Computación Gráfica

Trabajo práctico final

Gonzalo Castiglione (49138) - Braulio Sespede (51074)
Grupo 2

Índice

1. Opcionales implementados

- Depth of field
- Spherical camera

Se eligieron implementar estos opcionales ya que ambos estaban relacionados con la arquitectura de la camara por lo que fueron sencillos de implementar.

2. Depth of field

Se implementó una versión parecida a la de "*Ray tracing from the ground up*". Se recibe por linea de comandos los parametros fd y lr que representan la distancia al plano focal y el radio de apertura del lente.

El funcionamiento del algoritmo es parecido al de la camera pinhole excepto por el calculo de la posición de los rayos principales y su dirección. La posición de origen de los rayos es un punto en un disco de radio de un tamaño dado sobre la posición de la camara, estos puntos se obtienen moviendose al origen de la cámara y finalmente sumandole las coordenadas del sampleado de un disco de radio 1 multiplicado por el tamaño del lente. En cuanto al calculo de la direcciones, se calcula constructivamente el punto donde colisionaria con el plano focal y se obtiene la dirección desde el lente.

Finalmente se disparan los rayos como con la camara pinhole, haciendo un average del color obtenido.

En un principio se tomaban samples random sobre un cuadrado en vez de utilizar multijittered sampling sobre un disco, por lo que se notaban claros efectos de aliasing.

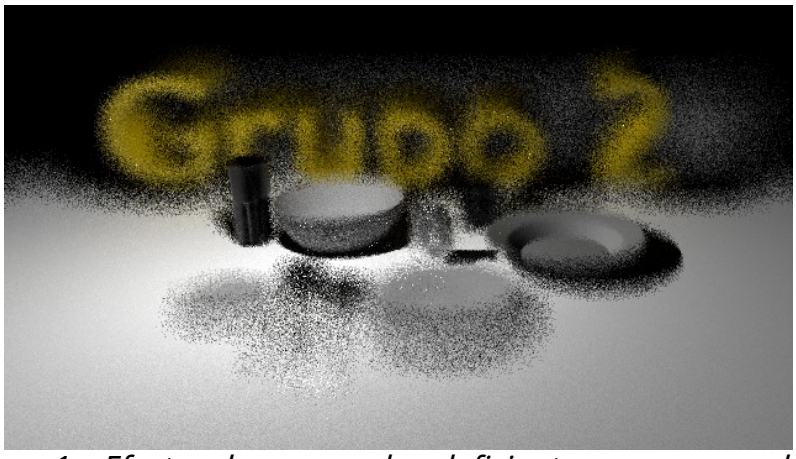


Imagen 1 – Efectos de un sampleo deficiente y pocos samples

2.1. Resultados

A continuación se muestra el efecto de los distintos parametros del depth of field sobre la escena "*metal2_scene/untitled.lxs*" con 4 hops maximos, 16 samples de aa (y de lente) y 100 de recasteo del mismo rayo.

Javar -jar

	Distancia focal	**Apertura**	**Tiempo de rendering (seg)**
dof1.png	11	1	1115
dof2.png	11	2	1026
dof3.png	11	3	1093
dof4.png	7	2	1075
dof5.png	13	2	1077

Las imagenes se encuentran en la carpeta final-images

Viendo las imagenes 2 a 4 se nota que a medida que aumenta la distancia focal, el plano focal se aleja de la camara, enfocando cada vez más lejos. Por otro lado, viendo las imagenes 3, 5, 6, se ve que al aumentar la apertura, el efecto de desenfoque lejos del plano focal es cada vez mayor.

