PRÉSENTATION DES MÉTHODES ET ALGORITHMES

Matthias Roves, Hugo Rens et Jehan-Antoine Vayssade

12 novembre 2017

Université Paul Sabatier (Toulouse III)

Analyse

- BSDFs Disney
- Sources ponctuelles très nombreuses
- Sources polygonales et linéaires

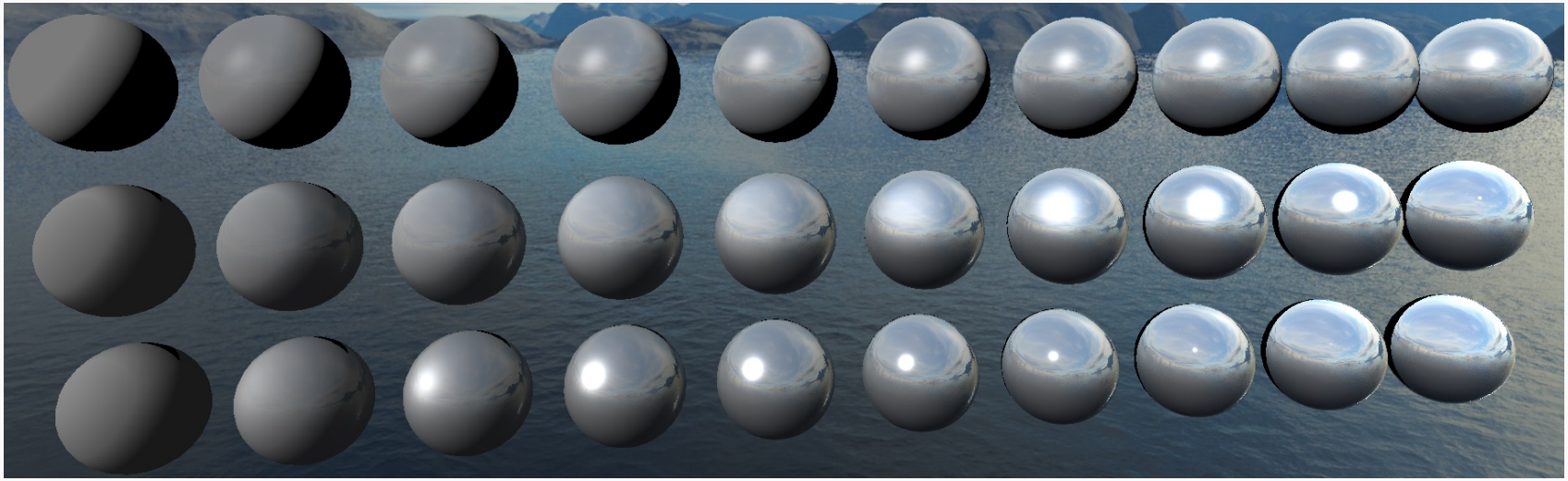
Plan de développement

- Étapes clés
- Risques
- Organisation

ANALYSE

BSDFs Disney

Modèle de Cook-Torrance



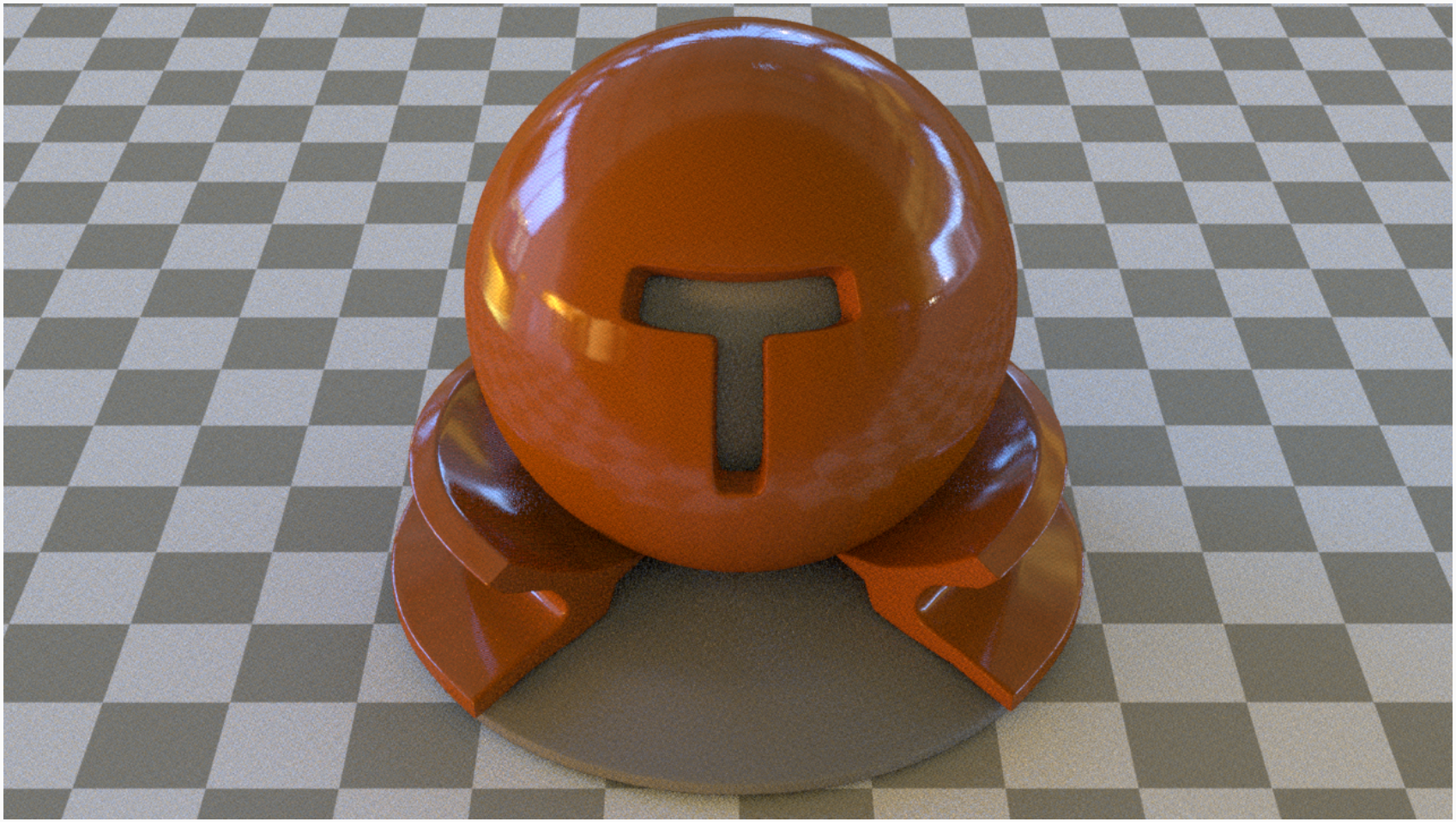
$$\rho_{ct}(X, w_i, w_o) = \frac{D * G * F}{4 * (w_o \cdot n_x) * (w_i \cdot n_x)} \tag{1}$$

