

# IMAGERIE NUMÉRIQUE

*C. Schlick*

## OBJECTIFS

- Maîtriser les ***modèles mathématiques*** et les ***techniques algorithmiques*** impliqués dans la manipulation d'images 2D et de scènes 3D
- Devenir un ***utilisateur avancé des logiciels*** de création ou de traitement d'images et de scènes, capable de comprendre les modèles et les algorithmes utilisés en coulisse par ces outils
- Etre capable de ***créer ses propres algorithmes*** de création, de traitement ou de manipulation d'images 2D et de scènes 3D

## ORGANISATION

- **CM** (*7 séances en amphi*)  
Semaine 1 : 3 séances aujourd'hui  
Semaines paires : mardi 14H00 – 15H20
- **TD** (*12 séances sur machine au CREMI*)  
G1 : mercredi 14H00 – 15H20  
G2 : mercredi 15H30 – 16H50  
G3 : mercredi 17H00 – 18H20

## ORGANISATION

- **Ressources pédagogiques**
  - WEB : [www.labri.fr/perso/schlick/inum](http://www.labri.fr/perso/schlick/inum)
  - MAIL : [inum@gmx.fr](mailto:inum@gmx.fr)
- **Contrôle des connaissances**
  - Projet en binôme (sujet 2D et sujet 3D) = 1 / 3
  - Examen final sur machine, en binôme = 2 / 3
  - Exercices hebdomadaires (2D ou 3D) = bonus
- **Dépôts sur votre espace [www](http://www) au CREMI**
  - Récupération automatique par robot logiciel

# OUTILS LOGICIELS

- **Partie 2D : Environnement JupyterLab**

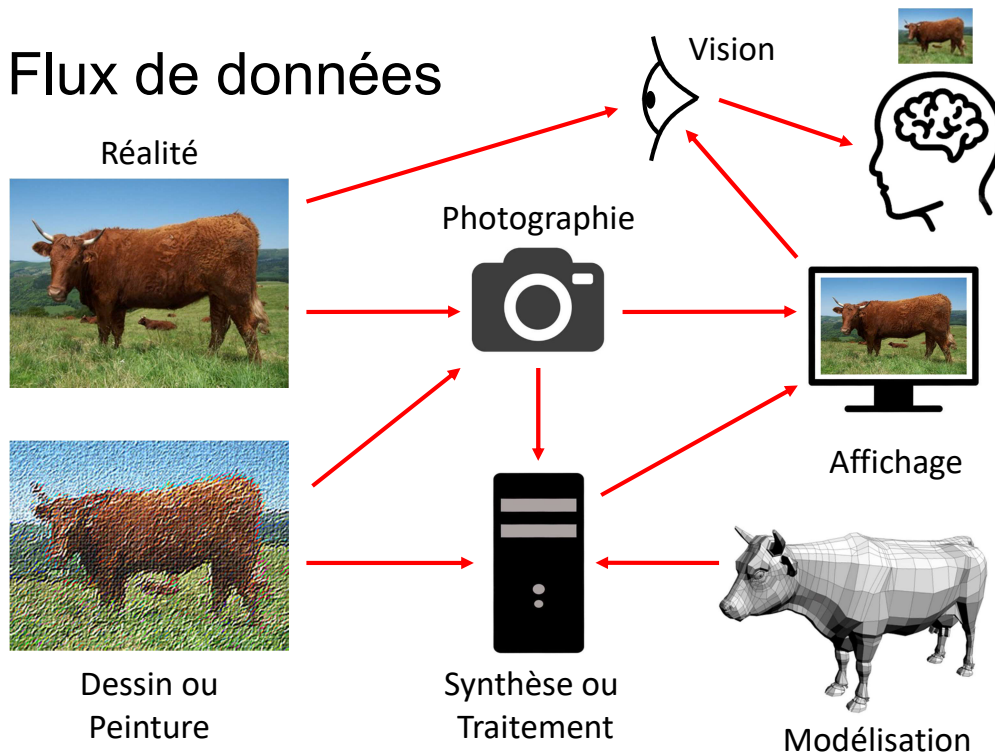
- Création de "notebooks" en Python *[.ipynb]*
- Utilisable en ligne à l'adresse :  
<https://jupyterhub.emi.u-bordeaux.fr>
- Installation sur poste personnel :  
<https://www.anaconda.com>

