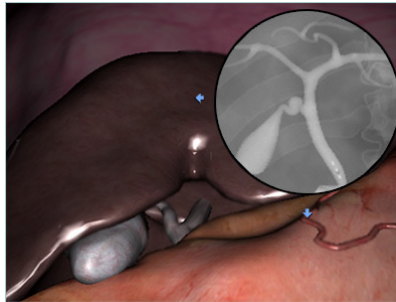
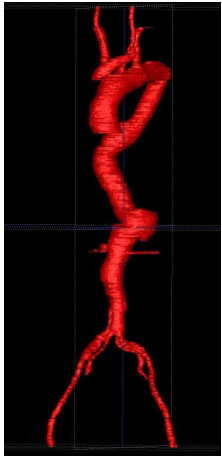
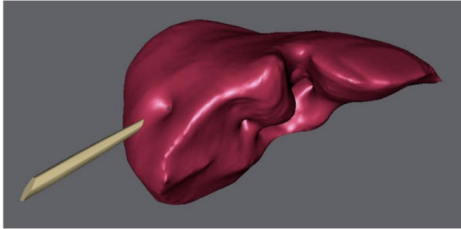


Informatique Graphique, Réalité Virtuelle et Applications médicales

Florence Zara

Université Claude Bernard Lyon 1



Plan du cours

- Informatique et image : différents domaines de recherche
 - Traitement d'images, synthèse d'images, Réalité Augmentée, Réalité Virtuelle
- Domaines d'applications de l'Informatique Graphique
- Comment créer des images virtuelles ?
 - Modélisation, animation, visualisation, pipeline graphique
- Comment interagir avec l'image créée ?
 - C'est la réalité Virtuelle
- Apport de la Réalité Virtuelle dans le monde médical
 - Conception de simulateurs d'apprentissage de gestes médicaux

Sources utilisées (à la base....) – Transparents de cours

Vincent Luboz - Pierre-Frederic Villard

Developing VR Surgical Simulation Software - Nombreuses images issues de ce cours
Department of Biosurgery and Surgical Technology - Imperial College London

Nicolas Holzschuch

Création d'images virtuelles (DEA IVR)
Rendu Volumique (DEA IVR)
ARTIS-INRIA Grenoble

Joëlle Thollot – Xavier Décoret – François Sillon

Modélisation Géométrique (DEA IVR) - Nombreuses images issues de ce cours
ARTIS-INRIA Grenoble

Alexandre Meyer

Pipeline du rendu projectif (Gamagora)
LIRIS – Lyon

Références de livres pour en savoir plus

Foley, van Dam, Feiner et Hugues

Computer Graphics - Principles and Practice

Foley, van Dam, Feiner, Hugues et Philipps

Introduction to Computer Graphics

Livres écrit sous la direction de Péroche et Bechmann

Informatique Graphique et Rendu

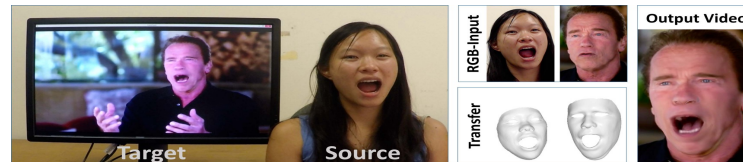
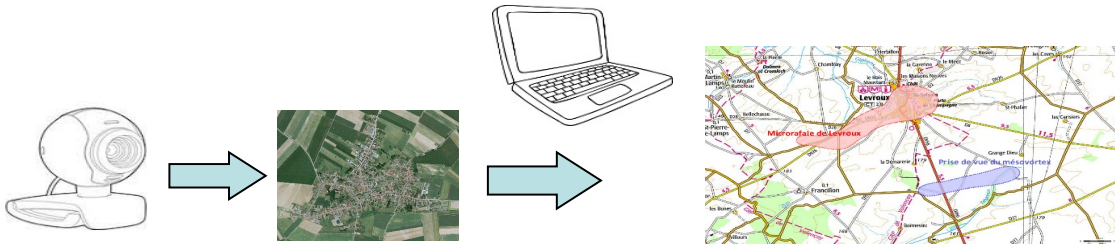
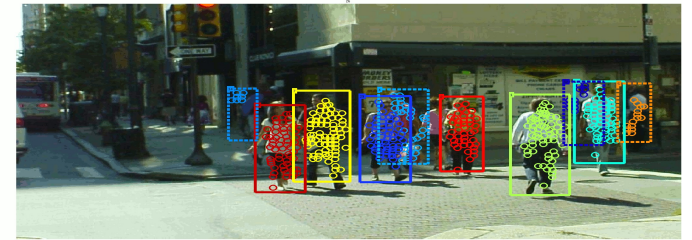
Informatique Graphique, modélisation géométrique et animation

