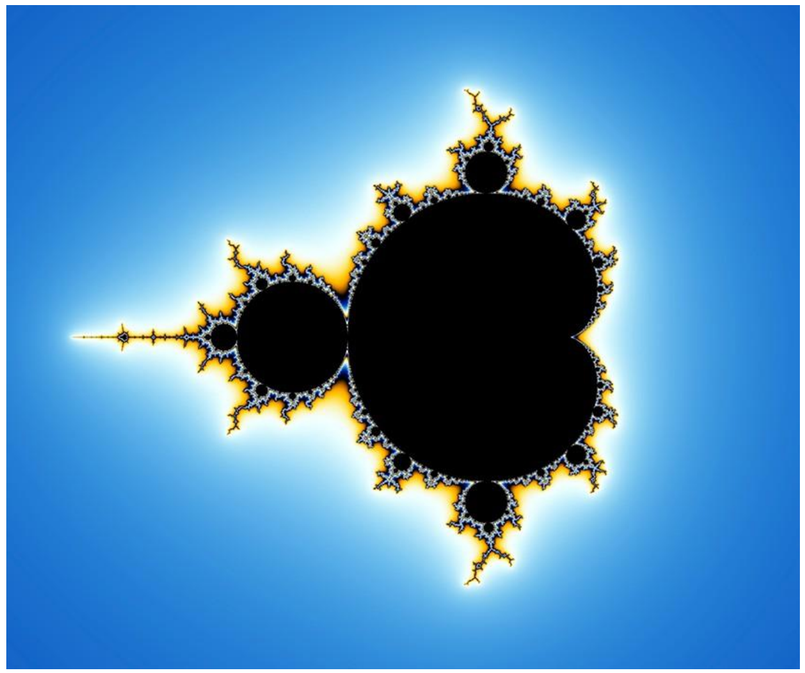
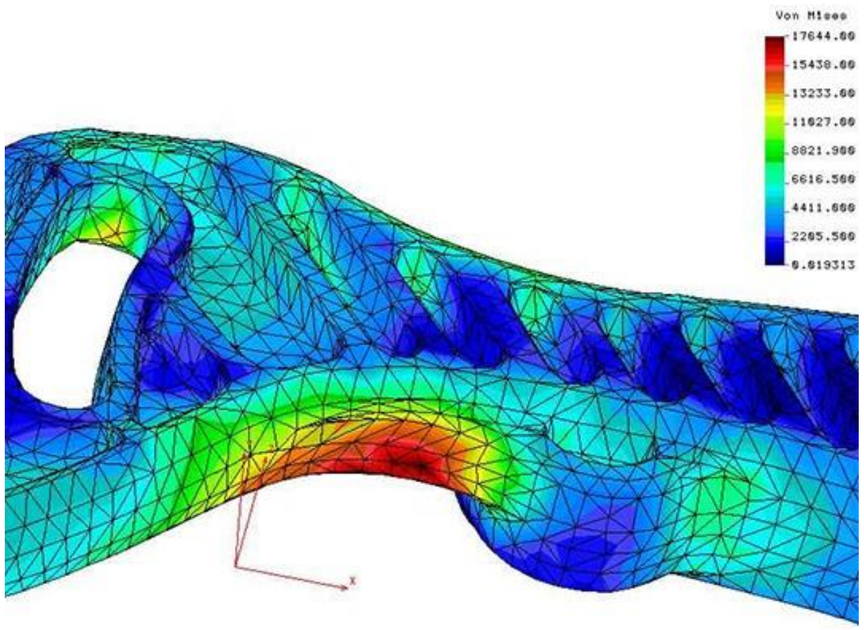
Introduction

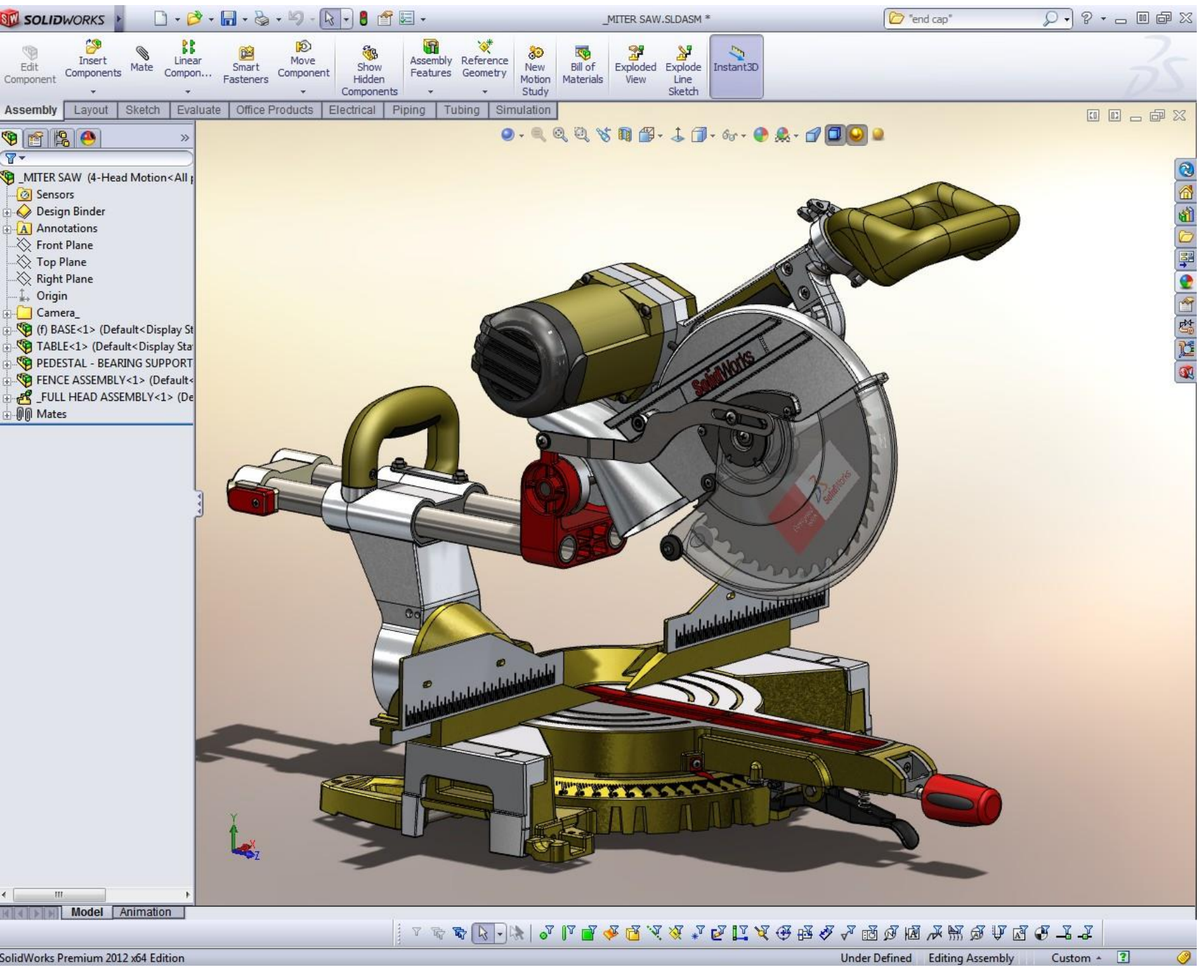
Modélisation et programmation 3D
HMIN212

Imagerie numérique

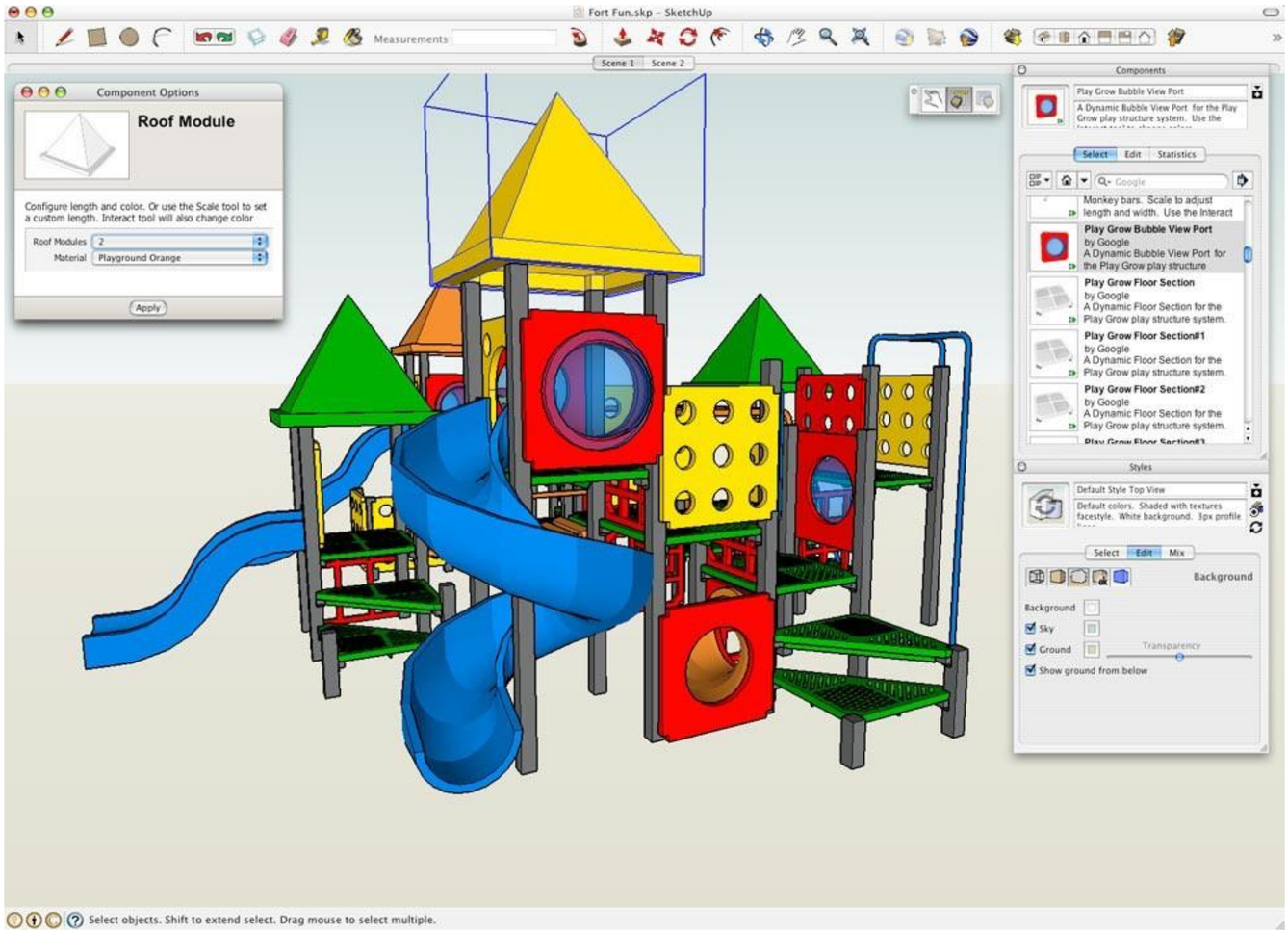




CAD & Design



SolidWorks



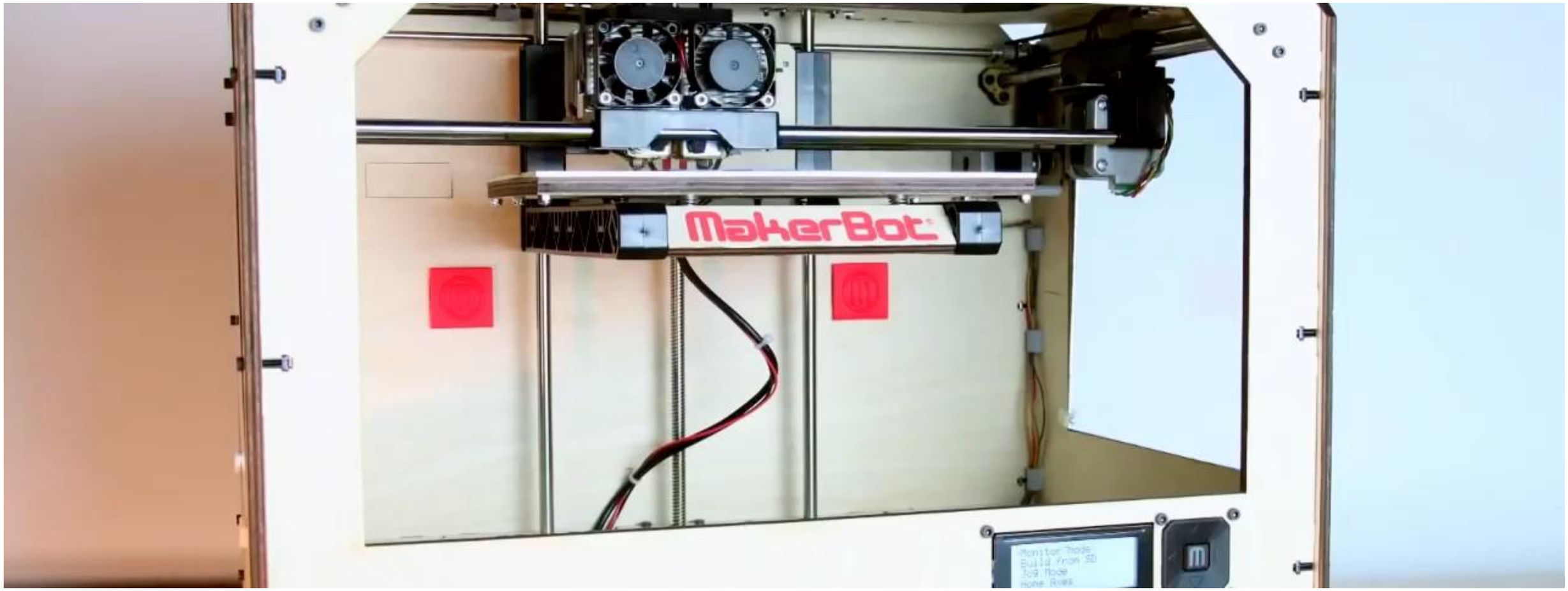
SketchUp

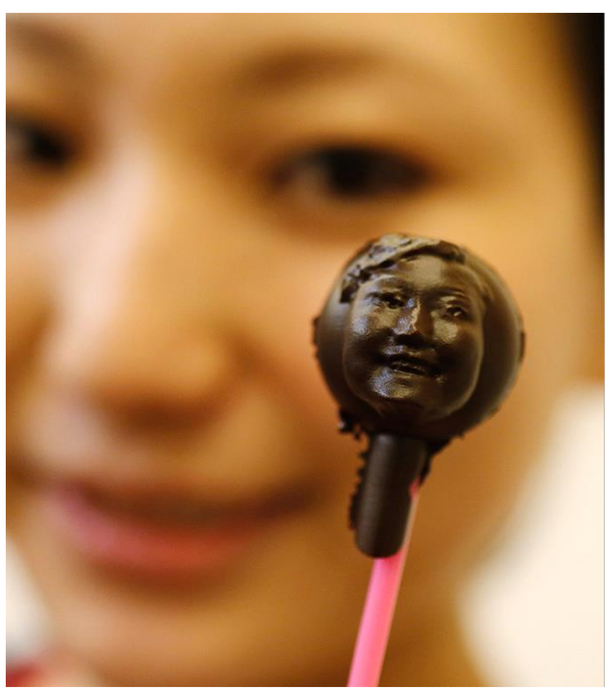
Pour la mechanique, l'architecture, electronique, optique, …

