

Mondes 3D

(synthèse d'images)

pierre.benard@labri.fr
gael.guennebaud@inria.fr

<http://www.labri.fr/perso/pbenard/teaching/mondes3d>

Organisation de l'UE

- **Site web :**
 - <http://www.labri.fr/perso/pbenard/teaching/mondes3d/>
- **TD machine**
 - en monôme
 - code à rendre avant la séance suivante
 - séances de **3h** : mini cours (10-20mn) + pratique
 - soyez à l'heure !!
- **Évaluation**
 - examen écrit (2/3)
 - contrôle continu (1/3)

Programme

- **Initiation au rendu 3D / synthèse d'images**
 - Comment produire une image (=tableau de pixel) à partir d'une représentation numérique d'une scène 3D
 - Notions :
 - Lancer de rayon
 - Modèle d'éclairage local, normales, BRDFs
 - Rasterisation: pipeline graphique, z-buffer, OpenGL, shaders
 - Transformations de l'espace
 - Textures, normales maps, réflexions & transparence
 - Intro modélisation géométrique
 - Maillage polygonal, surfaces de subdivisions
 - Intro animation
 - etc.

Jeux vidéos



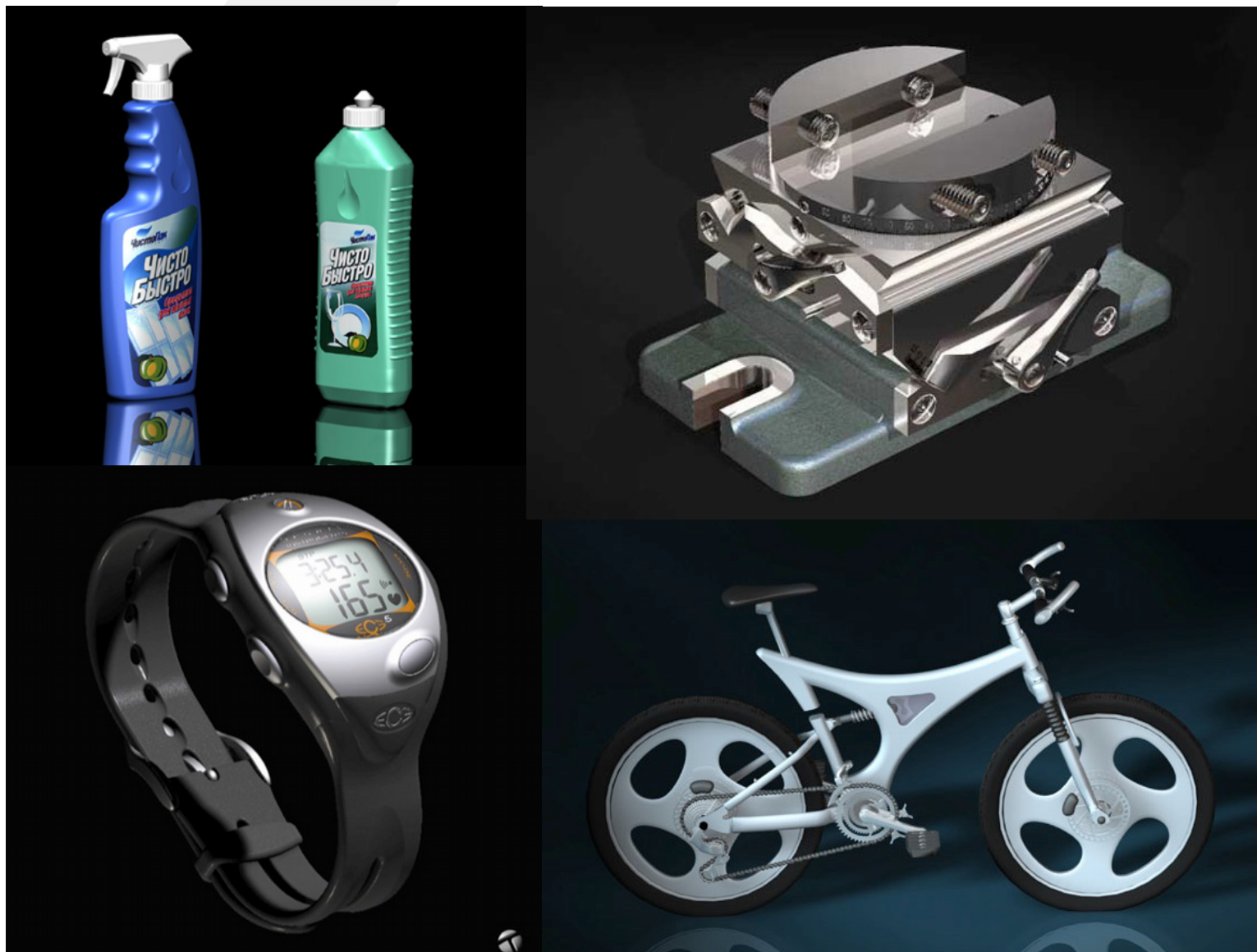
Films / Pub



Simulateurs



CAD & Design



CAD & Design



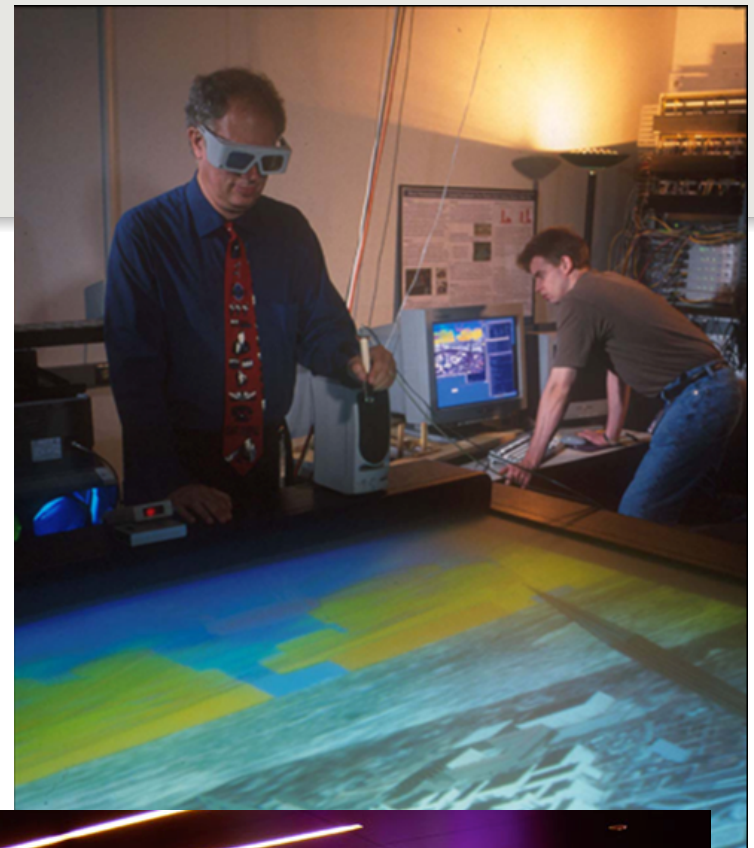
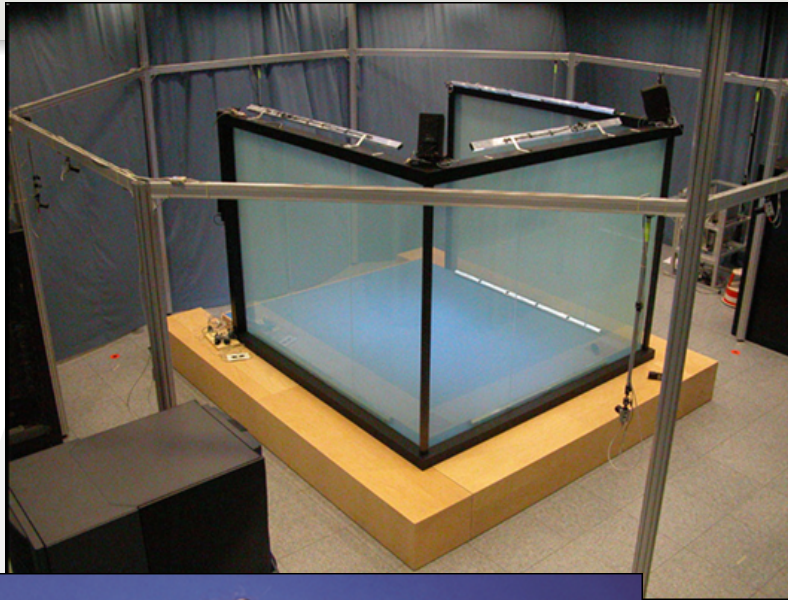
Architecture



