

# M2-Images

## Scènes complexes et pipeline graphique

J.C. Iehl

October 24, 2011

## Résumé des épisodes précédents

### scènes complexes (films d'animations) :

- ▶ volume de données,
- ▶ très petites primitives (>100M pour une image 2M pixels ?),
- ▶ beaucoup de textures (>100Go).

### solutions existantes :

- ▶ nier le problème : méthodes externes / out-of-core,
- ▶ réorganiser le pipeline graphique,
- ▶ construire des représentations multi-échelles des objets.

# Résumé

## méthodes externes / out-of-core :

- ▶ ne charger que les données nécessaires,
- ▶ re-ordonner les calculs pour limiter les chargements,
- ▶ == identifier les “calculs” utilisant les mêmes données, et les réaliser en parallèle lorsque les données sont disponibles.

## et alors ?

- ▶ méthodes utilisées pour le lancer de rayon (visibilité),
- ▶ reyes ?
- ▶ gpu ?

# Pipeline et Complexité

## complexité du rendu :

- ▶ chargement efficace des données,
- ▶ re-ordonnancement / équilibrage,
- ▶ “calculs” parallèles :
- ▶ propagation des rayons,  
intersections avec la géométrie == visibilité,
- ▶ interactions lumière / matière == éclairage ?

## quelle est la complexité du rendu ?

- ▶ fonction du nombre de pixels de l'image ?
- ▶ fonction du nombre d'objets ? de primitives ?

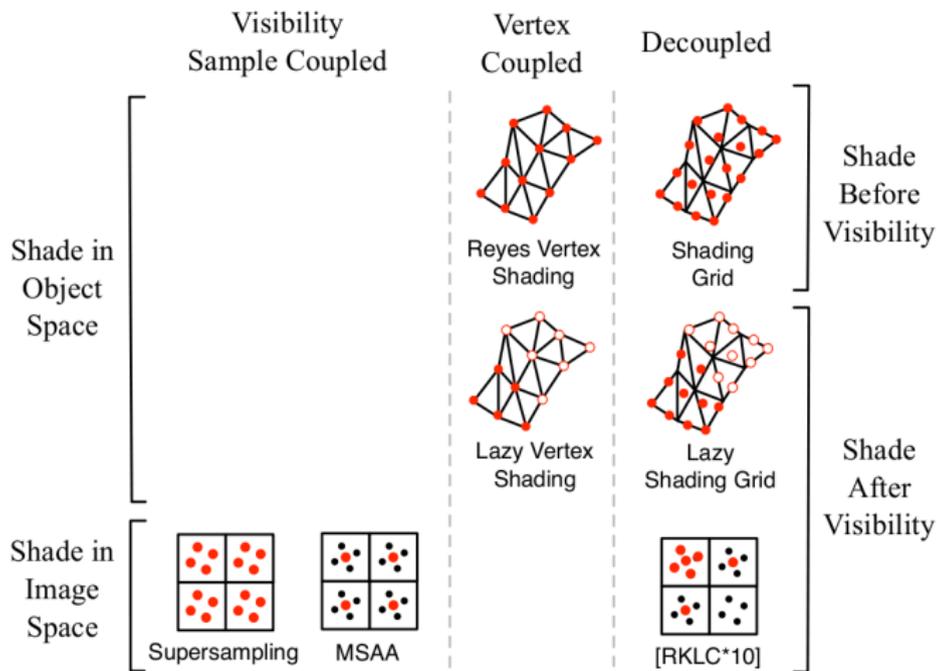
# Complexité et “éclairage”

problème :

- ▶ beaucoup de primitives visibles par pixel,
- ▶ combien de fois est calculé la visibilité ?
- ▶ combien de fois est calculé l'éclairage ?
- ▶ une fois par échantillon de visibilité et par pixel ?
- ▶ une fois par pixel ?

beaucoup trop...

# Découpler visibilité et éclairage



# Sur-échantillonnage

## sur-échantillonnage (FSAA) :

- ▶ solution naïve :
- ▶ calculer une image plus grosse,
- ▶ filtrer et réduire pour obtenir l'image finale.