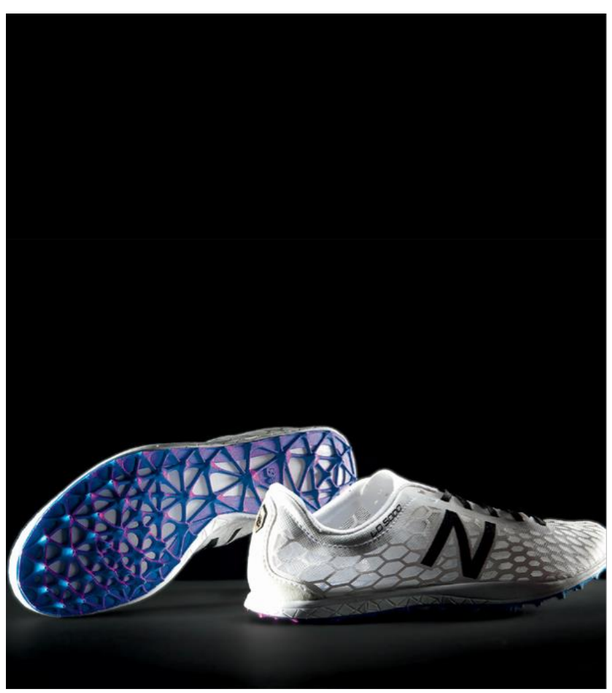
CAD & Design

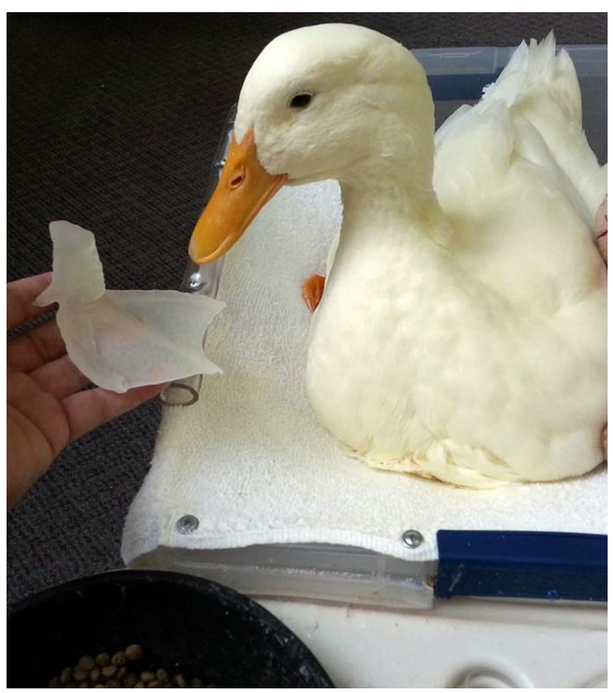
CAD & Design

Fabrication 3D

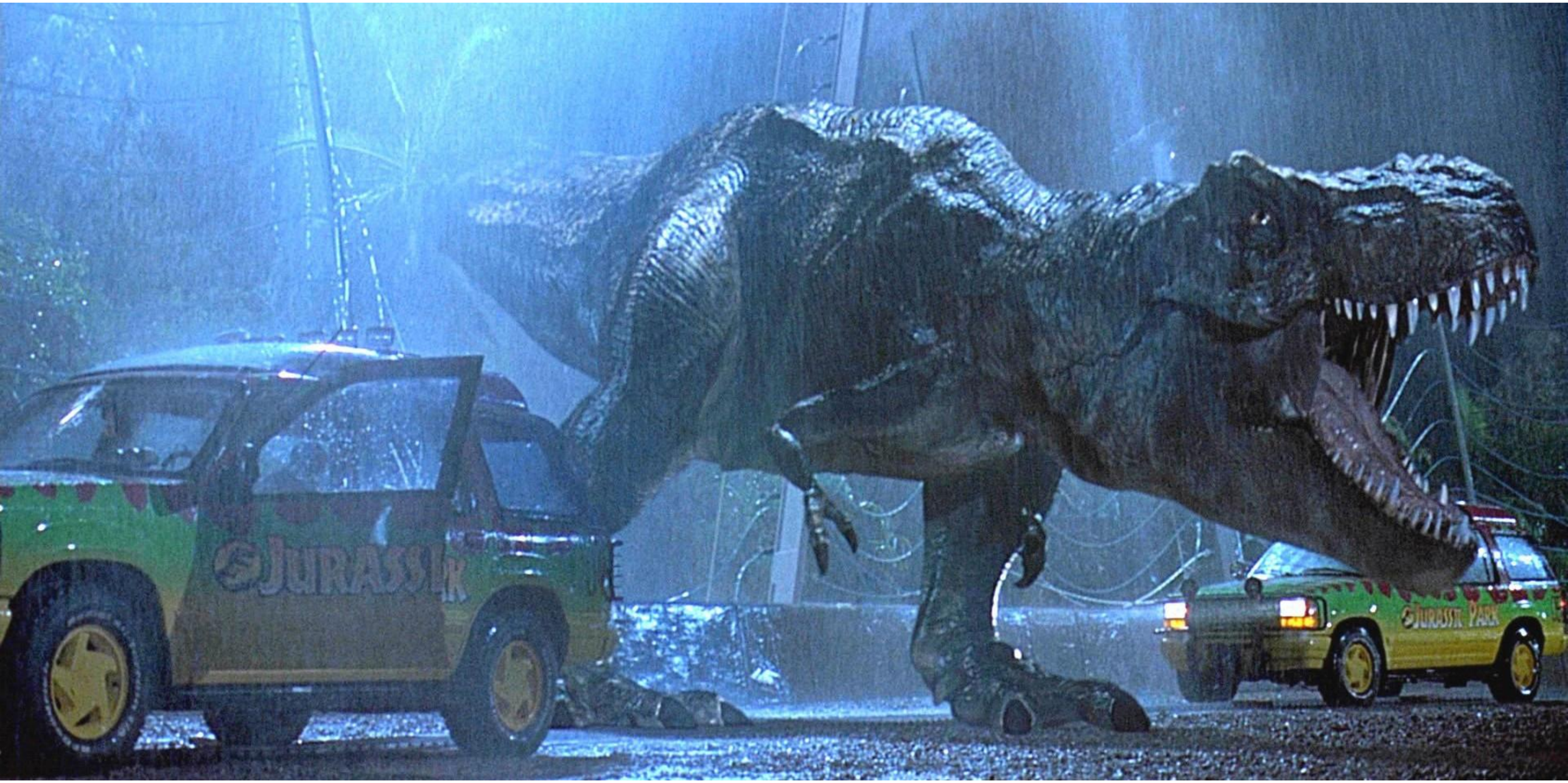






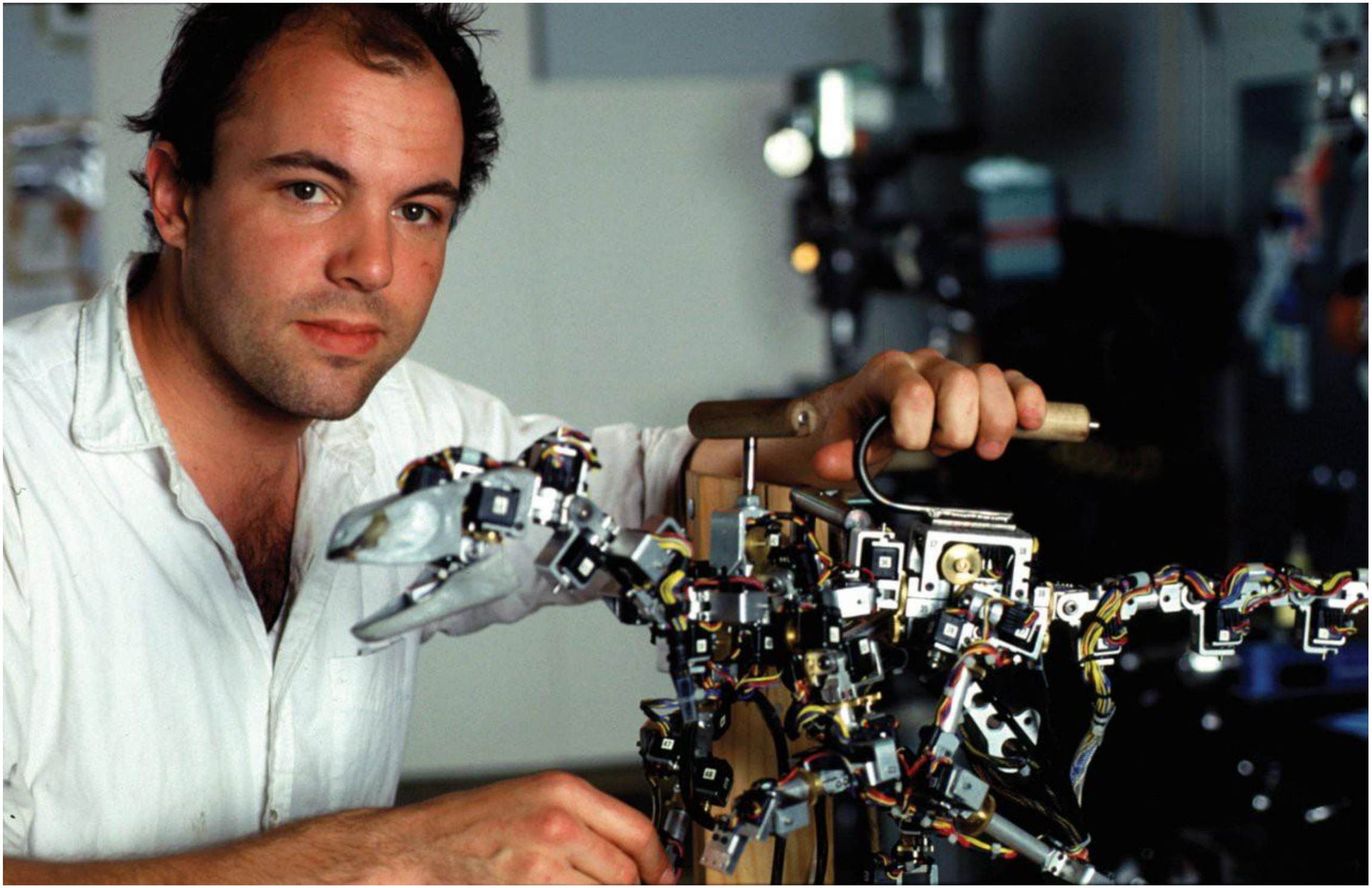


Les films



Jurassic Park (1993)

Les films



“Dinosaur Input Device”

Les films

The Matrix (1999)

Les films

The Matrix (1999)

Les films

Avatar (2009)

Les films

Avatar (2009)

Les films

Avatar (2009)

Films / Pub

Jeux vidéo

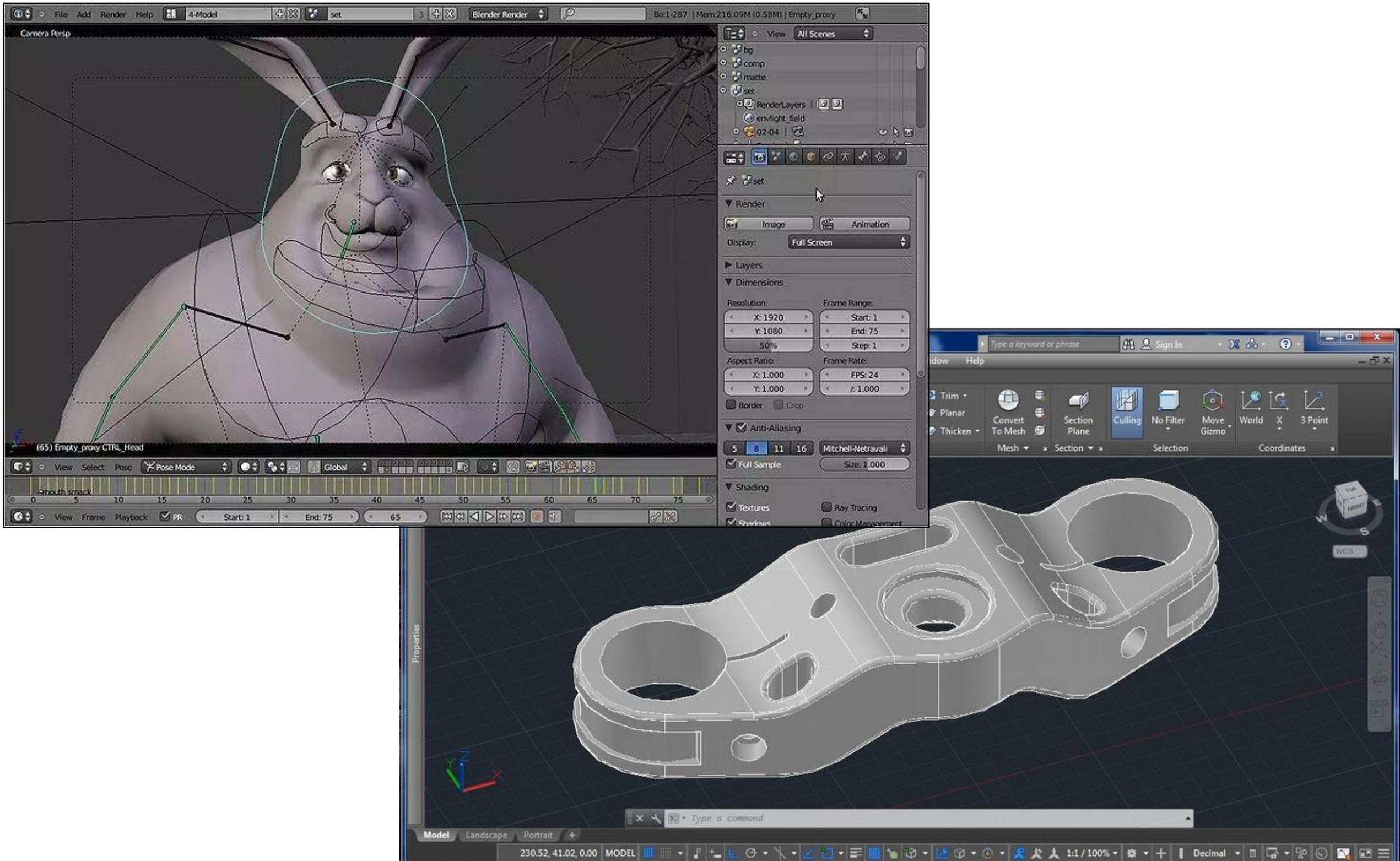
Jeux vidéo

Crysis 3 (2013)

Jeux vidéo

“Adam” Realtime Rendering Demo (Unity 2016)

Conception


File Add Render Help
4-Model
set
Blender Render
Camera Persp
View All Scenes
bg
comp
matte
set
RenderLayers
envlight_field
02-04
Render
Image
Animation
Display: Full Screen
Layers
Dimensions
Resolution:
X: 1920
Y: 1080
50%
Frame Range:
Start: 1
End: 75
Step: 1
Aspect Ratio:
X: 1.000
Y: 1.000
Frame Rate:
FPS: 24
Border
Crop
Anti-Aliasing
5 8 11 16
Mitchell-Netravali
Full Sample
Size: 1.000
Shading
Textures
Ray Tracing
(65) Empty_proxy CTRL_Head
View Select Pose
Pose Mode
Global
View Frame Playback
PR
Start: 1
End: 75
65
Type a keyword or phrase
Sign In
Help
Trim
Planar
Thicken
Convert To Mesh
Section Plane
Culling
No Filter
Move Gizmo
World
X
3 Point
Mesh
Section
Selection
Coordinates
Properties
Type a command
Model
Landscape
Portrait
230.52, 41.02, 0.00
MODEL
1:1 / 100%
Decimal

Conception de produits et visualisation

Ikea - 75% des catalogues sont des images de synthèse

Conception de produits et visualisation

Tesla Model X concept (2012)

Architecture

Architecture

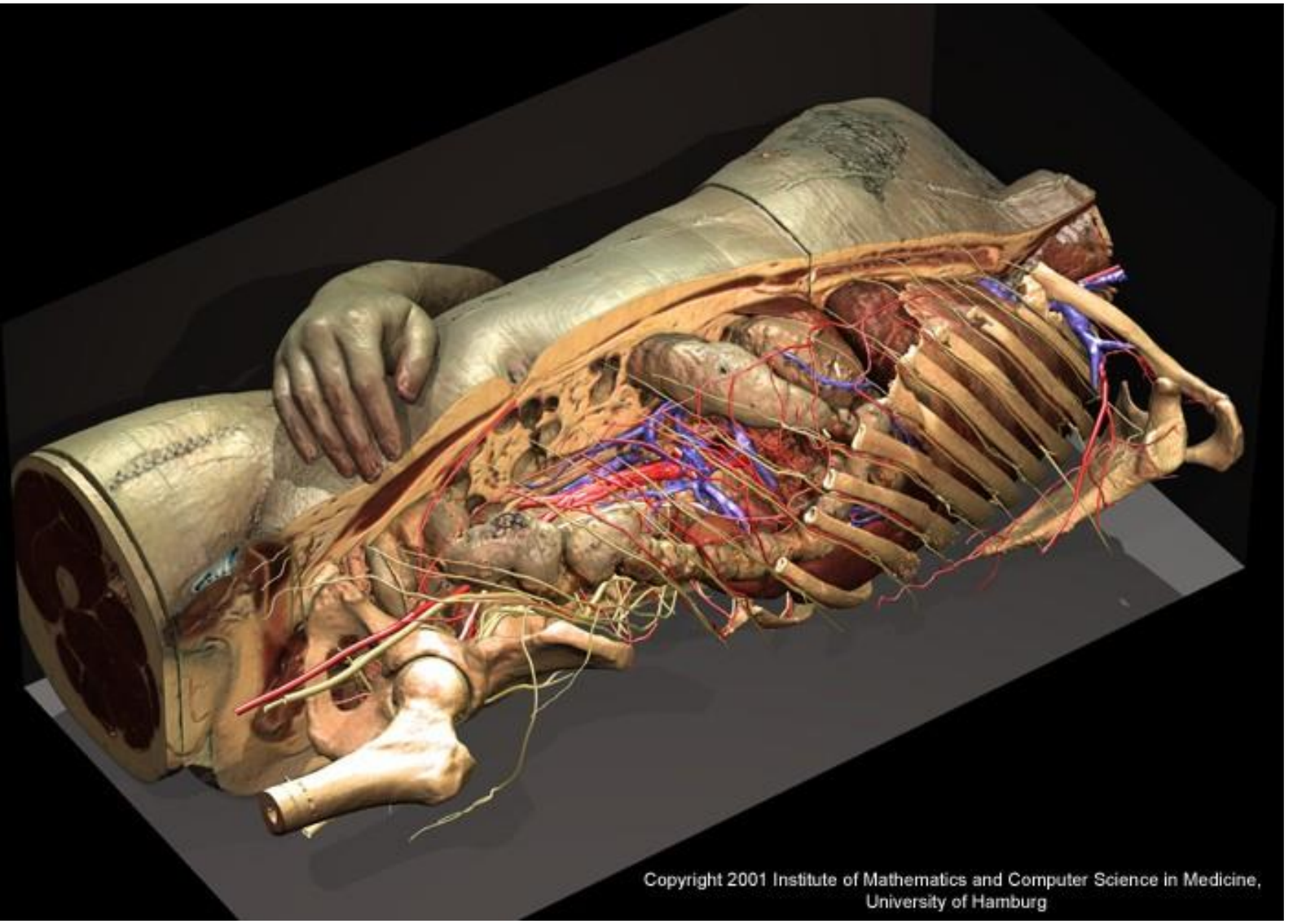
Architecture

Bilbao Guggenheim, Frank Gehry

Architecture

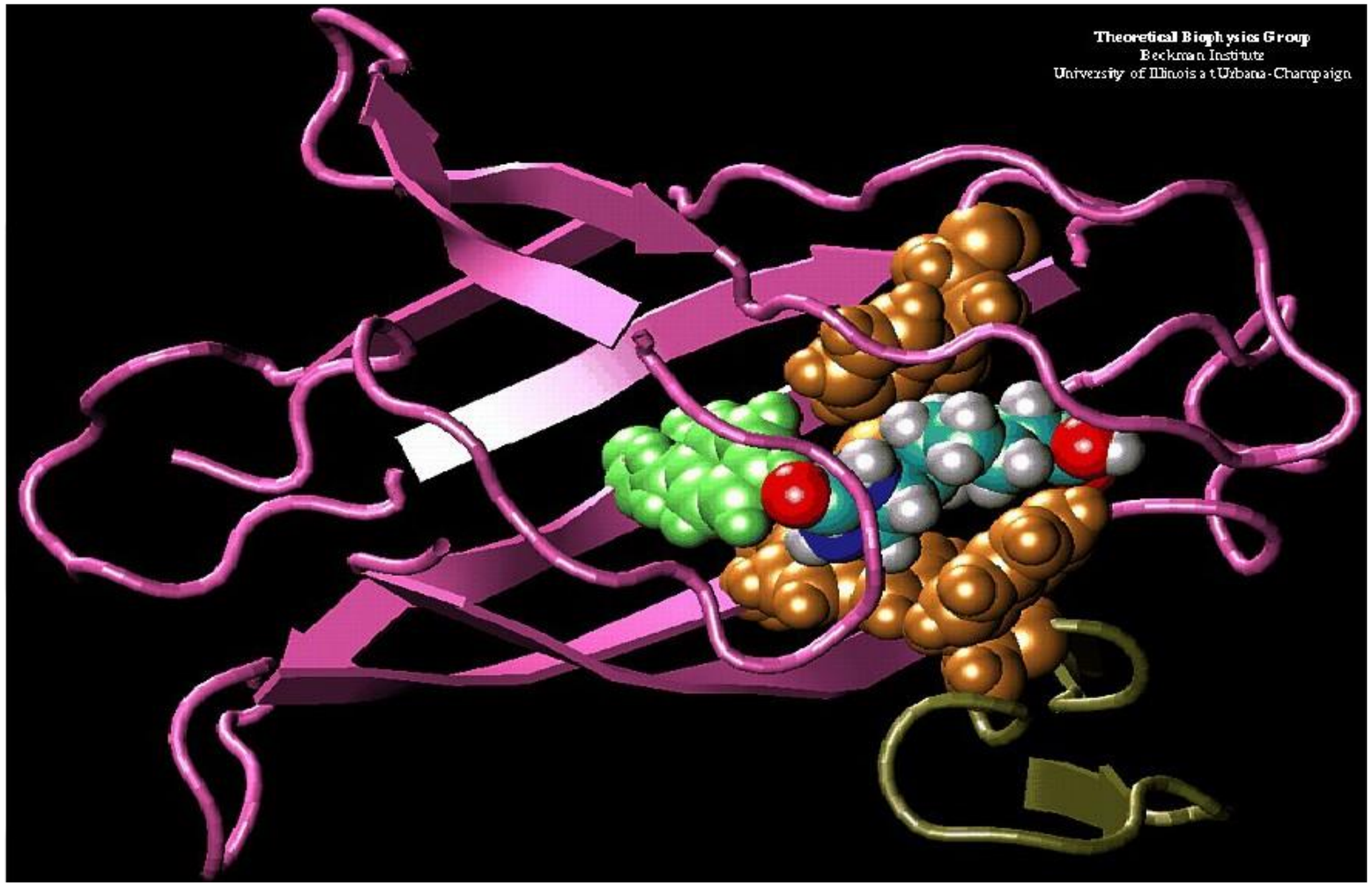
Heydar Aliyev Center, Zaha Hadid Architects