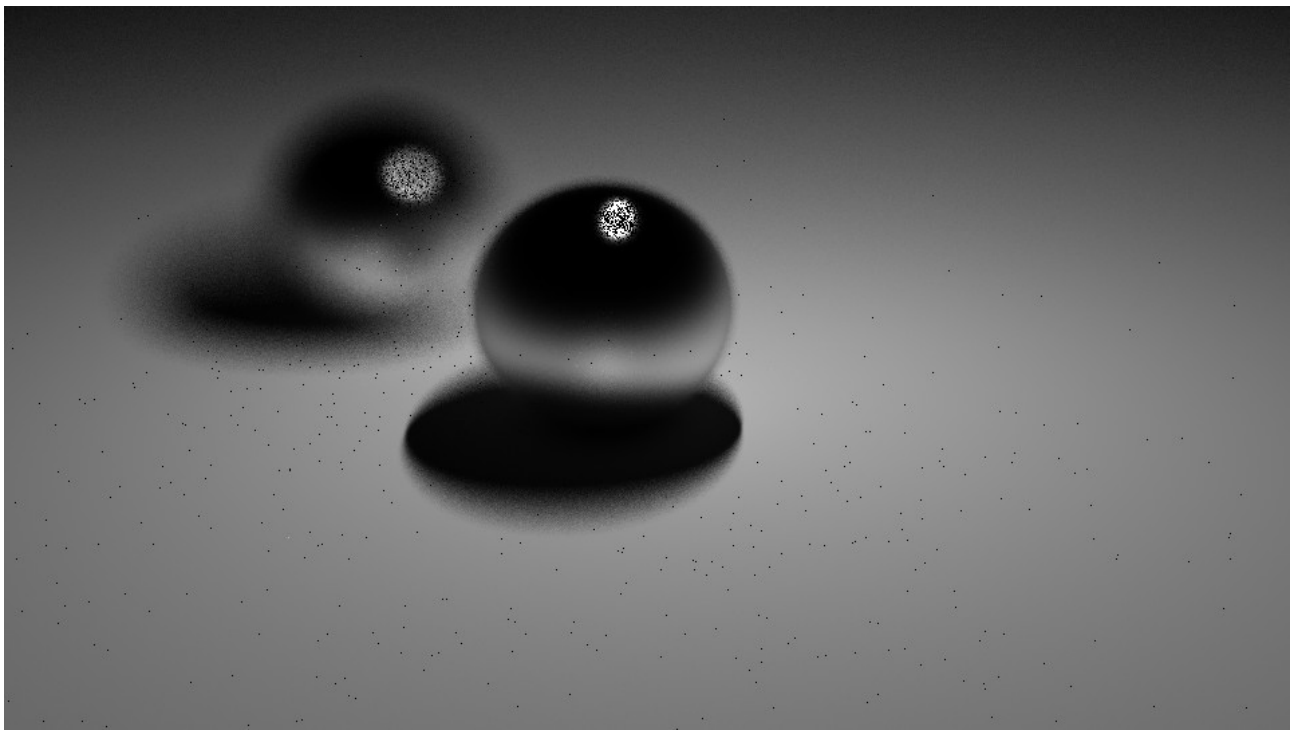


Imagen 2 - dof2.png con 100 samplesde aa, en vez de 16

También se puede observar que la cantidad de samples necesarios para obtener una imagen de buena calidad aumenta significativamente.

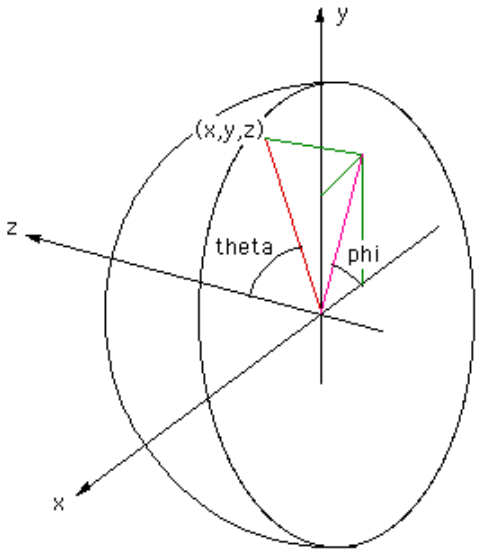
3. Spherical camera

Se implementaron las formulas encontradas en http://paulbourke.net/dome/fisheye/. Se recibe por linea de comandos los parametros phi, que representa el angulo máximo y va desde 0 a 360.

Para implementar la proyección “fisheye angular” solo fue necesario extender la clase Camera y cambiar la forma en que se calculan las direcciones de los rayos principales, de manera que se proyecten sobre el plano imagen de manera que desde el centro de la imagen sea proporcional al angulo desde la direccion de la camara. La idea principal es que la distancia desde el centro de la imagen (se renderea como una esfera en el plano) se mapee el angulo sobre de la proyección esferica en el plano.

Para lograr esto se siguieron los siguientes pasos:

1. Se normaliza las coordenadas del viewPort de -1 a 1
2. Se calcula r y el angulo proyectado (phi)
3. Aquellos pixeles que tengan r mayor que 1 son ignorados (estan afuera de la esfera proyectada)
4. Por último, con aquellos pixeles válidos, se mapea r sobre el angulo theta (ver imagen 2)
5. Se usa phi y theta para obtener el vector dirección deseado usando la base ortonormal definida por la camara.



theta = r * aperture/2

Direction vector (x,y,z)

x = sin(theta) cos(phi)
y = sin(theta) sin(phi)
z = cos(theta)

0 ≤ theta ≤ aperture/2
0 ≤ phi ≤ 2π

Imagen 3 – Cálculo del vector dirección

3.1. Resultados

A continuación se muestra el efecto de los distintos parametros del depth of field sobre la escena "*scene_hard/scene_hard.lxs*" con 4 hops maximos, 16 samples de aa (y de lente) y 100 de recasteo del mismo rayo.

	Phi máximo	**Tiempo de rendering (seg)**
fisheye1.png	45	6538
fisheye2.png	120	2286
fisheye3.png	240	1750

Las imagenes se encuentran en la carpeta final-images

4. Comentarios

Otro problema encontrado fue que la implementación de raycasting no anda correctamente con los nuevos tipos de camara, por lo que se hicieron las pruebas corriendo con pathtracing unicamente.

Por desición de implementación no se puede usar ambos tipos de cámaras a la vez.