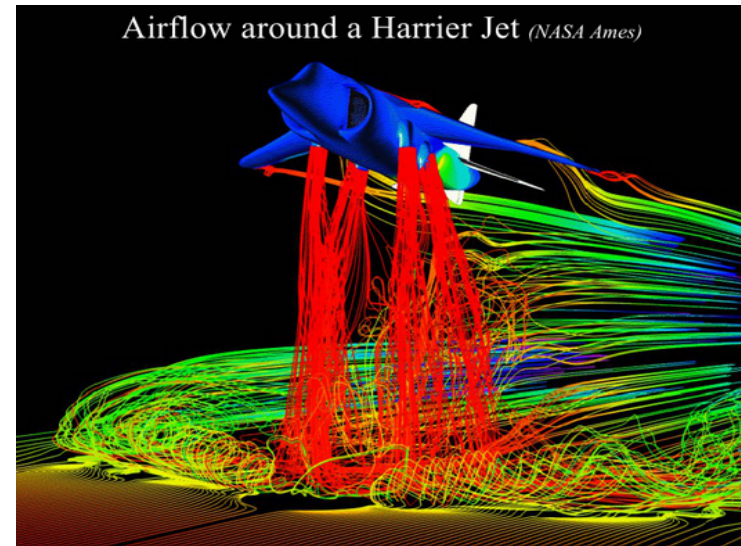
Architecture



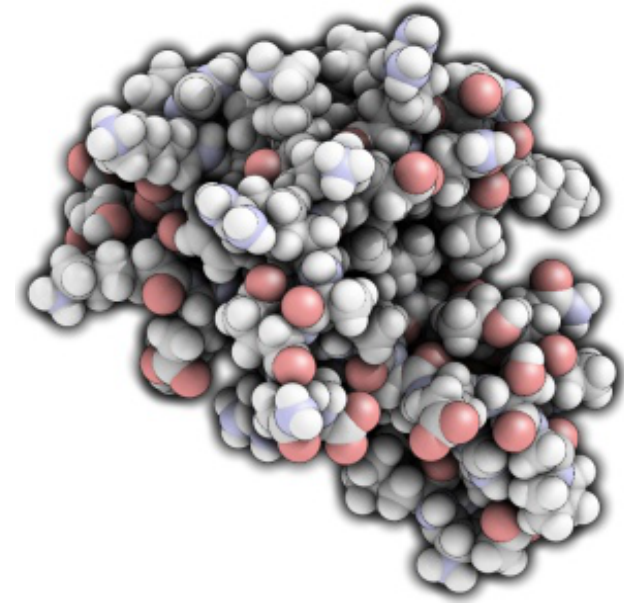
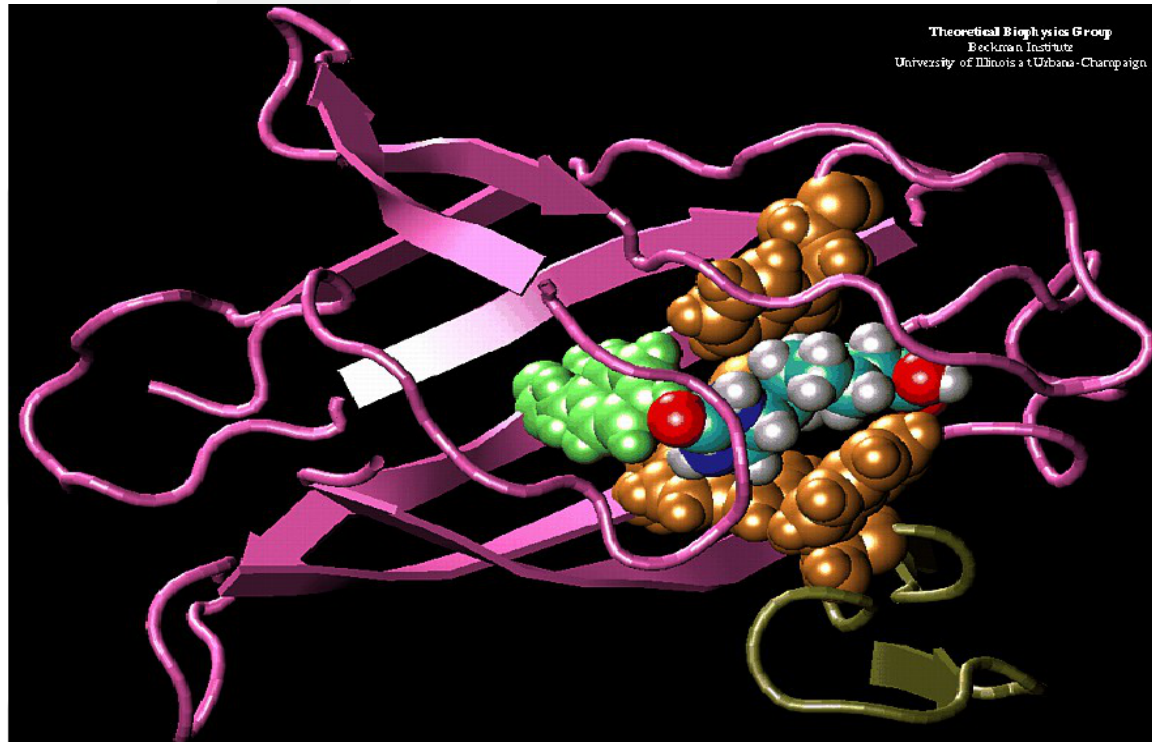
Réalité virtuelle



