color del pixel separadamente, teniendo en cuenta hacer los cálculos con el componente apropiado del color difuso y de la intensidad de la fuente de luz.

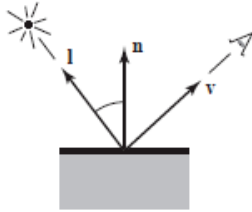


Figura 8. Geometría del modelo de iluminación Lambert.

2) Iluminación Blinn-Phong. El modelo de iluminación anterior es independiente del punto de observación, es decir, sin importar desde donde se observe no hay variación en el resultado. En la realidad hay objetos que presentan cierto grado de brillo dependiendo desde dónde se observe, produciendo destellos o *reflexiones especulares*, que parecen moverse a medida que se mueve el observador. Para esto el modelo Blinn-Phong produce reflexión en su punto más brillante cuando \mathbf{v} y \mathbf{l} están posicionados simétricamente respecto a la normal de la superficie, que es el punto en que se consigue el efecto espejo, y a medida que los vectores se alejan de esta posición la reflexión disminuye.

Para saber qué tan alejados están los vectores de la posición de espejo comparamos el vector bisector \mathbf{h} , también unitario, con la normal de la superficie. Mientras más cerca esté \mathbf{h} el componente especular se debe ser brillante y volverse opaco a medida que se alejan.

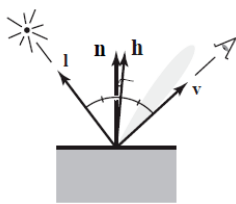


Figura 9. Geometría del modelo de iluminación Blinn-Phong.

Éste resultado se logra calculando el producto punto entre \mathbf{h} y \mathbf{n} , cuyo máximo valor se alcanza cuando éstos son iguales, luego elevando este resultado a una potencia $p > 1$ para que decrezca más rápido. El valor de p o *Exponente de Phong*,

controla qué tan brillante es la superficie. Tenemos entonces la ecuación del modelo:

$$L = k_d I \max(0, \mathbf{n} \cdot \mathbf{l}) + k_s I \max(0, \mathbf{n} \cdot \mathbf{h})^p$$

Donde k_s es la *constante de reflexión especular*, o el color especular de la superficie. El vector bisector \mathbf{h} se puede obtener fácilmente dado que ambos vectores son de igual magnitud la suma de ambos es un vector bisector del ángulo y sólo hace falta normalizarlo de la siguiente manera:

$$\mathbf{h} = \frac{\mathbf{v} + \mathbf{l}}{\|\mathbf{v} + \mathbf{l}\|}$$

Como se ve en la ecuación de éste modelo, la iluminación Lambert se encarga del cálculo de la parte difusa y la iluminación Blinn-Phong de la parte especular.

VIII. OTROS ALGORITMOS: VENTAJAS Y DESVENTAJAS

Existen otros algoritmos que mediante los cuales se hace el renderizado de imágenes y cada uno tiene características y procesos diferentes dentro de las cuales se pueden resaltar tres factores principales que los diferencian como los son: el resultado obtenido en las imágenes finales, el costo computacional que requiere y la complejidad de su modelamiento e implementación. Mencionaremos dos algoritmos aparte del Raytracing dando una breve descripción y en qué grado manejan estos tres factores principales, estos son:

1) Raycasting. En realidad de este algoritmo es de donde se desprende el Raytracing, siendo la diferencia principal el hecho de que el algoritmo Raycasting sólo se encarga de determinar las superficies visibles y aplicar el modelo de iluminación correspondiente para aplicar el color al pixel pero no se trata de un algoritmo recursivo con el que se pueden conseguir efectos como reflexión y refracción. Sin embargo esto implica que el costo computacional es relativamente bajo.

2) Radiosidad. Éste algoritmo ya trabaja de forma muy diferente. En principio, subdivide la escena en una serie de sub-fragmentos para posteriormente mediante ecuaciones lineales se calcula cómo la luz viaja desde una fuente a través de estos fragmentos. El modelamiento de este método es algo más complejo ya que requiere ciertas técnicas para lograr un correcto cálculo de la transmisión de la luz. Sin embargo no posee efectos como reflejos especulares y la refracción. Muchas veces para lograr resultados más realistas se suelen combinar la técnica de Radiosidad con la de Raytracing, ya que la primera presenta algunas mejoras en la iluminación y el sombreado.

Después de lo mencionado se pueden identificar las siguientes ventajas y desventajas que presenta el algoritmo de Raytracing:

- Ventajas:
 - El modelo de su funcionamiento es relativamente sencillo y facilita su implementación.
 - Durante el proceso normal se obtienen naturalmente efectos como la reflexión y la refracción con muy buenos resultados en relación a otras técnicas.
 - Es posible combinar este algoritmo con otros para mejorar algunas de sus falencias.
 - Es un algoritmo altamente paralelizable.
- Desventajas:
 - El costo computacional es relativamente alto debido a la cantidad de cálculos realizados para las intersecciones.
 - Por sí solo no consigue efectos demasiado realistas en las imágenes resultantes.
 - Si existen objetos muy pequeños es posible que no puedan ser alcanzados por ningún rayo durante el proceso.

- [2] P. Shirley, S. Marschner. *Fundamentals of Computer Graphics, Third Edition*. Taylor & Francis Ltd., 2009.
- [3] K. Suffern. *Raytracing from the Ground Up*. A K Peters, 2007.
- [4] J. A. Nin. *Métodos para testear intersecciones*. Facultad de ingenierías, Universidad de la república. Enlace: <<http://www.fing.edu.uy/inco/cursos/cga/Clases/2011/Metodosparatestearintersecciones>
<http://www.fing.edu.uy/inco/cursos/cga/Clases/2011/Metodosparatestearintersecciones.pdf.pdf>>
- [5] J. Foley, A. Van Dama, S. Feiner, J. Hughes, R. Phillips. *Computer Graphics: Principles and Practice*. Massachusetts: Wesley Publishing Company, 1996.
- [6] P. Gonzáles, P. Diego, D. Lusquiños, D. Marín, S. Regueiro. *Técnicas de iluminación*. Universidad de Coruña, España. Enlace: <<http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/modelosIlumionacion/index.html>>

Actualmente para compensar el consumo de recursos de cómputo necesarios para la ejecución del algoritmo se aprovecha una de sus ventajas como lo es la facilidad para paralelizar el proceso, obteniendo así una forma de mejorar el rendimiento de la ejecución de este algoritmo. Para otras de sus desventajas es posible combinar otras técnicas en conjunto con el fin de lograr los mejores resultados posibles.

I. CONCLUSIONES

- El proceso para determinar el color de cada pixel en este algoritmo consiste en determinar primero cuál es la superficie visible y luego aplicar un modelo de iluminación en base a las propiedades del material.
- La recursividad para producir efectos de visualización como la reflexión y la refracción es la principal característica del Raytracing.
- El modelamiento del proceso de este algoritmo es relativamente sencillo y presenta facilidades para la implementación.
- Existen otras técnicas utilizadas para el renderizado de imágenes diferentes a la de Raytracing.
- El costo computacional para la ejecución de este algoritmo es elevado debido a sus cálculos, pero es posible utilizar técnicas para mejorar el rendimiento.
- Aunque el algoritmo no presenta los mejores resultados para algunos efectos, presenta facilidades para su utilización en conjunto con otras técnicas para mejorar el resultado.

REFERENCIAS

- [1] P. Shirley, R. K. Morley. *Realistic Ray Tracing, Second Edition*. A. K. Peters, Ltd., 2003.