- **Partie 3D : Logiciel Wings3D**

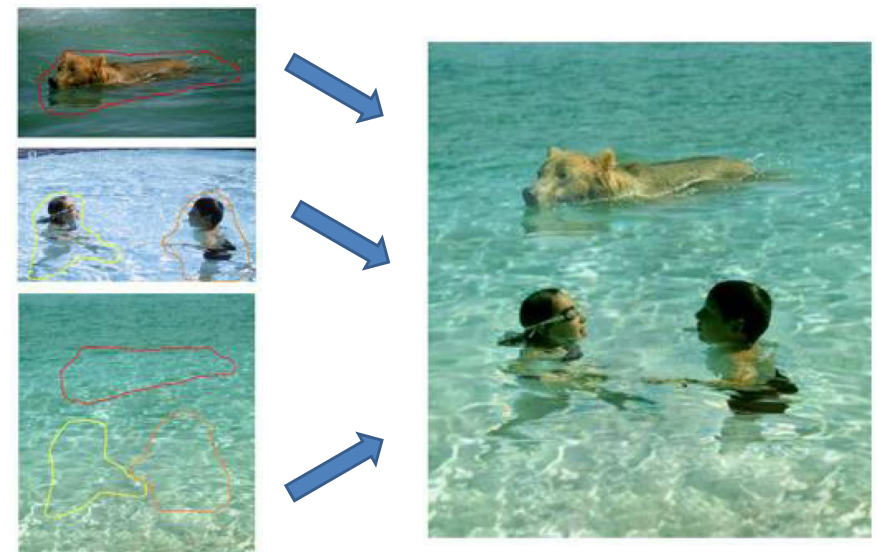
- Modeleur 3D = création interactive de scènes
- Installation sur poste personnel :  
<https://www.wings3d.com>

## Quelques exemples de techniques abordées

### Flux de données



### Composition d'images (seamless)



## Transfert de style entre images



Ansel Adams, *Clearing Winter Storm*

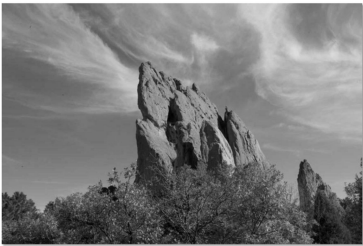
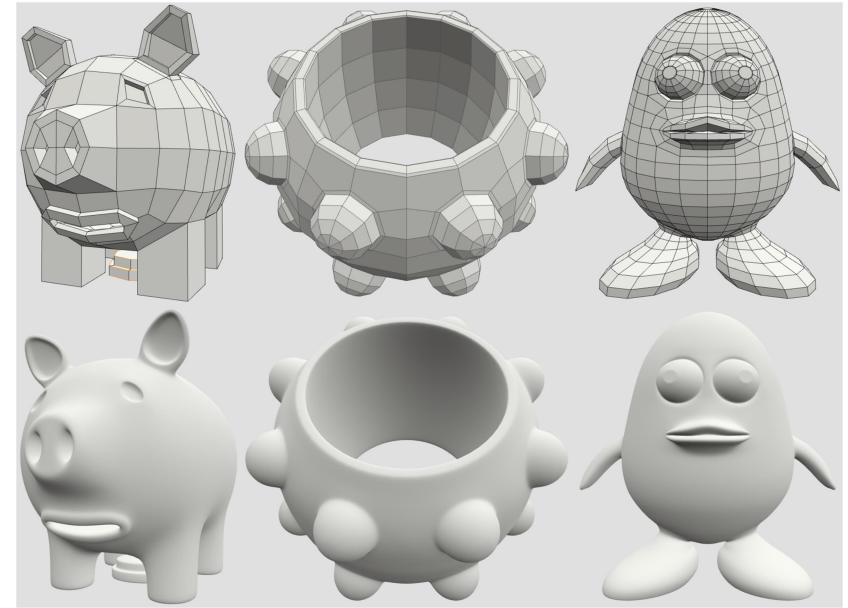