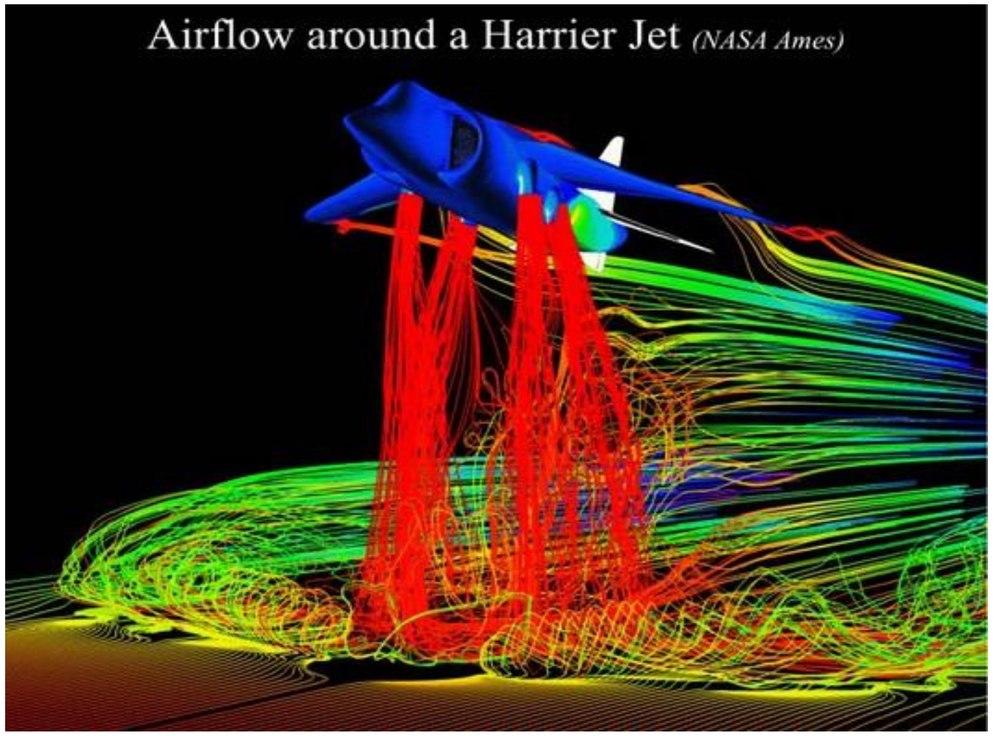
Visualisation

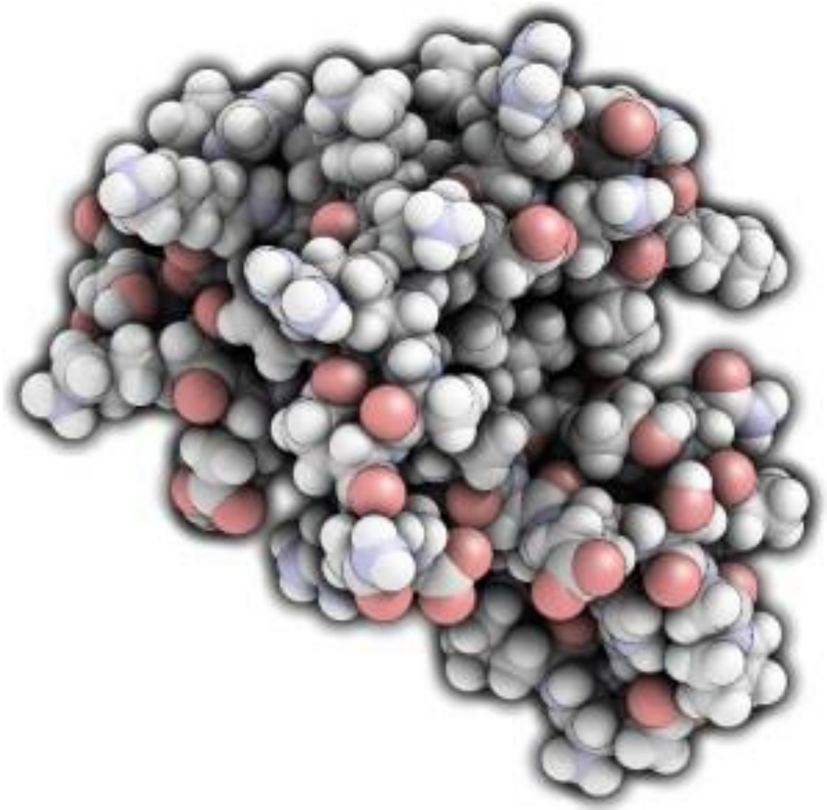


Science, ingénierie, médecine, journalisme, …

Visualisation scientifique







Simulation virtuelle

Driving simulator
Toyota Higashifuji Technical Center

da Vinci surgical robot
Intuitive Surgical

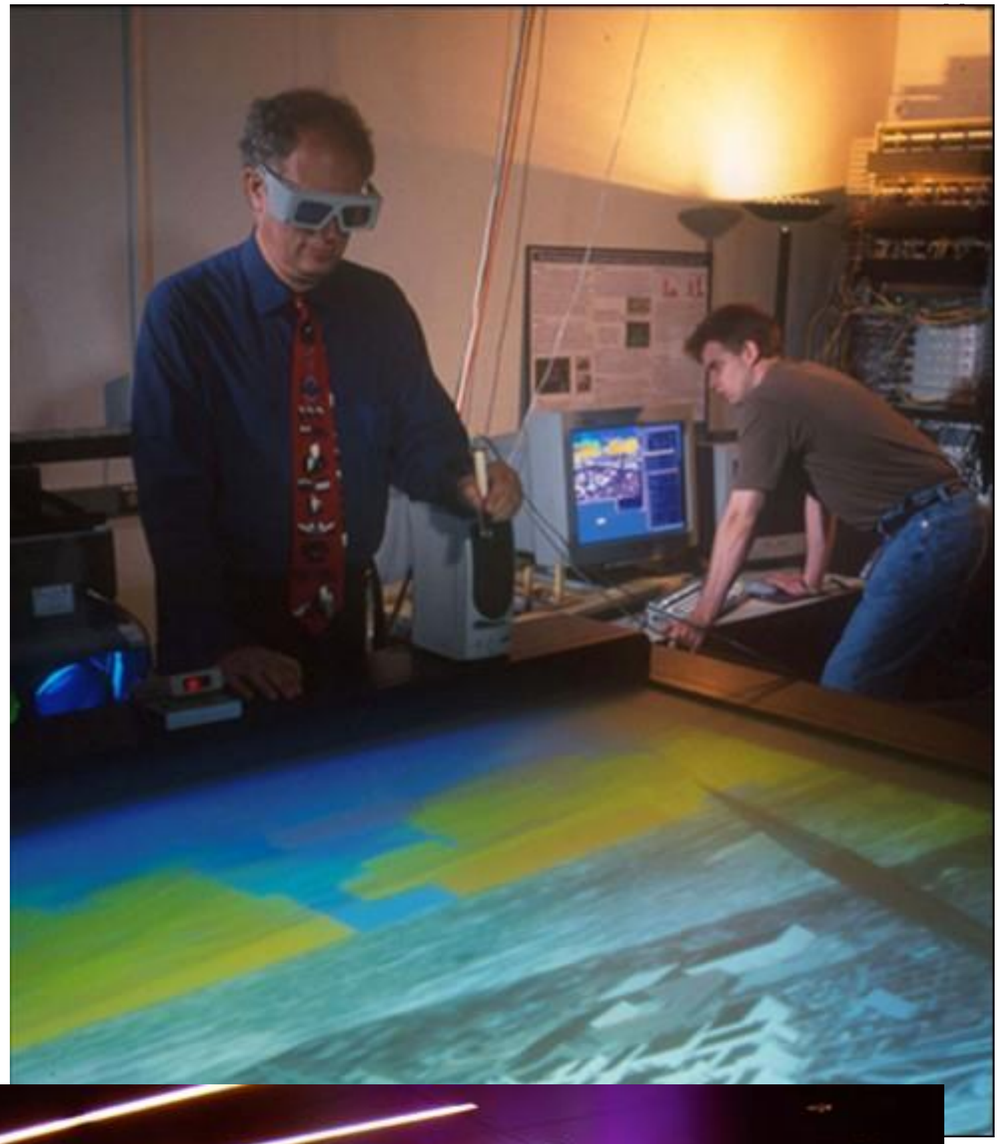
Vol, conduite, chirurgie…

Simulateurs

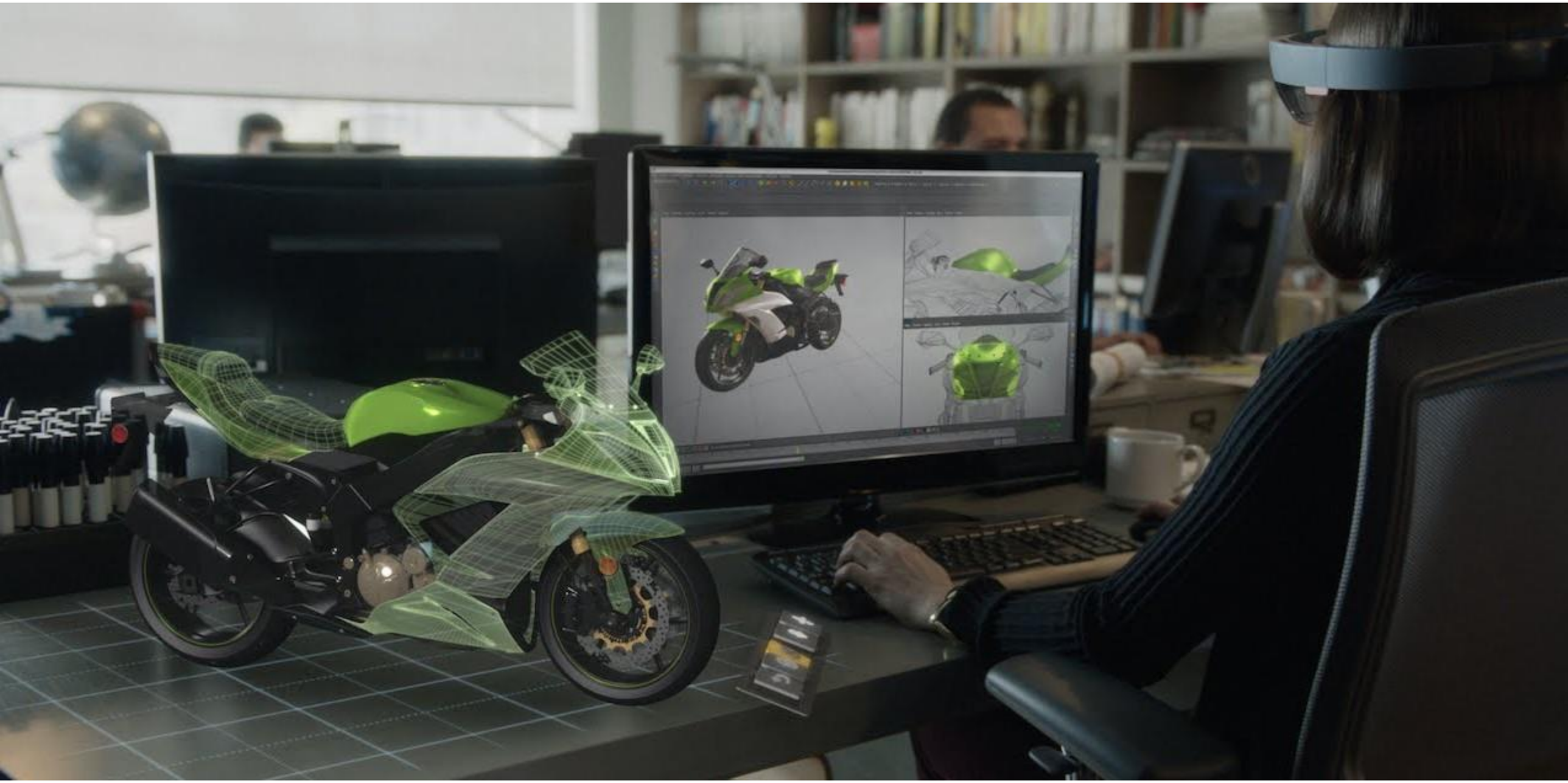
Réalité virtuelle

HTC Vive headset and controllers

Réalité virtuelle

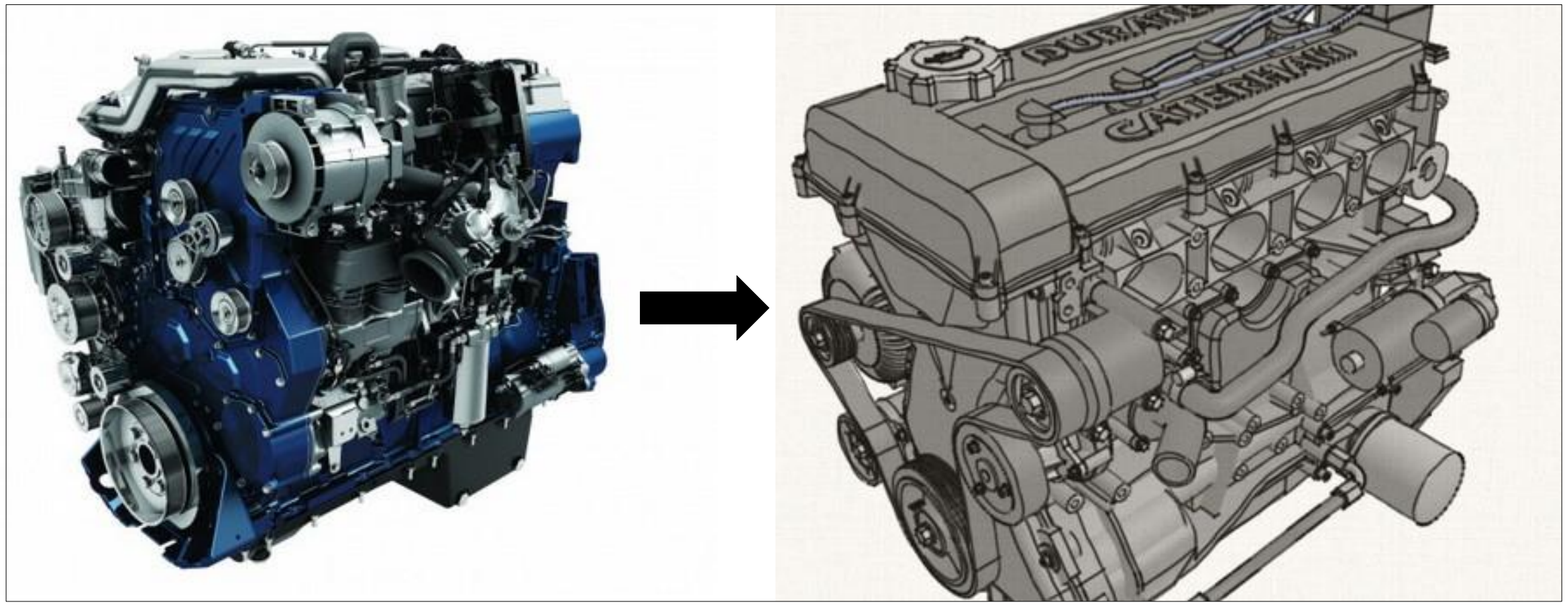


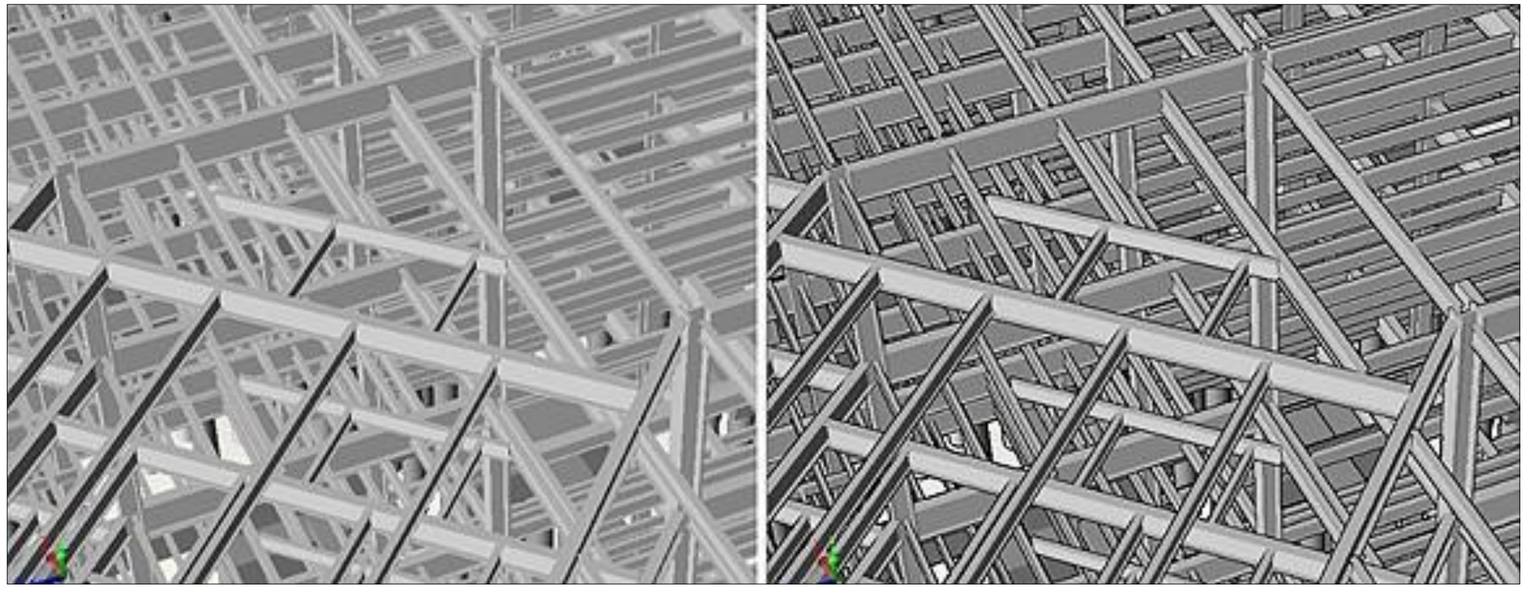
Réalité augmentée



Microsoft Hololens augmented reality headset concept

Rendu illustratif/expressif





Synthèse d’images

Ces applications nécessitent une théorie et des systèmes sophistiqués

Science et mathématiques

- Physique de la lumière, de la couleur, de l'optique,…
- Mathématiques des courbes, surfaces, géométrie, perspective,…

Technologie et systèmes

- Périphériques d'entrée, GPU, écrans,…

Art et psychologie

- Perception : couleur, stéréo, mouvement, qualité d'image,…
- Art et design : composition, forme, éclairage, ...