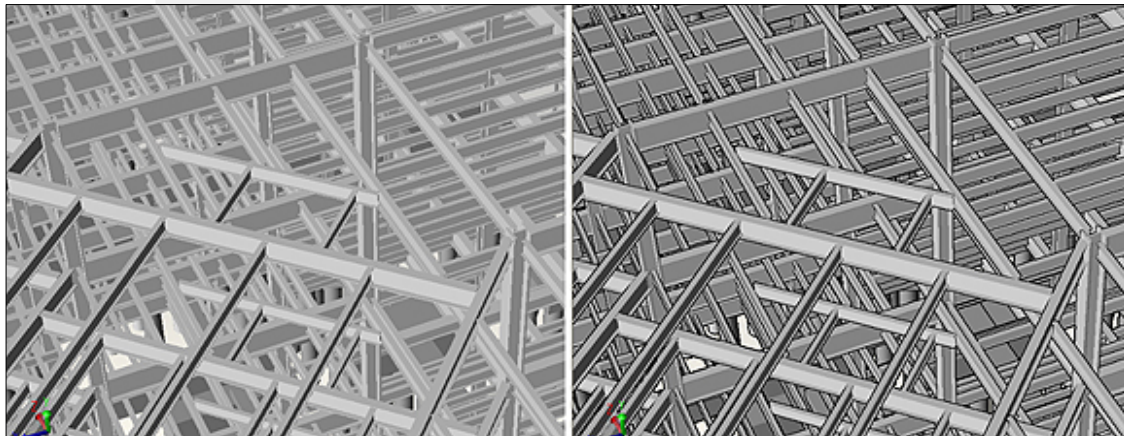
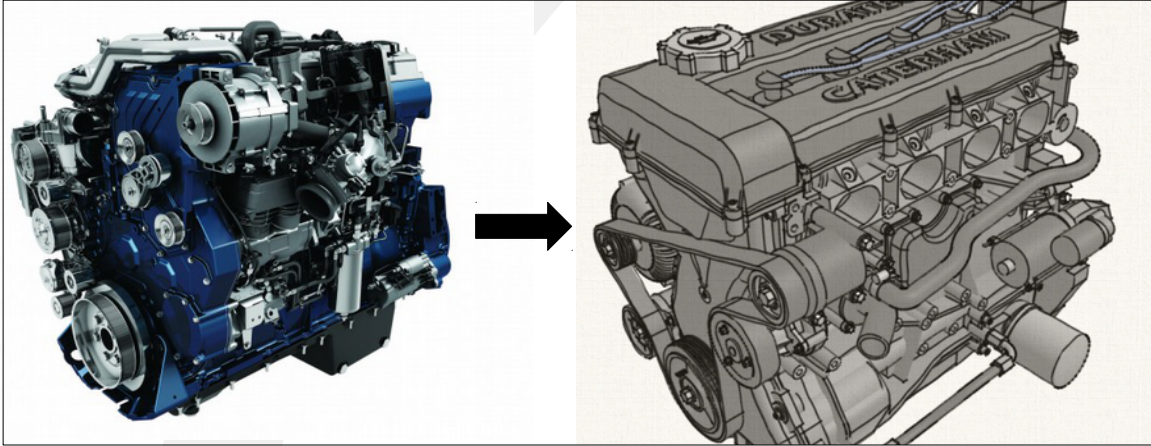
Visualisation scientifique



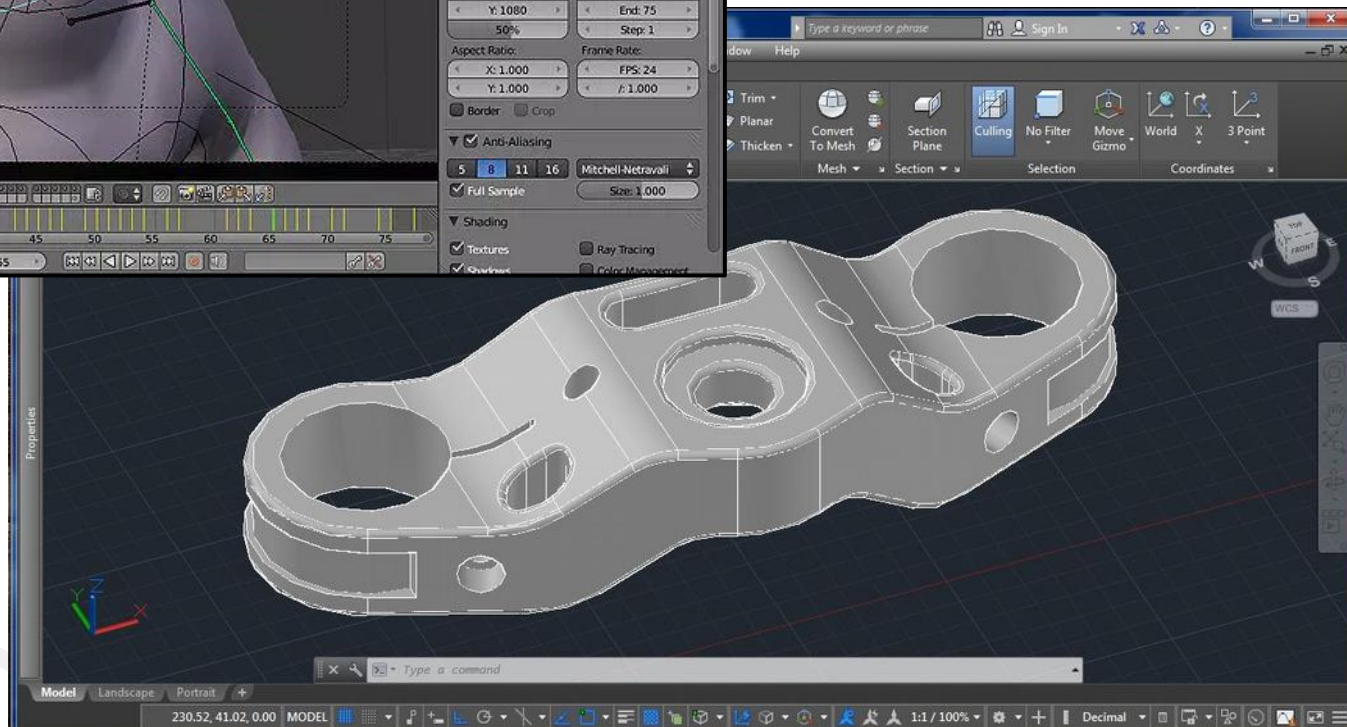
Theoretical Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champaign



Rendu illustratif/expressif



Conception



Synthèse d'Images

- **3 classes d'algorithmes**

- rendu temps réel (jeux, simulateur, modeleur, etc)

