



# Approximation d' éclairage indirect en temps réel: méthodes et algorithmes

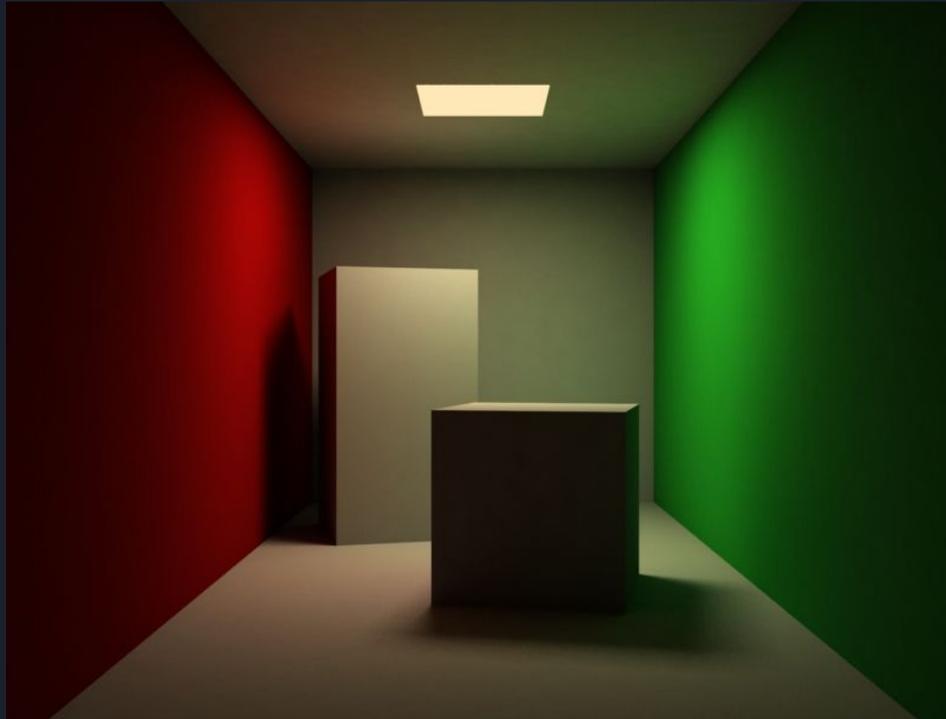
Blaise Cardonne, Gauthier Bouyjou, Valentin Camus, Rihab Elrifai, Sylvain Durand  
Encadrant/Client : François Desrichard



# Axes de la présentation

1. Méthodes et algorithmes:
  - Introduction à la problématique: l'éclairage indirect
  - Le problème de la visibilité
  - Les gaussiennes sphériques
  - Gestion de scènes
2. Plan de développement:
  - Moyens de communications
  - Gestion de versions
  - Gestion des tests
3. Conclusion:
  - Décomposition en tâches et planning préliminaire