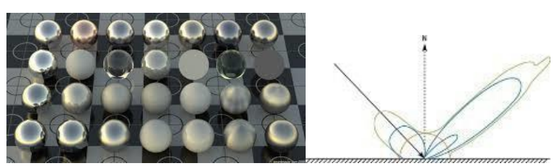
→ **Une discipline de l’Informatique Graphique**

Synthèse d'Image

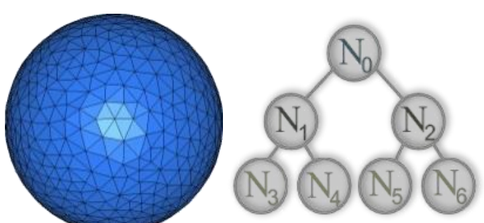
Modèles

Capteur

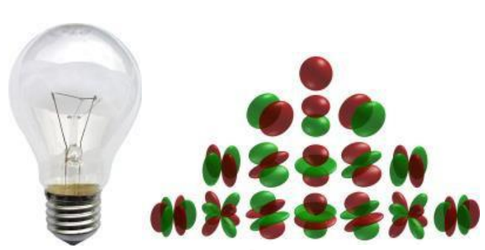
Mouvement



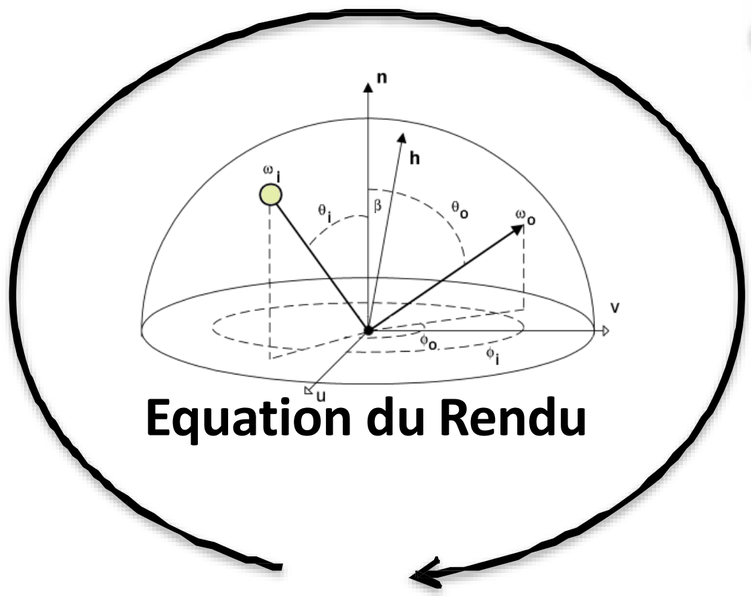
Apparence



Forme

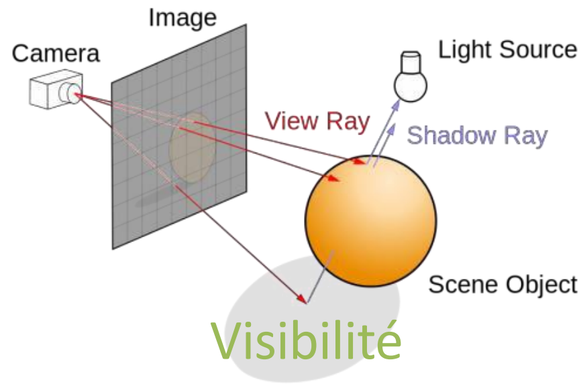


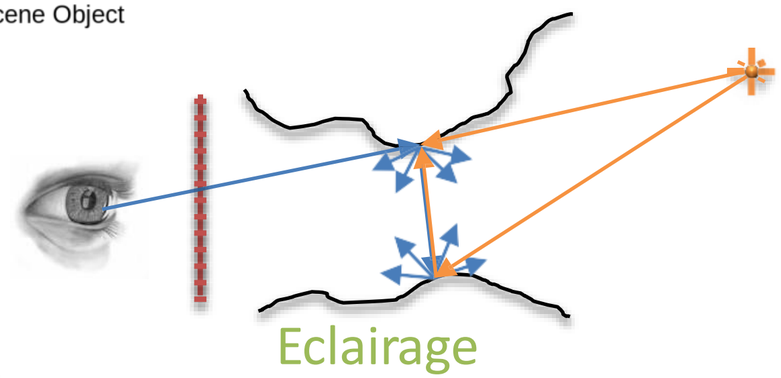
Lumière



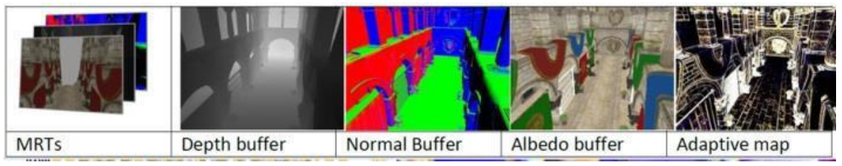
Equation du Rendu

Algorithmes





Eclairage



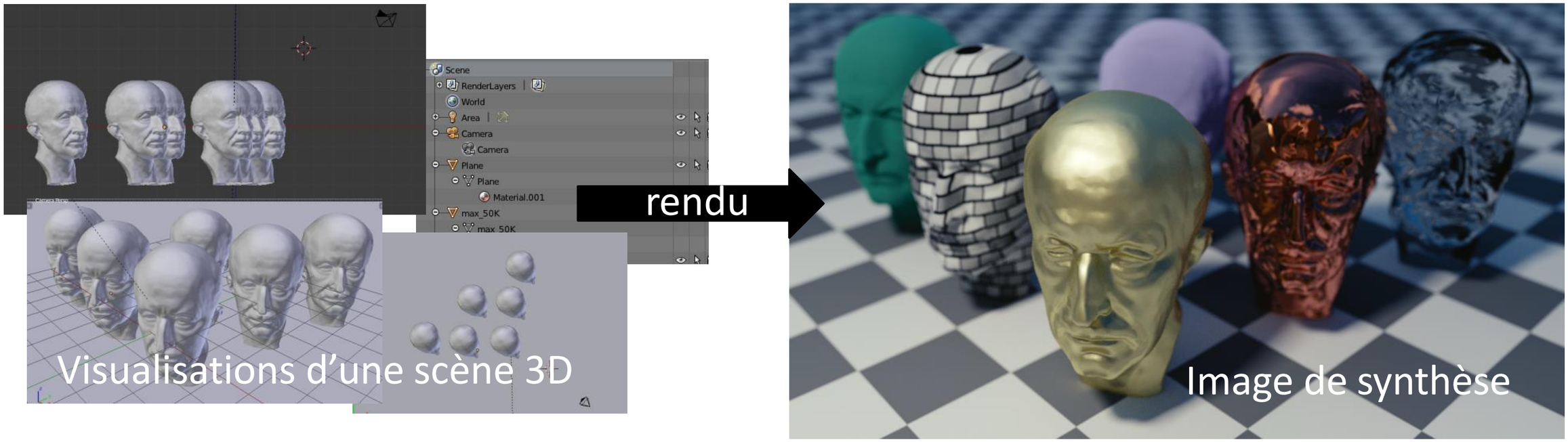
Post-processing

Pipeline GPU

Rendu Temps-Réel

Implémentations

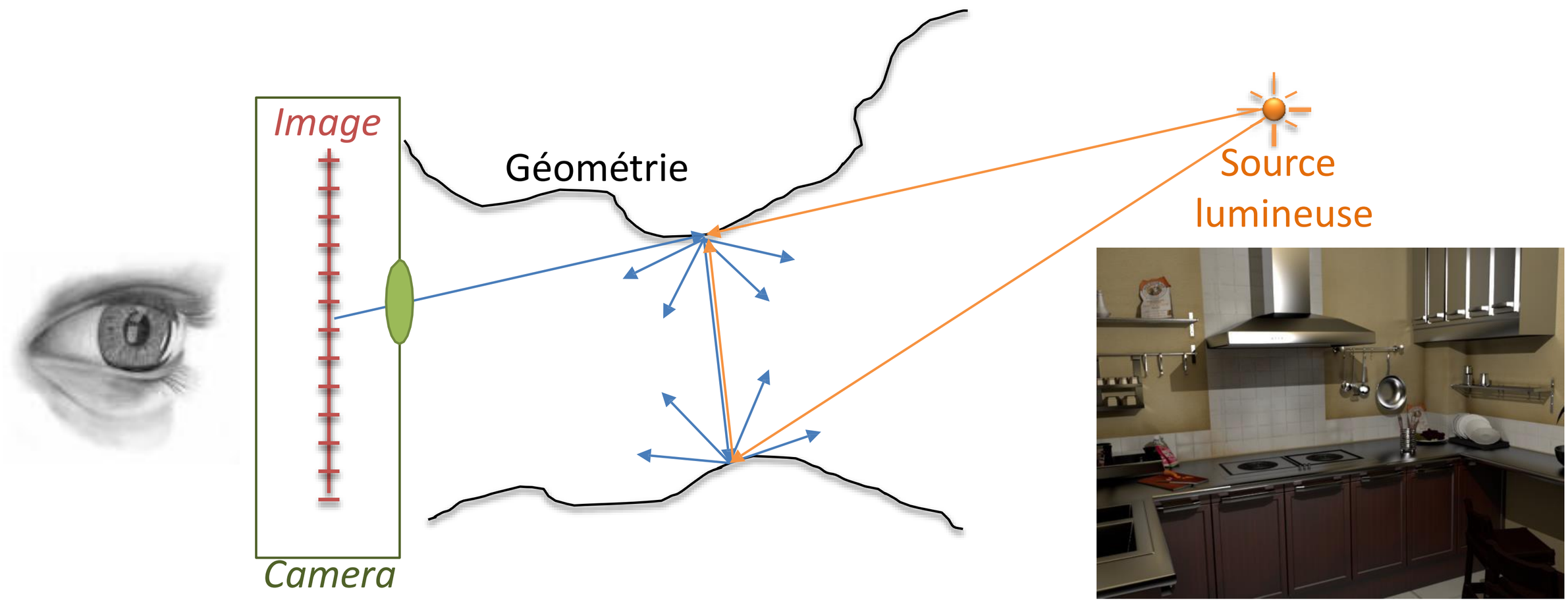
Un processus de simulation



- Rendu = Synthèse d'Image
- Génération d'une image numérique à partir d'une scène 3D
- Réaliste (physiquement plausible) ou expressif

Rendu basé physique

- Simulation du transport de la lumière
  - De la source au capteur
  - Dans une scène virtuelle
  - Pixel de l’image du capteur = mesure radiométrique

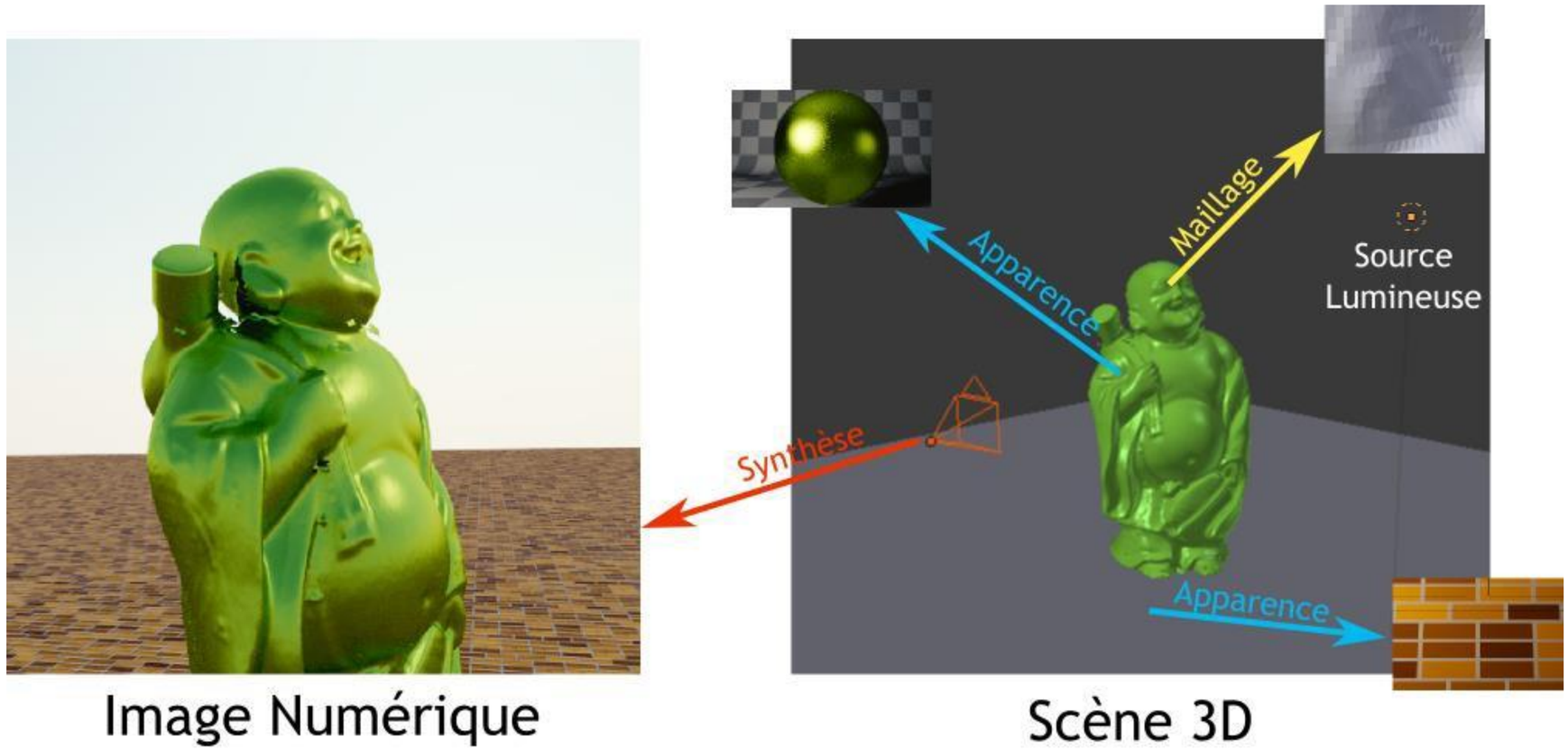


Rendu temps-réel

- Pour les systèmes interactifs
- Approximation géométrique et radiométrique de la scène
- Calcul parallèle (GPU)

SCÈNE ET GÉOMÉTRIE

Modèles de Scène 3D



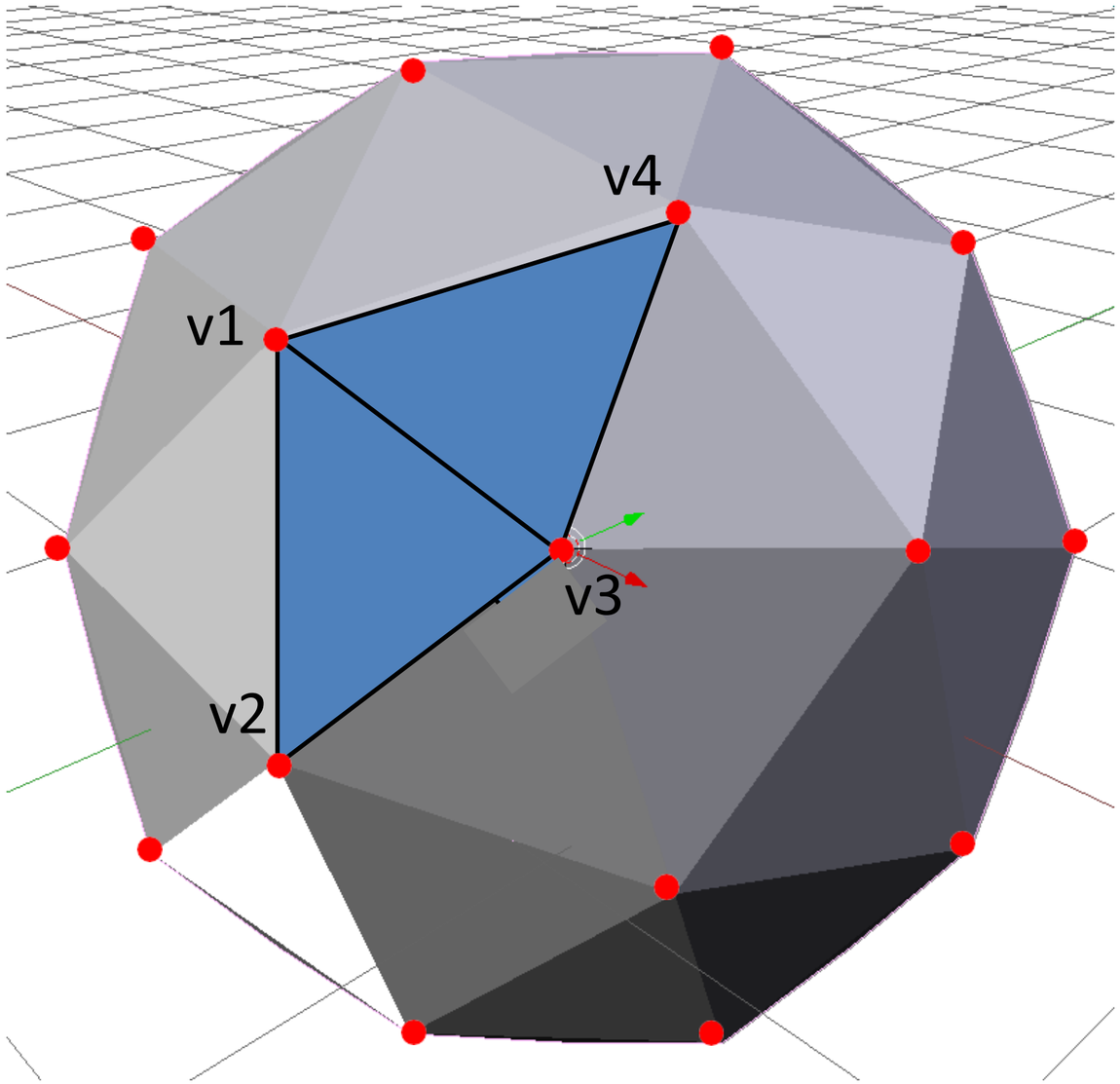
- Une collection de modèles :
  - Capteur (caméra)
  - Géométries
    - Maillages, particules, iso-surfaces, etc
  - Apparence
    - Matériaux, textures
  - Lumières
  - Animation
    - Évolution temporelles des paramètres des autres modèles
    - Physique solide, fluides, corps déformables
    - Interactivité et actuators
- Une structure entre ces modèles
  - Appartenance et hiérarchie
  - Données et instances

Surface Maillée

- Maillage:
  - modèle géométrique dominant en rendu
  - Génération possible à partir des autres modèles

- Définition: un ensemble de faces polygonales F indexant un ensemble de sommets V.
- V: Ensemble de sommets (géométrie)
  - v1 (x, y, z)
  - v2 (x, y, z)
  - v3 (x, y, z)
  - v4 (x, y, z)
- F: Ensemble de faces (topologie)
  - (v1, v2, v3)
  - (v1, v3, v4)

- Outre la position, chaque sommet peut porter d'autres attributs:
  - vecteur *normales,* critiques en rendu.
  - *Couleur par sommet*
  - *Coordonnées de textures (UV)*

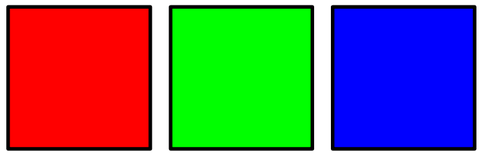


IMAGERIE

Image Numérique

- Grille de pixels couleurs
- Pixel:
  - Photographie numérique : correspond à un point une petite région sur le capteur photosensible
  - Image de synthèse :
    - correspond à un point le capteur virtuel
    - correspond à un ensemble de chemins lumineux
    - correspond à un point 3D de la scène
    - correspond à une collection d’attributs interpolés

