

Implementación del algoritmo de path tracing en la GPU

Joaquim Romo

Treball Final de Grau / 2014

DIRECTOR
Javier Agenjo Departamento GTI



Resumen

El presente trabajo tiene como objetivo el estudio y la implementación de un algoritmo de renderizado con raytracing estocástico en la unidad de procesamiento de gráficos (GPU). Se ha elegido realizar la implementación en una arquitectura de este tipo debido a las ventajas que ofrece en cuanto a tiempo de ejecución, gracias a la gran capacidad de cómputo en paralelo que ofrecen las arquitecturas de GPU actuales.

La primera parte del trabajo se dedica al estudio teórico del algoritmo de pathtracing, se comentan algunos conceptos físicos básicos relacionados con el transporte de luz y su interacción con los materiales así como las ecuaciones matemáticas y las técnicas estadísticas necesarias para la comprensión y correcta implementación del algoritmo.

En una segunda parte se discuten las tecnologías involucradas, concretamente el uso que se hace de la arquitectura CUDA, la librería OptiX y su funcionamiento y la implementación del algoritmo que se ha realizado sobre estas.

Índice

Índice de figuras	VII
Índice de tablas	IX
1. INTRODUCCIÓN	1
1.1. Contexto	1
1.2. Algoritmos de iluminación global	3
1.2.1. Radiosity	3
2. FUNDAMENTOS TEÓRICOS	5
2.1. Unidades Radiométricas	5
2.1.1. Flujo	5
2.1.2. Irradiancia	5
2.1.3. Angulo solido	6
2.1.4. Radiancia	7
2.2. BRDF	8
2.2.1. Propiedades de la BRDF	9
2.3. Ecuación de renderizado	10
2.4. El metodo de montecarlo	11
2.4.1. Muestreo de importancia	11
3. FRAMEWORK	13
3.1. Cuda	13
3.2. La libreria OptiX	13
3.2.1. Host	13
3.2.2. Device	13
Bibliography	15

Índice de figuras

2.1. Definición de angulo solido [Haade, 2007]	6
2.2. BRDF $l = \omega_i, v = \omega_o$	8

Índice de cuadros

Capítulo 1

INTRODUCCIÓN

1.1. Contexto

En el entorno de la imagen generada por computador siempre ha sido un reto tratar de generar imágenes lo más realistas posibles. Para ello un gran número de investigadores se han dedicado a diseñar algoritmos que simulan o imiten el comportamiento y la interacción de la luz con los materiales. Estos algoritmos que tratan de simular de forma realista el comportamiento de la luz son generalmente conocidos como algoritmos de iluminación global.

Estos algoritmos, por lo general, suelen tener una complejidad computacional muy elevada y el tiempo de cómputo necesario para obtener un resultado satisfactorio en escenas complejas era un factor limitador en su aplicación práctica. Por ello las aplicaciones que hacen uso de gráficos 3D en tiempo real típicamente se centran en la iluminación local o directa de los objetos de la escena y simulan la iluminación indirecta mediante técnicas que aun sin tener un fundamento físico ofrecen una mayor credibilidad para el ojo humano. Estas técnicas suelen ser algoritmos de postprocesado que se aplican en espacio de pantalla, por ejemplo “ambient occlusion” o “directional occlusion”.

Sin embargo en los últimos años se han realizado grandes avances en las arquitecturas de las unidades de procesamiento de gráficos (GPUs), en especial la gran capacidad de cómputo en paralelo debido al elevado número de microprocesadores que forman estos dispositivos. Con tal de aprovechar estos avances en el hardware, los fabricantes de GPU han desarrollado librerías de computo generico (OpenCL, CUDA) que ofrecen gran libertad al programador para implementar sus propios algoritmos.

Estas mejoras han permitido realizar implementaciones de algoritmos de iluminación global en las GPUs que son mucho más rápidos que las implementaciones típicas en la CPU permitiendo reducir el tiempo de cómputo de varias horas o

días a minutos e incluso a ratios interactivos dependiendo de la GPU y algoritmos utilizados.

1.2. Algoritmos de iluminación global

Se conoce como algoritmos de iluminación global aquellos que tratan de simular distintos aspectos del comportamiento de la luz en su interacción con los objetos de una escena tridimensional. Algunos de ellos están pensados y optimizados para fenómenos concretos mientras que otros tratan de recrear fielmente todos los aspectos del transporte de luz.

En esta sección revisaremos por encima algunos de los algoritmos clásicos.

1.2.1. Radiosity

Radiosity fue el primero de los algoritmos de iluminación global que se desarrollaron. Inicialmente el algoritmo fue desarrollado en los años 1950 para aplicarlo al problema de la transferencia de calor. En 1984 fue modificado y adaptado por Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg y Bennett Battaille, investigadores de la universidad de Cornell para su aplicación en la generación de imagen sintética.

Este algoritmo trata de resolver el problema de la iluminación indirecta entre superficies puramente difusas o Lambertianas.

