Projet 150h

Lancer de rayons en temps réel

Nathan Roth

Encadré par M. Théry Sylvain et M. Pascal Guehl

Université de Strasbourg

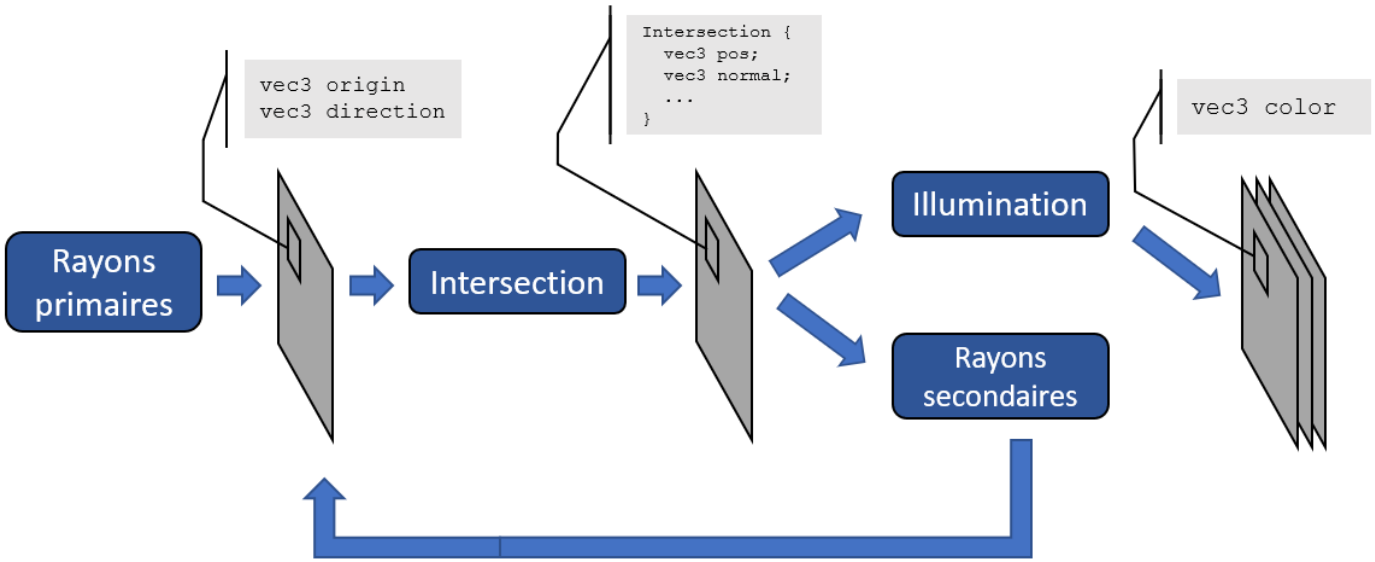
Contexte

Battlefield V (Electronic Arts)

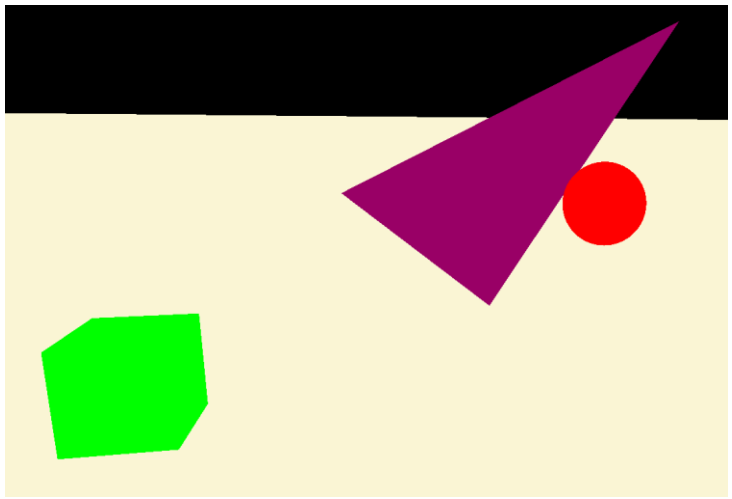
Date	GPU	TFLOPS (32bits)
Juin 2015	GTX 980 ti	6
Mars 2017	GTX 1080 ti	11,3
Sept. 2018	RTX 2080 ti	14,2