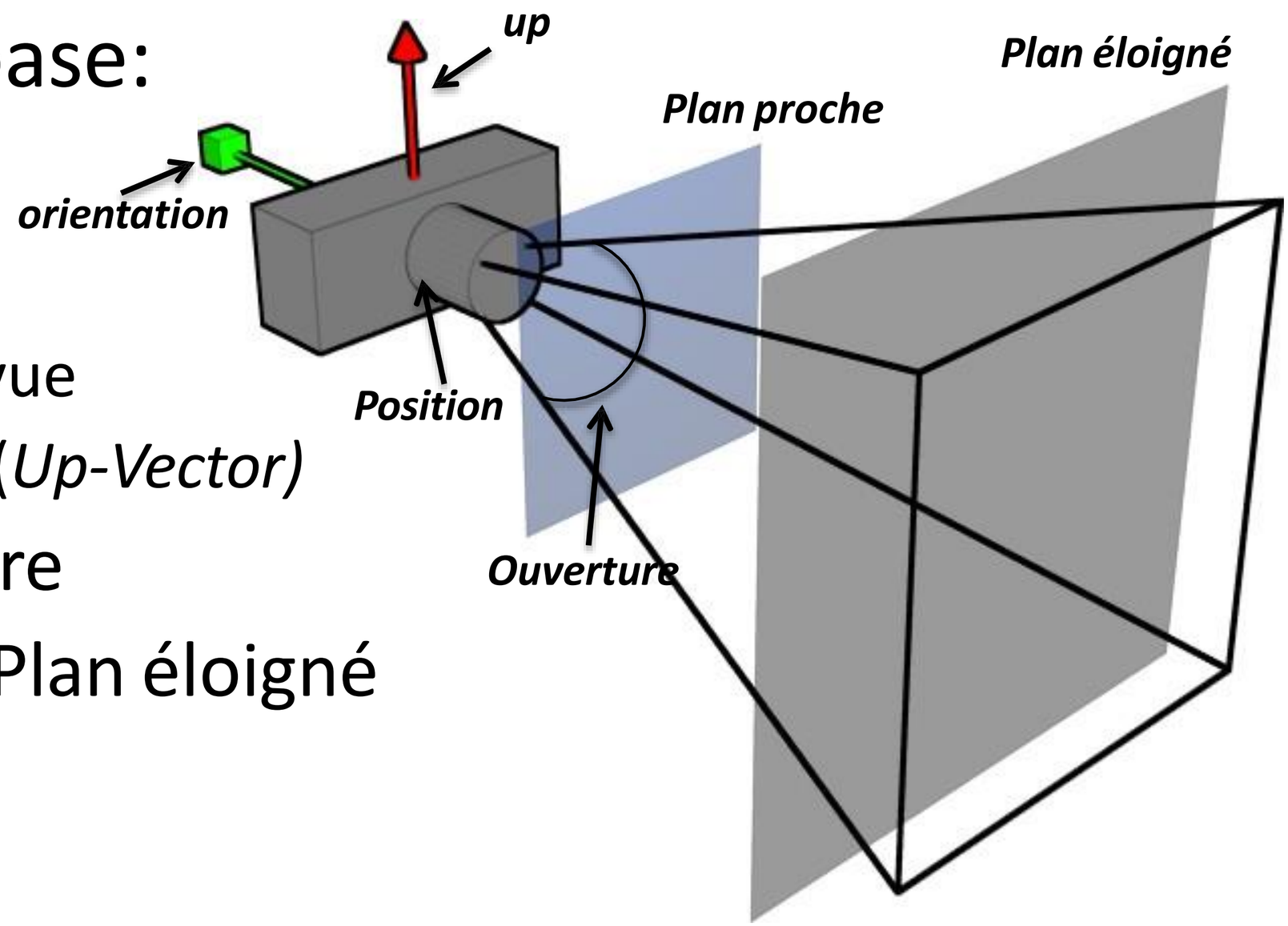
Couleur en image de synthèse

- Espace RGB 
- Précision variable
  - 8bits/composante : standard des APIs, des GPU, du rendu et des écrans
  - 16/32bits par composante : imagerie à haute dynamique (HDR)
    - Nécessite une conversion avant affichage (Tone Mapping)
- RGB**A** : transparence *alpha* par pixel
  - Pose un problème d'ordonnancement
    - Pas directement compatible avec les algorithmes de rendu par projection (Z-Buffer)

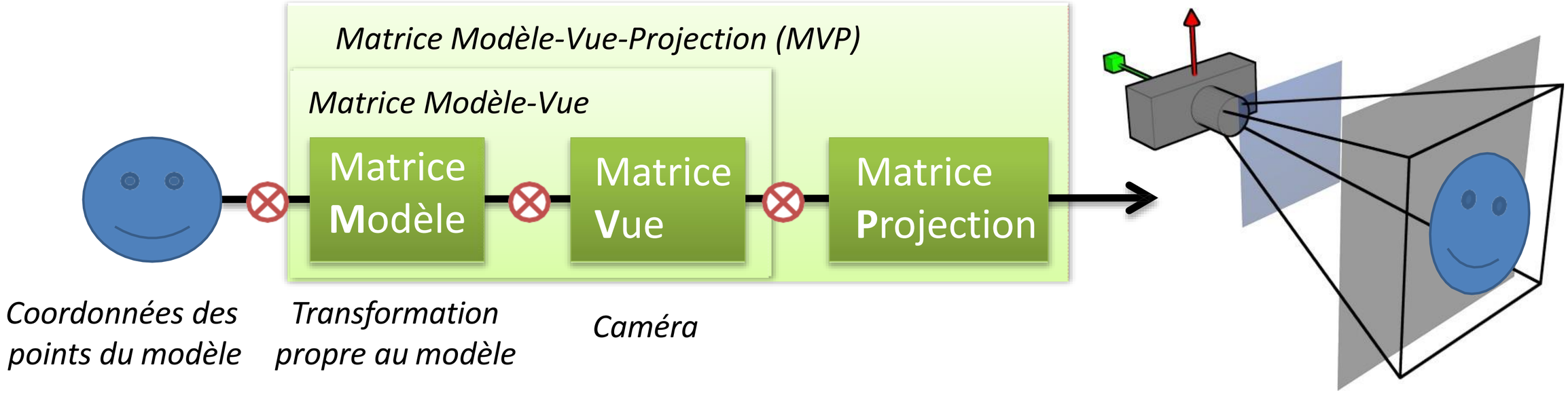
CAMÉRA ET TRANSFORMATIONS

Modèle de Caméra

- En général: *pinhole* camera
- Projection perspectivique
- Paramètre de base:
  - Position
  - Orientation
    - Direction de vue
    - Vecteur haut (*Up-Vector)*
  - Angle Ouverture
  - Plane proche/Plan éloigné



Transformation et Projections

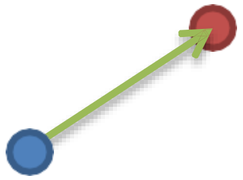


- Représentation par une matrice 4x4
- Transformation rigide
  - Translation
  - Rotation
  - Echelle
- Utilisation: changement de repère pour le placement des géométries dans le repère de la caméra et leur projection

Transformation Affine

Translation

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



Rotation

$$R_X(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \quad R_Y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \quad R_Z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Scaling

$$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

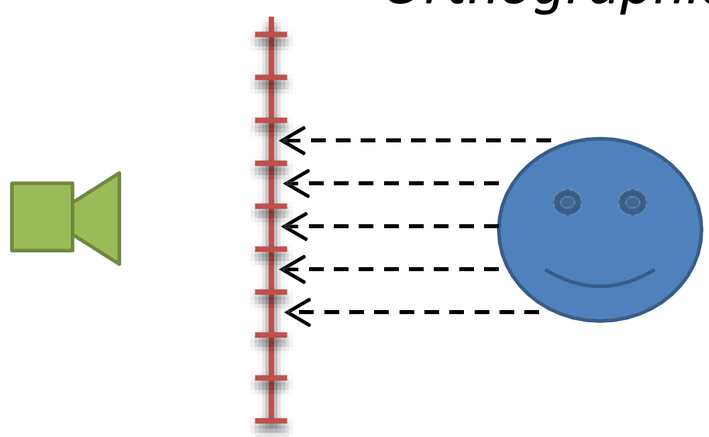
Transformations appliquées pour :

- *Déplacer les sommets des points 3D (matrice modèle)*
- *Les placer dans le repère de la caméra (matrice vue)*

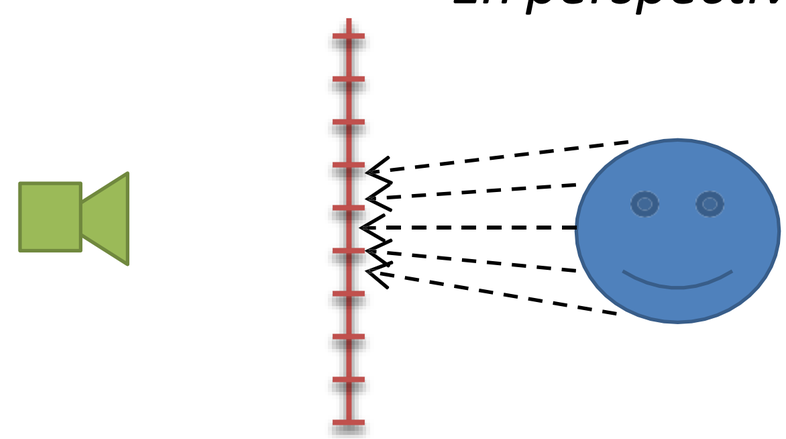
Projection

- Projeter les sommets des polygones (transformés) dans le plan de l'image.
- 2 types de projections:

Orthographique



En perspective



- Encore une fois exprimable à l'aide d'une matrice 4x4 : la **matrice de projection**

Sujets abordés

But du cours

Aperçu des principales idées en matière de graphisme et d'imagerie

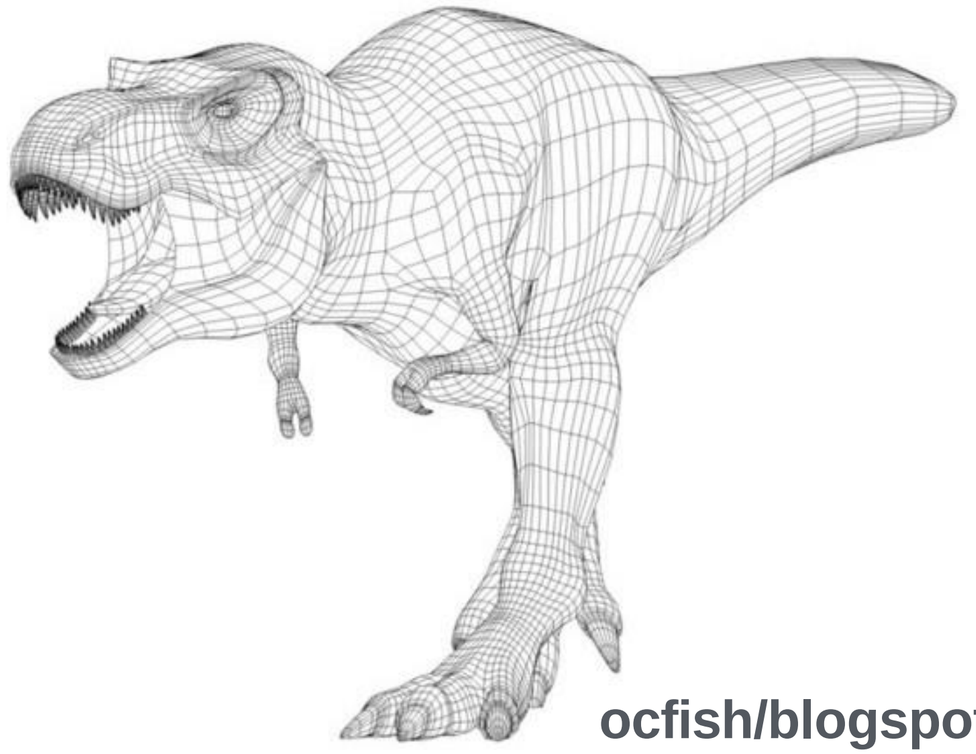
- Modélisation du monde, synthèse d'images
- Graphiques 3D: géométrie, rendu, animation

Acquérir des concepts et des compétences de base

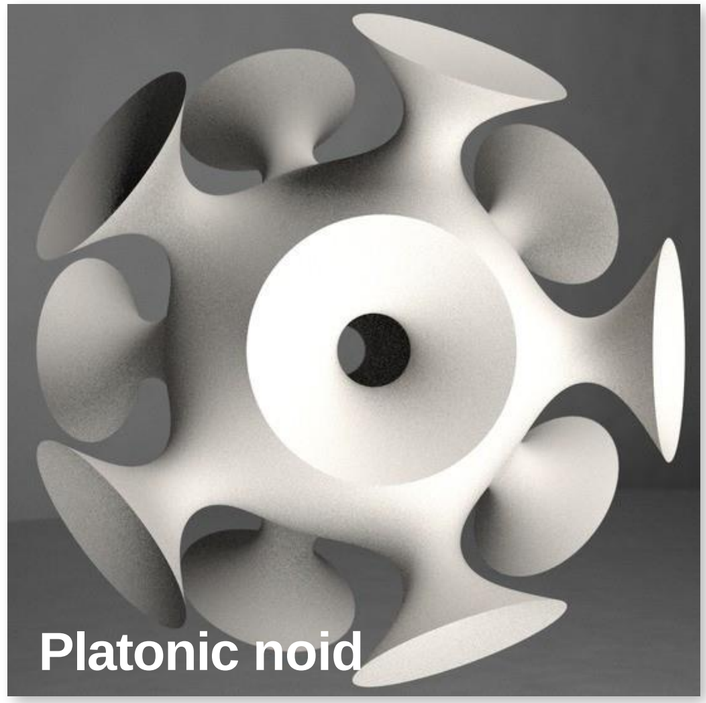
- Représentations (géométrie, images, transformations,…)
- Algorithmes (échantillonnage, subdivision, lancer de rayons,…)
- Technologie (GPU, écrans, caméras,…)

Modélisation géométrique

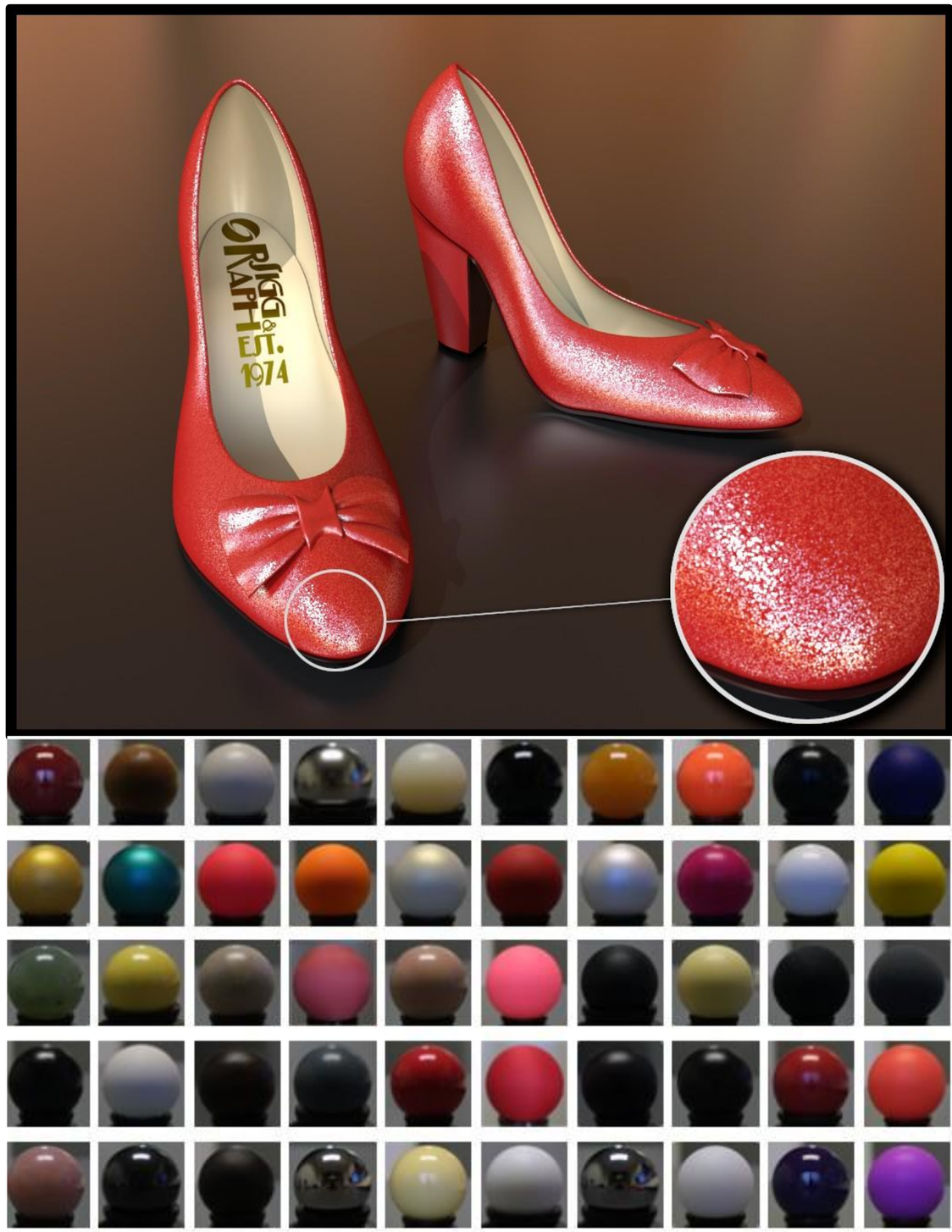
(Stanislav Orekhov)


ocfish/blogspot

[Kaldor 2008]


Platonic noid

Modéliser les propriétés des matériaux



Modéliser la lumière

Transport de la lumière et image de synthèse

Photograph (CCD) vs. computer rendering

Animation et simulation physique

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
  - Rasterisation: pipeline graphique, z-buffer, OpenGL, shaders
  - Transformations de l'espace
  - Textures, normales maps, reflexions & transparence
  - Intro modélisation géométrique
    - Maillage polygonal, surfaces de subdivisions
  - Intro animation
  - etc.