BSDF PLASTIQUE

La BSDF de type Plastique se base sur un modèle de matériau diélectrique standard. Il sera donc modélisé avec un modèle de diffusion Lambertien et un spéculaire basé sur un Fresnel diélectrique.

$$diff_{Lambert}(x, w_i, w_o) = \frac{1}{\pi} \tag{2}$$

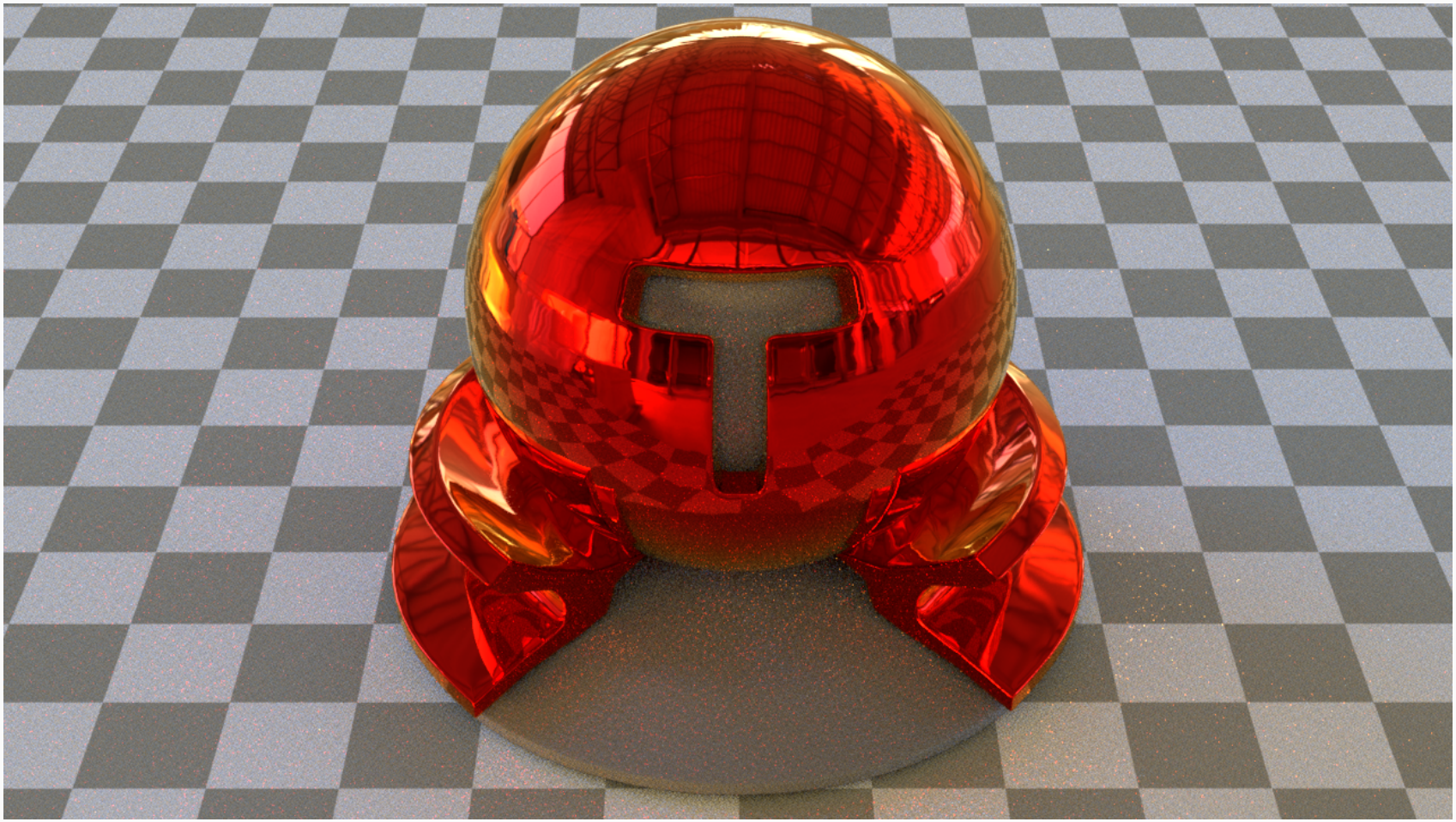
BSDF MATTE



BSDF MÉTALLIQUE

La BSDF Métallique quant à elle représente des objets conducteurs, elle ne possède donc pas de terme diffus et son spéculaire suit un Fresnel conducteur.

BSDF MÉTALLIQUE



La BSDF Matte représente un objet complètement diffus, elle ne possède donc aucun terme spéculaire et ne se modélise qu'avec un modèle Oren-Nayar.

