

Programmation Graphique Haute Performance

2016

Pierre.Benard@labri.fr
Gael.Guennebaud@inria.fr

Organisation de l'UE

Site web :

<http://www.labri.fr/perso/pbenard/teaching/pghp/>

TD machine

- en monôme
- code à rendre avant la séance suivante

Évaluation

- examen écrit (2/3)
- contrôle continu (1/3)

Jeux vidéo



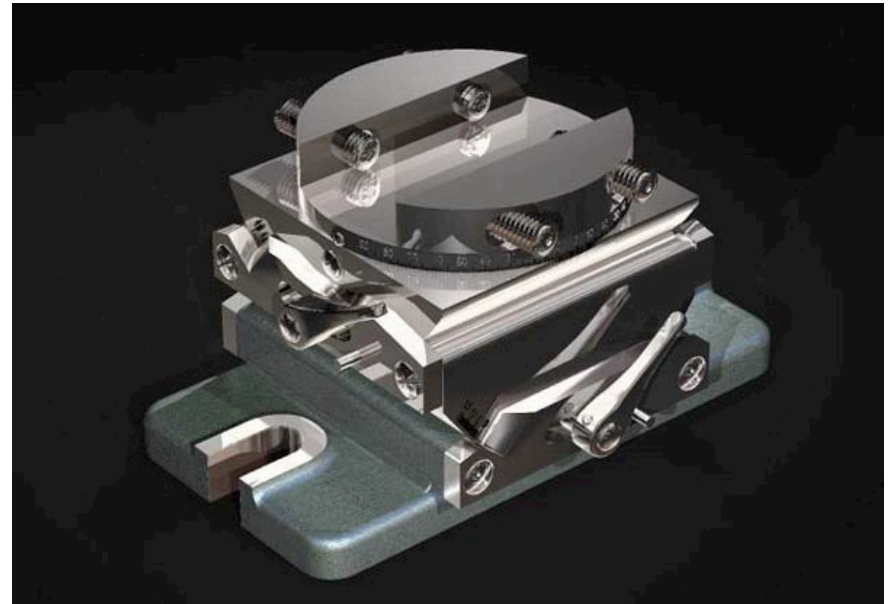