Informatique et Images :
différents domaines de recherche

Informatique et images : différents domaines de recherche

Analyse d'images (*Image Analysis*)

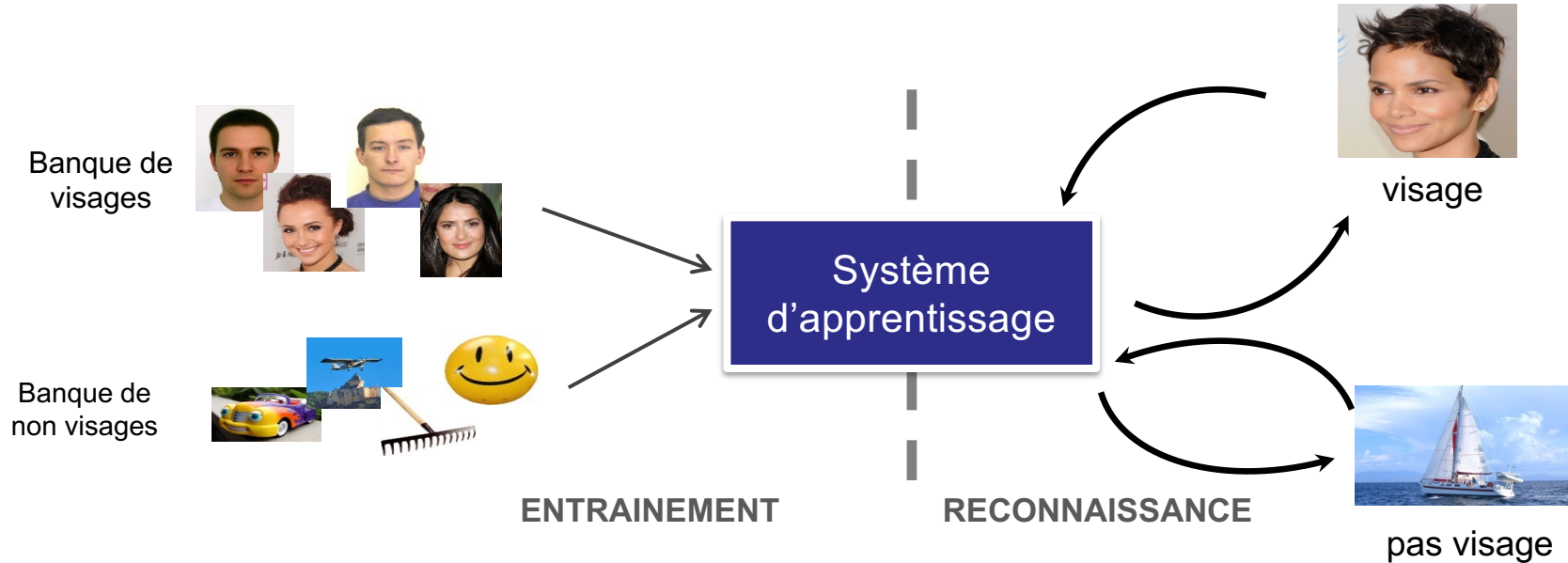
- Traitement d'images (*Image Processing*)
- Reconnaissance des formes (*Pattern Recognition*)
- Vision par ordinateur (*Computer Vision*)



Informatique et images : différents domaines de recherche

Apprentissage automatique pour le traitement d'images

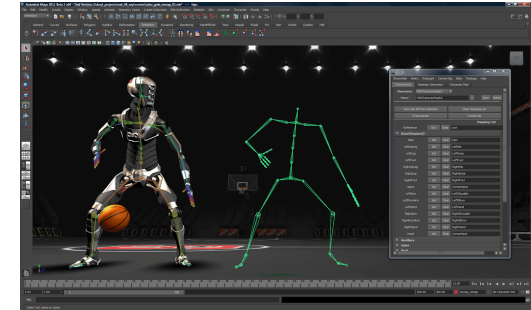
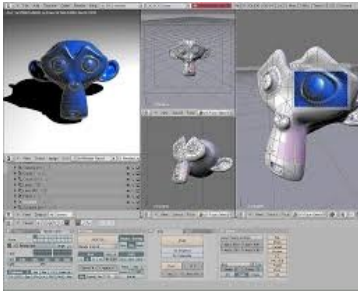
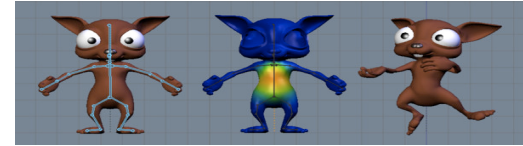
A partir d'une banque d'exemples, l'ordinateur apprend à classer les éléments