$$s = (w_i \cdot w_o) - (n_x \cdot w_i) * (n_x / cdot w_o)$$

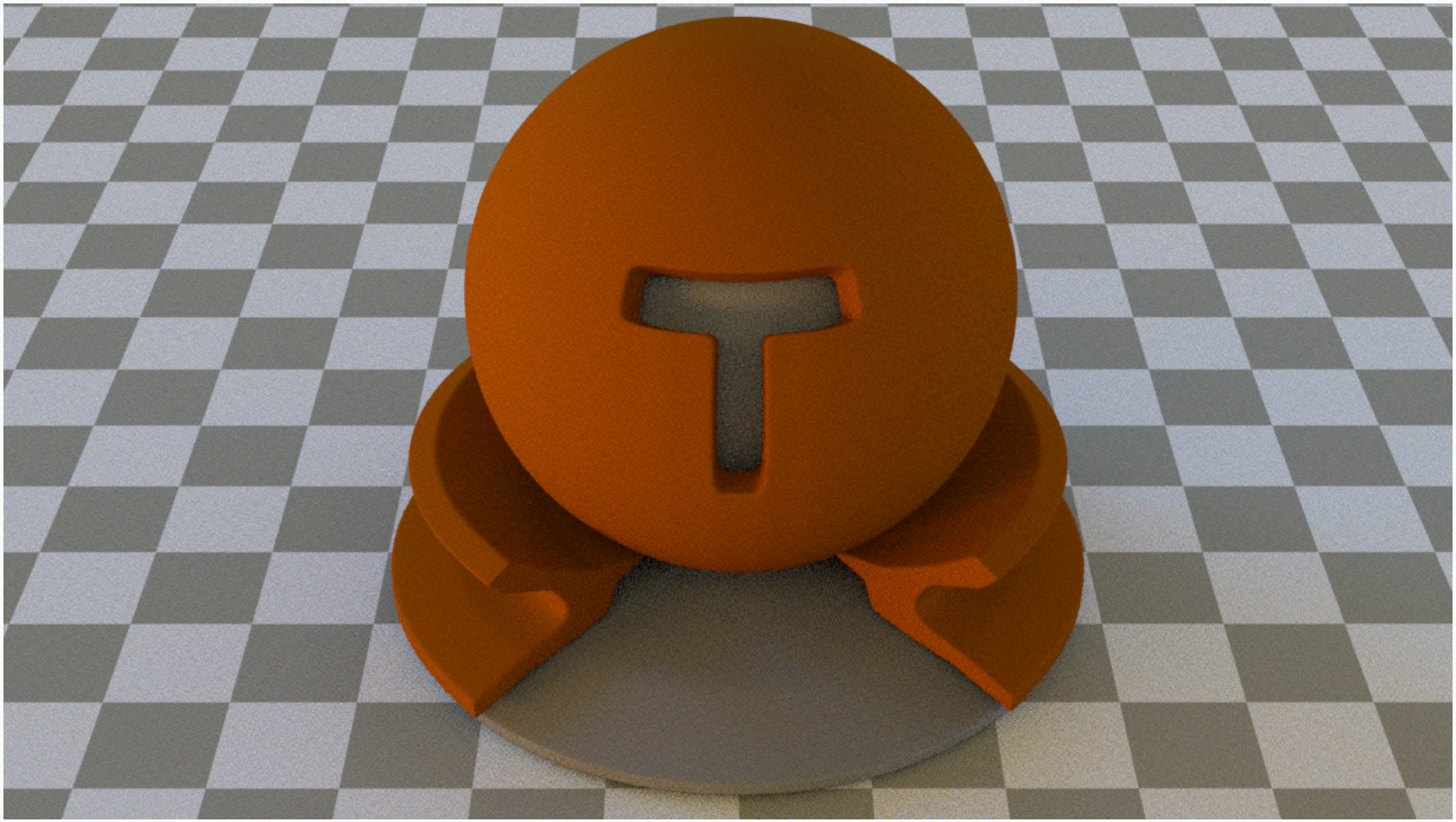
$$t = \begin{cases} 1 & si \quad s \leq 0 \\ max(n_x \cdot w_i, n_x \cdot w_o) & si \quad s > 0 \end{cases}$$

$$A = \frac{1}{\pi + (\frac{\pi}{2} - \frac{2}{3}) * \alpha}$$

$$B = \frac{\sigma'}{\pi + (\frac{\pi}{2} - \frac{2}{3}) * \alpha}$$

$$diff_{Matte}(x, w_i, w_o) = c_{d,x}(n_x \cdot w_i) * (A + B * \frac{s}{t}) \qquad (3)$$