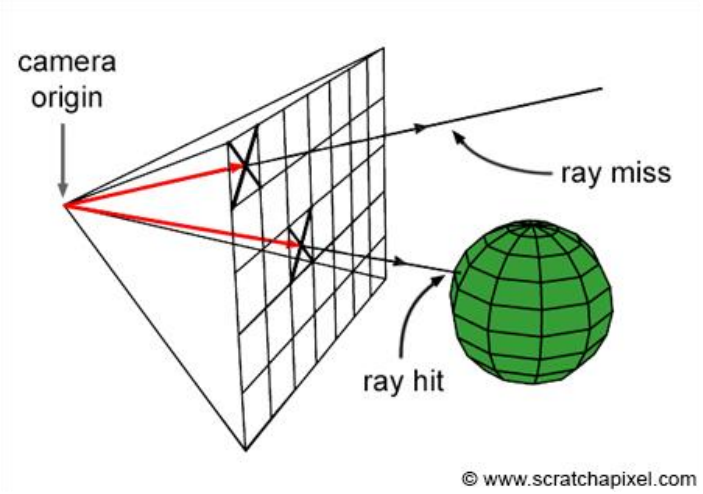
Minecraft (Sonic Ether's mod)

Programme


vec3 origin
vec3 direction
Intersection {
vec3 pos;
vec3 normal;
...
}
vec3 color
Rayons primaires
Intersection
Illumination
Rayons secondaires

Lancer de rayons

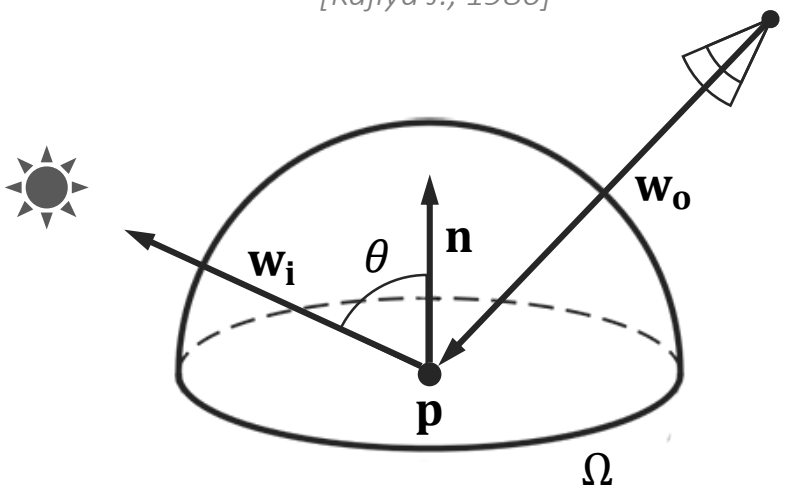




Estimateur de Monte-Carlo

$$L_o(\mathbf{p}, \mathbf{w_o}, \mathbf{w_i}) = L_e + \int_{\Omega} L_i(\mathbf{p}, \mathbf{w_i}) f_r(\mathbf{w_o}, \mathbf{w_i}) \cos\theta \, d\omega$$

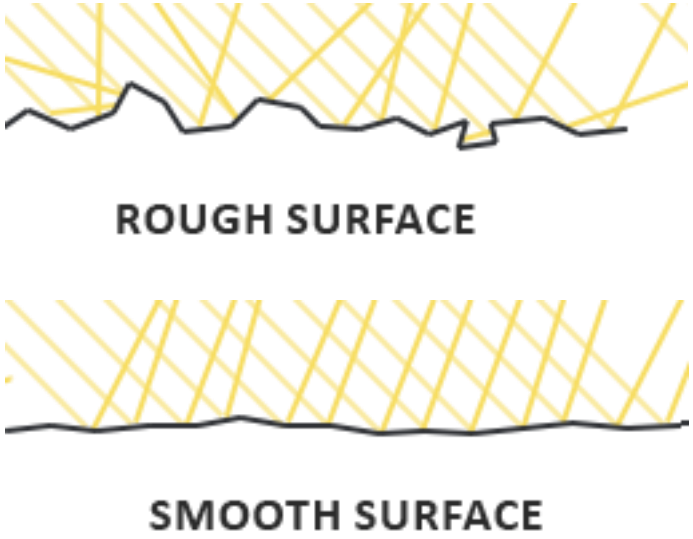
[Kajiya J., 1986]