Films – Publicité



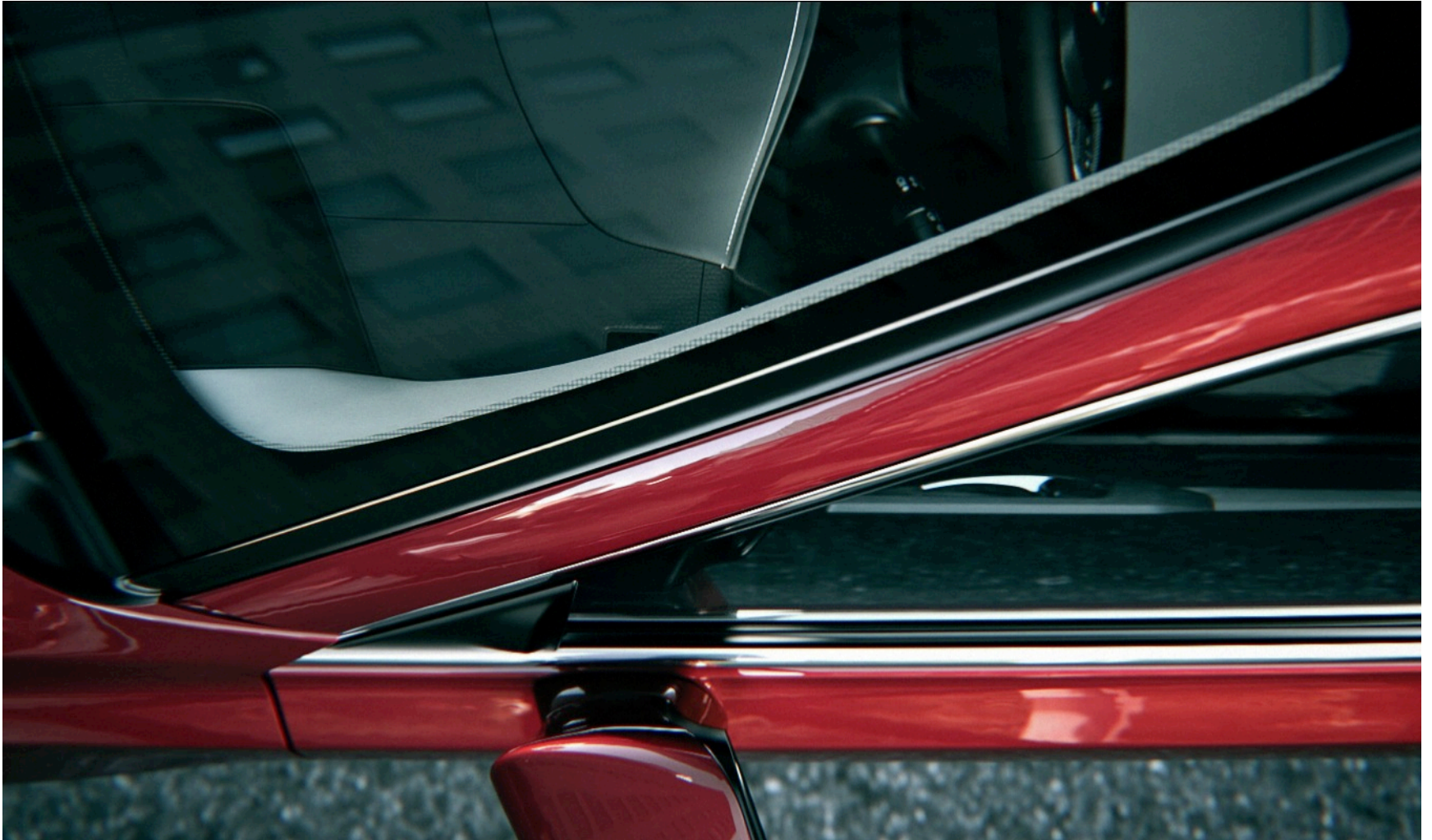
Simulation



CAD - Design



CAD - Design



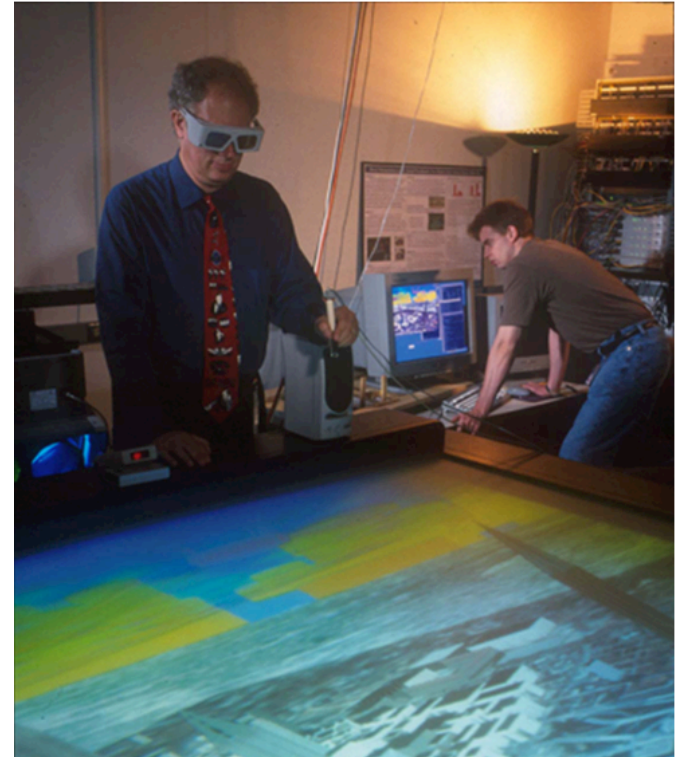
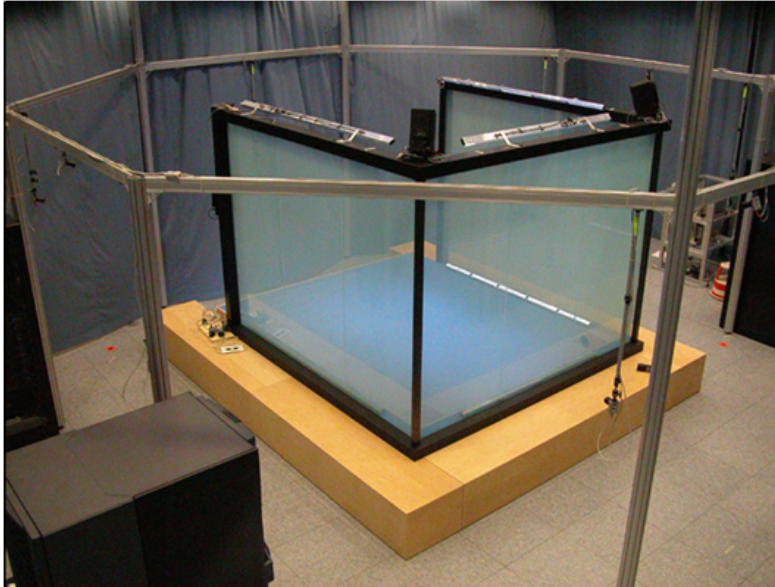
Architecture



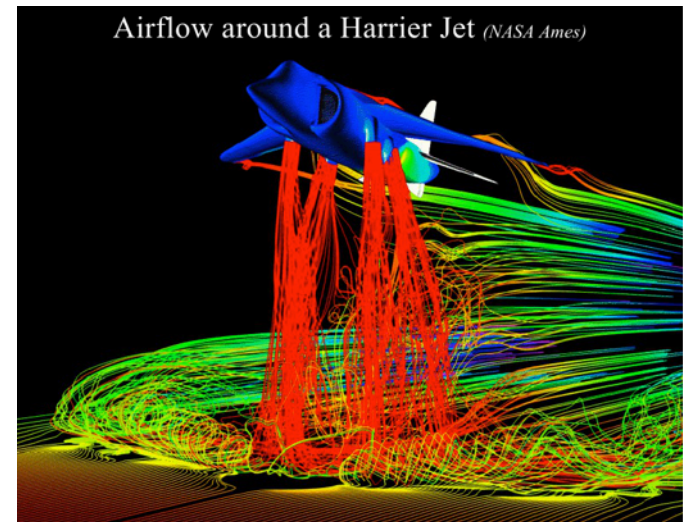
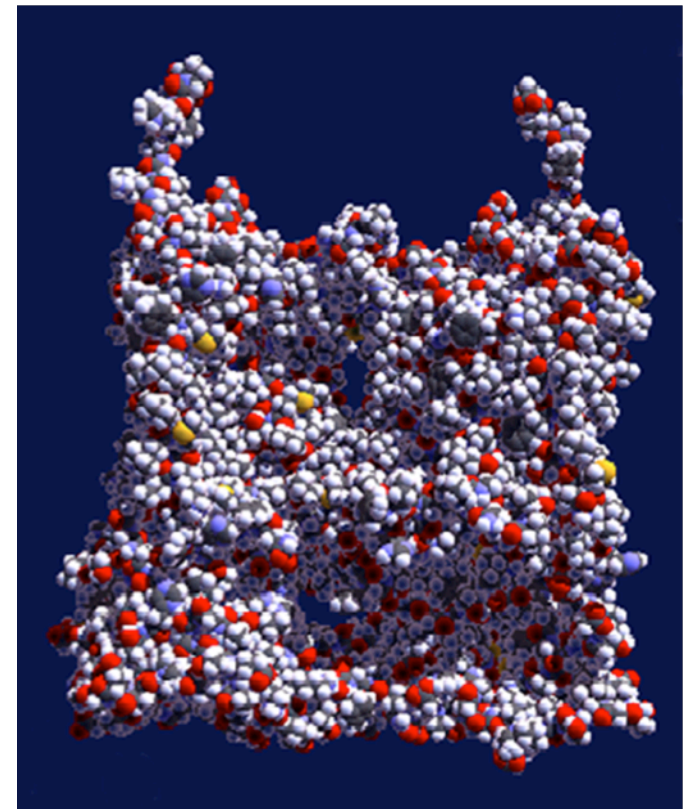
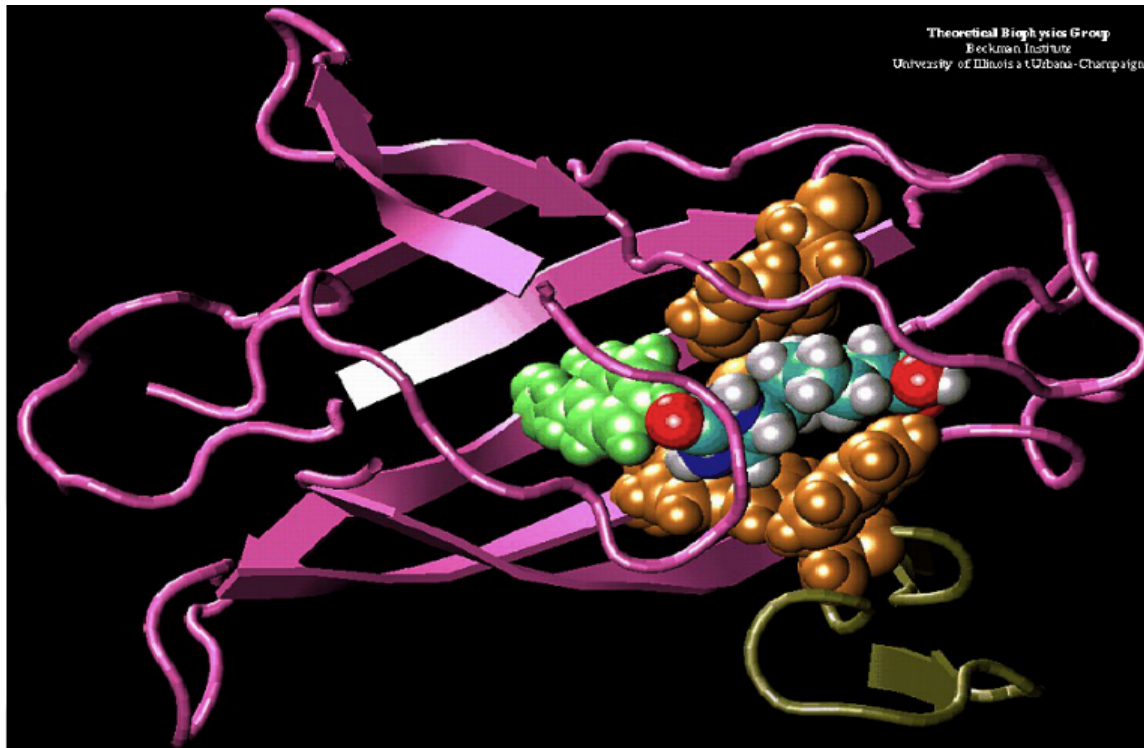
Architecture