## bilan :

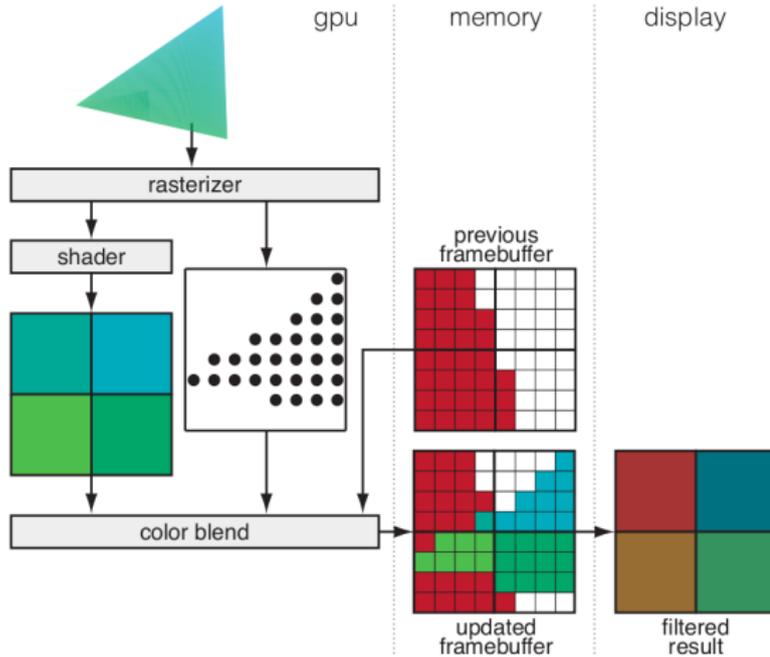
- ▶ l'éclairage est évalué pour chaque échantillon de visibilité,
- ▶ simple, mais inefficace.

# Sur-échantillonnage de la visibilité

sur-échantillonnage de la visibilité (MSAA) :

- ▶ par pixel :
- ▶ plusieurs échantillons de visibilité,
- ▶ une seule évaluation de l'éclairage (= couleur),
- ▶ la couleur du pixel est la moyenne des couleurs des primitives pondérées par l'aire occupée dans le pixel.

# Déterminer la couleur d'un pixel



"Decoupled Sampling for Graphics Pipelines", siggraph 2011

## Sur-échantillonnage de la visibilité (MSAA) :

bilan :

- ▶ nettement mieux,
- ▶ 8, 16, etc. échantillons de visibilité par pixel, (approximation de l'aire occupée par la primitive),
- ▶ 1 seule évaluation de l'éclairage par primitive et par pixel.

# Et alors ?

quelle est la complexité du rendu ?

- ▶ en fonction du nombre de pixels ?
- ▶ en fonction du nombre de primitives ?
- ▶ quelle réduction a-t-on obtenu avec MSAA et FSAA ?

## ...dans l'espace objet

dans quel espace travailler ?

- ▶ dans l'espace de l'image : sur les pixels ?
- ▶ à la surface des objets visibles ?

évaluer l'éclairage avant ou après le résultat du test de visibilité complet ?

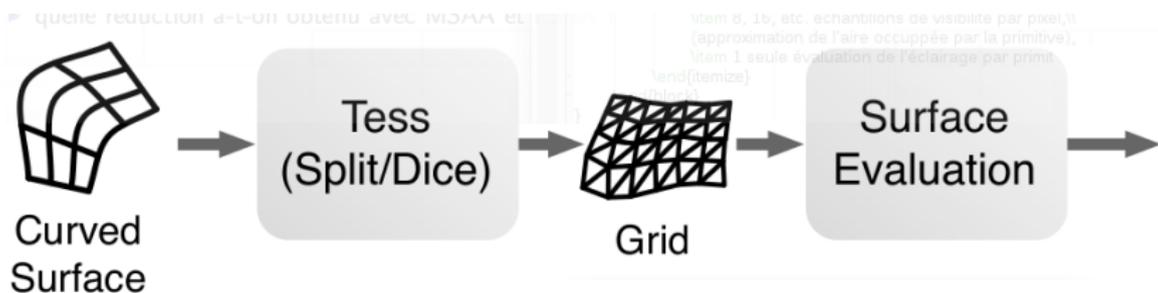
liberté complète sur la méthode de calcul ou restriction en fonction du pipeline ?

# Découpler primitives et éclairage

rappels :

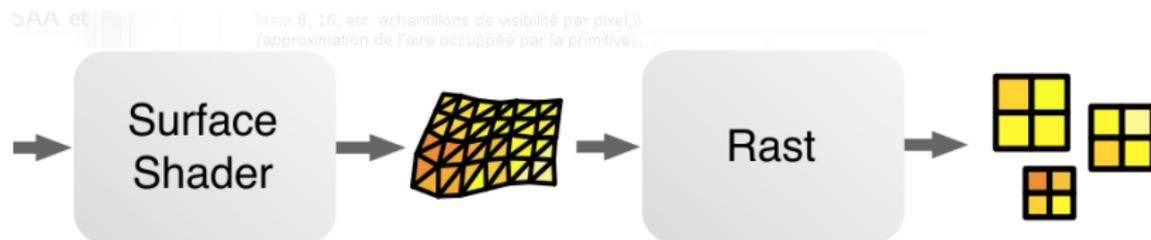
- ▶ pipeline reyes :
- ▶ subdiviser les objets (split),
- ▶ échantillonner en grille de micro-polygones (dice),
- ▶ de taille équivalente à 1 pixel,
- ▶ évaluer l'éclairage des sommets des micro-polygones,
- ▶ tester la visibilité des sommets des micro-polygones.