Capítulo 2

FUNDAMENTOS TEÓRICOS

2.1. Unidades Radiométricas

Se conoce como radiometría al estudio de las radiaciones electromagnéticas. Ya que la luz visible es una onda electromagnética los algoritmos de renderizado que buscan el realismo se fundamentan sobre conceptos radiométricos. Por ello en esta sección haremos una pequeña introducción sobre algunos conceptos básicos que nos permitan entender mejor los algoritmos de iluminación global.

2.1.1. Flujo

El flujo radiométrico mide la cantidad de energía radiante por unidad de tiempo. Sus unidades son Watts o Joules/segundo.

$$\Phi = \frac{dQ(t)}{dt} \quad (2.1)$$

2.1.2. Irradiancia

La irradiancia representa el flujo incidente en una superficie y se mide como el flujo radiante por unidad de área y sus unidades son de W/m^2

$$E = \frac{d\Phi}{dA} \quad (2.2)$$

2.1.3. Angulo solido

El angulo solido no es una unidad radiométrica en si mismo pero es un concepto geométrico necesario para poder explicar otros conceptos radiométricos además de otros apartados del presente trabajo.

Podemos entender el concepto de angulo solido como la extension del angulo a las tres dimensiones.

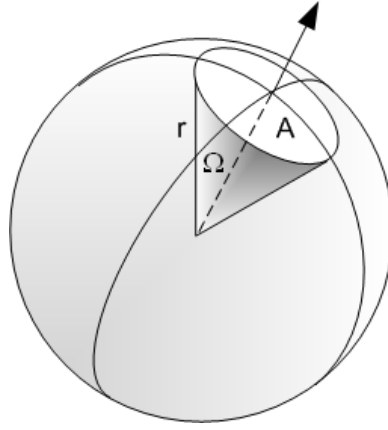


Figura 2.1: Definición de angulo solido [Haade, 2007]

El angulo solido se mide como el área medida sobre una esfera de radio unitario. Sus unidades son adimensionales y son llamadas stereorradianes $[sr]$.

$$\Omega = \frac{A}{r^2} \quad (2.3)$$

Usando coordenadas esféricas $\Theta = (\phi, \theta)$ podemos definir el angulo solido diferencial como:

$$d\omega_{\Theta} = \sin \theta d\theta d\phi \quad (2.4)$$

2.1.4. Radiancia

La radiancia, también llamada intensidad por algunos autores [Kajiya, 1986, Immel et al., 1986], es probablemente la unidad radiométrica mas importante en lo que concierne al presente trabajo.

Esta unidad mide la irradiancia por unidad de angulo solido.

$$L = \frac{dE}{d\omega} = \frac{d^2\Phi}{d\omega dA \cos \theta} \quad (2.5)$$

2.2. BRDF

La función de distribución de reflectancia bidireccional (de ahora en adelante BRDF, por sus siglas en inglés), definida por primera vez por [Nicodemus, 1965] Nicodemus (1965), es un función que define la respuesta a la luz de una superficie opaca, tomando como parámetros dos vectores unitarios que definen las direcciones de entrada y salida de la luz. Más formalmente, la BRDF mide la relación entre la radiancia diferencial reflejada en la dirección de salida y la irradiancia diferencial entrante en el ángulo sólido diferencial alrededor del vector de entrada

$$f(x, l, v) = \frac{dL(x \rightarrow v)}{dE(x \leftarrow l)} \quad (2.6)$$

donde l es el vector unitario que apunta en la dirección opuesta a la de entrada de la luz y v es el vector unitario que apunta en la dirección de salida de la luz.

La BRDF solo esta definida para vectores l y v tales que $n \cdot v > 0, n \cdot l > 0$, siendo n la normal de la superficie.

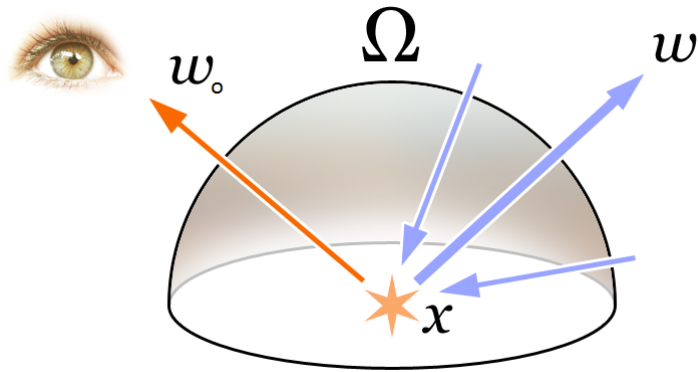


Figura 2.2: BRDF $l = \omega_i, v = \omega_o$

Para obtener la radiancia total reflejada en un punto x en la dirección saliente v es necesario integrar sobre el angulo solido en el dominio de la hemiesfera centrada en x .