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 maitrisé SFX,(films)
    - qques minutes par images
    - haute qualité visuel en un temps raisonnable

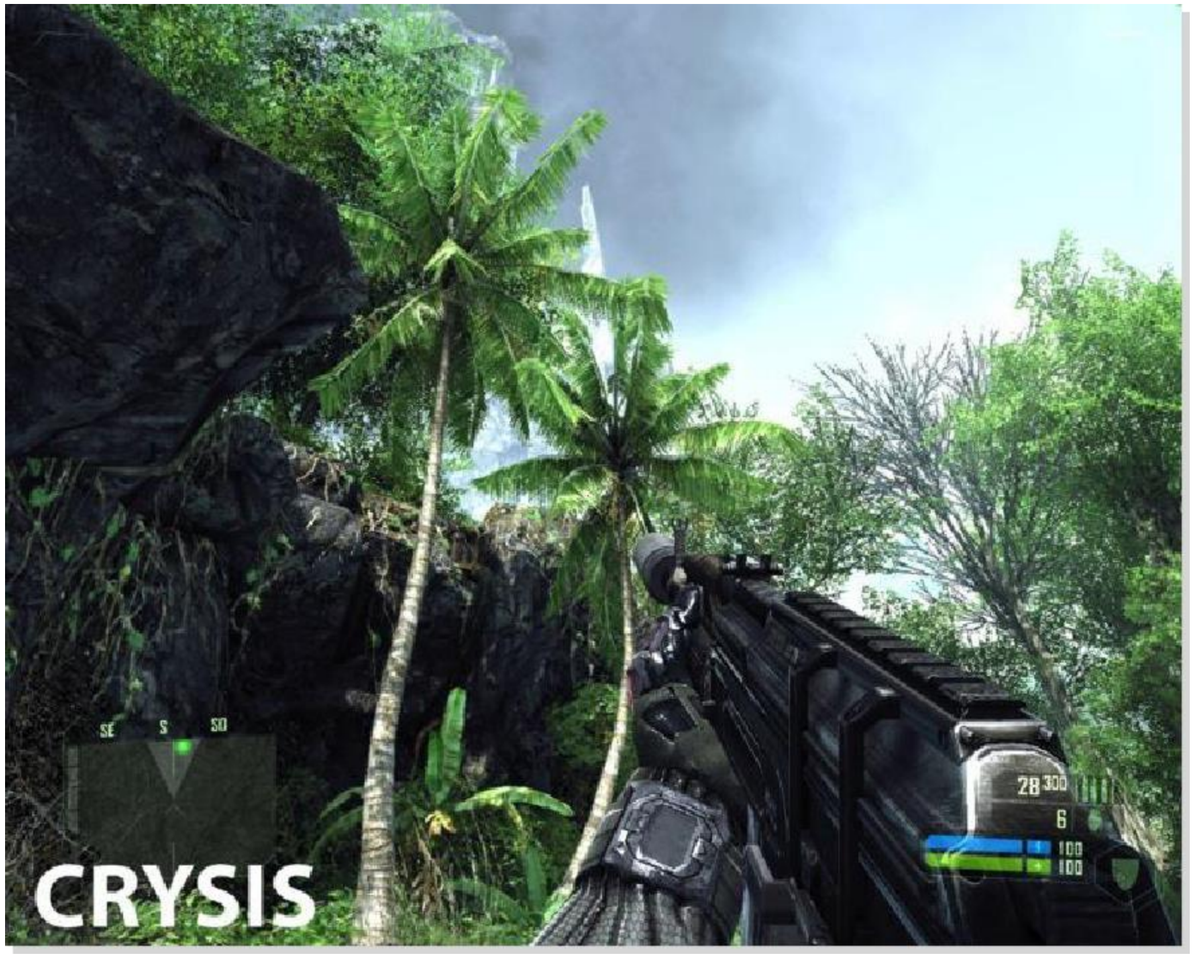
    } **Pipeline maison**
    (+OpenGL)

  - rendu physiquement réaliste (conception, biologie, etc)
    - de qques heures à quelques jours par images...
    - maitrise de l'erreur

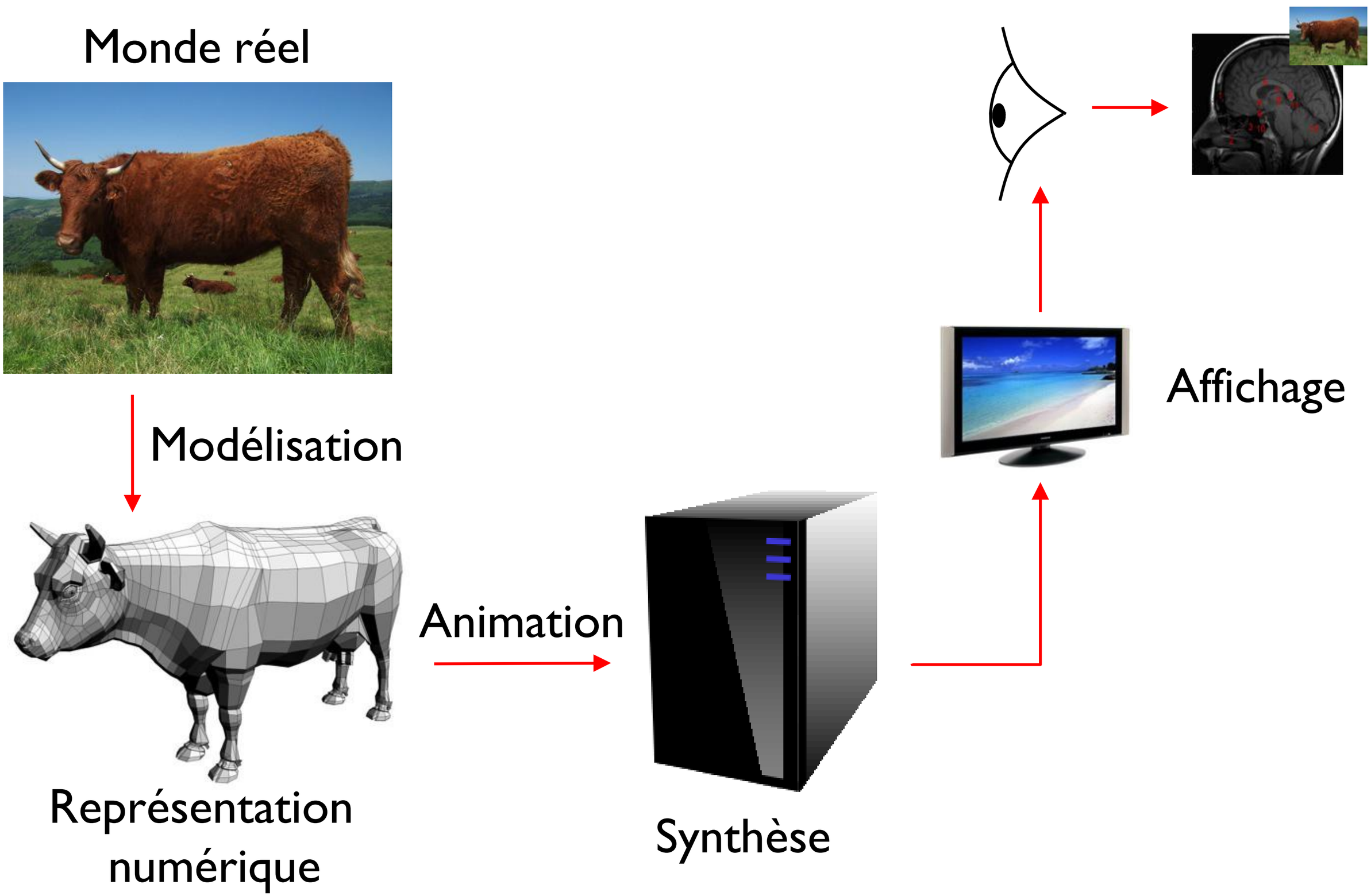
    } **Pipeline maison**

Question

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



Problèmes à résoudre



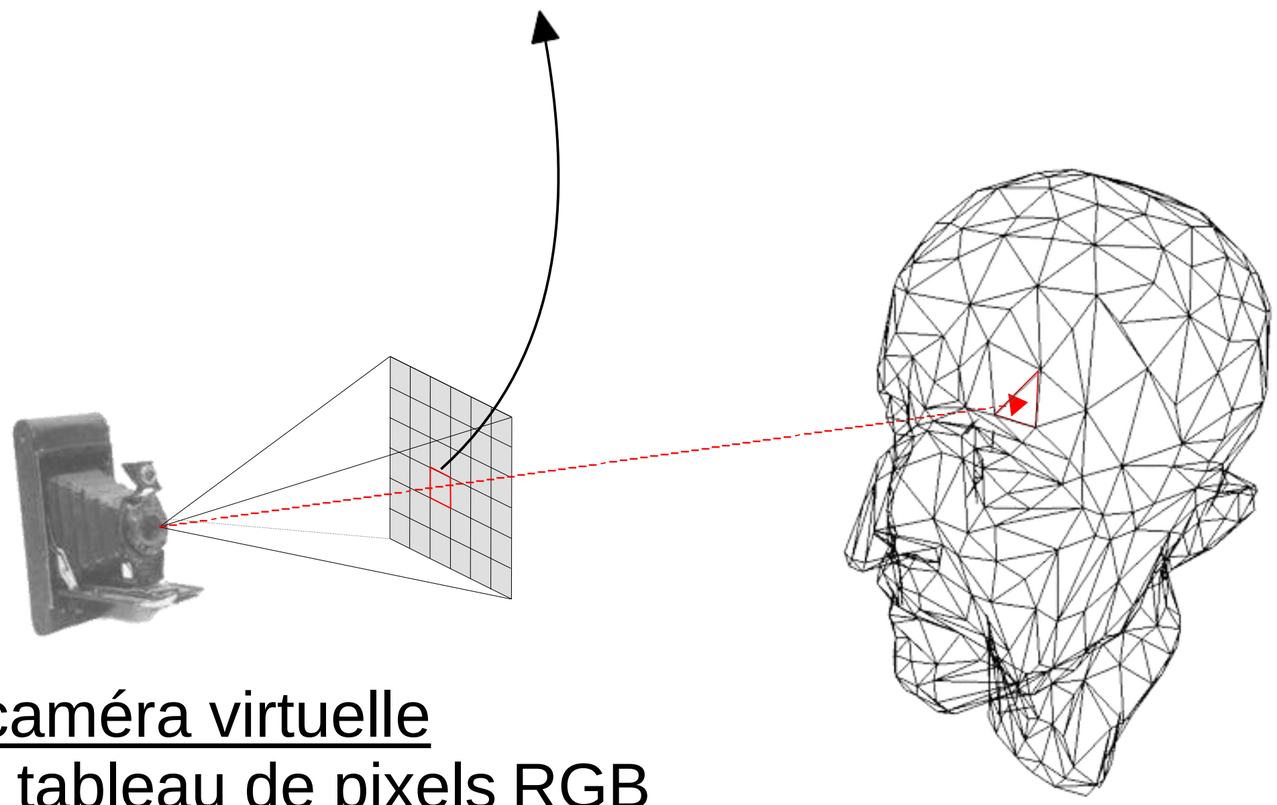
Problèmes à résoudre

- **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 ?

<u>problématique :</u>

- quelle est la couleur de chacun des pixels ?

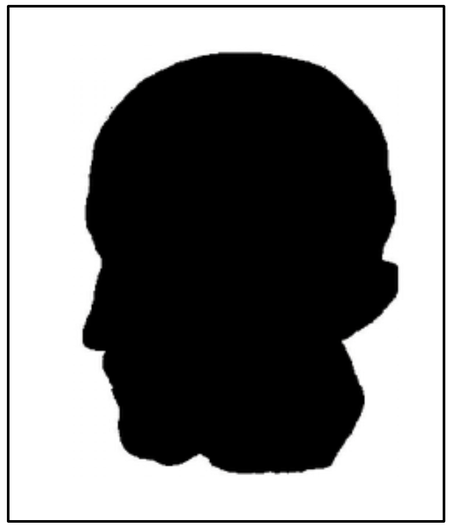


<u>caméra virtuelle</u>

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

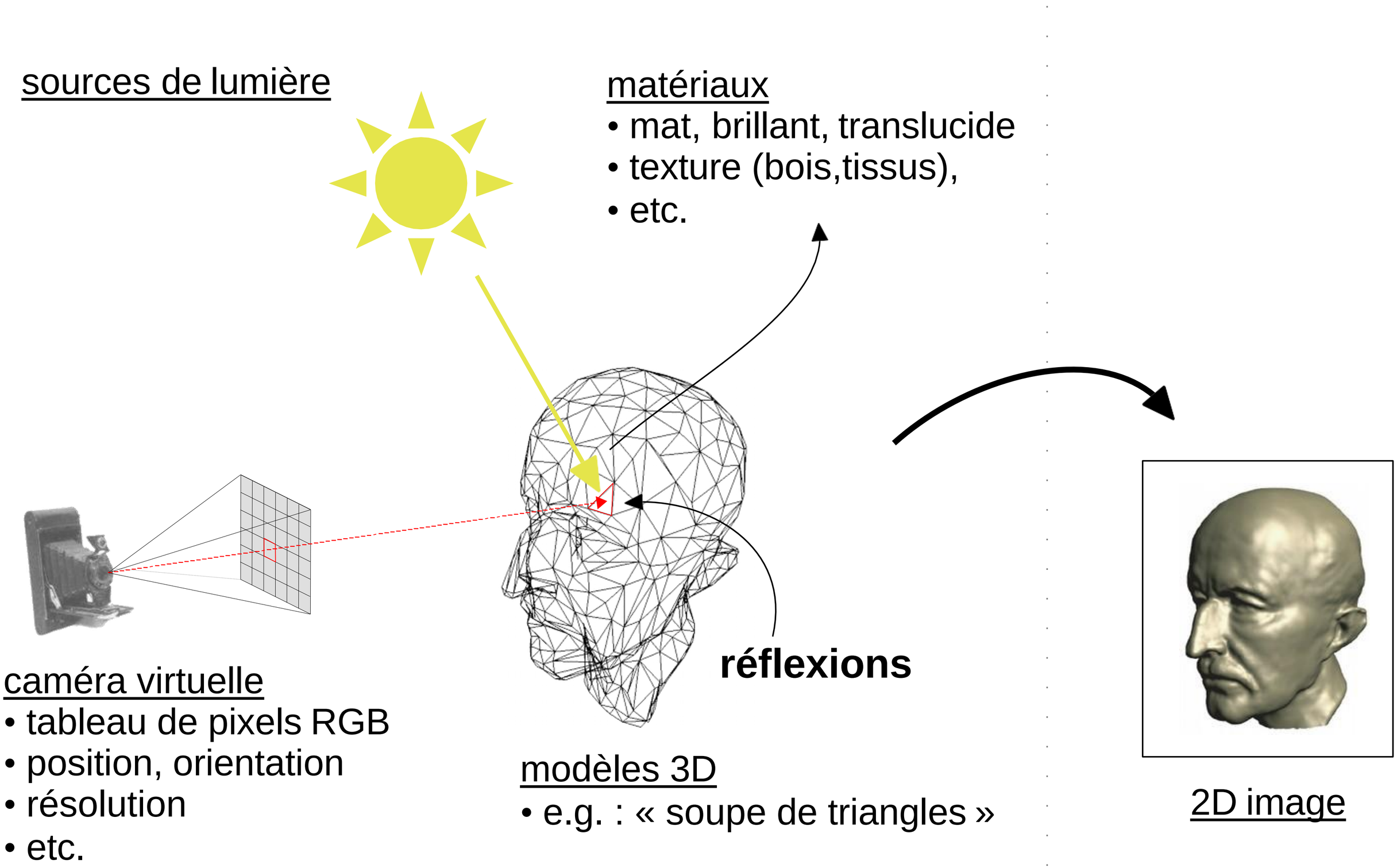
<u>modèles 3D</u>

- e.g. : « soupe de triangles »



<u>2D image</u>

Comment synthétiser une image ?



Synthèse d'images

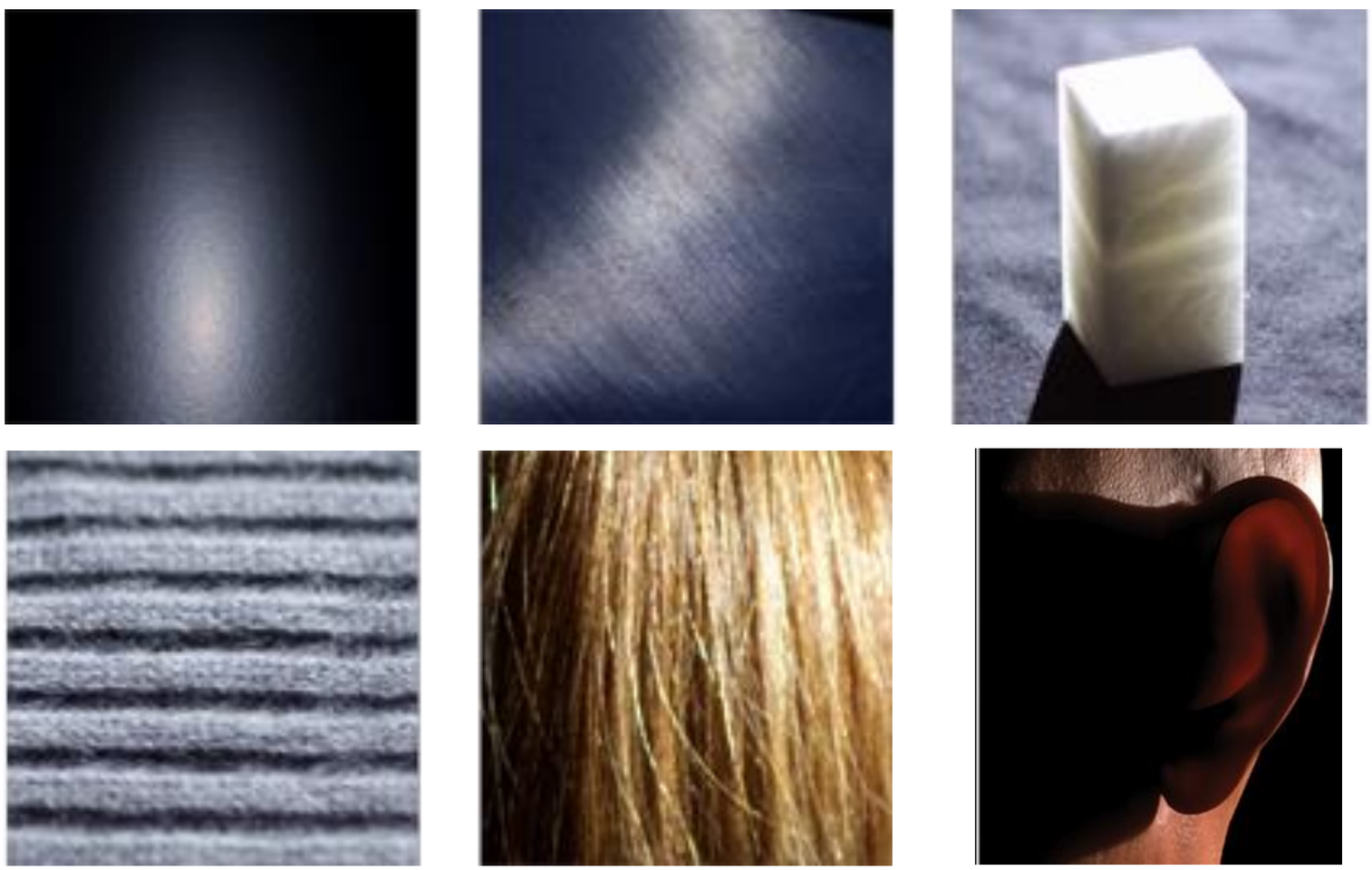
$\leftrightarrow$

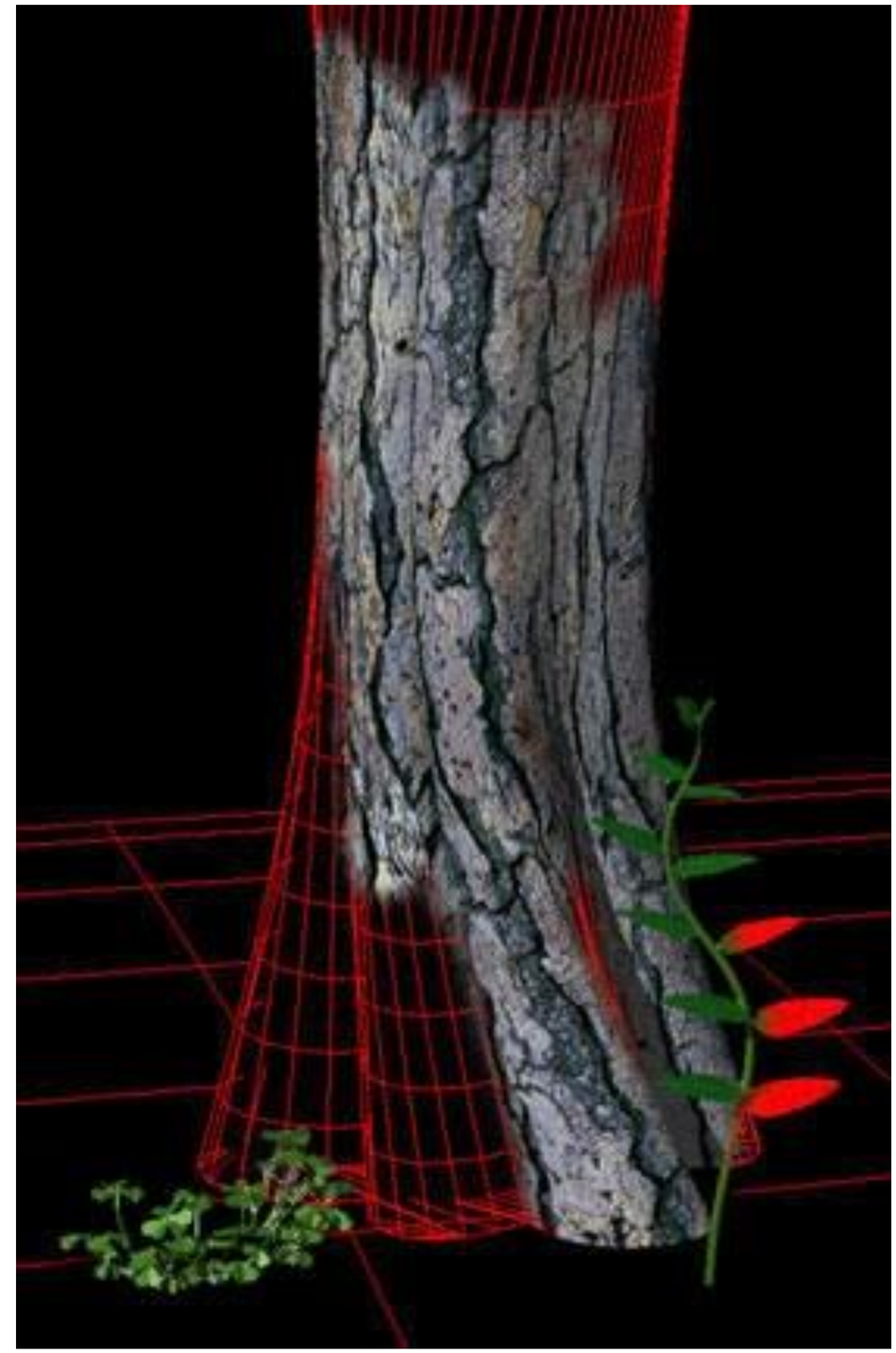
Simulation des interactions entre la lumière, la matière, les formes, et le milieu ambiant