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



OpenGL
DirectX
Consols

- rendu en temps maîtrisé SFX,(films)

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



Pipeline maison
(+OpenGL)

- rendu physiquement réaliste (conception, biologie, etc)

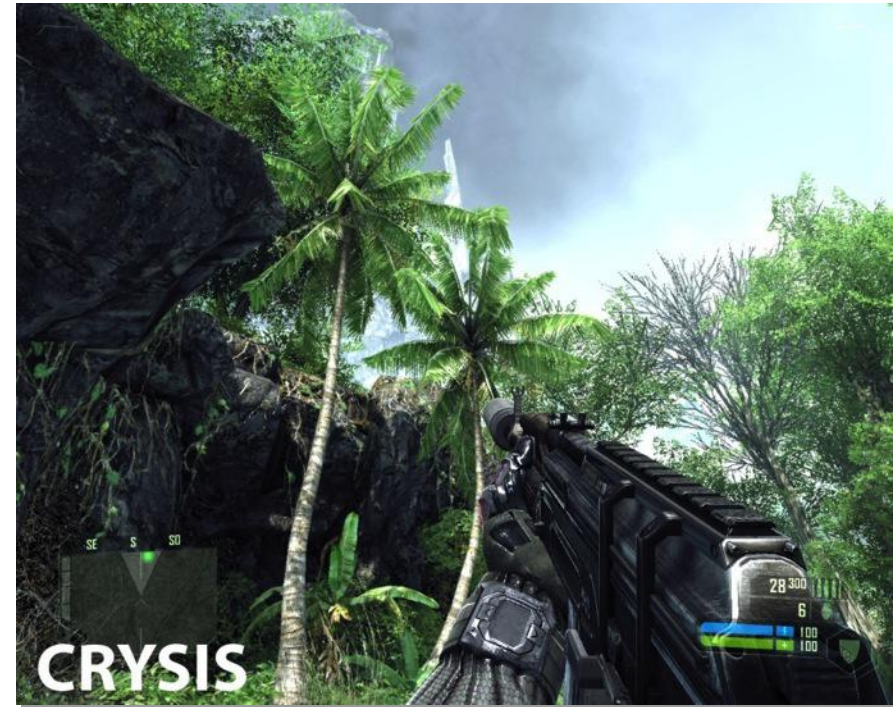
- de qqes heures à quelques jours par images...
- maîtrise de l'erreur



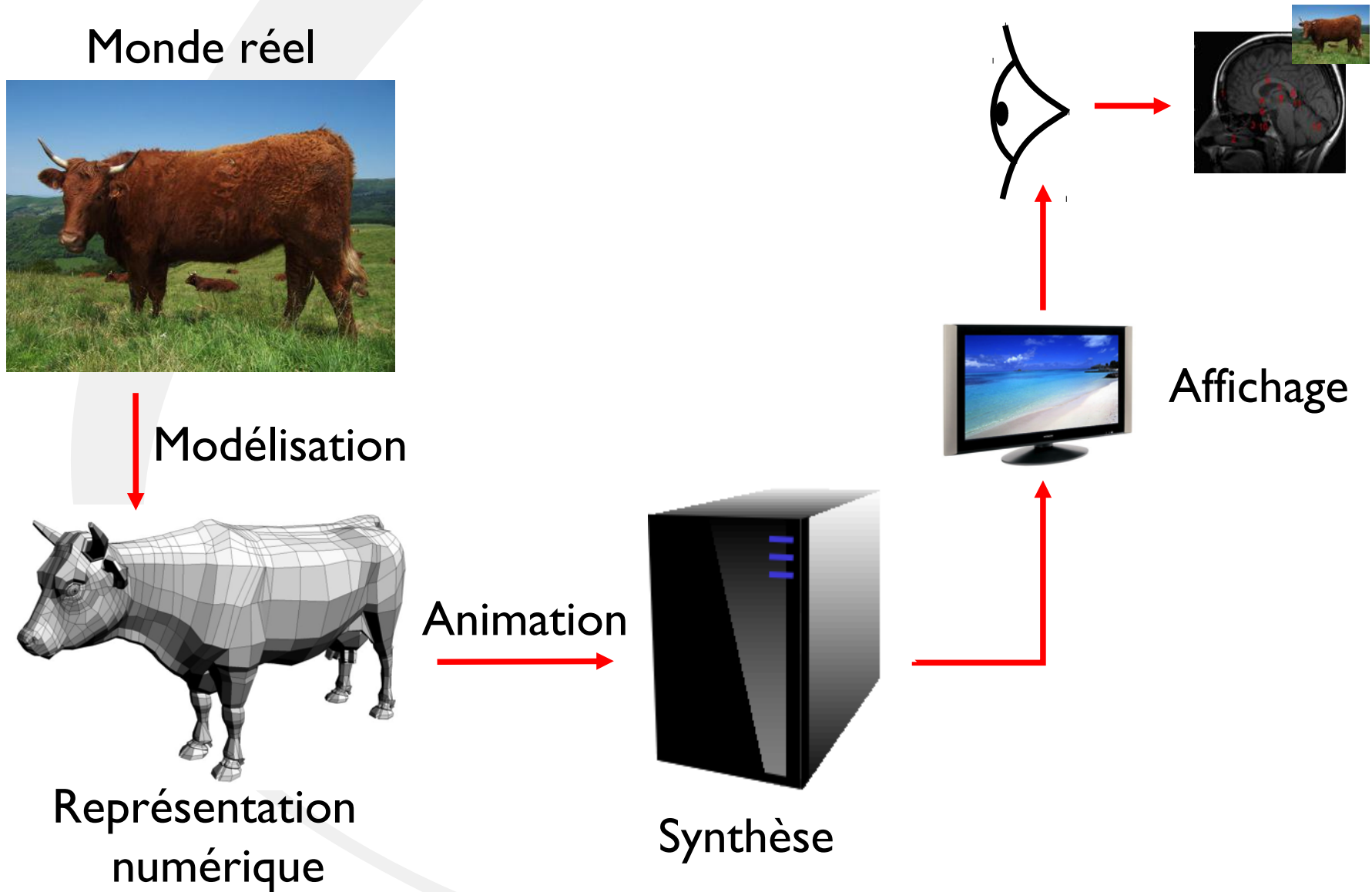
Pipeline maison

Question

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



Problèmes à résoudre



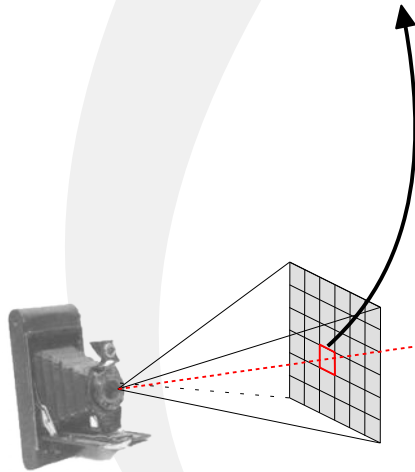
Problèmes à résoudre

- **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
- **Autres**
 - interactions avec le monde virtuel
 - game-play, « intelligence artificielle », etc.