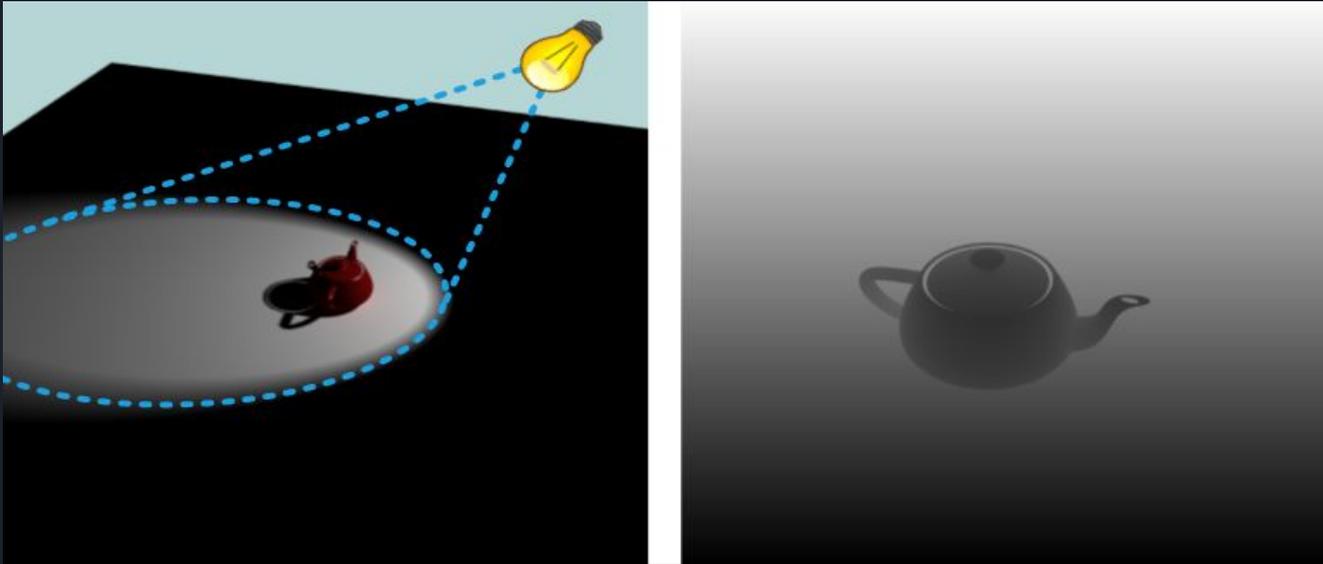
# L'éclairage indirect



Cornell box ©Wikipedia

# Shadow map

- Génération de cartes de profondeurs depuis les sources de lumières



[http://www.downloads.redway3d.com/downloads/public/documentation/bk\\_re\\_shadow\\_mapping\\_detailed.html](http://www.downloads.redway3d.com/downloads/public/documentation/bk_re_shadow_mapping_detailed.html)

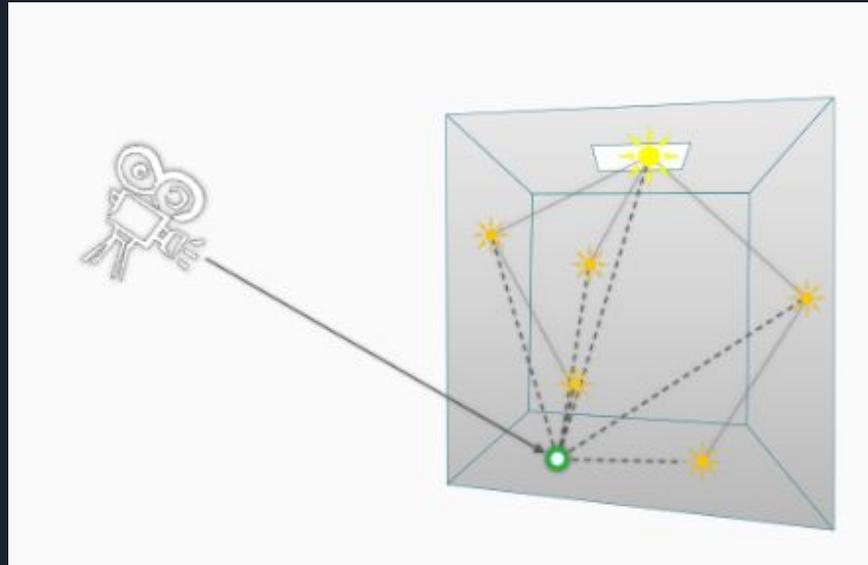


# Le problème de la visibilité

- “It has been observed that for many purposes, global illumination solutions do not need to be precise, but only plausible.”
  - Carsten Dachsbacher et Marc Stamminger, 2005
  
- Plusieurs approches basées sur *Instant radiosity* (Alexander Keller, 1997):
  - *Reflective shadow maps*, Carsten Dachsbacher et Marc Stamminger, 2005
  - *Imperfect shadow maps for efficient computation of indirect illumination*, Ritschel et al., 2008

# Les VPLs

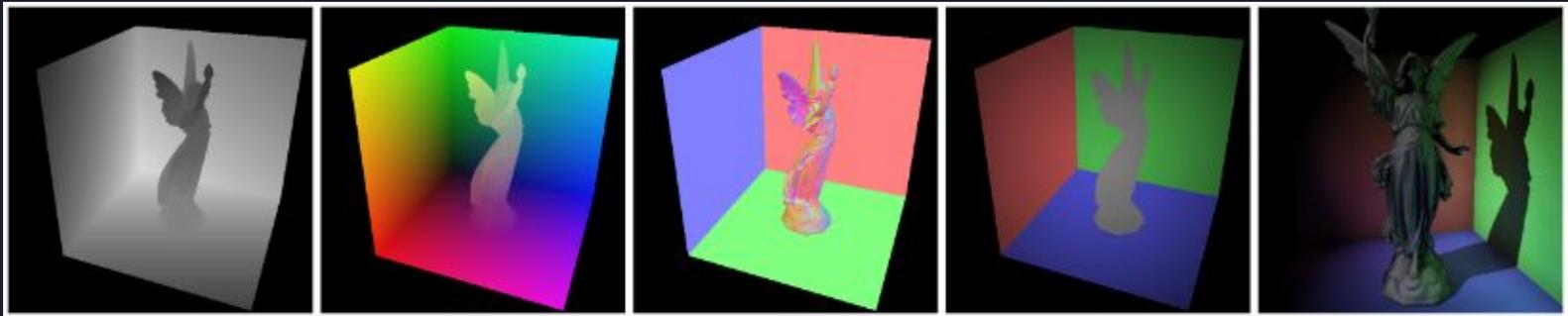
- Approximation à base de *Virtual Point Lights (VPLs)*
  - *Instant radiosity*, Alexander Keller, 1997