Apparence, matériaux

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

$\Longrightarrow$ **m2 (Informatique graphique)**

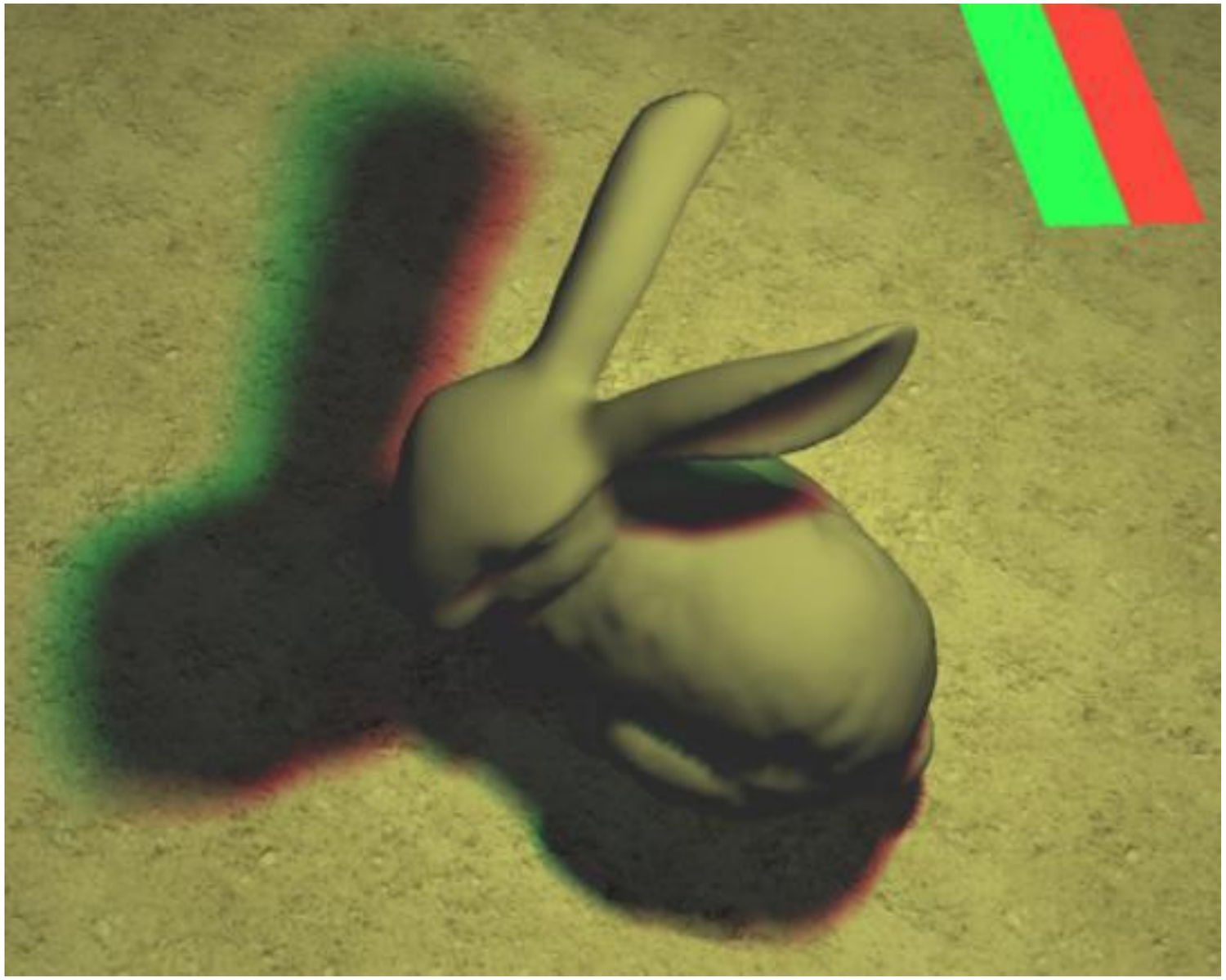




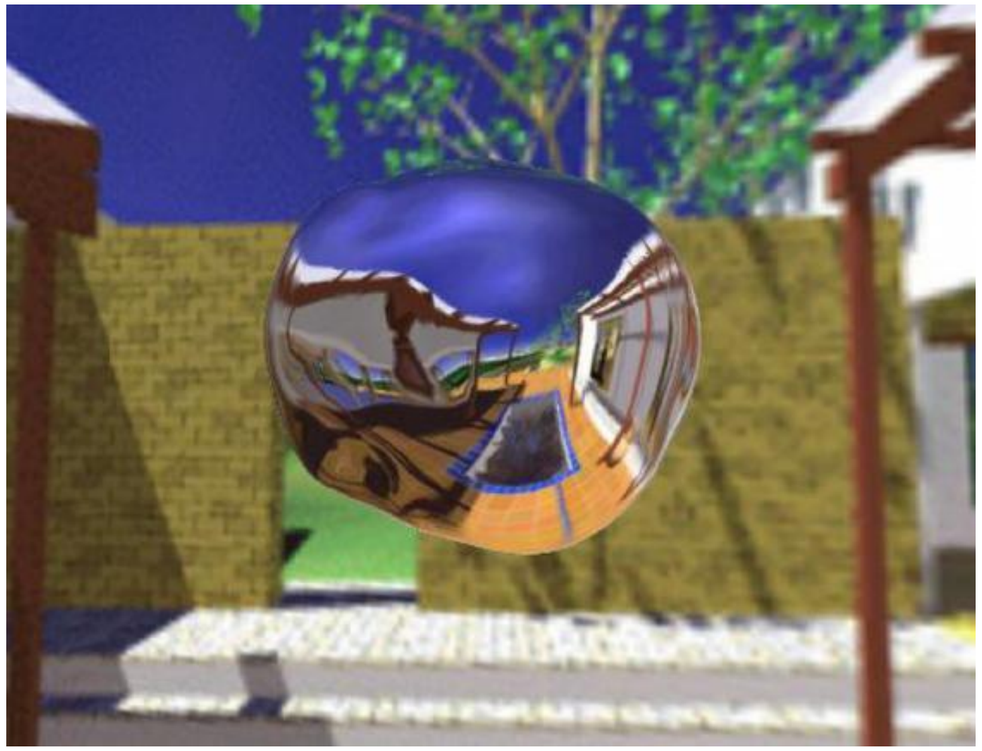
Ombres et lumière

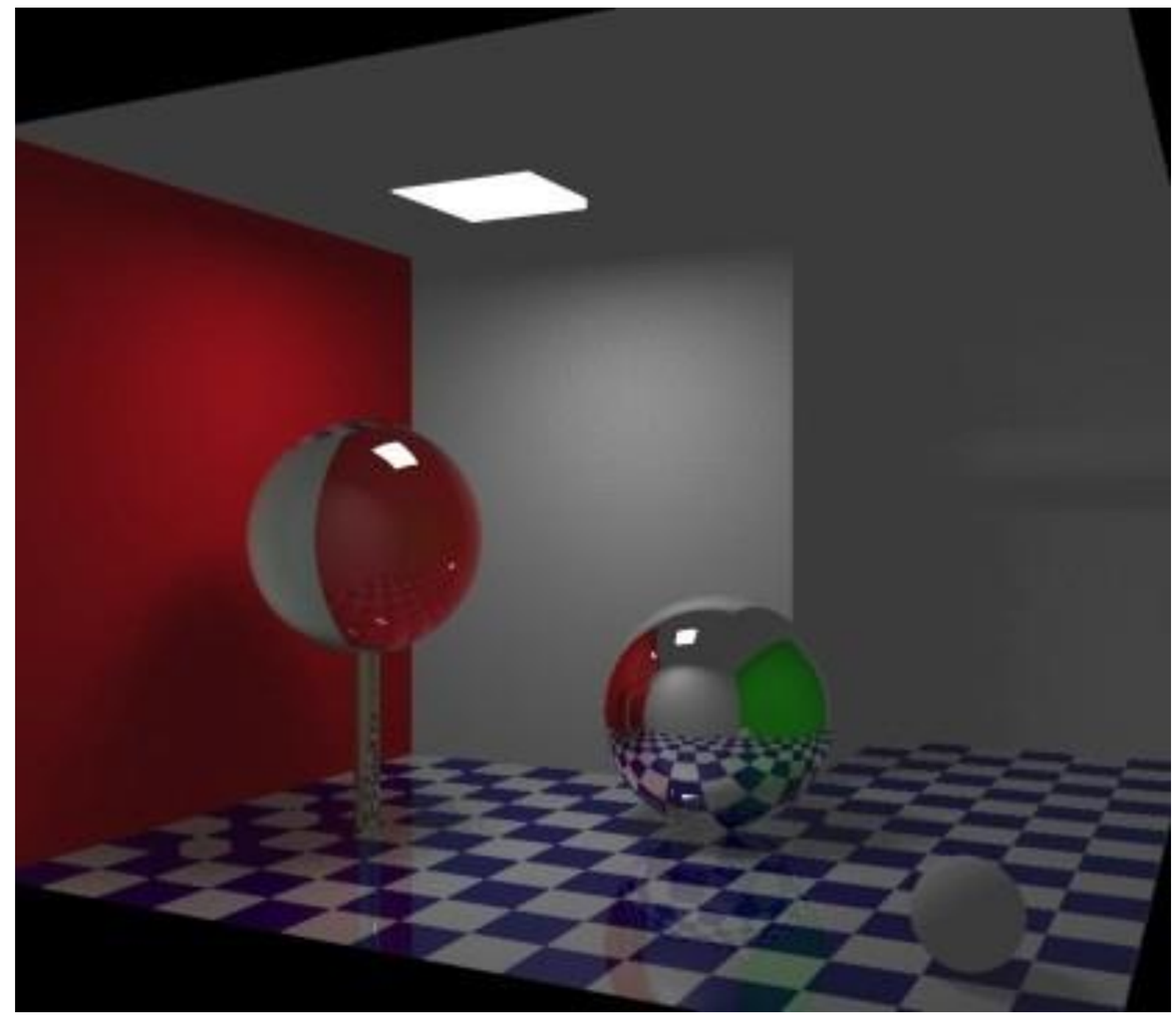
Qui est éclairé par une source?