BSDF MATTE

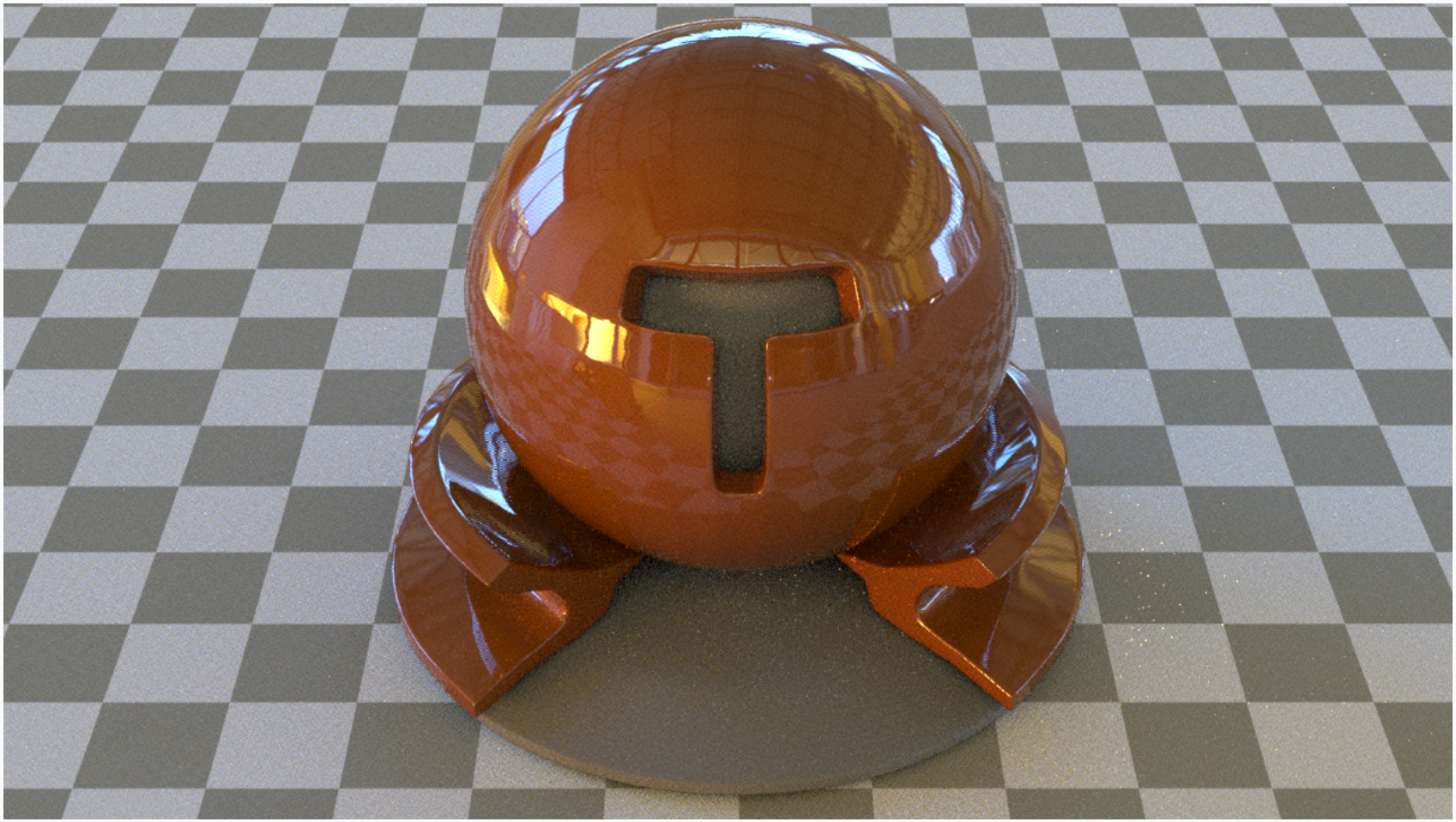


Contrairement au modèle Plastique, le modèle Substrate permet aussi de représenter des interactions lumineuses sur plusieurs couches de matériaux.

$$diff_{SubS} = \frac{28 * c_{d,x}}{23 * \pi} * (1 - c_{s,x}) * (1 - 0.5 * |w_i \cdot n_x|)^5 * (1 - 0.5 * |w_o \cdot n_x|)^5 \tag{4}$$

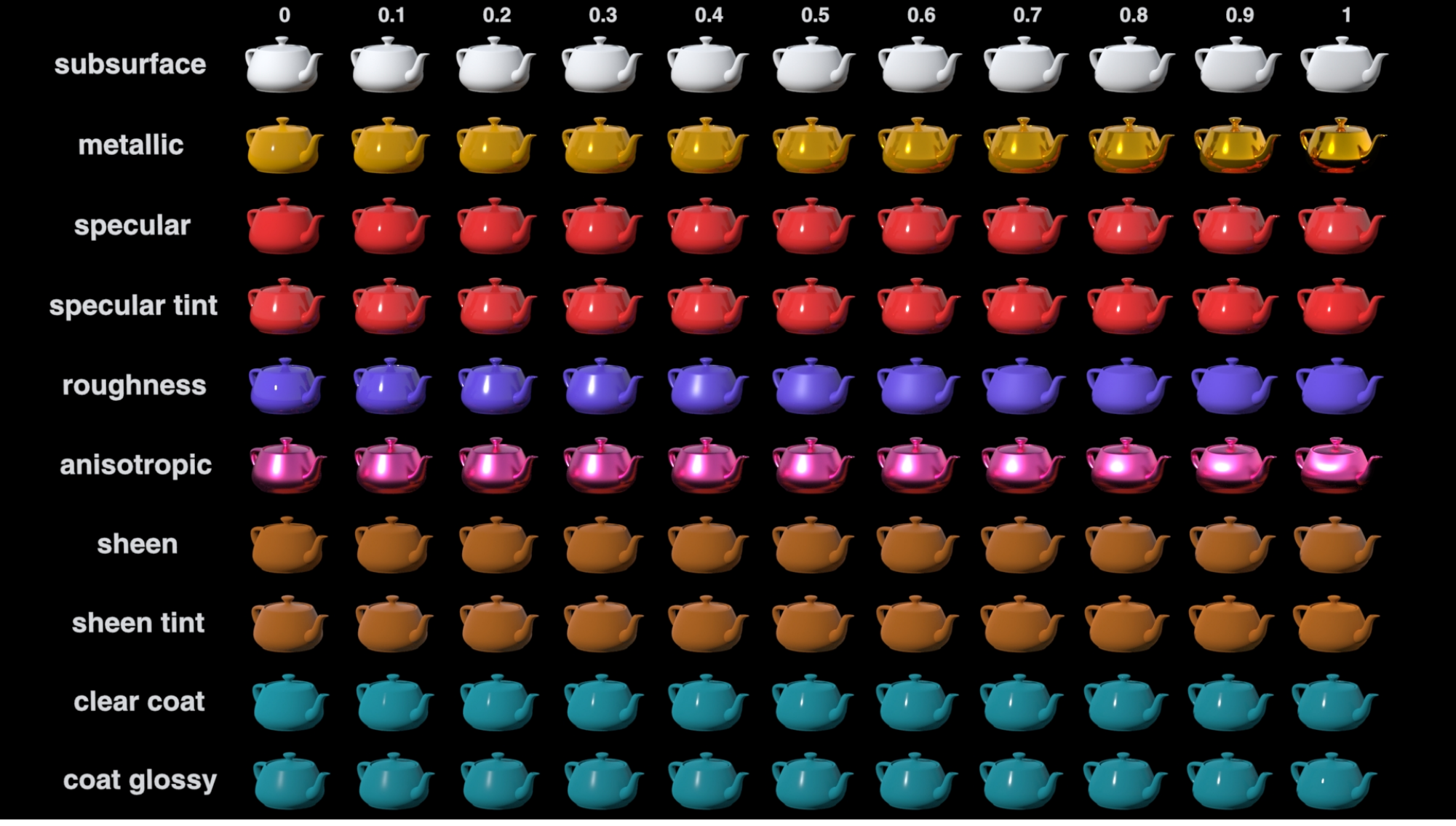
$$spec_{SubS} = \frac{D(\alpha_x, \alpha_y, w_h) * F(c_{s,x}, w_i \cdot w_h)}{4 * (w_i \cdot w_h) * \max(|w_i \cdot n_x|, |w_o \cdot n_x|)} \tag{5}$$