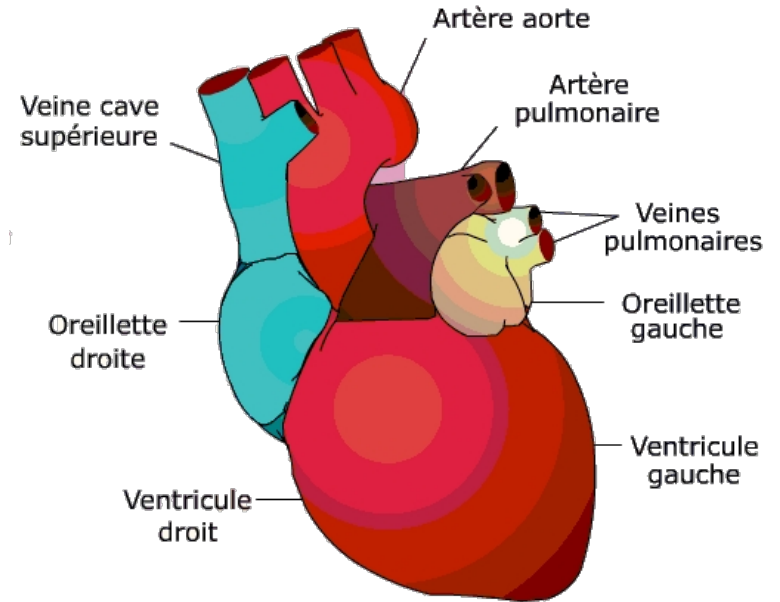
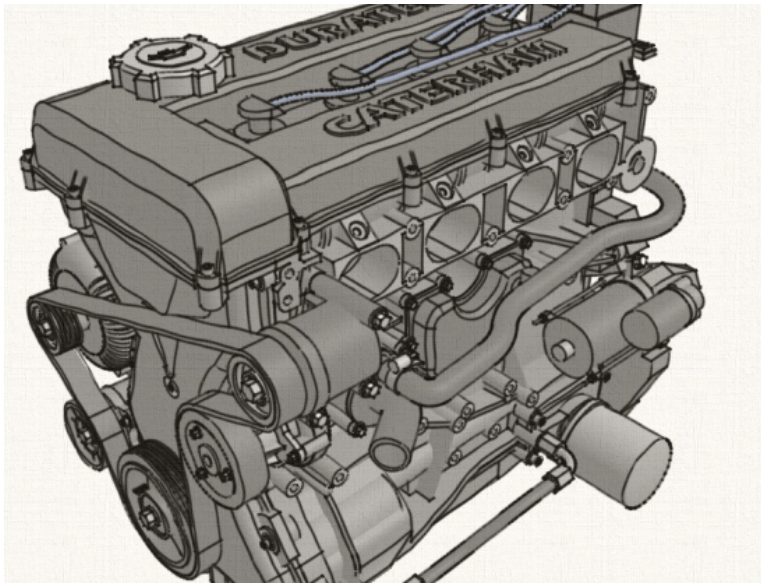
Réalité virtuelle



Visualisation scientifique



Rendu expressif / illustratif



3 classes d'algorithmes

Rendu temps réel (jeux, simulateur, modeleur)

- 30 images/secondes
- représentations légères, beaucoup d'« astuces »

OpenGL
DirectX
Consoles

Rendu en temps maîtrisé (SFX, films d'anim.)

- quelques minutes par images
- haute qualité visuel en un temps raisonnable

Pipeline maison
(+ OpenGL)

Rendu physiquement réaliste

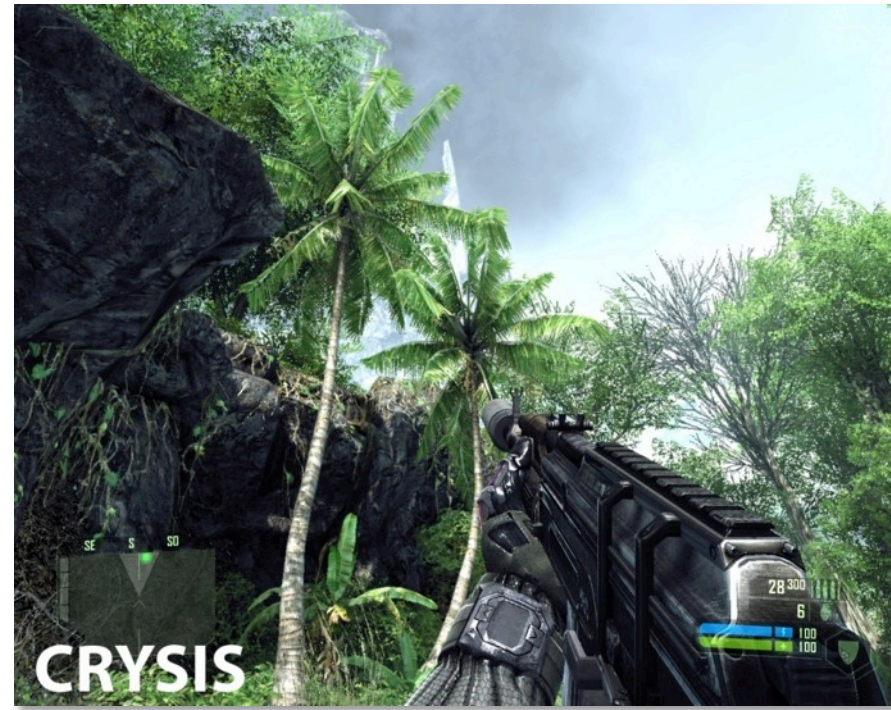
(conception, biologie, etc.)