- visibilité
- géométrie



Réflexions miroirs / réfractions



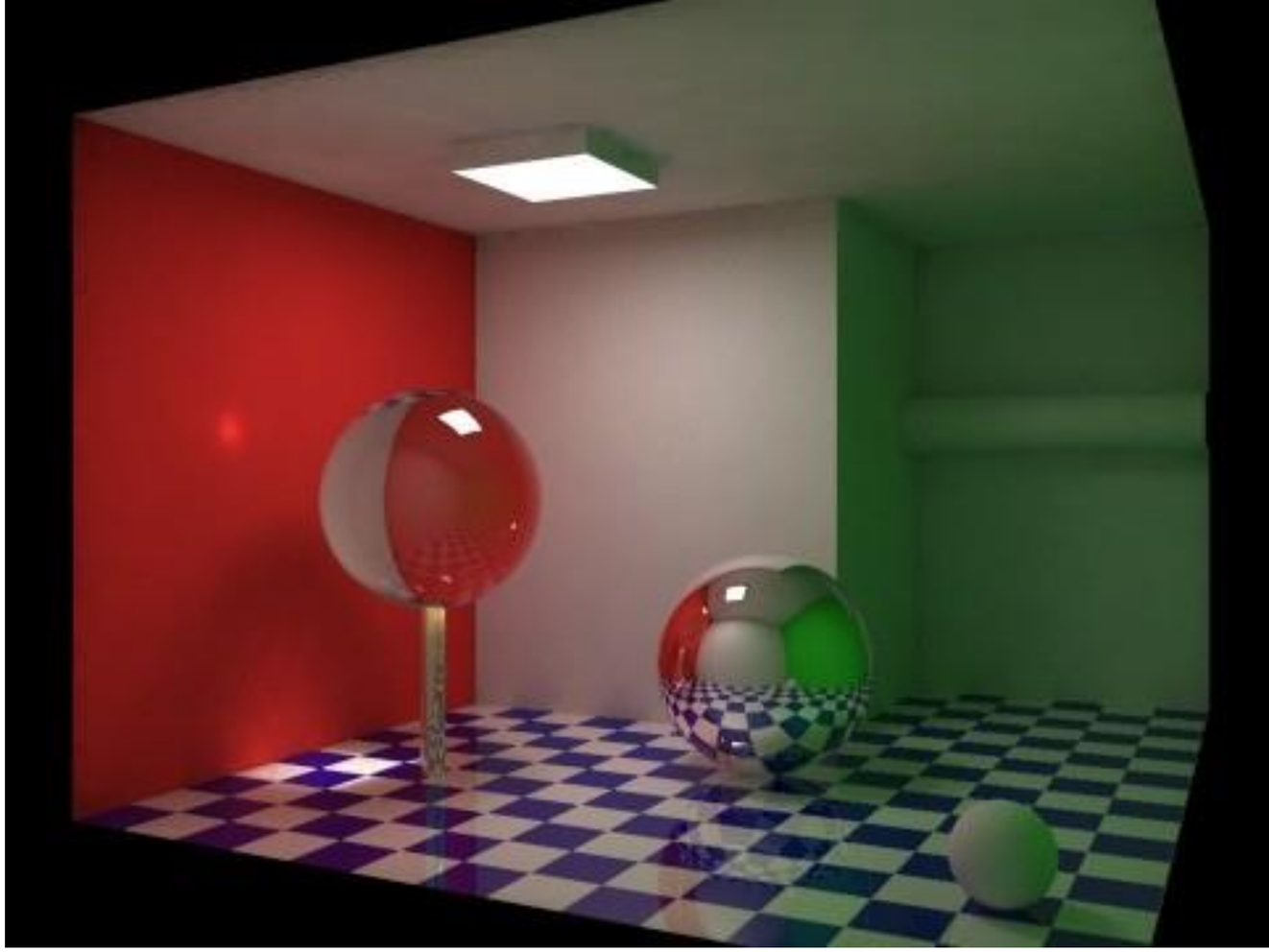


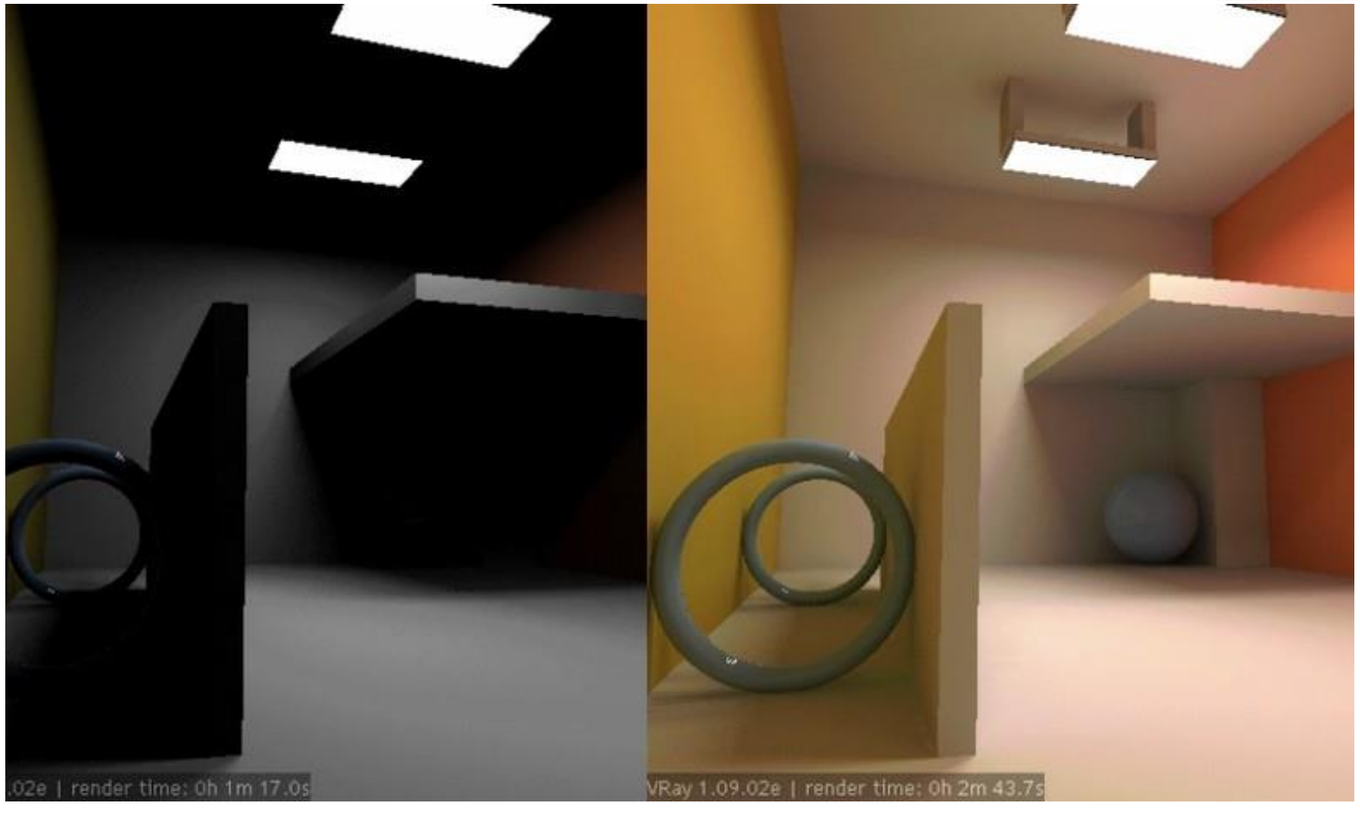
Éclairement direct vs indirect

Inter-réflexions

Sources étendues

Illumination globale

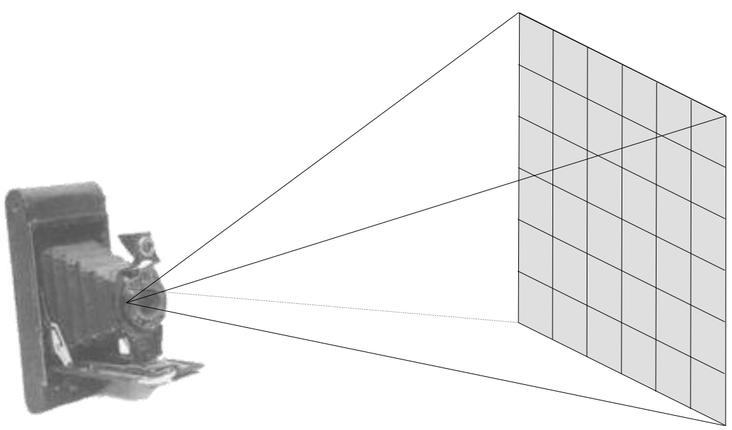


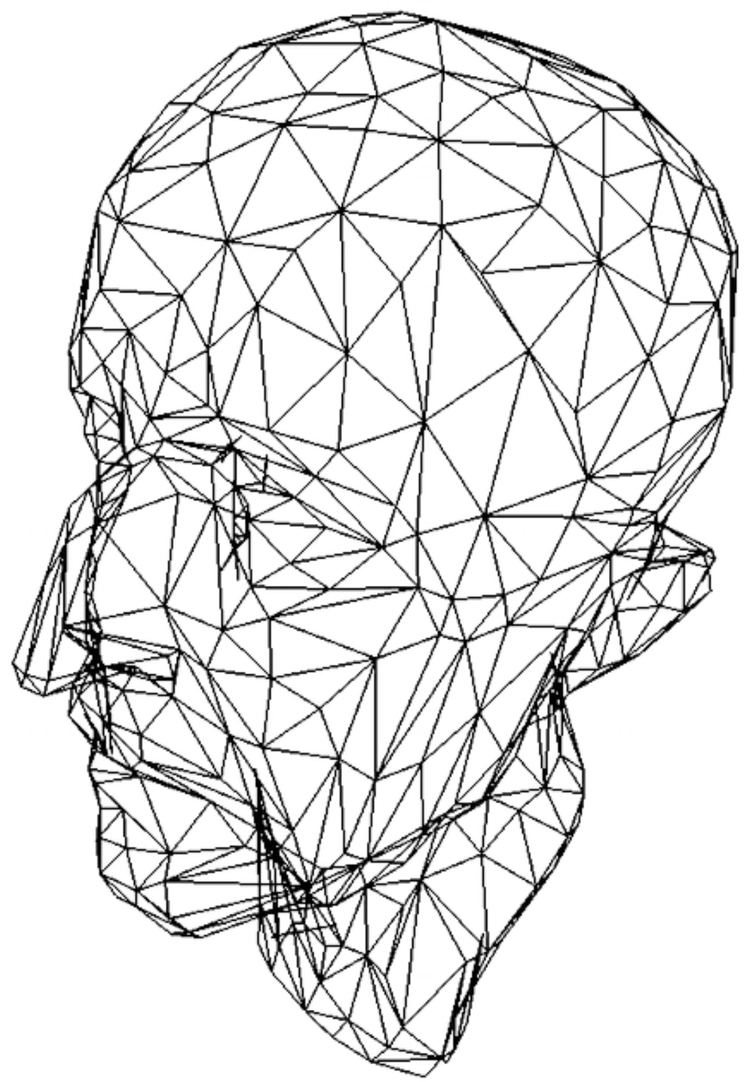


direct + indirect

carte d'environement

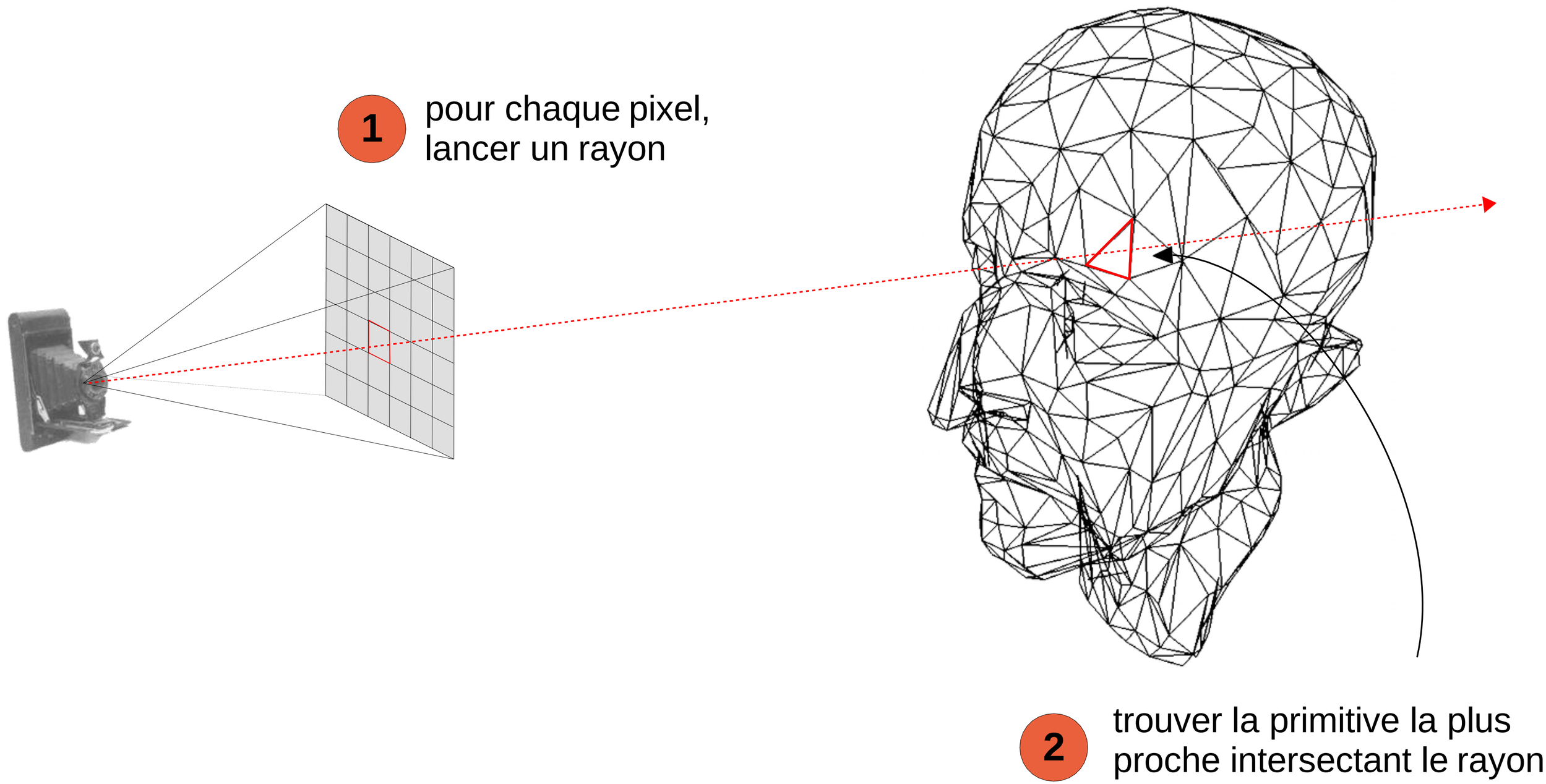
Visualiser une scène 3D





Visualiser une scène 3D : deux approches

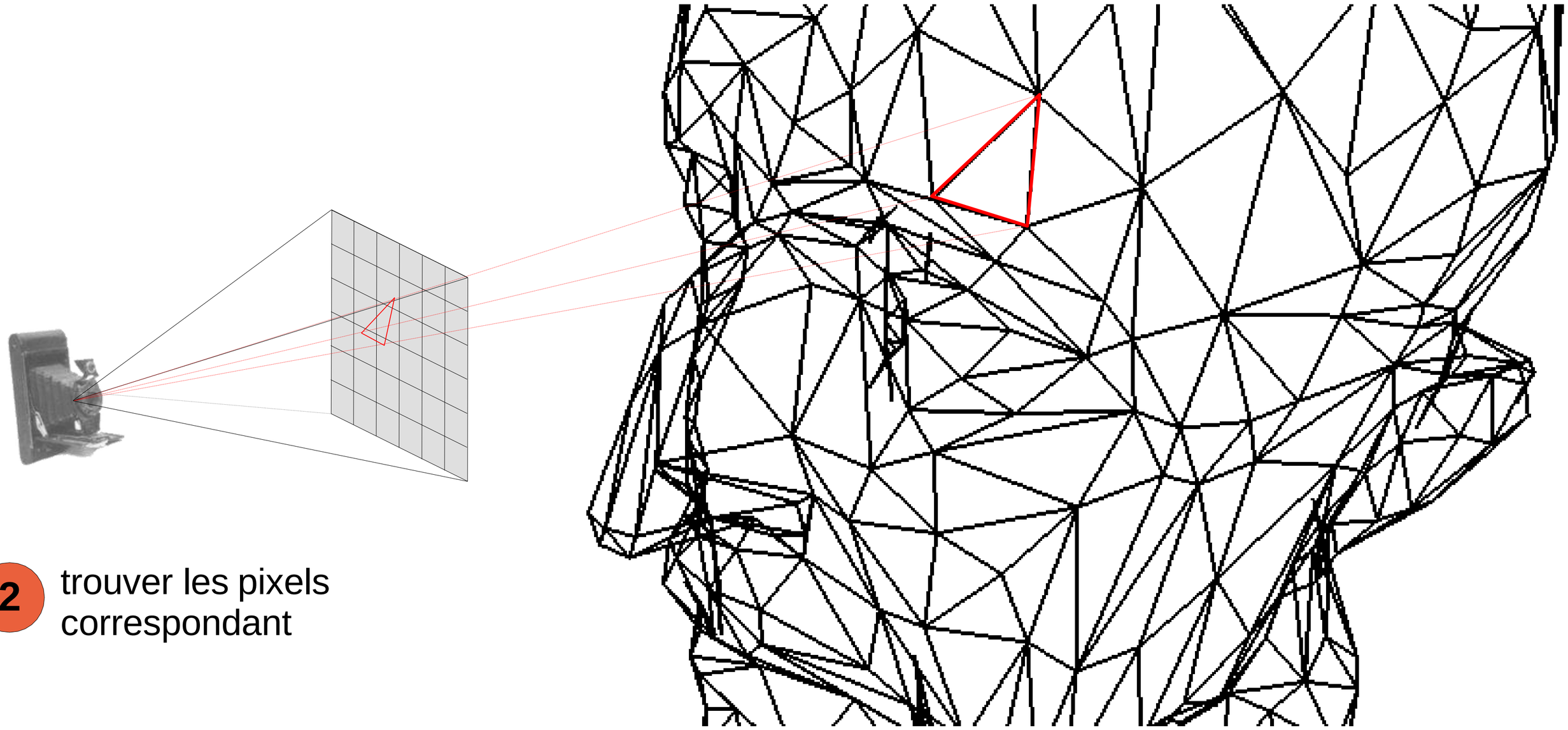
- **1ère approche : lancer de rayons (Ray-tracing – Ray casting)**



Visualiser une scène 3D : deux approches

- **2ème approche : rastérisation**

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



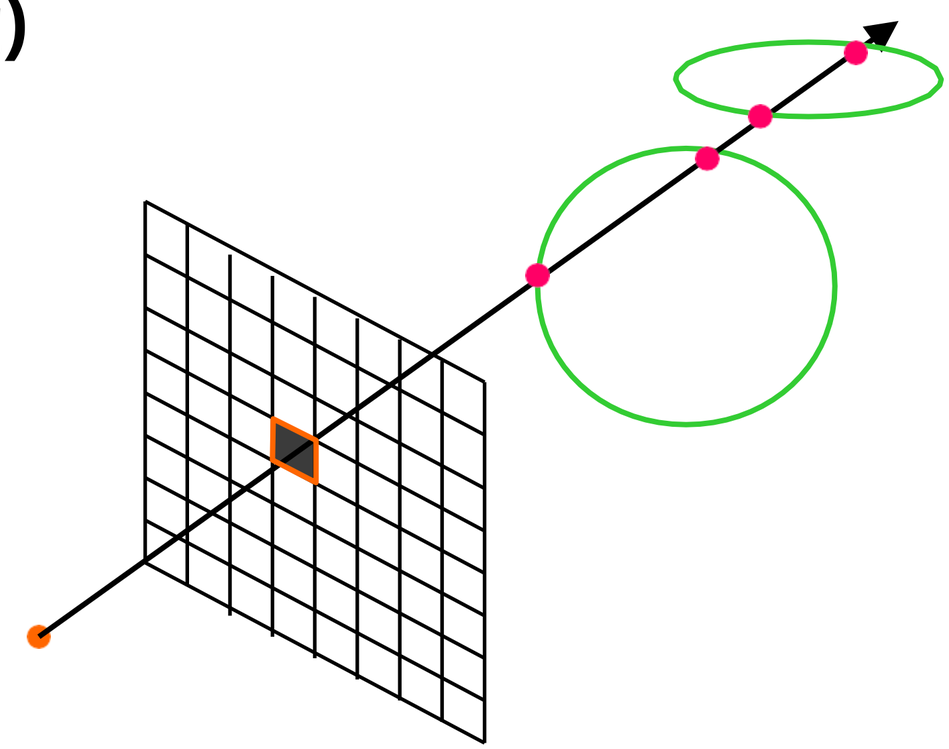
2 trouver les pixels
correspondant

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**

Rayons lumineux

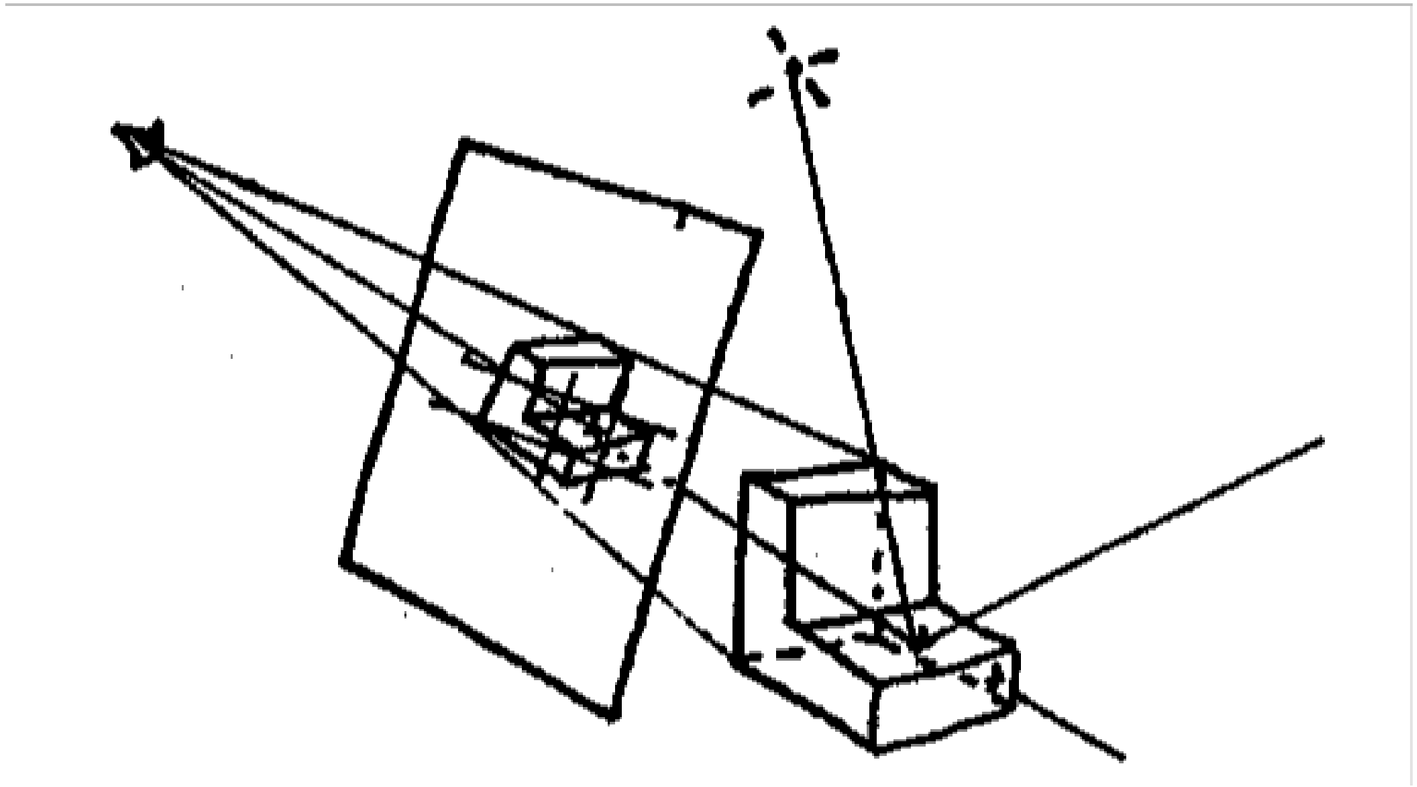
Trois idées sur les rayons lumineux

1. La lumière se déplace en ligne droite (principalement)
2. Les rayons lumineux n'interfèrent pas les uns avec les autres s'ils se croisent (les photons n'ont pas d'interactions; ignorant les effets des vagues comme les interférences)
3. Les rayons lumineux se déplacent des sources lumineuses à l'œil (mais la physique est invariante sous inversion de chemin - réciprocité).

Lancer de rayon

Appel 1968 - Ray casting

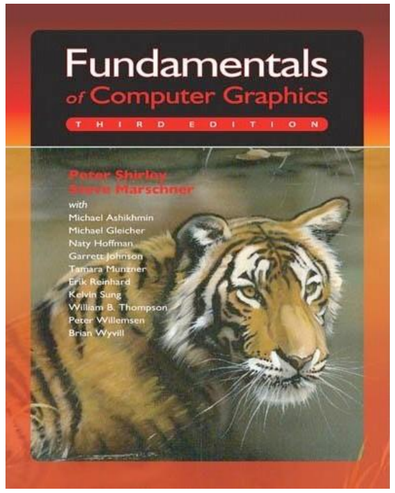
1. Générer une image en lançant un rayon par pixel
2. Vérifier les ombres avec un rayon lancé vers la lumière

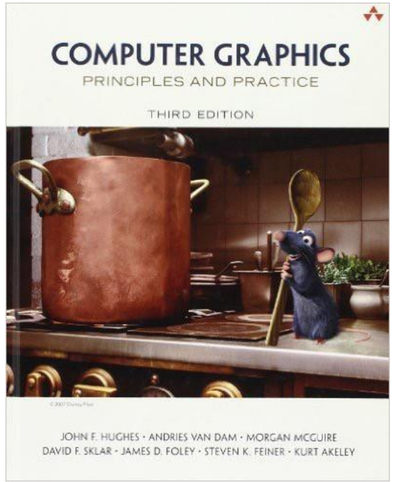


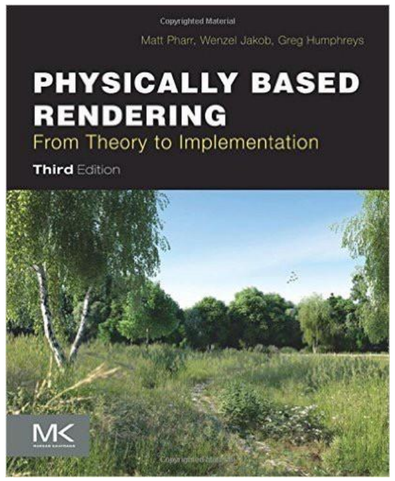
Ressources

Livres

- Fundamentals of Computer Graphics by P. Shirley, S. Marschner, et al.
- Computer Graphics: Principles and Practice (3rd Edition) by Hughes, van Dam, et al.
- Physically Based Rendering, Third Edition: From Theory to Implementation by Pharr, Jakob and Humphreys







Ressources

Cours de

- Jonathan Ragan-Kelley & Ren Ng de Berkeley
- Tamy Boubekeur Télécom ParisTech
- Pierre Benard Université de Bordeaux

Questions ?