BSDF SUBSTRATE



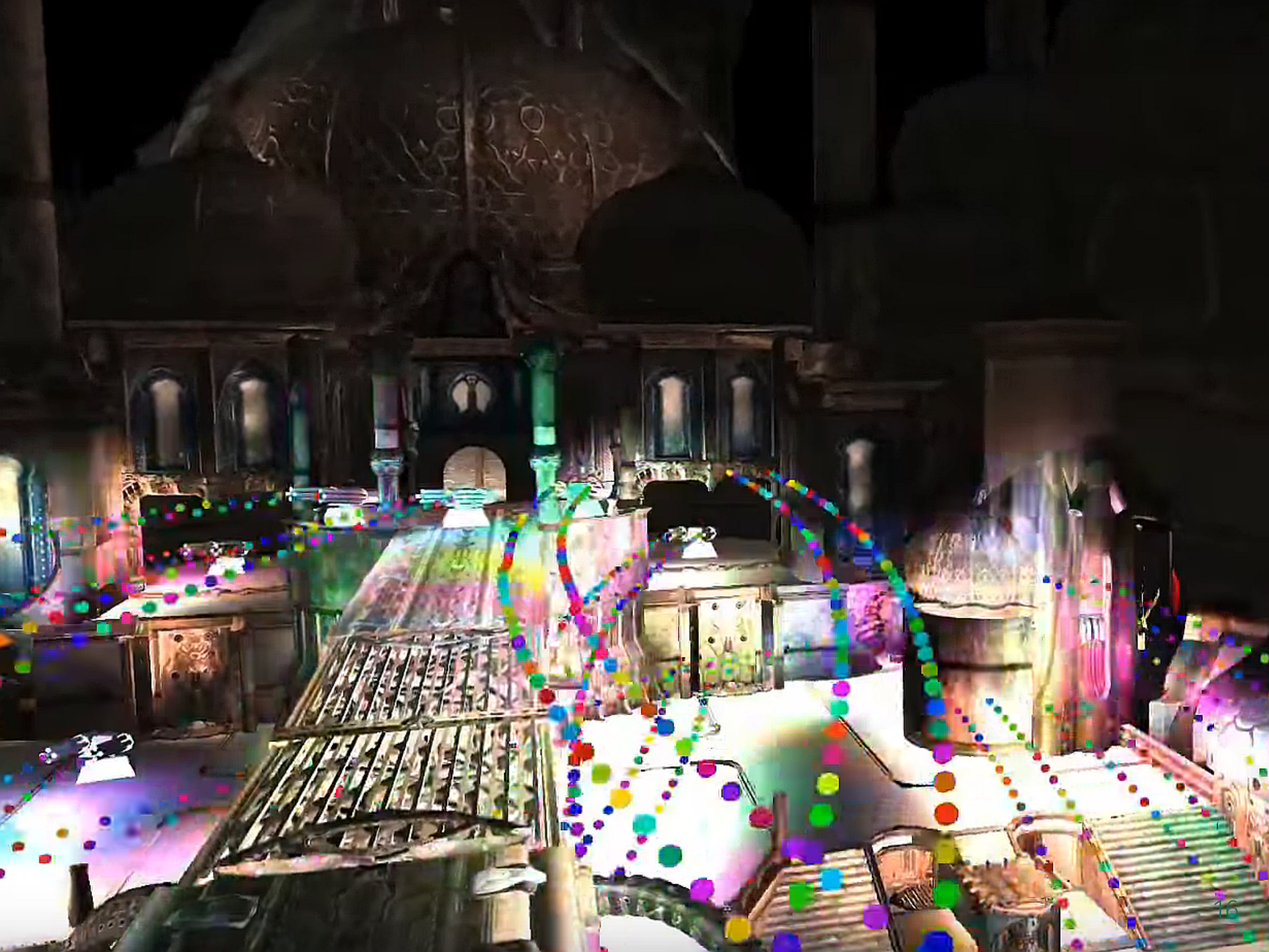
BSDF PRINCIPLED (DISNEY)

Le Principled Shader est basé sur la volonté de représenter un grand nombre de matériaux à l'aide d'une unique BSDF. Il est donc construit comme une composition de plusieurs modèle paramétré afin de couvrir une large gamme de modèles lors des rendus.



Quelques exemples avec un même modèle.

Sources ponctuelles très nombreuses



- Rendu **temps-réel** (~60fps).
- Accumulation des contributions des lumières **séquentielle** [1].

> Mais...
>
> Comment gérer beaucoup de lumières tout en gardant le caractère temps réel ?

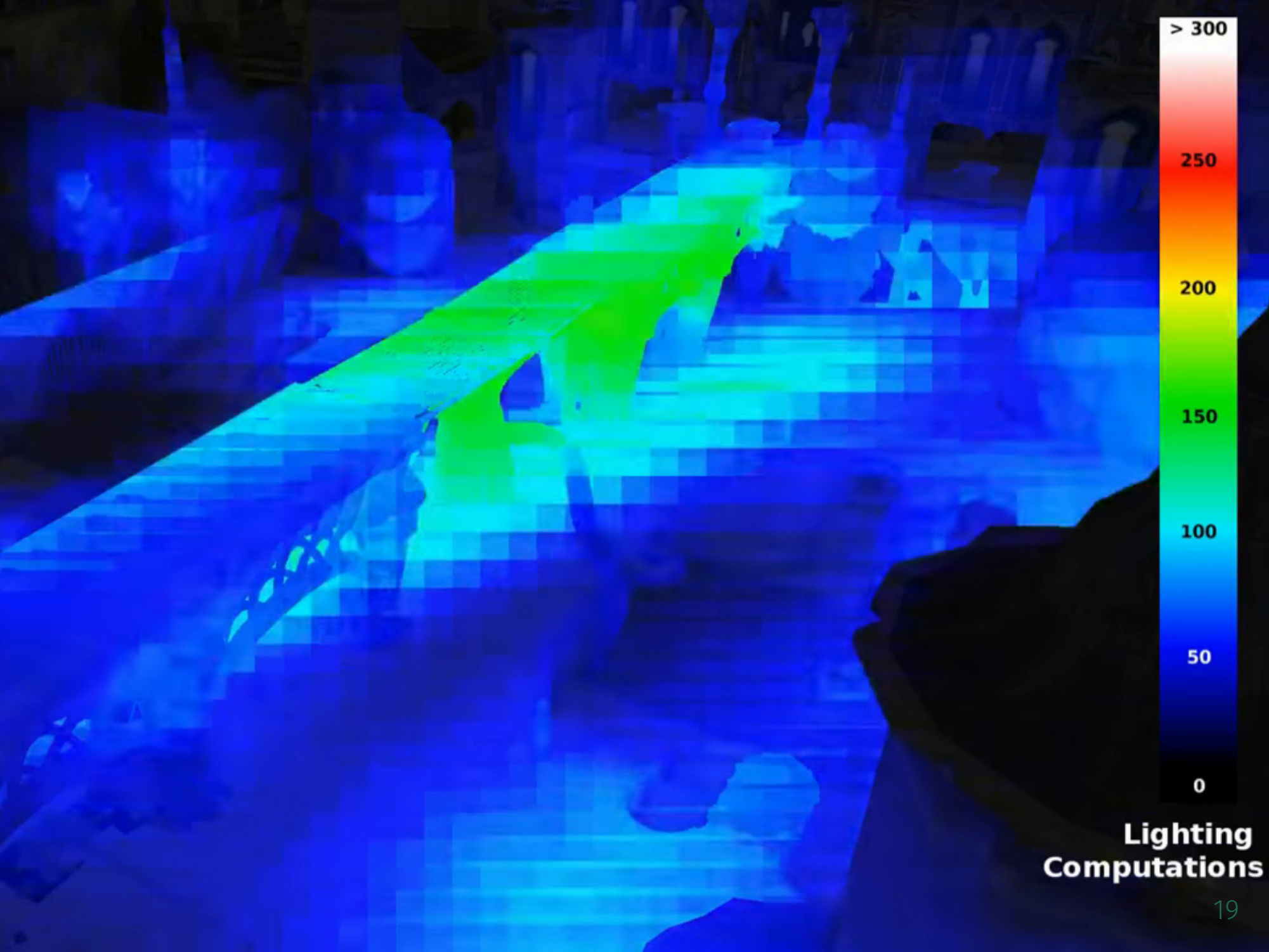
1. en parallèle pour chaque pixel, tout de même