- de quelques heures à quelques jours par images
- maîtrise de l'erreur

Pipeline maison

Question

Que faut-il faire pour créer un film d'animation ou un jeu vidéo ?

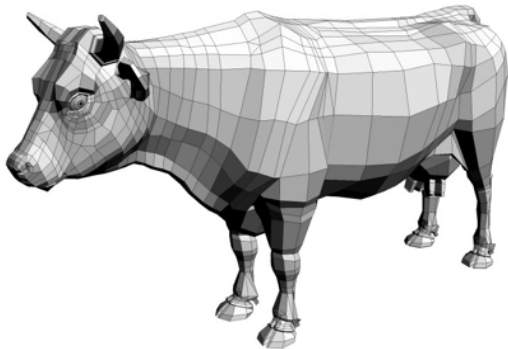


Problèmes à résoudre

Monde réel



Modélisation

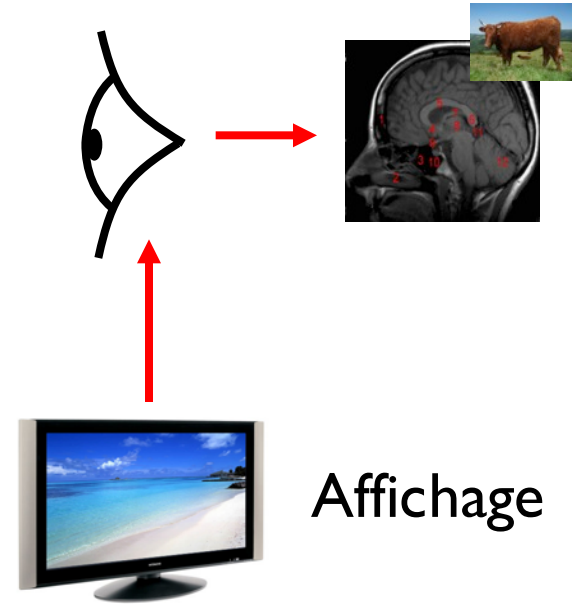


Représentation
numérique

Animation



Synthèse



Affichage

Problèmes à résoudre

Modélisation

des objets, d'une scène

Synthèse d'une image

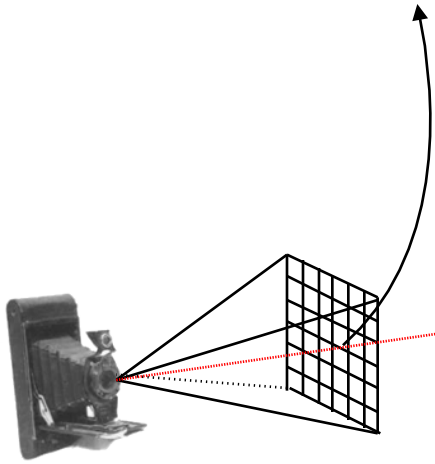
à partir des objets, matières, éclairages, caméras...

Animation

spécifier ou calculer des mouvements
et des déformations

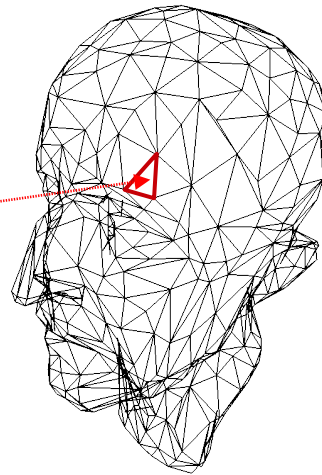
Synthèse d'image

problématique : quelle est la couleur de chacun des pixels ?



caméra virtuelle

- tableau de pixels RGB
- position, orientation
- résolution
- etc.



modèles 3D

e.g. : « soupe de triangles »

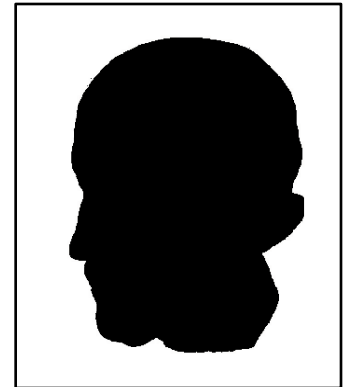
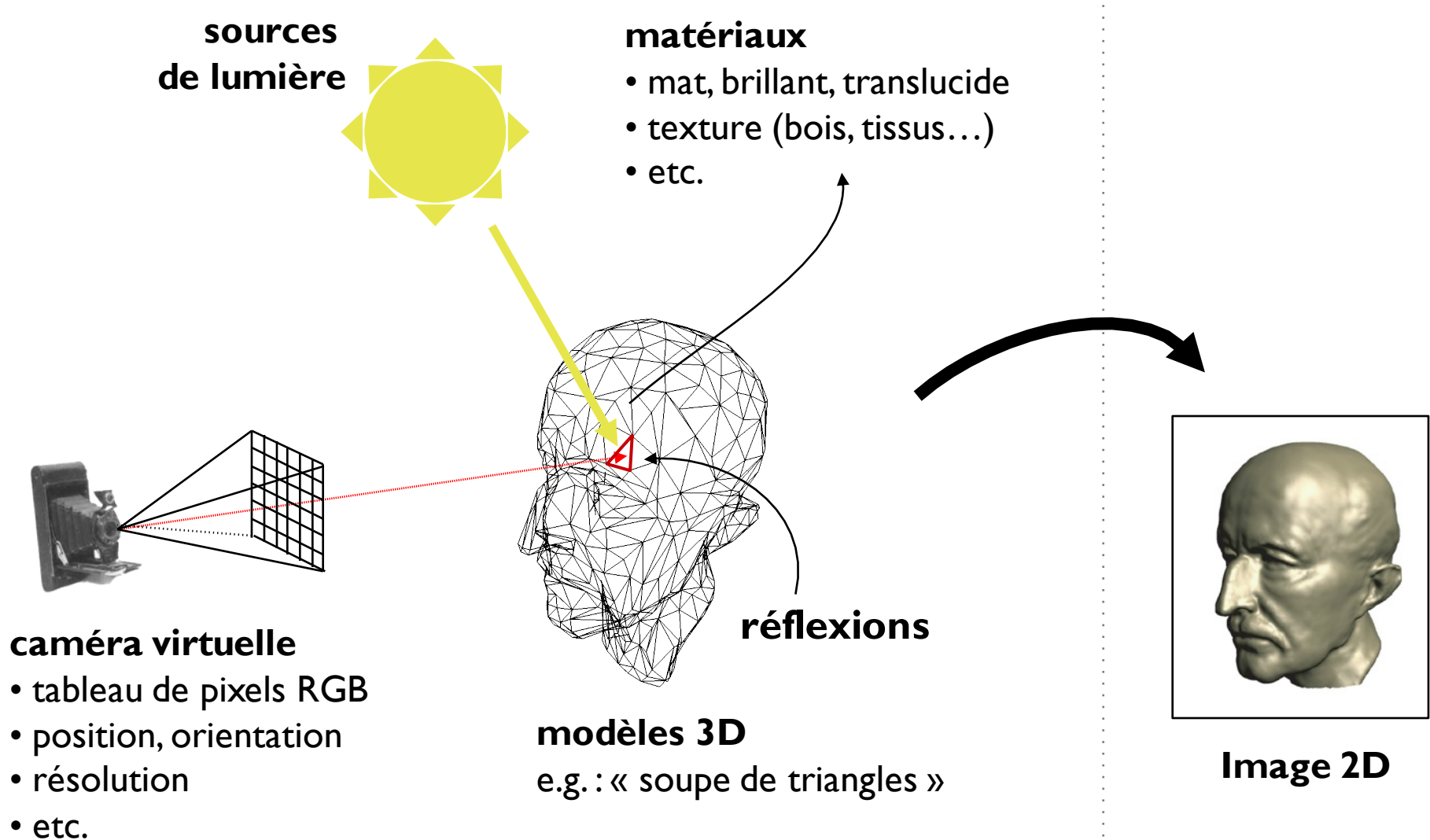