Comment synthétiser une image ?

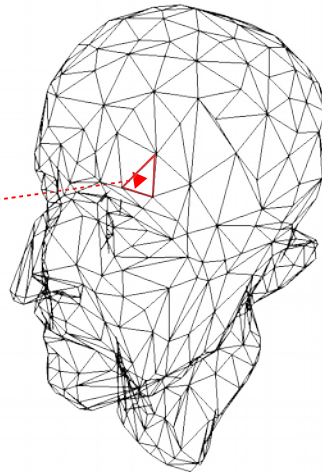
problématique :

- quelle est la couleur de chacun des pixels ?



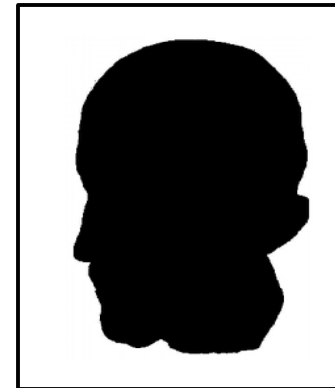
caméra virtuelle

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



modèles 3D

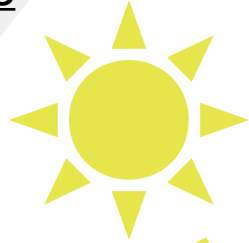
- e.g. : « soupe de triangles »



2D image

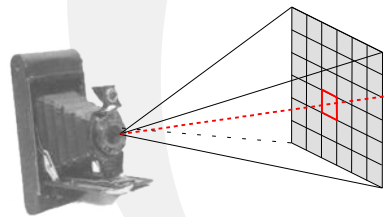
Comment synthétiser une image ?

sources de lumière



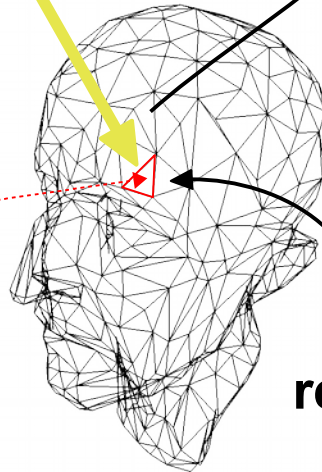
matériaux

- mat, brillant, translucide
- texture (bois, tissus),
- etc.



caméra virtuelle

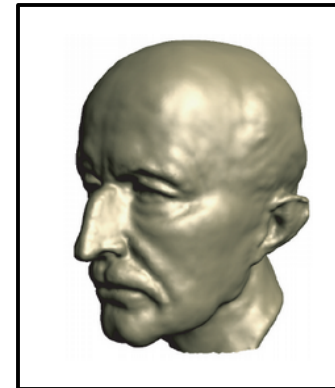
- tableau de pixels RGB
- position, orientation
- résolution
- etc.



réflexions

modèles 3D

- e.g. : « soupe de triangles »



2D image

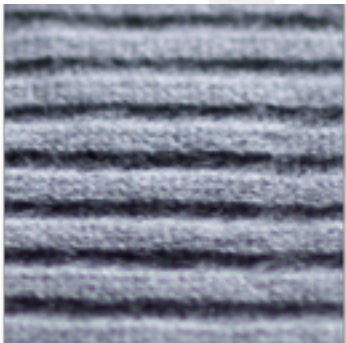
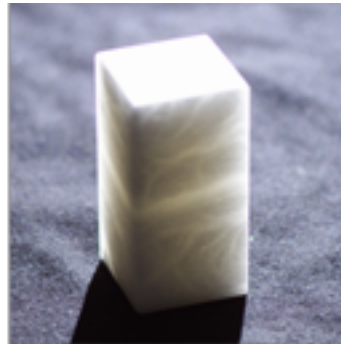
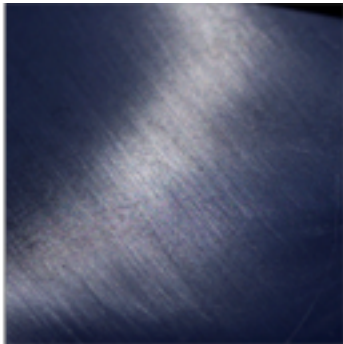
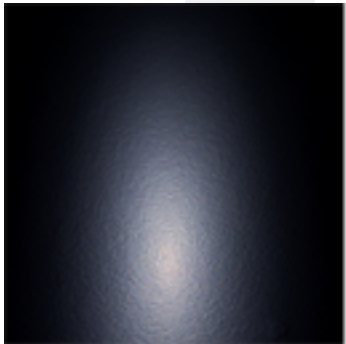
Synthèse d'images



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

Apparence, matériaux

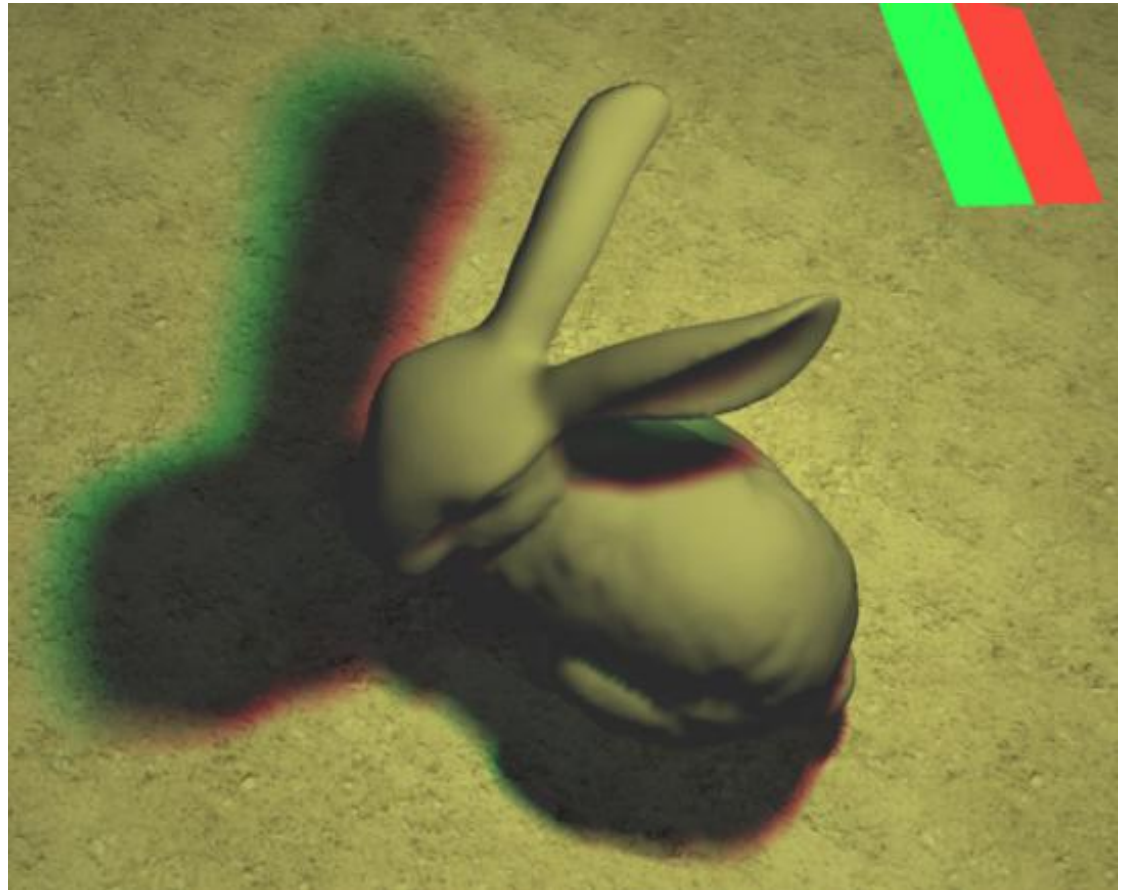
- **Textures** : couleur, effets de formes (normales, champ de hauteur, ...), environnement lumineux
- **Fonctions de réflectance** :
 - BRDF, SVBRDF, BTDF, BTF, ...
⇒ **master 2 (SIA)**