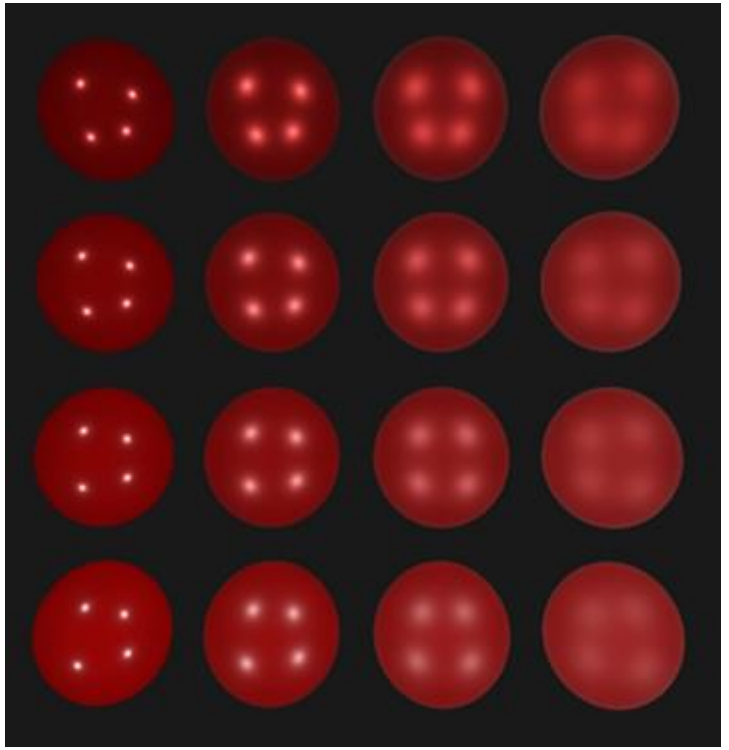


$$L_o(\mathbf{p}, \mathbf{w_o}) = \frac{1}{N} \sum_{k}^{N} \frac{L_i(\mathbf{p}, \mathbf{w_k}) f_r(\mathbf{w_o}, \mathbf{w_k}) \cos\theta}{P(\mathbf{w_k})}$$

[M. Pharr, « PBR »]