Photo d'amateur (un peu pourrie)



Après transfert de style (nettement moins pourrie)

## Modélisation géométrique



## Modélisation optique



## Simulation des interactions lumineuses





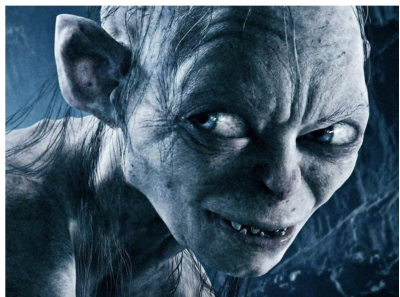
# AVERTISSEMENT

Même si les exemples présentés ici peuvent donner l'impression d'un contenu artistique, voire ludique, il est important de réaliser que l'enseignement de l'UE inclut :

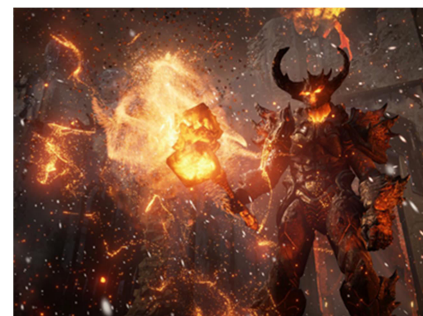
- **une composante mathématique**  
calcul matriciel, équations intégrales ou différentielles  
simulations numériques, géométrie analytique...
- **une composante algorithmique**  
vectorisation des algorithmes, convolutions discrètes,  
structures hiérarchiques, algorithmes récursifs...
- **des environnements informatiques nouveaux**  
jupyterlab, numpy, scipy, pillow, scikit, wings3D...