Image 2D

Synthèse d'image



Synthèse d'image

=

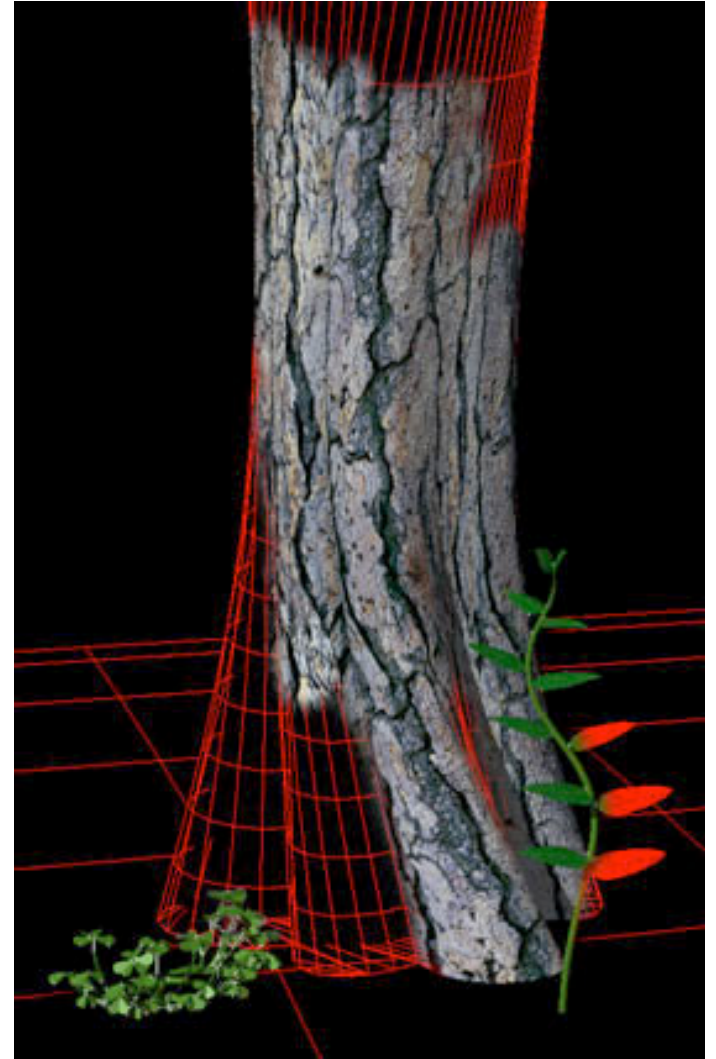
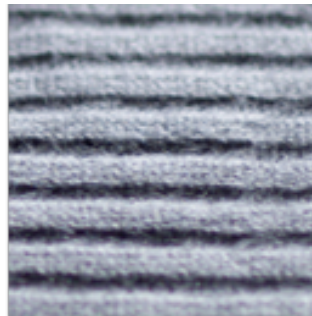
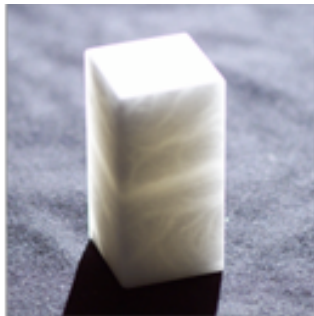
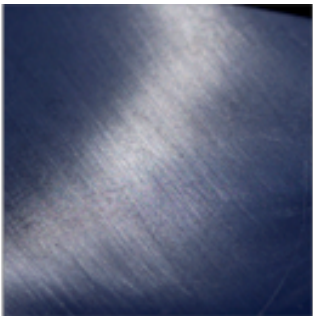
Simulation des interactions entre
la lumière, la matière, les formes,
et le milieu ambiant

Apparence, matériaux

Textures : couleur, effets de formes (normales, champ de hauteur, ...), environnement lumineux

Fonctions de réflectance :
BRDF, SVBRDF, BTDF, BTF...