Informatique et images : différents domaines de recherche

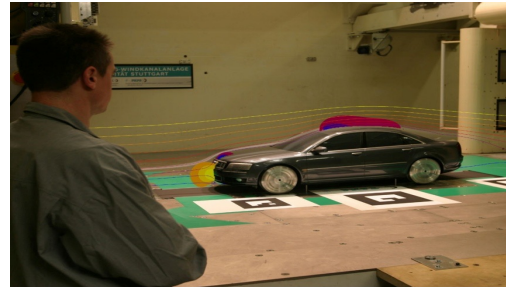
Synthèse d'images (*Computer Graphics*)

- Modélisation
- Animation
- Rendu



Informatique et images : différents domaines de recherche

Réalité Augmentée : mélange des 2 = image réelle + image virtuelle



Informatique et images : différents domaines de recherche

Réalité Virtuelle : synthèse d'image + matériel de vision + robotique



©Tous droits réservés UTBM

Immersion + interaction

Domaines d'applications de l'Informatique Graphique (Computer Graphics)

Pourquoi créer des images virtuelles ?

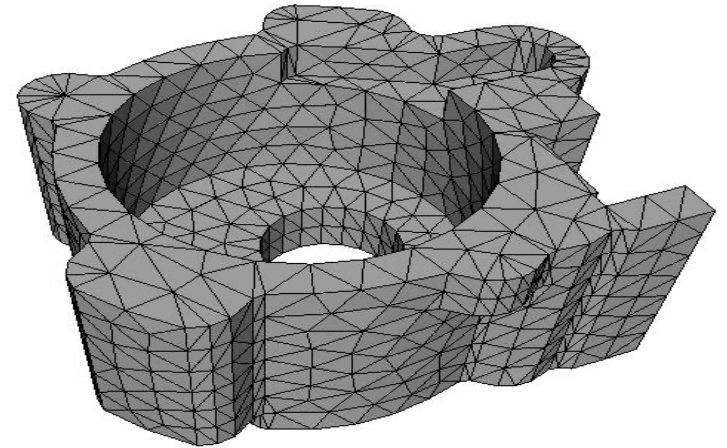
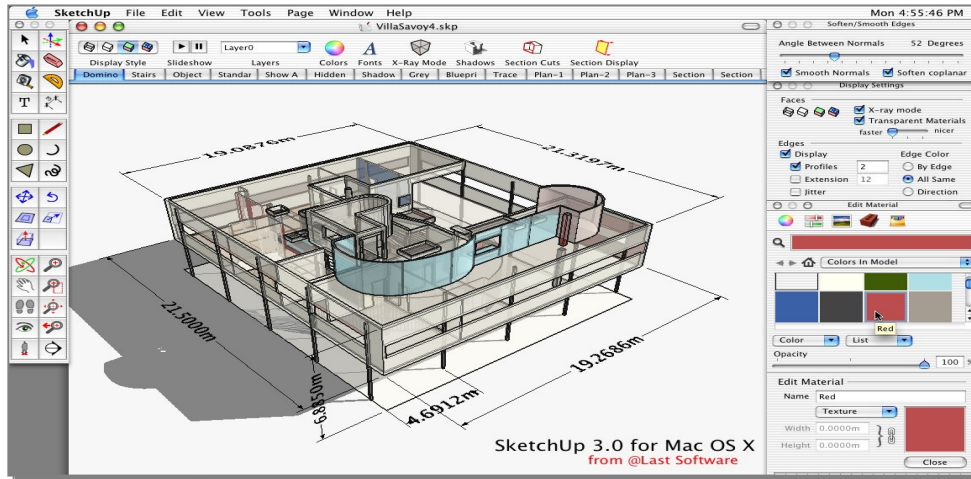
De nombreux domaines d'application pour l'informatique graphique

Illustration de quelques exemples :