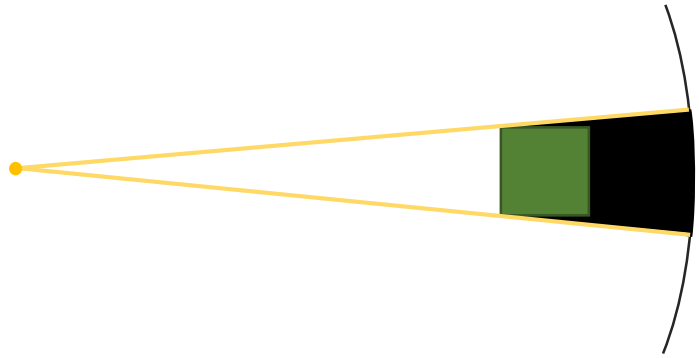
Modèle micro-facettes

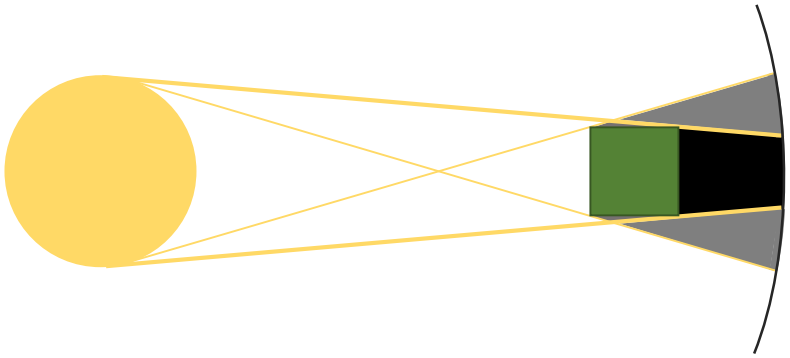


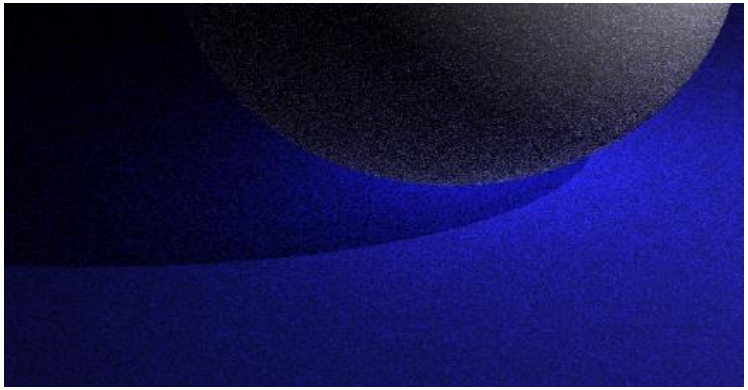


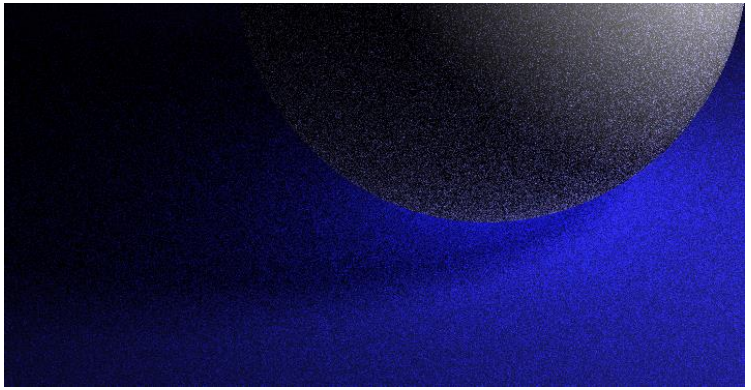
(learnopengl.com)