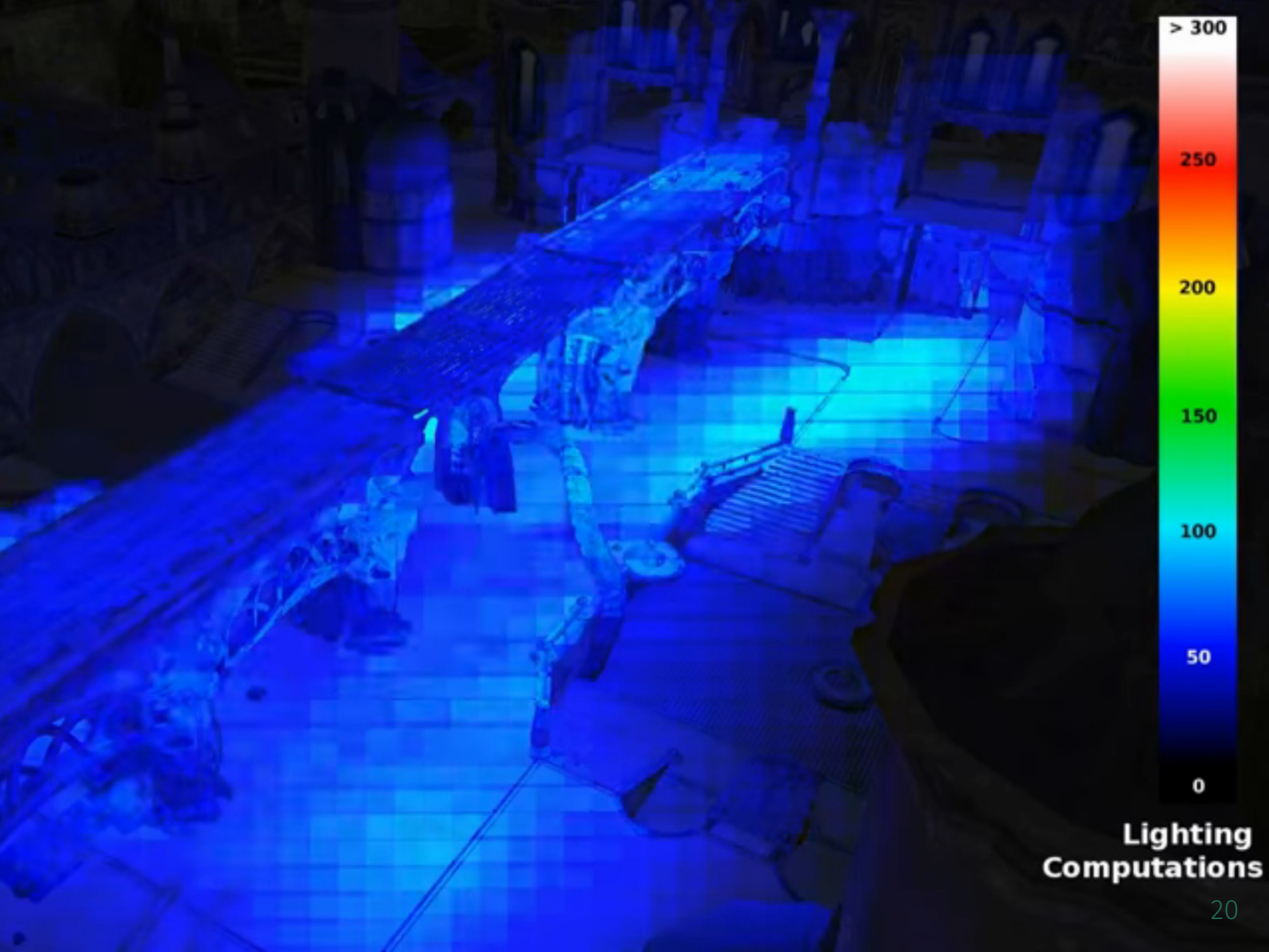
Réduire le nombre de lumières à regarder pour chaque fragment.

- Tiled shading
- Clustered shading

Réduire le nombre de lumières à regarder pour chaque fragment.

- Tiled shading
- Clustered shading


> 300
250
200
150
100
50
0
Lighting
Computations


> 300
250
200
150
100
50
0
Lighting
Computations

- Regroupement des fragments selon un ensemble de critères.
- Réduit considérablement le nombre de sources par fragment.
- Plusieurs types de critères :
  - Position (dans quel espace ?)
  - Normales

- Regroupement des fragments selon un ensemble de critères.
- Réduit considérablement le nombre de sources par fragment.
- Plusieurs types de critères :
  - Position (dans quel espace ?)
  - Normales

BOUCLE DE RENDU

```
0  Z Prepass (Zmin ET Zmax)
1  Clustering (compute shader)
2  for each object: // Direct
3      for each fragment:
4          Discard if Zfail // Attention au depth figthing
5          for each light in polygons:
6              Accumulate direct lights
7  for each object: // Indirect
8      for each fragment:
9          Discard if Zfail // Attention au depth figthing
10         for each light in the fragment many-lights cluster:
11             Accumulate indirect lights
```

Sources polygonales et linéaires

Calculer l'éclairage produit par une source polygonale revient à intégrer une fonction d'éclairage dans la projection de ce polygone sur la sphère.

Mais...