## *Ubiquité de l'imagerie numérique au quotidien*

### Films – Publicité



### Jeux vidéo





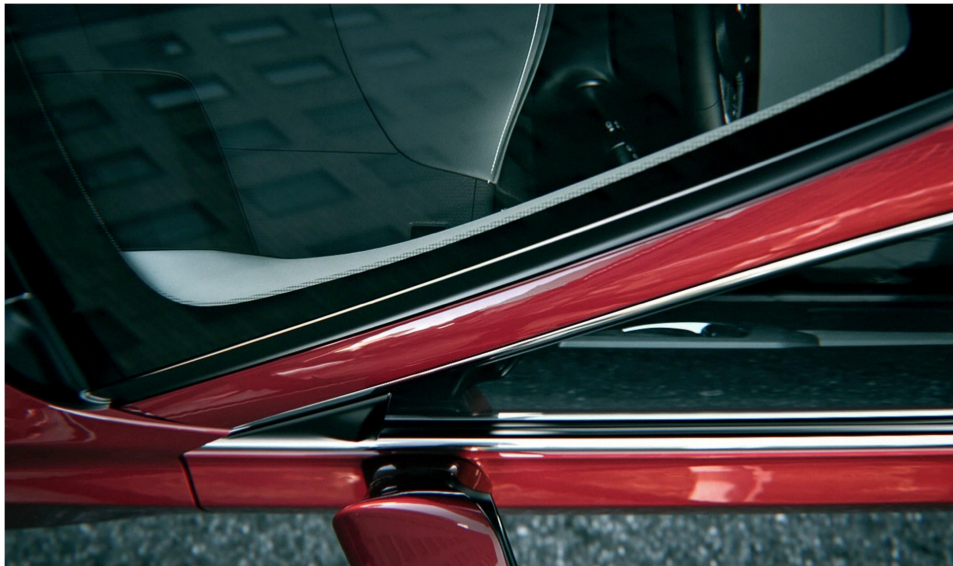
## Simulation



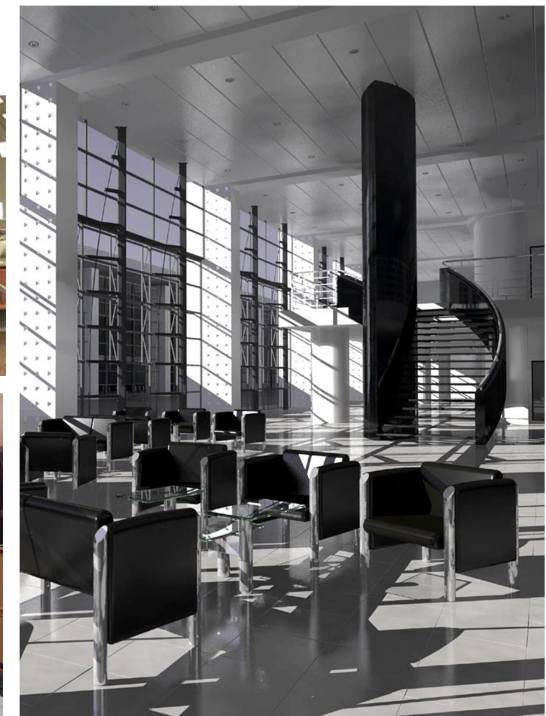
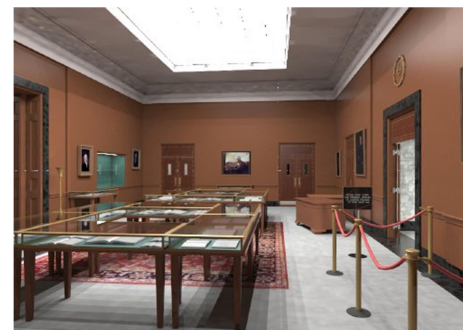
## Design



## Design



## Architecture

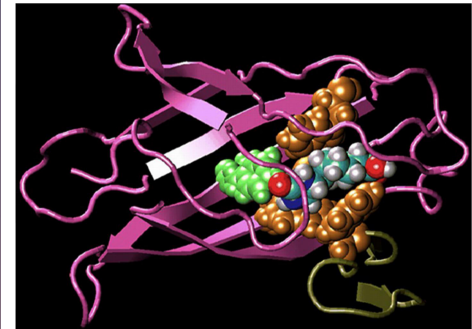
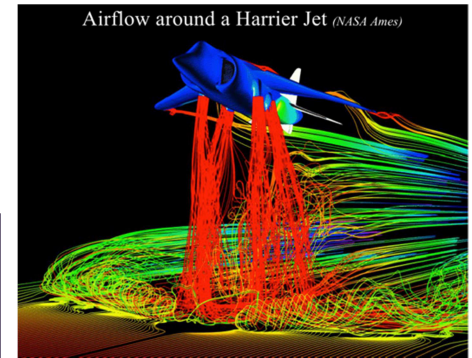
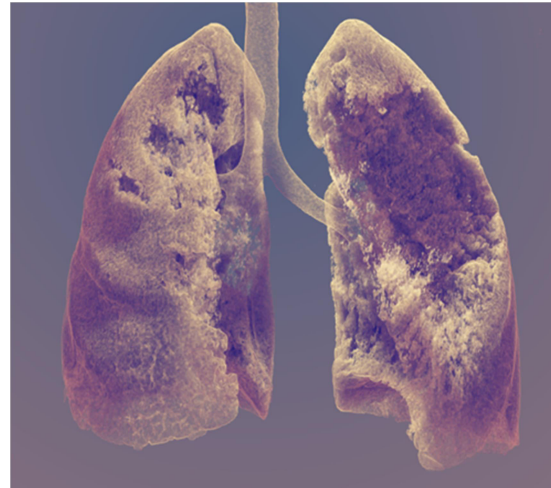