Soft shadows

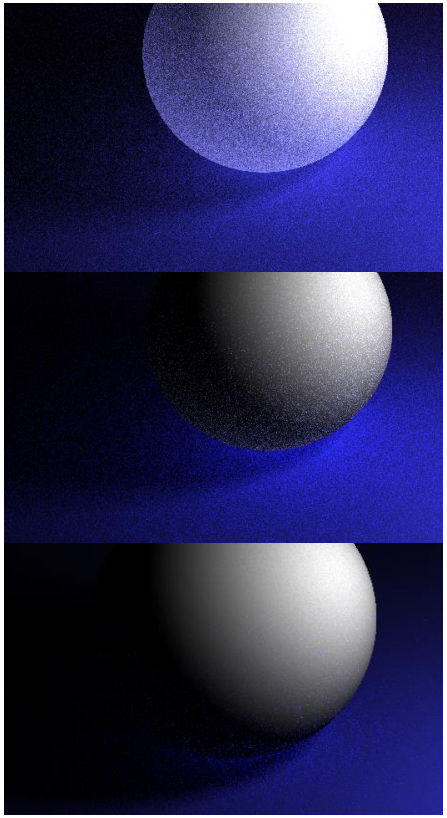


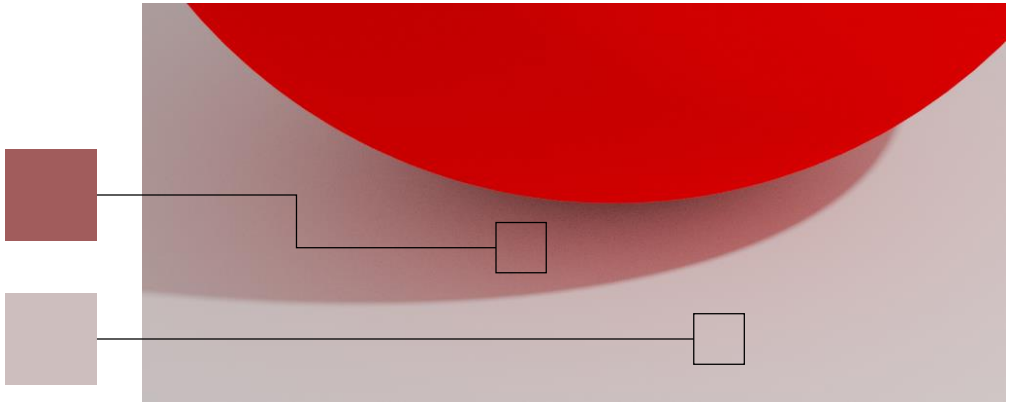


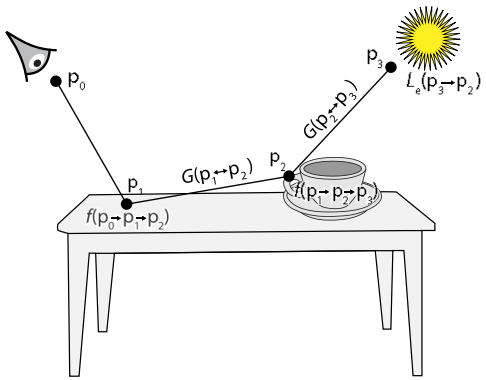


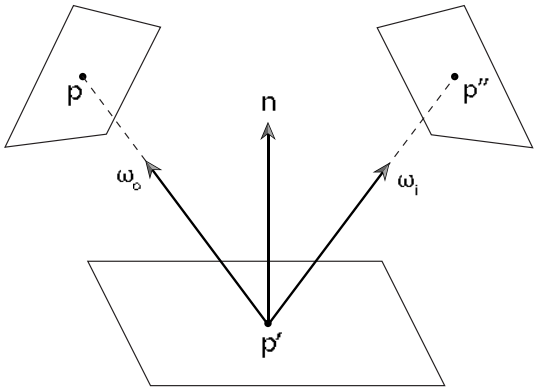


Path tracing



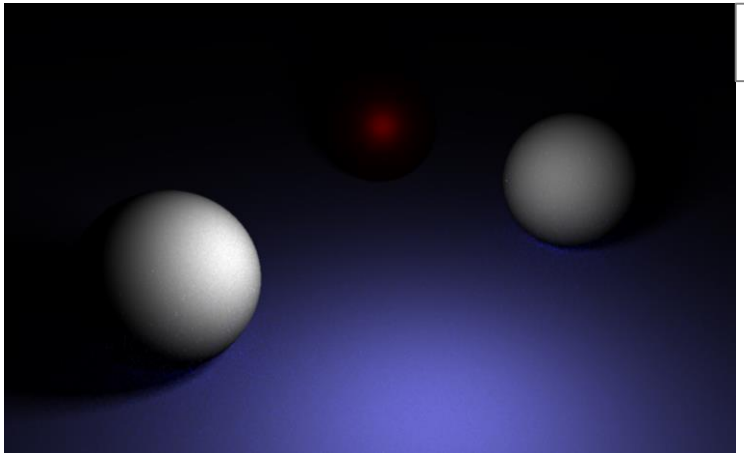




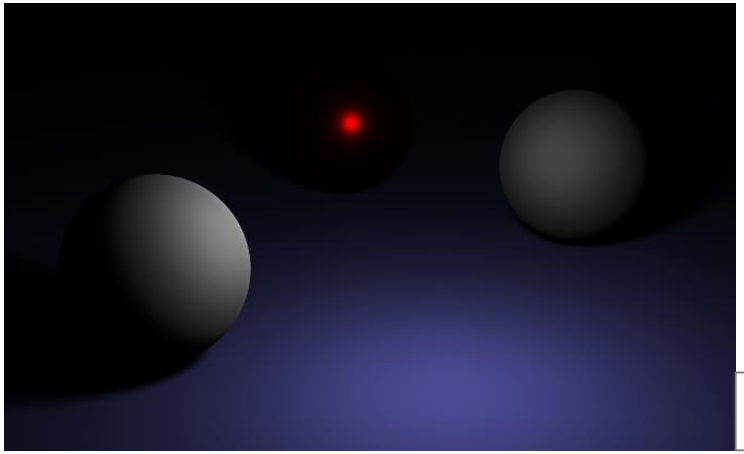


(M. Pharr, « PBR »)

Performances



Notre moteur



Blender

Illumination directe

Rayons	Rebonds	Lumière	IPS
5	1	1	>60
10	1	1	>60
15	1	1	>60
20	1	1	>60
10	5	1	>60
15	5	1	30
20	10	1	27
10	5	5	30
15	5	10	20
15	5	15	14

Conclusion et ouverture

Multiple Importance Sampling
PDF générale pour Cook-Torrance

Références

Webographie

- TechPowerUp, www.techpowerup.com/gpu-specs/
- 2009 MINECRAFT VS 2019 MINECRAFT RAY TRACING - Minecraft with RTX Youtube, www.youtube.com/watch?v=NehSihoHCpc
- Scratchpixel, www.scratchapixel.com
- J. d. Vries, «Learn OpenGL», learnopengl.com

Bibliographie

- The rendering Equation, James T. Kajiya, *ACM Siggraph, vol. 20, n°4, pp 143-150, 1986*
- R. L. Cool et K. E. Torrance, «A Reflectance Model for Computer Graphics» *ACM Transaction on Graphics, vol. 1, n° 11, pp. 7-24, 1982.*
- M. Pharr, W. Jakob et G. Humphreys, « Physically Based Rendering: From Theory To Implementation»