# Découpler primitives et éclairage



"A Lazy Object-Space Shading Architecture With Decoupled Sampling", hpg 2010

# Découpler primitives et éclairage



"A Lazy Object-Space Shading Architecture With Decoupled Sampling", hpg 2010

# Découpler primitives et éclairage

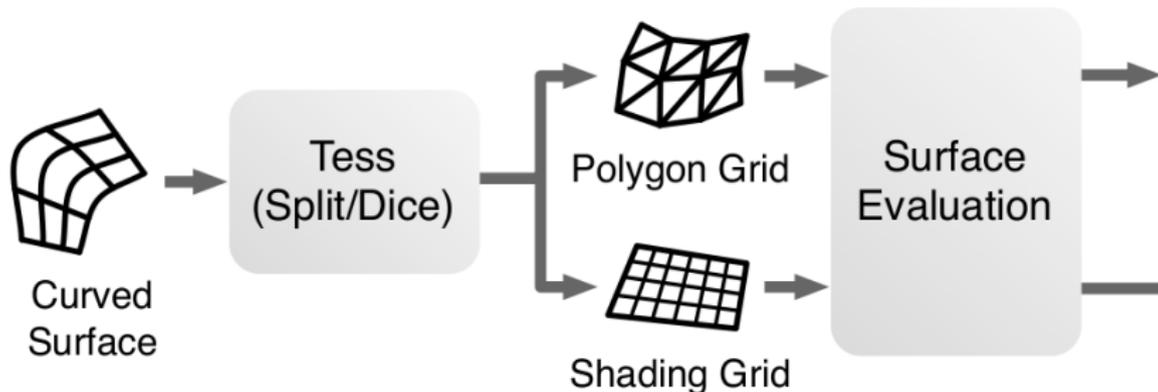
## principe :

- ▶ ne pas générer un sommet / micro-polygone par pixel,
- ▶ (subdiviser en fonction de la forme de l'objet),
- ▶ mais générer une évaluation de l'éclairage par pixel,
- ▶ + conserver les résultats ré-utilisables (pixels voisins),
- ▶ évaluation de l'éclairage dans le domaine paramétrique des objets / triangles,
- ▶ l'éclairage n'est évalué que pour les points visibles.

“A Lazy Object-Space Shading Architecture With Decoupled Sampling”

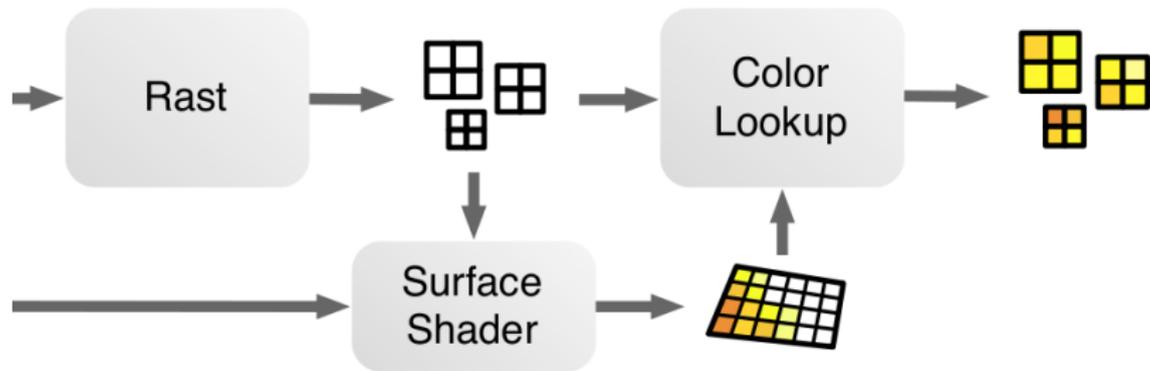
C. Burns, K. Fatahalian, W. Mark, hpg 2010

# découpler primitives et éclairage



"A Lazy Object-Space Shading Architecture With Decoupled Sampling", hpg 2010

# découpler primitives et éclairage



"A Lazy Object-Space Shading Architecture With Decoupled Sampling", hpg 2010

# et alors ?

avantages :

- ▶
- ▶

inconvénients :

- ▶
- ▶

## et alors ?

### avantages :

- ▶ moins de subdivisions, moins de micro-polygones,
- ▶ les micro-polygones sont plus gros ( $> 5$  pixels),
- ▶ les tests de visibilité sont plus efficaces (moins de primitives par pixel),
- ▶ l'éclairage n'est évalué qu'une fois par pixel et lorsque la primitive est visible...

### inconvénients :

- ▶ efficace sur un point de vue. pour les ombres, les reflets ?
- ▶ principes utilisables en lancer de rayons ?

## et alors ?

à lire :

“Decoupled Sampling for Graphics Pipelines”

J. Ragan-Kelley, J. Lehtinen, J. Chen, M. Doggett, F. Durand,  
siggraph 2011