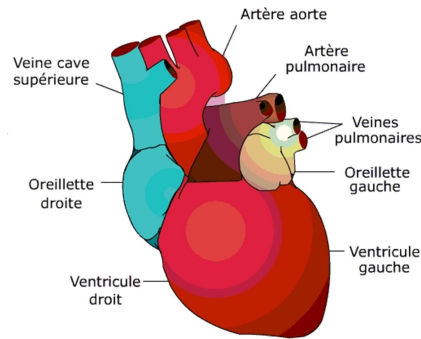
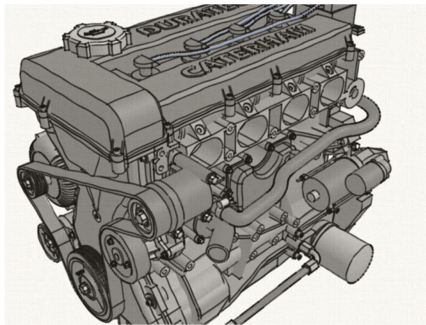
# Urbanisme



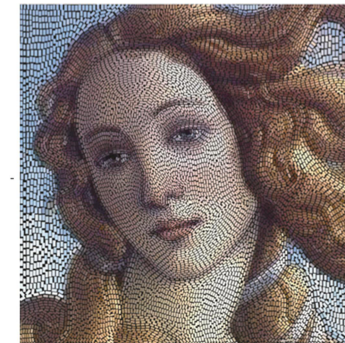
# Visualisation scientifique



# Visualisation expressive



# Visualisation artistique

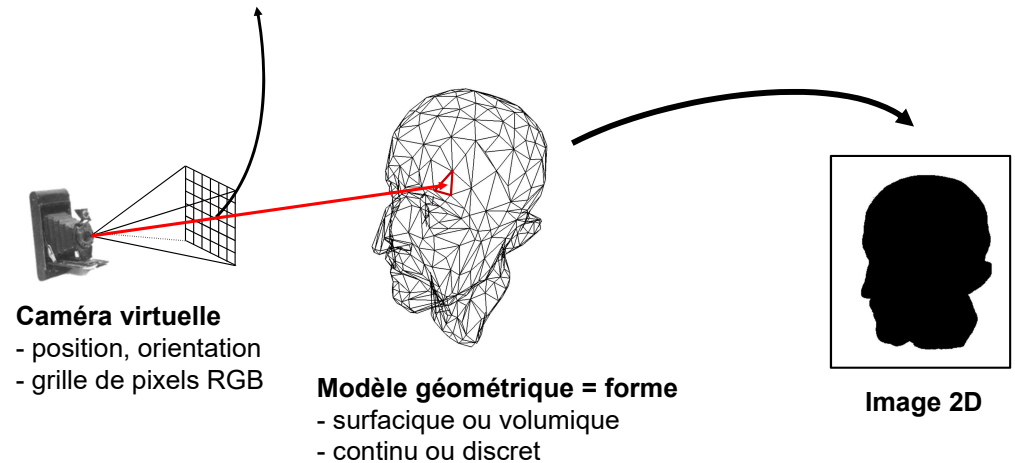




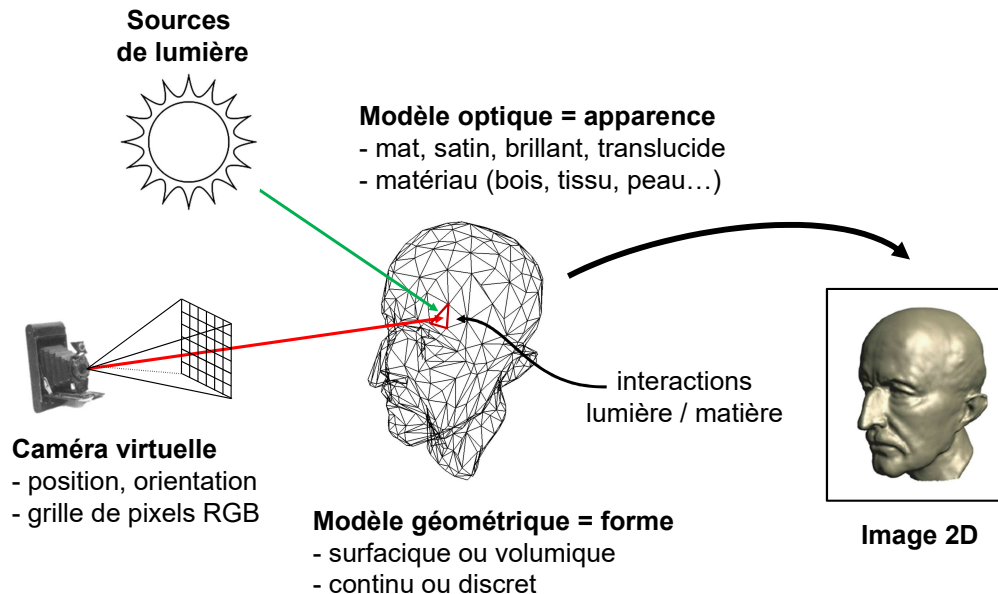
## Passage du 3D au 2D

## Synthèse d'images

**Problème :** quelle est la couleur de chacun des pixels ?



## Synthèse d'images

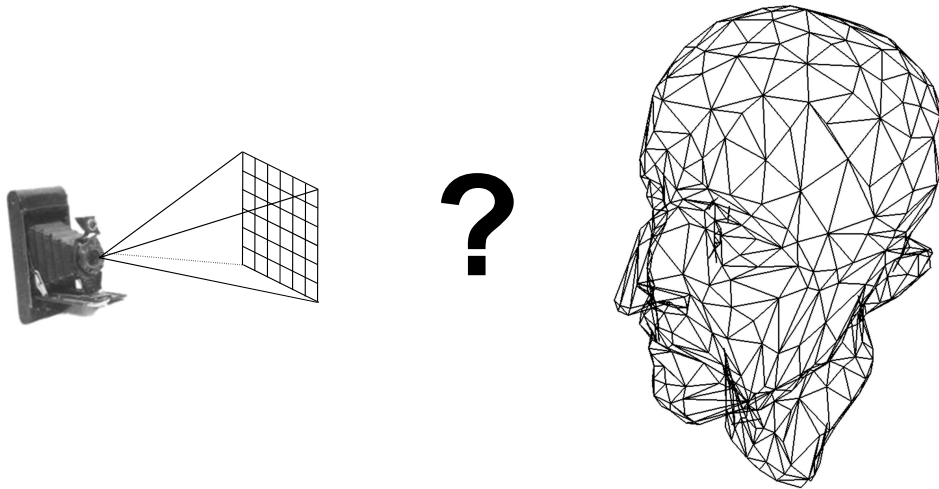