$$L_o = \int_{\Omega_x} f(x, l, v) L_i(l) (l \cdot n) d\omega_i \quad (2.7)$$

2.2.1. Propiedades de la BRDF

Una BRDF debe cumplir ciertas propiedades para que sea físicamente plausible. En primer lugar debe cumplir la ley de conservación de la energía. En el caso que nos ocupa esto significa que una superficie puede absorber luz, transformándola en calor, o puede reflejarla pero en ningún caso puede reflejar mas energía lumínica que la que recibe.

$$\forall l, \int_{\Omega_x} f(x, l, v)(n \cdot v) d\omega_o \leq 1 \quad (2.8)$$

Además también debe ser reciproca, esto significa que si intercambiamos los vectores l y v su valor se mantiene. Este hecho cobra sentido si pensamos que la BRDF es una característica intrínseca de cada material y que al intercambiar los vector v y l el angulo entre ellos sigue siendo el mismo.

$$f(x, l, v) = f(x, v, l) \quad (2.9)$$

2.3. Ecuación de renderizado

La ecuación de renderizado fue desarrollada en los años 80 simultaneamente y de forma independiente por distintos autores [Kajiya, 1986, ?]. Se trata de una ecuación integral que unifica y formaliza los distintos algoritmos de renderizado, ya que hasta ese momento no existía un marco de trabajo teórico común.

Esta ecuación se puede encontrar en muchas formas distintas según el autor que la cite pero en la forma propuesta por el autor original es la siguiente:

Donde $I(x, x')$ es la intensidad de la luz que llega del punto x' al punto x , (x, x') es la intensidad de la luz emitida en el punto x' hacia el punto x . (x, x', x'') es una función de distribución que determina qué proporción de la luz incidente en x' proveniente de x'' es rebotada hacia x . Esta función depende de las características de cada material y es comúnmente conocida como BRDF de sus siglas en inglés Bidirectional Reflectance Distribution Function.

El dominio de la integral, S , es la unión de todas las superficies de la escena $S = \bigcup S_i$ $g(x, x')$ es un término geométrico que determina la visibilidad entre los puntos x, x' .

A primera vista esta ecuación puede parecer imponente pero si nos olvidamos por un momento de los formalismos matemáticos y nos centramos en su significado, lo que viene a decir es que la luz que llega al punto x en la dirección $x'x$ es igual a la luz emitida en el punto x' en la dirección $x'x$ más la integral de toda la luz que llega al punto x' y es dispersada en la dirección $x'x$.

Lo significativo de esta ecuación es que resulta muy intuitivo derivar algoritmos de renderizado de la misma: se evalúa para cada punto a pintar y se evalúa $I(x', x'')$ recursivamente hasta que se cumpla determinada condición.

2.4. El metodo de montecarlo

El método de montecarlo se trata de un método de integración numérico para integrales definidas sobre un dominio de dimension arbitraria, del tipo:

$$I = \int_D f(x)dx, D \subseteq \mathbb{R}^m \quad (2.10)$$

Sabemos que la esperanza de una función continua se define como la integral de la función por la probabilidad de x . Y que podemos estimar la esperanza calculando la media de los valores que toma la función en puntos aleatorios escogidos independientemente y con la misma distribución.

$$E(f(x)) = \int f(x)p(x)dx \approx \frac{1}{N} \sum_{i=1}^N f(x_i) \quad (2.11)$$

El método de montecarlo se basa en este hecho para estimar el valor de una integral definida tomando muestras aleatorias sobre el dominio $x_1, x_2, \dots, x_n \in D$ y aplicando:

$$I = \int_D f(x)dx \approx \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)} \quad (2.12)$$

Siendo $p(x_i)$ la probabilidad de tomar una muestra x_i concreta de entre todas las posibles en el dominio D . En el caso de tomar las muestras sobre una distribución uniforme:

$$p(x_i) = \frac{1}{\int_D dx} \quad (2.13)$$

$$I \approx \frac{\int_D dx}{N} \sum_{i=1}^N f(x_i) \quad (2.14)$$

2.4.1. Muestreo de importancia

Capítulo 3

FRAMEWORK

3.1. Cuda

3.2. La librería OptiX

3.2.1. Host

clase Material

clase Geometry

3.2.2. Device

Intersection programs

Closest hit programs

Any hit programs

Bibliografía

- [Goral et al., 1984] Goral, C. M., Torrance, K. E., Greenberg, D. P., and Battaile, B. (1984). Modeling the interaction of light between diffuse surfaces.
- [Haade, 2007] Haade (2007). Solid_Angle.
- [Immel et al., 1986] Immel, D., Cohen, M., and Greenberg, D. (1986). A radiosity method for non-diffuse environments. *ACM SIGGRAPH Computer ...*, 20(4):133–142.
- [Kajiya, 1986] Kajiya, J. (1986). The rendering equation. *ACM Siggraph Computer Graphics*, 20(4):143–150.
- [Nicodemus, 1965] Nicodemus, F. (1965). Directional reflectance and emissivity of an opaque surface. *Applied Optics*, 4(7):767–773.