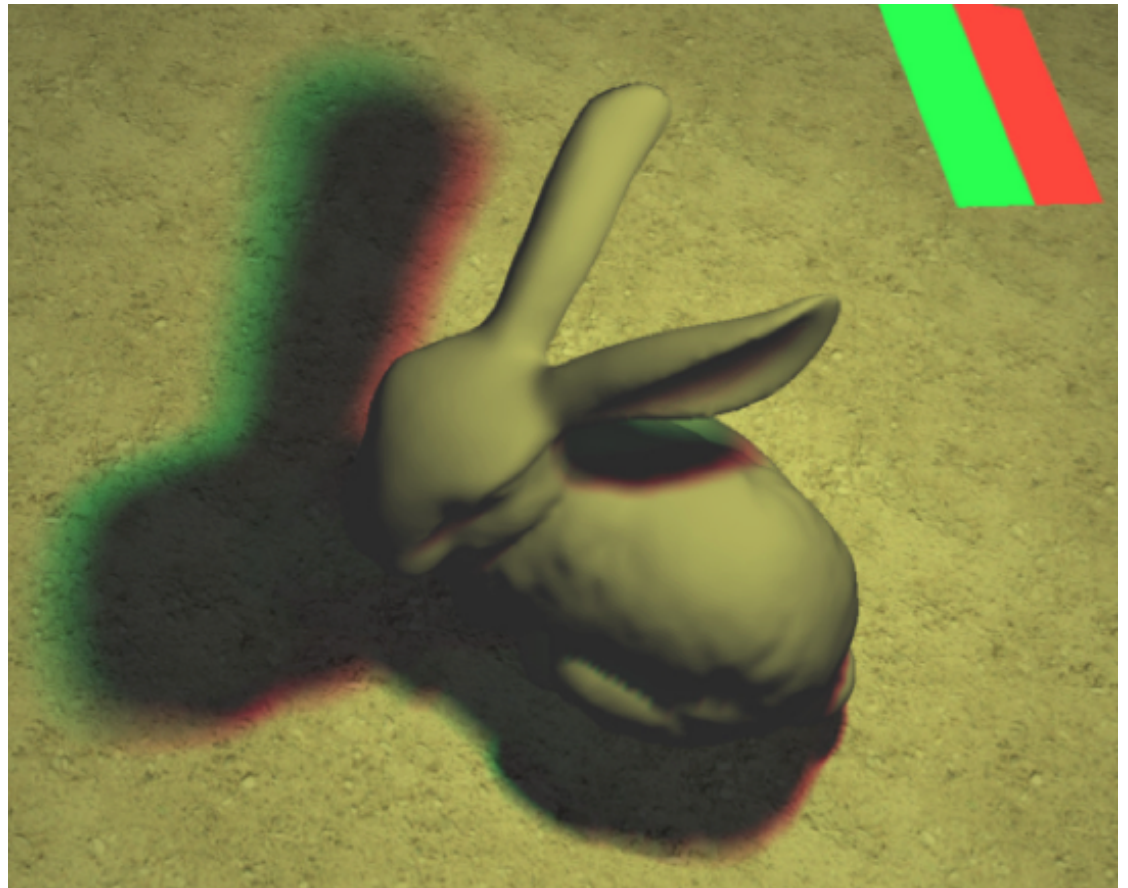
⇒ master 2 (SIA)



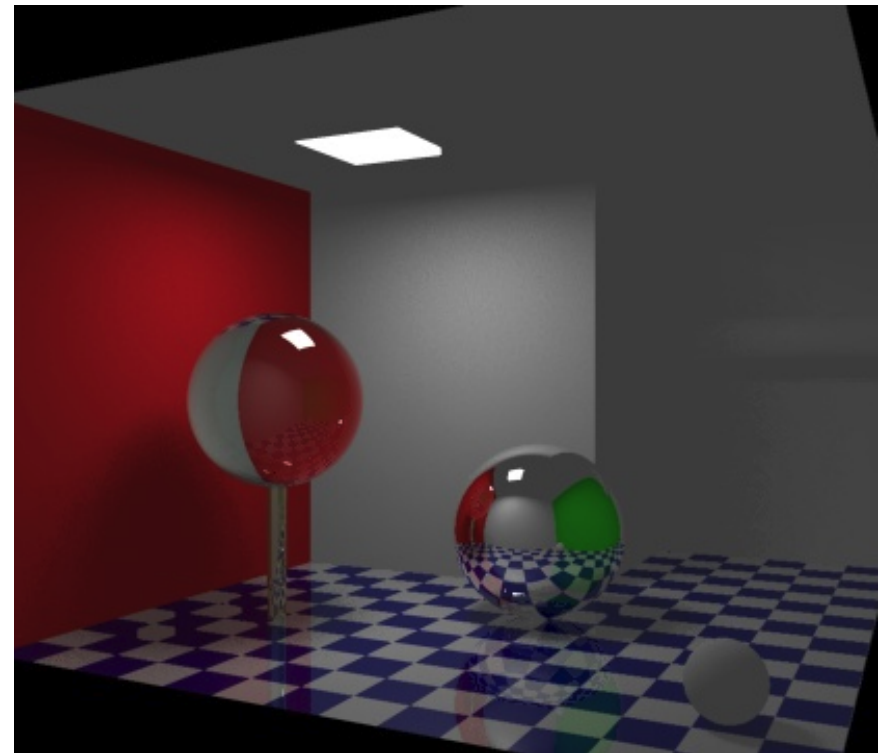
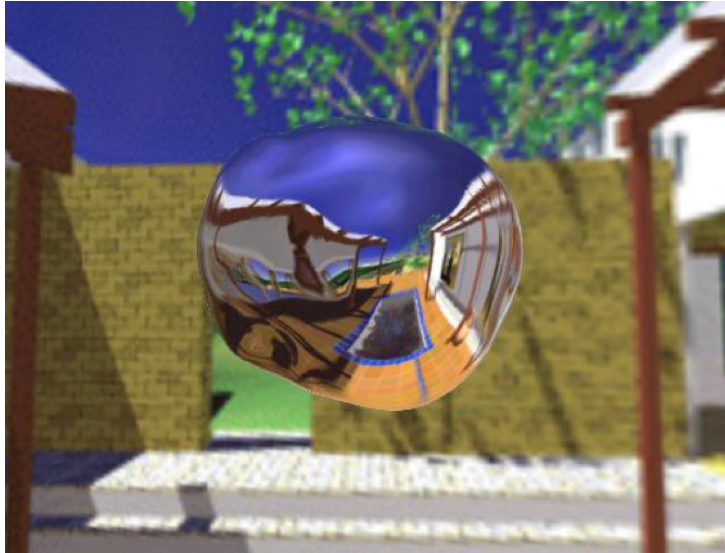
Ombres et lumière

Qui est éclairé par une source?