## Synthèse d'images

**Principe :** *Effectuer une simulation numérique des phénomènes qui se déroulent dans le monde réel*

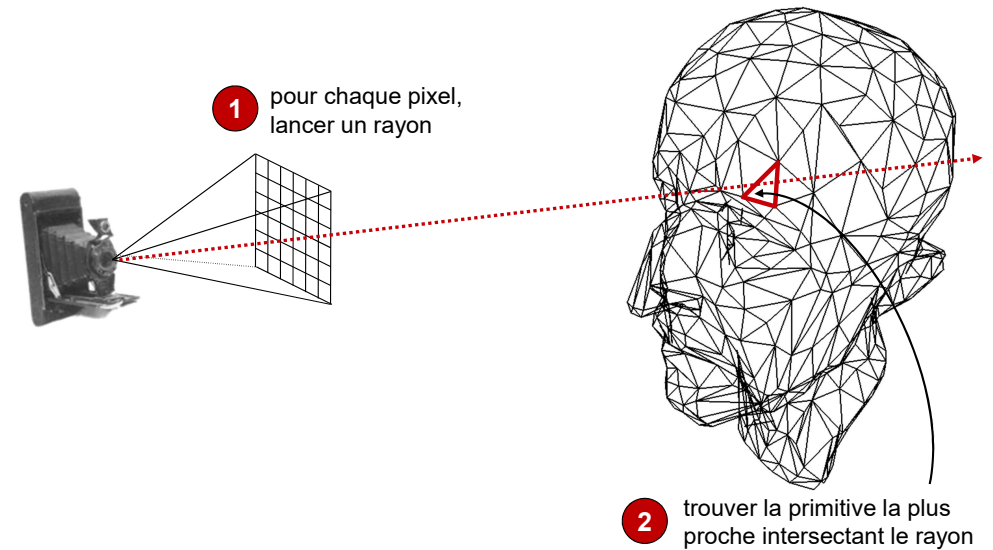
- La lumière émise par les sources (soleil, lampe) se propage dans l'espace et subit diverses réactions au contact des matériaux des objets de la scène 3D
- Un observateur va capter une partie de ces rayons lumineux pour former son image rétinienne 2D

# Visualiser une scène 3D



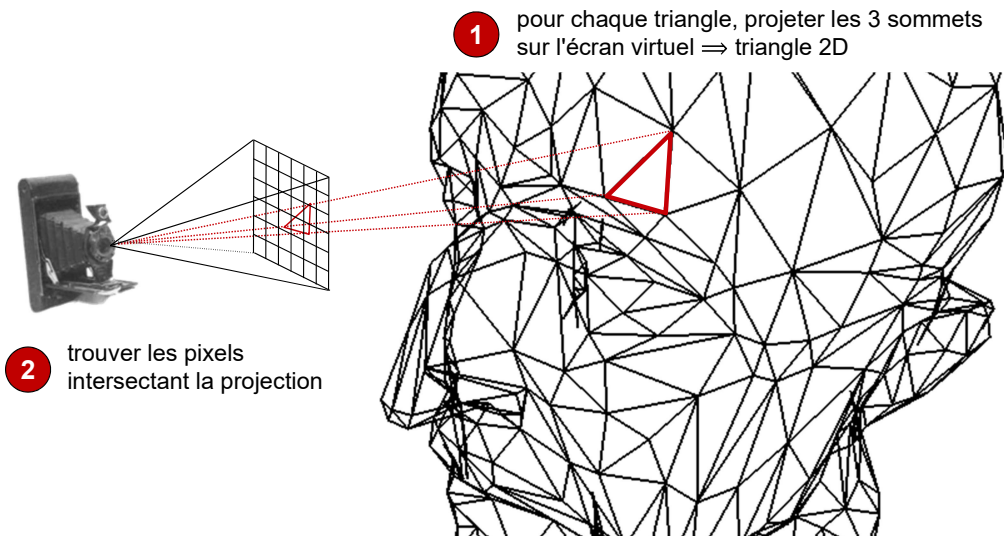
# Visualiser une scène 3D

1<sup>ère</sup> approche : **lancer de rayons**



# Visualiser une scène 3D

2<sup>ème</sup> approche : **projection discrète**



# Visualiser une scène 3D

## Lancer de rayons (ray tracing / ray casting)

- Propagation d'un rayon à partir de chaque pixel
- Pour chaque rayon, identifier l'intersection avec la primitive la plus proche de la caméra virtuelle
- Scène 3D = ensemble de primitives "**intersectables**"

## Projection discrète (rasterization)

- Projection de chaque primitive sur l'écran virtuel
- Pour chaque projection, identifier les pixels recouverts totalement ou partiellement par la primitive projetée
- Scène 3D = ensemble de primitives "**projetables**"