Difficile en temps réel pour des formes particulières.
Quasiment impossible pour un polygone quelconque.

APPROXIMATION EN TANT QUE SOLUTION

- Fonction d'éclairage inconnue (ou compliquée).
- Approximation simple à l'aide d'une fonction de base transformable.

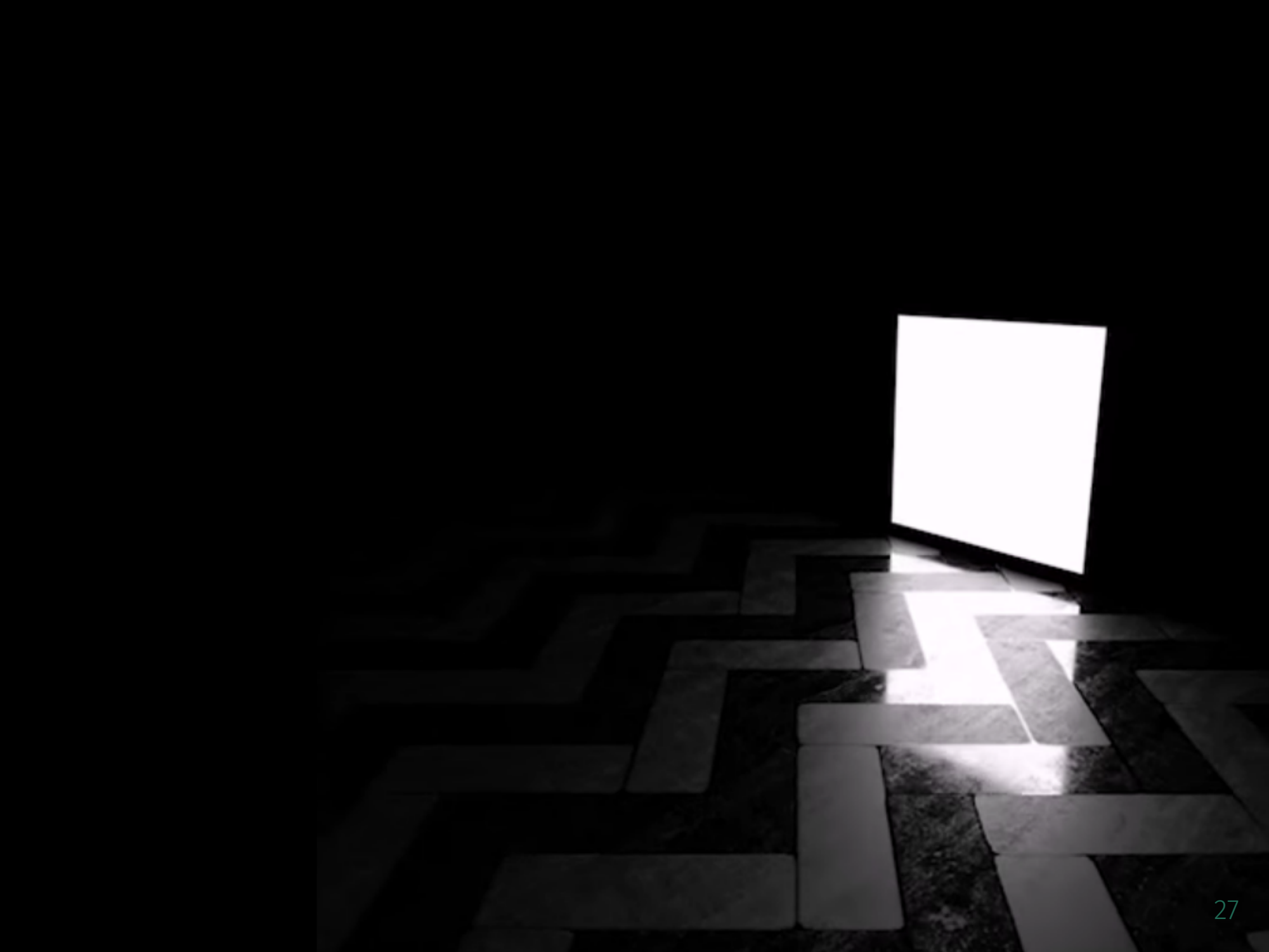
2. Linearly Transformed Spherical Distribution

- Fonction d'éclairage inconnue (ou compliquée).
- Approximation simple à l'aide d'une fonction de base transformable.

LTSDs [2]

Famille de fonctions aux propriétés très pratiques.