- visibilité
- géométrie



Réflexions miroirs / réfractions



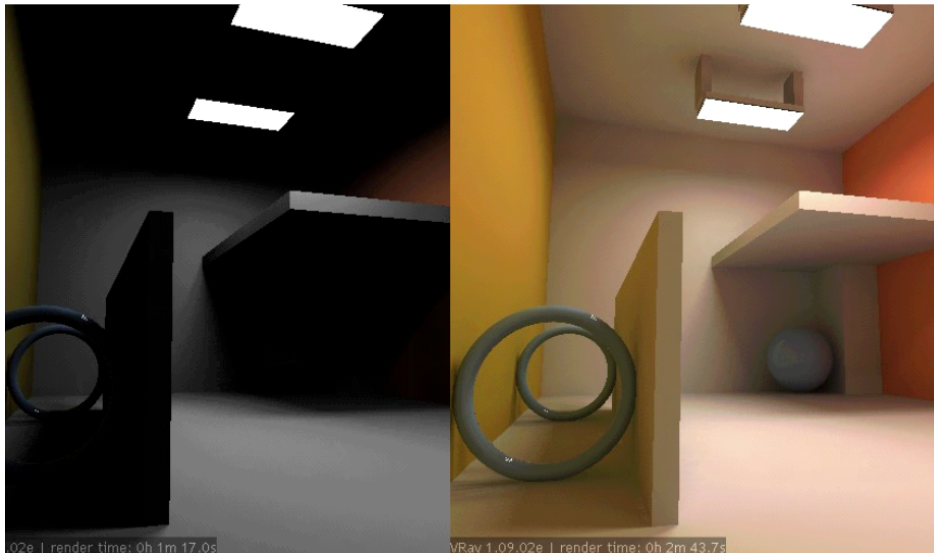
Rebonds de la lumière

Inter-réflexions

Sources étendues

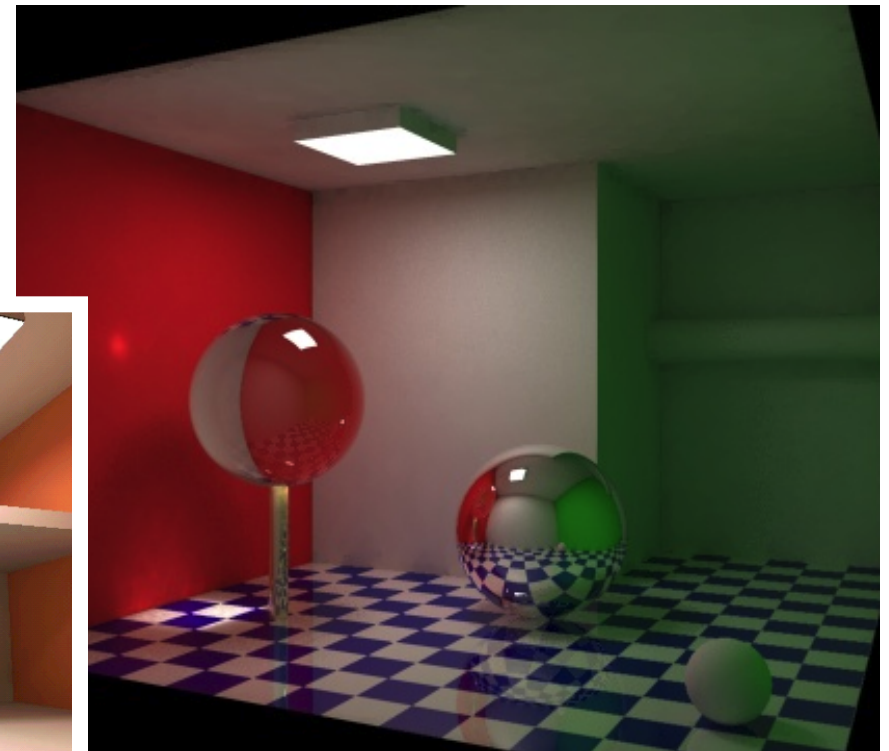
Illumination globale

⇒ **master 2 (SIA)**

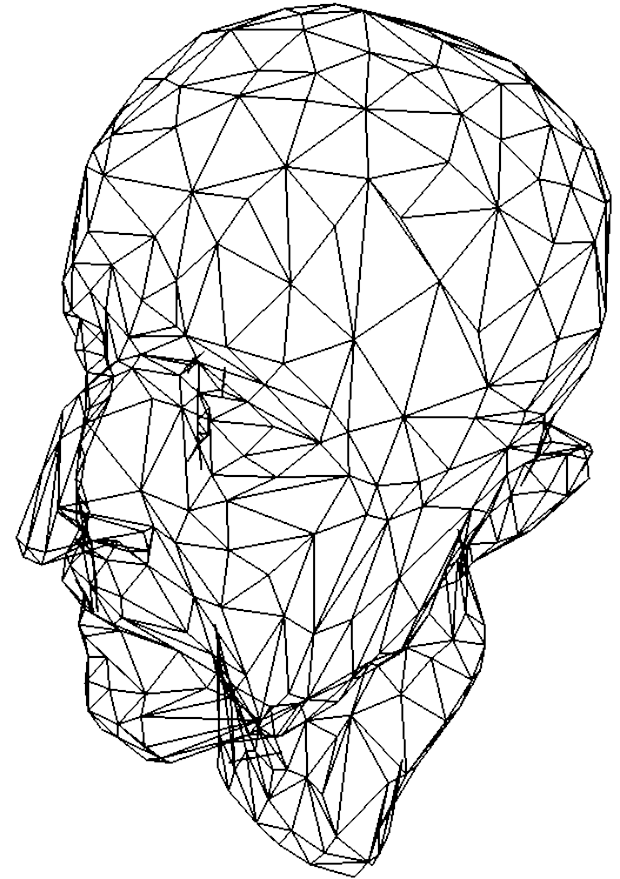
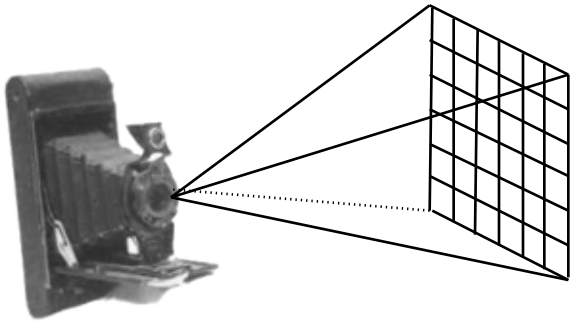