©Mathias Paulin

# Reflective shadow maps

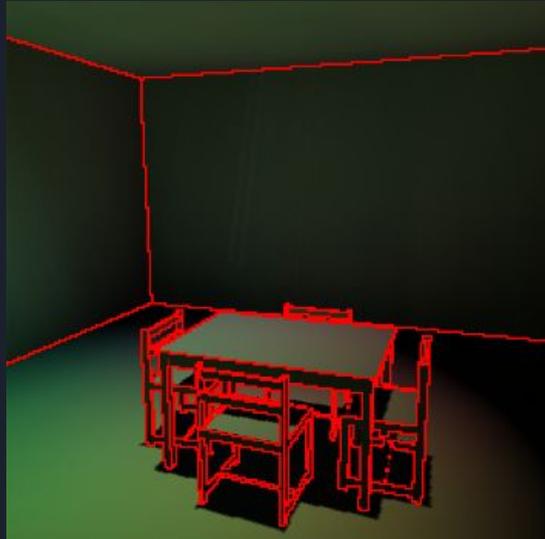
- De l'article *Reflective shadow maps*, par Carsten Dachsbacher et Marc Stamminger en 2005
- Complète l'approche des cartes de profondeurs traditionnelles avec de nouvelles informations:
  - les coordonnées en espace monde
  - les normales
  - le "flux radiant" ( $\sim c_{\text{source}} * c_{\text{diffuseM}}$ )



©Carsten Dachsbacher et Marc Stamminger

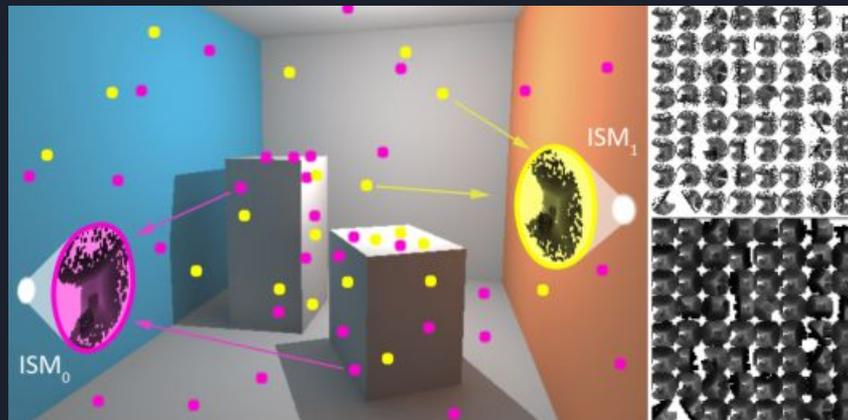
# Reflective shadow maps

- Ignore complètement le problème de la visibilité entre un point de la scène et une VPL
- Considère un sous ensemble de VPLs par point de la scène
- Effectue tout ce processus une première fois à basse résolution pour interpoler si c'est raisonnable l'éclairage indirect



# Imperfect shadow maps

- De l'article *Imperfect shadow maps for efficient computation of indirect illumination* par Ritschel et al. en 2008
- Répond au problème de visibilité entre un point de la scène et une VPL via des cartes de profondeurs basses résolutions



©Ritschel et al.

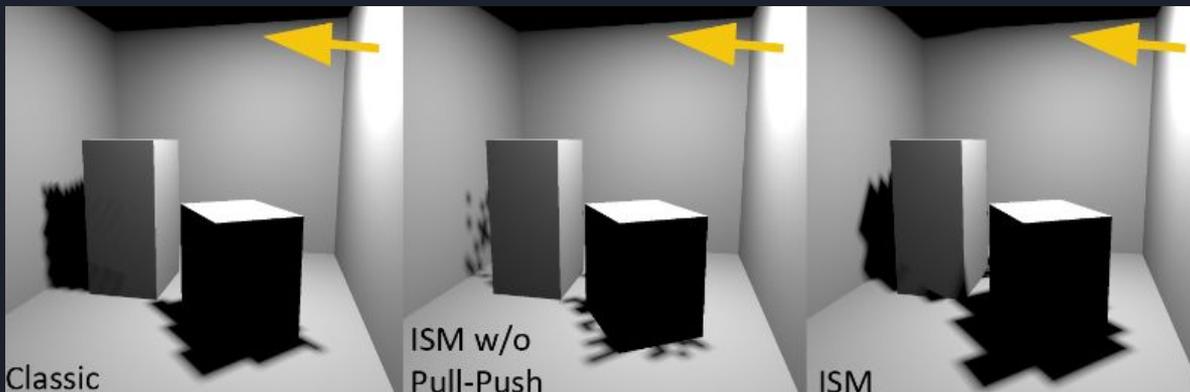
# Imperfect shadow maps

- La méthode:
  - création d'un nuage de points représentant la scène dans une phase de pré-traitement
  - calcul des cartes de profondeurs à partir d'un sous ensemble du nuage de point
  - étape de "pull/push" pour corriger les artéfacts liés à la représentation sous forme de nuages de points