2. Linearly Transformed Spherical Distribution



DISTRIBUTIONS SPHÉRIQUES

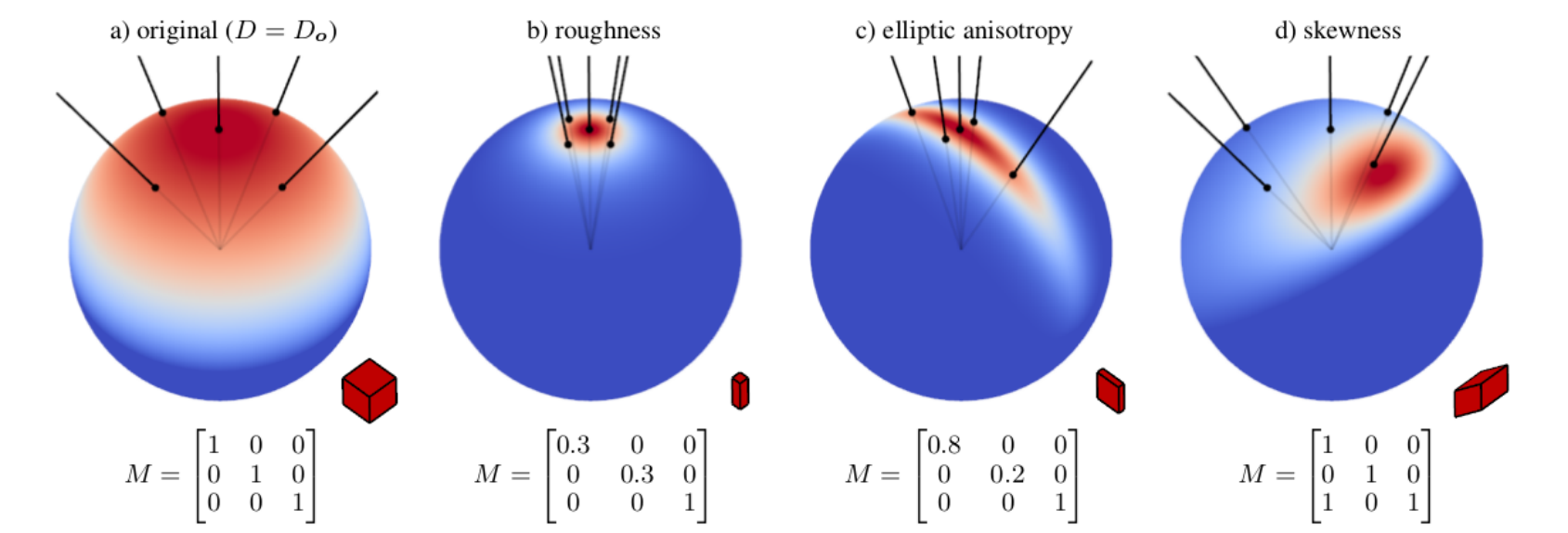


FIGURE 1 – La fonction de base D_o (gauche) et ses transformations, permettant de simuler des BRDF de type *GGX*

UNE PROPRIÉTÉ SALVATRICE

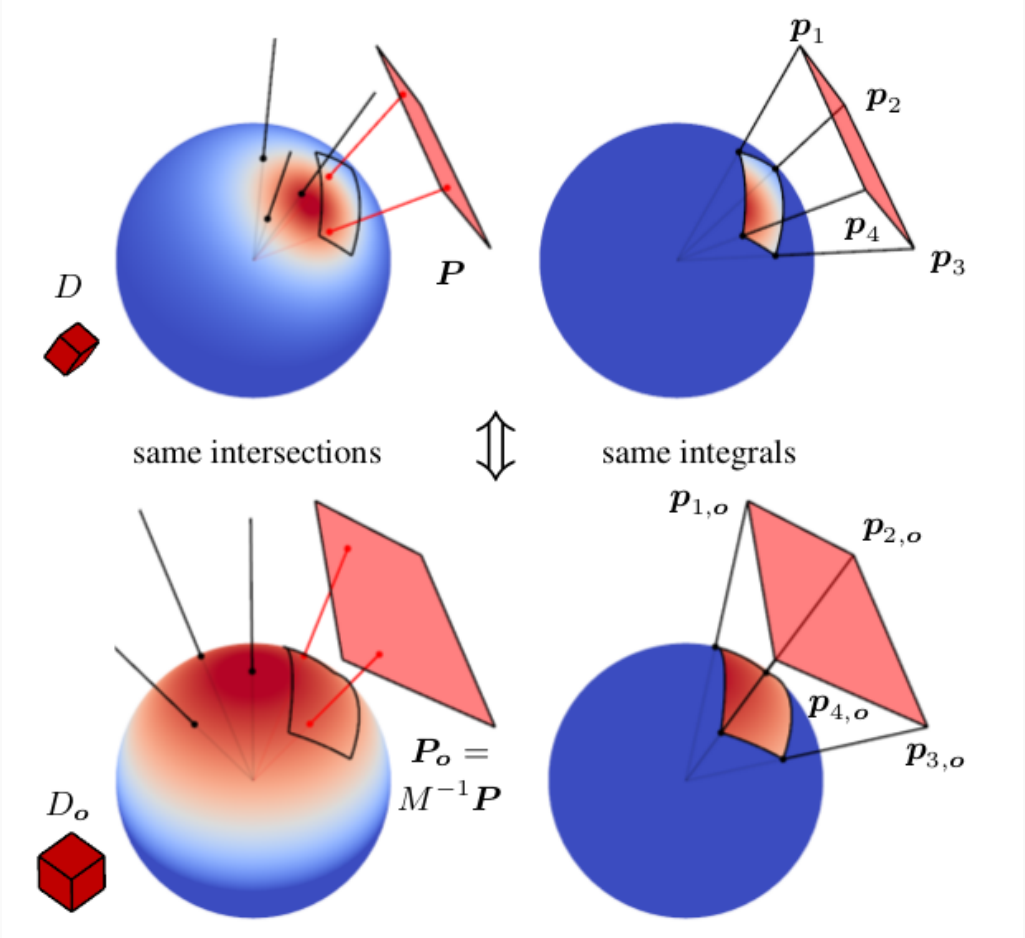


FIGURE 2 – Des intégrales identiques !

Pour une luminosité constante, on peut obtenir une expression analytique à l'aide du théorème de Stokes[3] :

$$\int_{P_o} D_o(\omega_o) d\omega_o = \frac{1}{2\pi} \sum_{i=1}^{n} acos(\langle p_i, p_j \rangle) \left\langle \frac{p_i \times p_j}{||p_i \times p_j||}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\rangle \quad (6)$$

où :

- $j = (i+1)\%n$, pour créer des couples de points successifs
- $\langle a, b \rangle$ est le produit scalaire

3. Baum et al., 1989, *Improving radiosity solutions through the use of analytically determined form-factors*

$$M = \begin{bmatrix} a & 0 & b \\ 0 & c & 0 \\ d & 0 & 1 \end{bmatrix} \tag{7}$$

- M^{-1}
- 4 éléments + la norme pour un (θ, α) [4]
- Précalcul dans des tables [5] 64×64

4. Avec θ l'angle incident et α la rugosité de la BRDF
5. Textures

PLAN DE DÉVELOPPEMENT

Étapes clés

1. Analyse
2. Prototypage
3. Développement
   - Code
   - Site
4. Recette

Étapes clés

1. Analyse
2. Prototypage
3. Développement
   - Code
   - Site
4. Recette

Jalons

1. Algorithmes identifiés
2. Conception et spécifications
3. Renderer de base
4. Clustering
5. Matériaux
6. Lumières polygonales
7. Many-lights

Radium Engine

L'existant

- Gestion avancée des shaders
- Caméras, séléction d'objets
- FrameBuffers
- Gestion des objets
- Interface graphique (Qt)
- Librairies : OpenGL, Eigen, Qt 5

Radium Engine

À développer

- Matériaux avancés : paramètres
- Éclairage direct/indirect
- Clustering
- Librairies : aucune de plus

Risque principal

Risque important si les clusters ne fonctionnent pas.

- Pas de sources pounctuelles indirectes
- Des sources polygonales, mais en non clusterisées

Communication Discord, emails

Versionnage GitHub (fork du dépôt upstream)

Tests Scènes de tests en `.obj`, et données calculées avec PBRT v3

Documentation LaTeX

Questions?