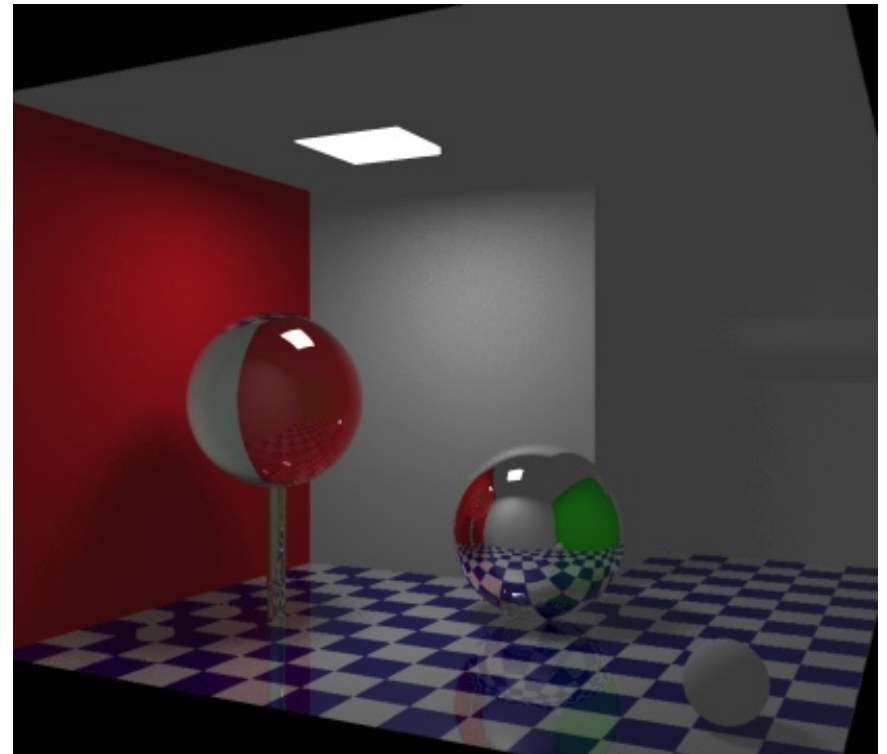
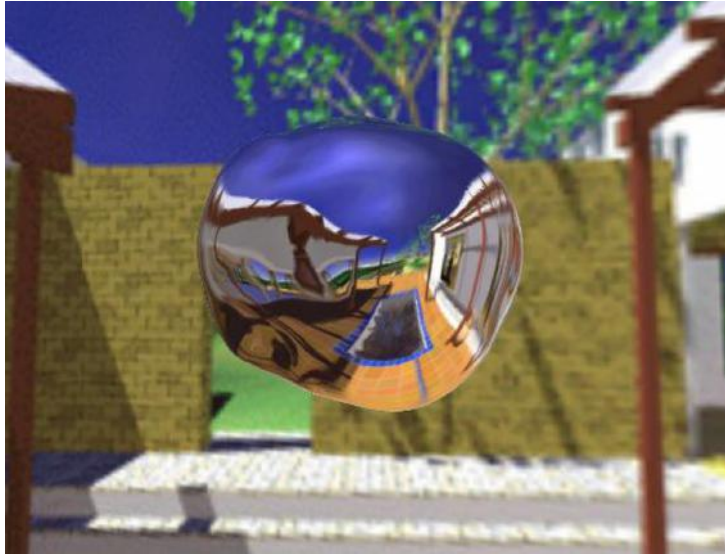
Ombres et lumière

Qui est éclairé par une source?