# Imperfect shadow maps

- La méthode:
  - création d'un nuage de points représentant la scène dans une phase de pré-traitement
  - calcul des cartes de profondeurs à partir d'un sous ensemble du nuage de point
  - étape de "pull/push" pour corriger les artéfacts liés à la représentation sous forme de nuage de points



©Ritschel et al.

# Imperfect shadow maps



©Ritschel et al.

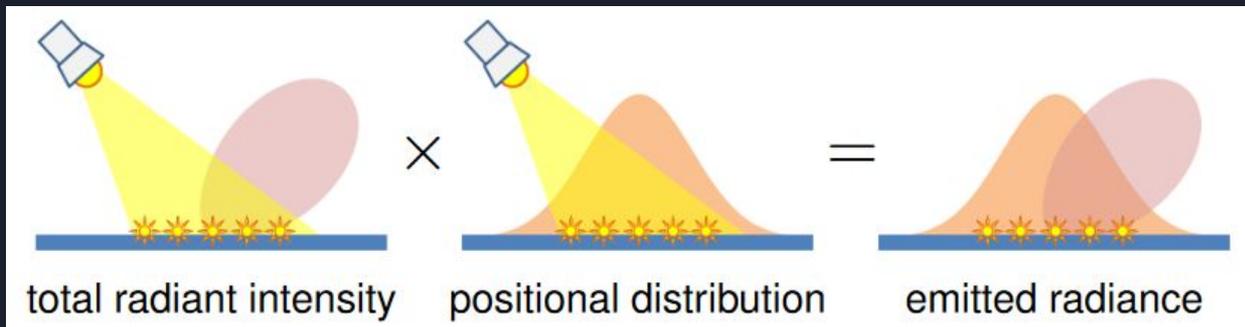
# Les gaussiennes sphériques

- De l'article *Virtual spherical gaussian lights for real-time glossy indirect illumination* par Yusuke Tokuyoshi en 2015
- Des propriétés intéressantes
  - Produits de gaussiennes → une gaussienne
  - Expression analytique pour l'intégrale
- Implantation de l'article basée sur *Reflective shadow maps*, par Carsten Dachsbacher et Marc Stamminger



# Les gaussiennes sphériques

- Utilise une distribution gaussienne pour représenter spatialement un ensemble de VPLs
  - produit de gaussiennes
- Exploite le *mipmapping* matériel des GPUs pour sommer plusieurs VPLs et fonctionner en temps réel sur des scène dynamiques



©Yusuke Tokuyoshi

# La gestion de scènes

- Primitives d'éclairage
  - lumières ponctuelles 💡
  - lumières directionnelles ✨
  - lumières de type projecteur 🔦
- Chargement de modèles arbitraires
  - gestion de maillages triangulaires
  - gestion de nuages de points (*ISM*)
- Gestion de matériaux
  - textures de couleurs, normales, ...
  - modèle de BRDF physiquement réaliste (micro-facettes)

$$f_r = k_d \frac{c}{\pi} + k_s \frac{DFG}{4(\omega_o \cdot n)(\omega_i \cdot n)}$$

©codinglabs.net



# Les bibliothèques

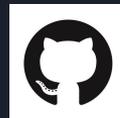
- OpenGL : Calcul d'image 2D ou 3D
- GLFW : Création des fenêtres et la réception des entrées et des événements
- ImGui : Interface utilisateur
- GLM : Algèbre linéaire
- STB : Chargement d'images via des fichiers
- Assimp : Chargement de géométries 3D via des fichiers

# Plan de développement

- Moyens de communication:
  - En interne → Slack
  - Avec le client → e-mails et réunions
- Gestion de versions: Github
- Gestion des tests: tests unitaires par module
- Répartition du travail par tâches: Trello
- Gestion des bugs: issue github



©Slack



©Github



©Trello



# Conclusion

- Découpage en sous tâches:
  - Développer une base de moteur de rendu  
→ deadline mi-janvier
  - Implanter les *reflective shadow maps*  
→ deadline fin janvier
  - Ajouter le support des gaussiennes sphériques  
→ deadline début février
  - Compléter l'approche par *reflective shadow maps* par les *imperfect shadow maps*  
→ deadline mi-février



Merci de votre attention.

Des questions ?