Direct

+ Indirect



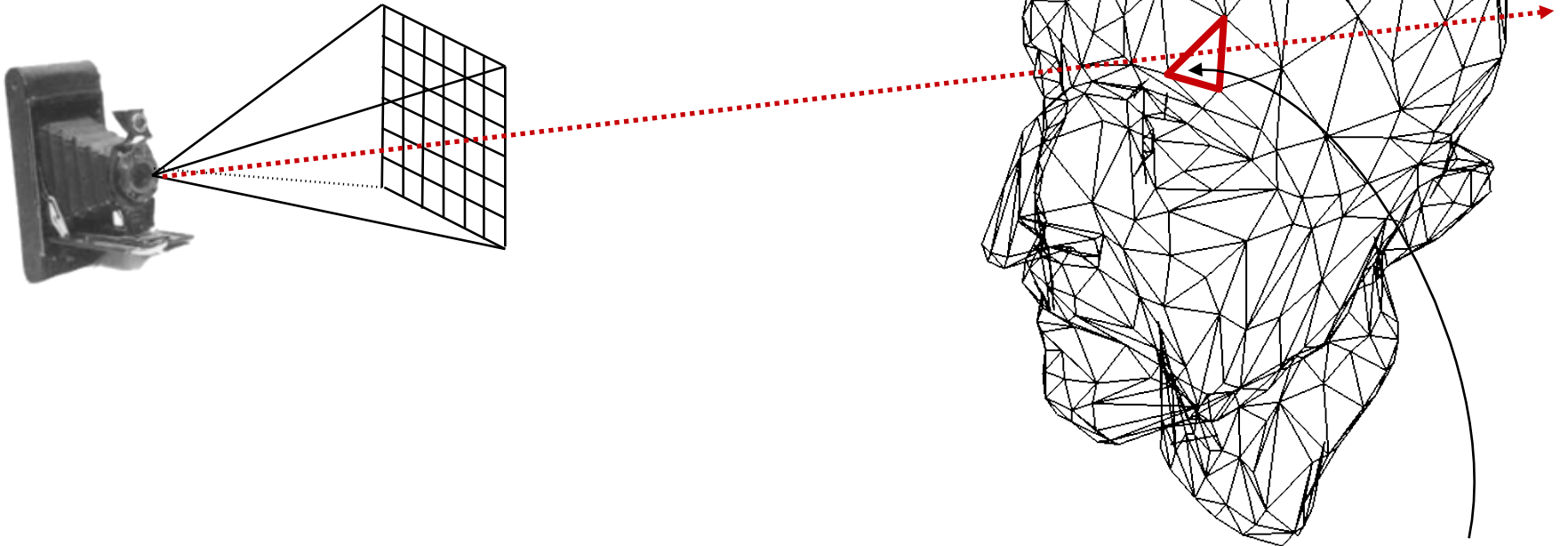
Visualiser une scène 3D



Visualiser une scène 3D

1^{ère} approche : **lancer de rayons**

1 pour chaque pixel,
lancer un rayon

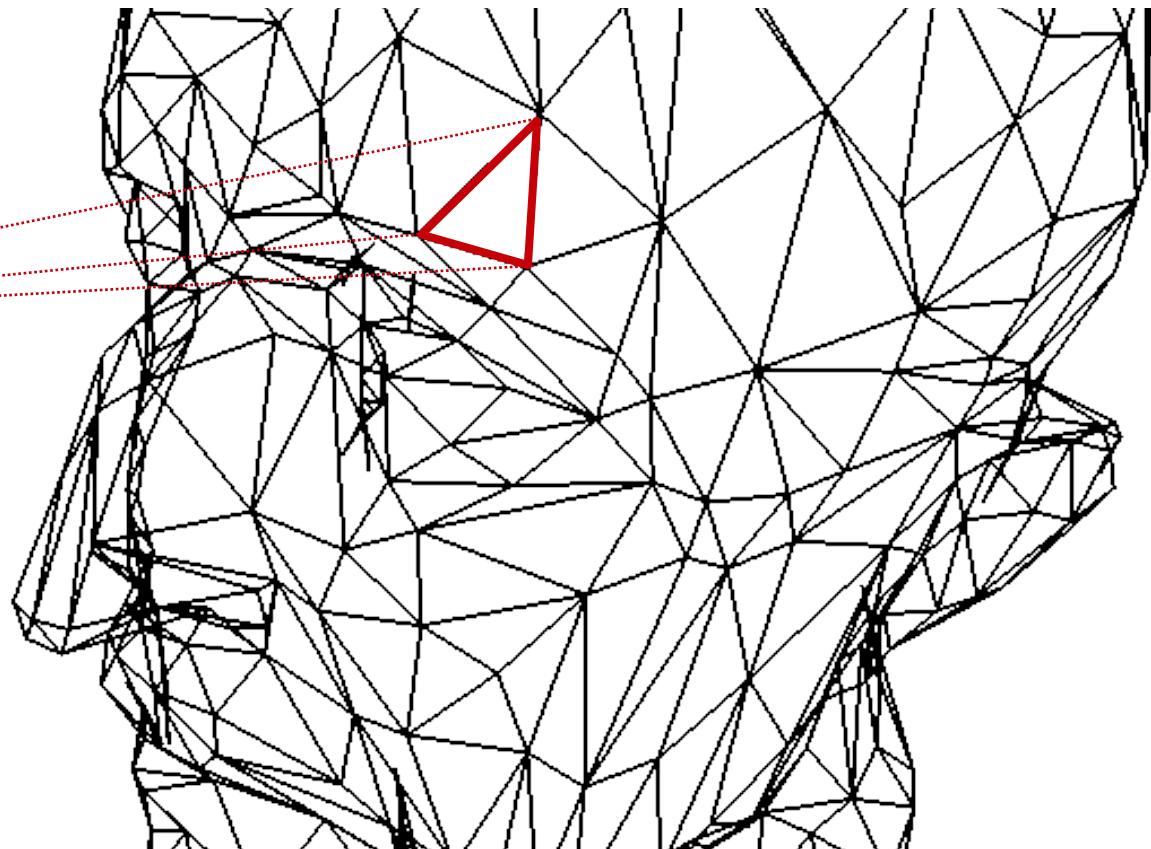
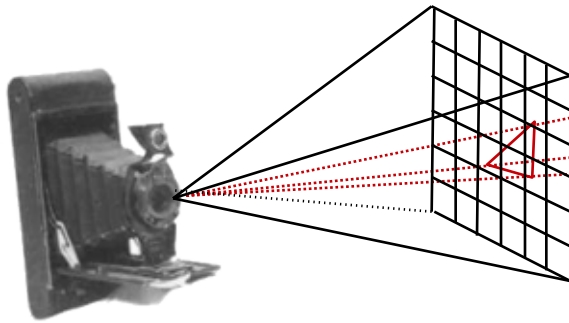


2 trouver la primitive la plus
proche intersectant le rayon

Visualiser une scène 3D

2^{ème} approche : **rastérisation**

1 pour chaque triangle, projeter les 3 sommets sur le plan image \Rightarrow triangle 2D



2 trouver les pixels correspondant

Visualiser une scène 3D

Lancer de rayon (ray tracing / ray casting)

- Pour chaque rayon, trouver la primitive la plus proche intersectant le rayon
- Propagation d'un rayon à partir des pixels
- Scène = ensemble de primitives « intersectables »

Rastérisation

- Pour chaque primitive, trouver les rayons l'intersectant
- Rendu en deux étapes :
 - projection des primitives sur l'écran (*forward projection*)
 - discrétisation (conversion des primitives 2D en pixels)
- Scène = ensemble de primitives « rastérisables »