- visibilité
- géométrie



Réflexions miroirs / réfractions



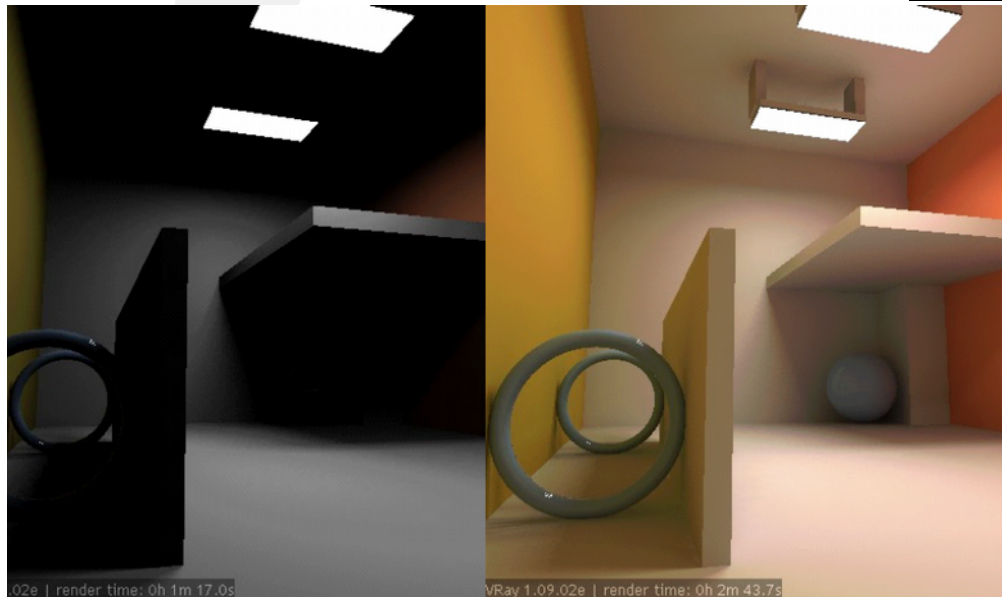
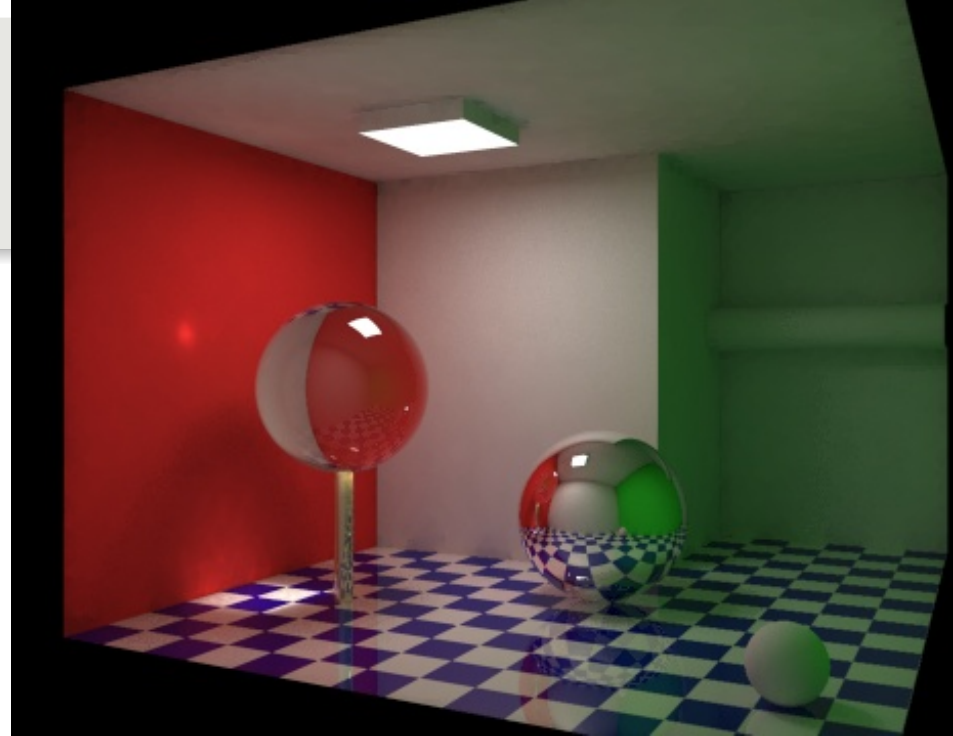
Éclairage direct vs indirect

Inter-réflexions

Sources étendues

Illumination globale

⇒ master 2 (SIA)



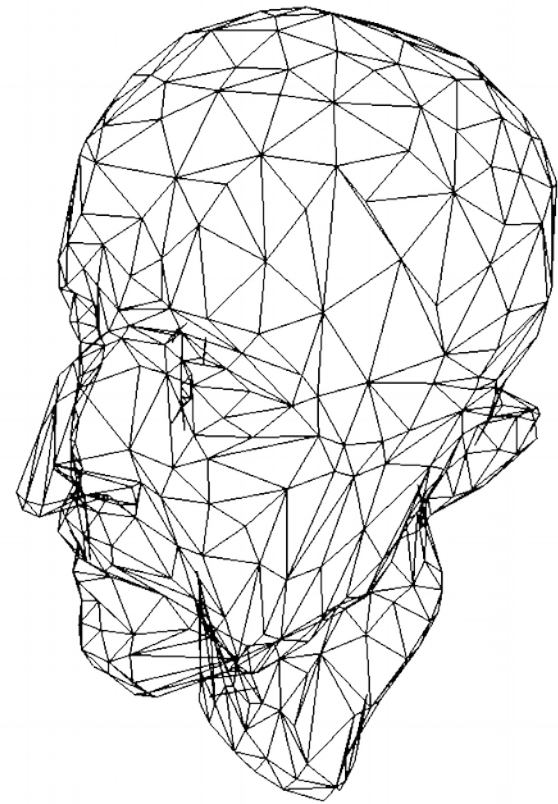
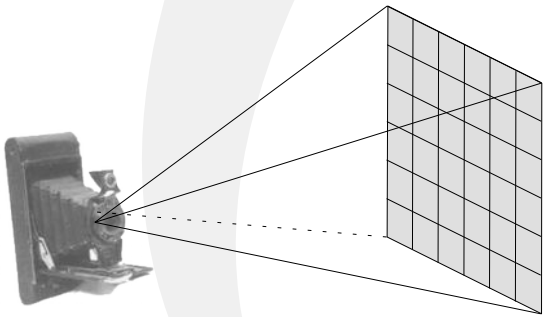
direct

+ indirect



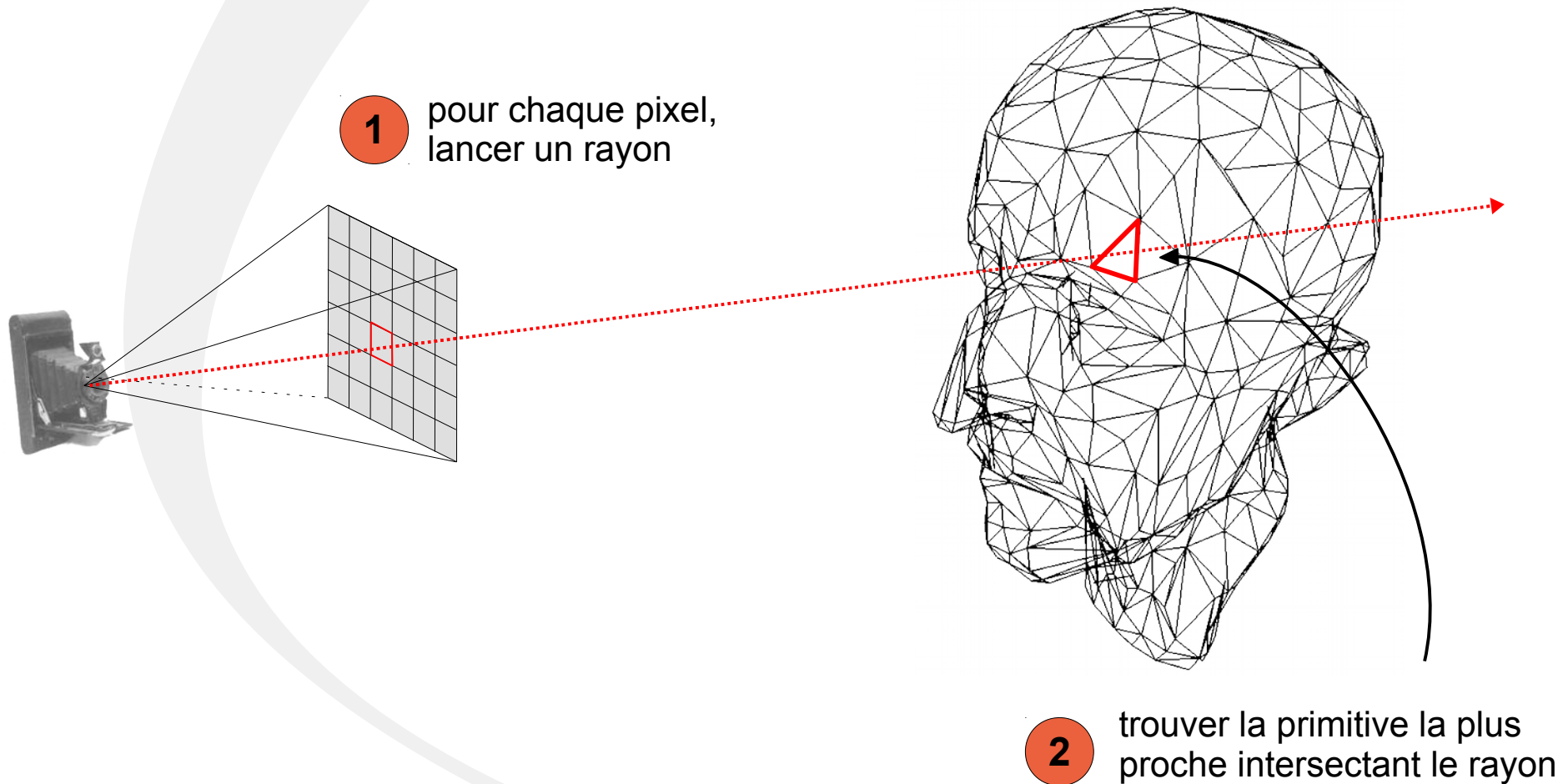
carte d'environnement

Visualiser une scène 3D



Visualiser une scène 3D : deux approches

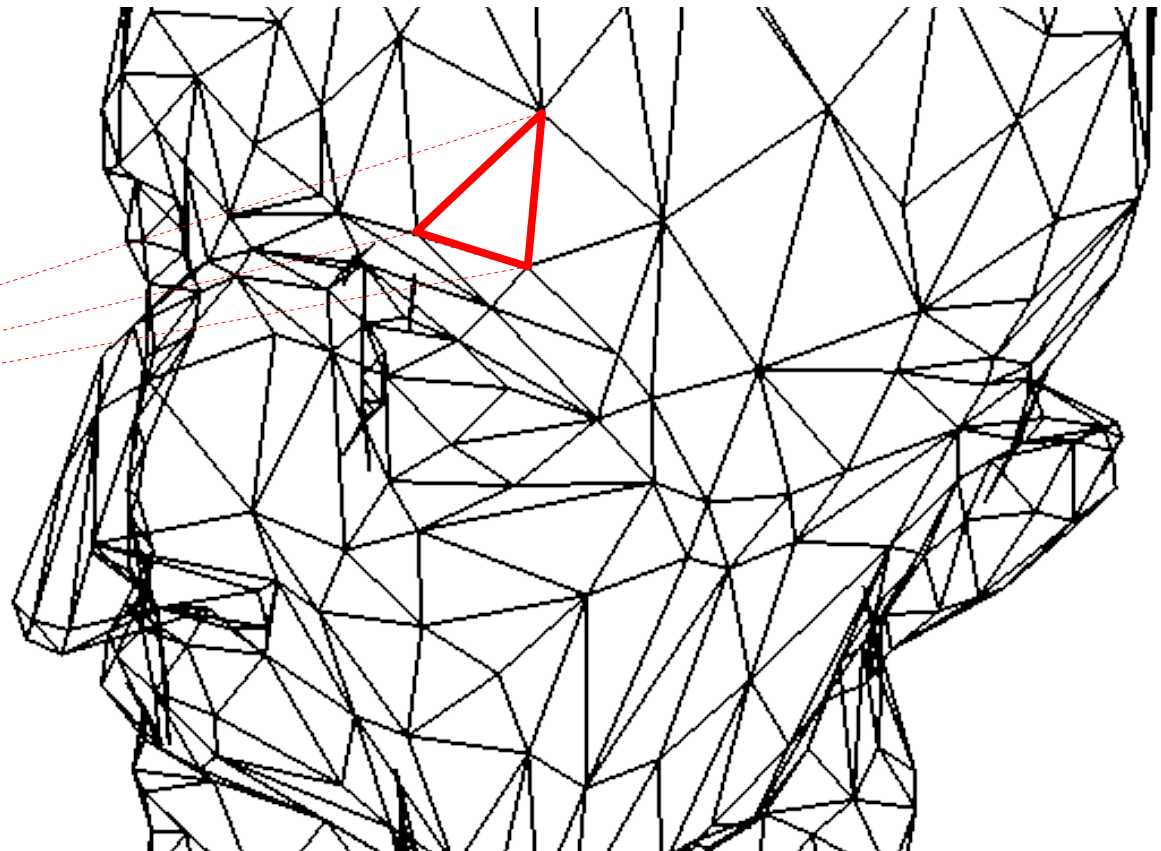
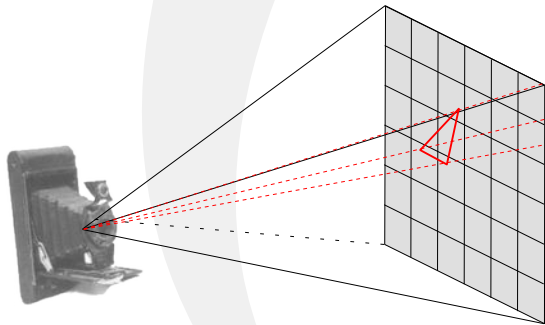
- 1ère approche : lancer de rayons (Ray-tracing – Ray casting)



Visualiser une scène 3D : deux approches

- 2ème approche : rastérisation

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



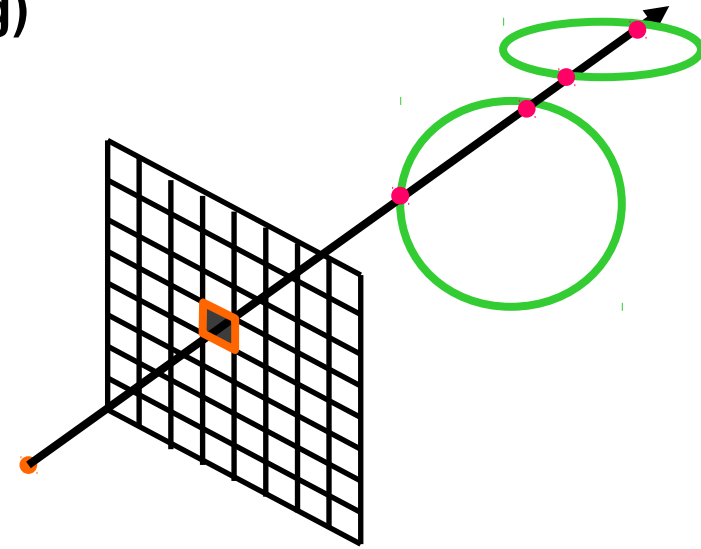
2 trouver les pixels
correspondant

Visualiser une scène 3D : deux approches

- **Tracé de Rayon (Ray-tracing – Ray casting)**

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

→ **approche suivie pour le réalisme**



- **Rastérisation**

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

→ **approche suivie pour le temps réel**