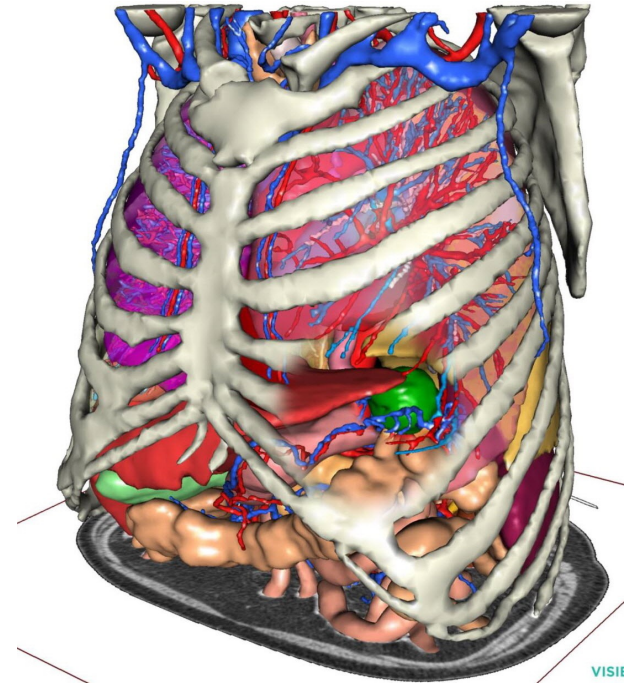
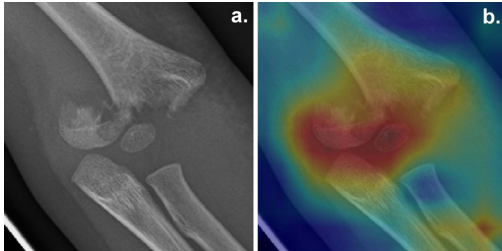
- Architecture / Conception Assistée par Ordinateur (CAO)
- Applications pour le médical
- Visualisation scientifique
- Loisirs numériques : films d'animation, jeux vidéos, effets spéciaux

Mais beaucoup d'autres...

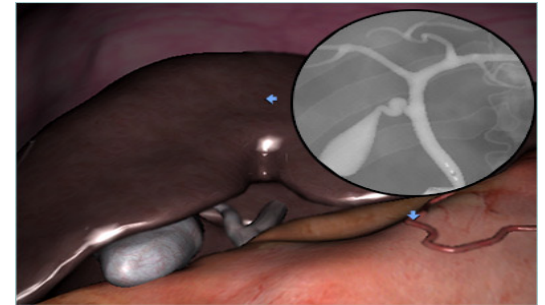
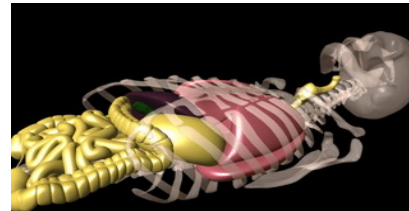
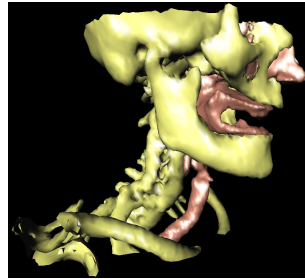
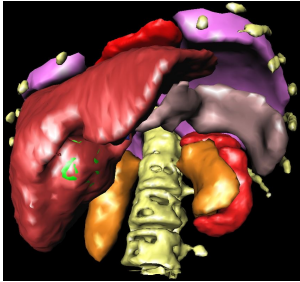
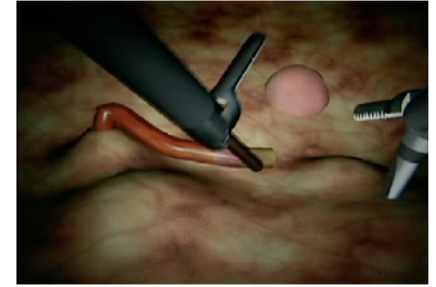
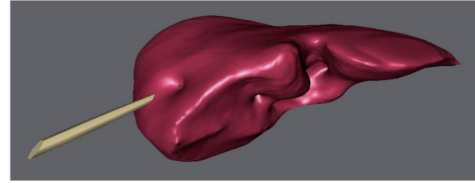
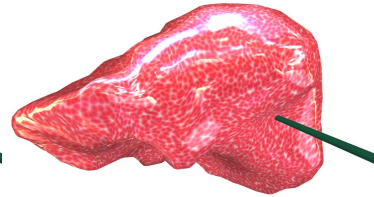
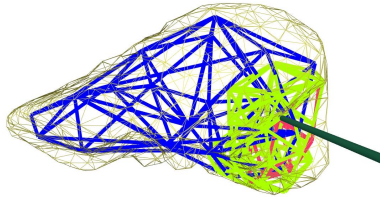
Domaines d'applications - Architecture / CAO



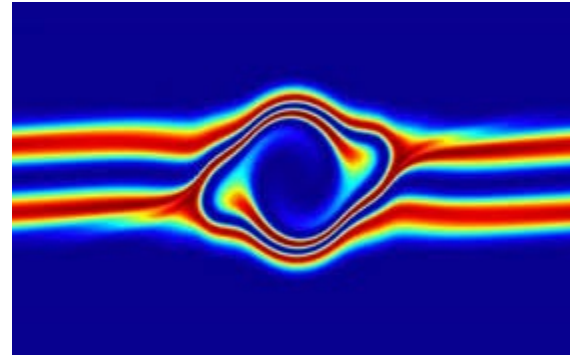
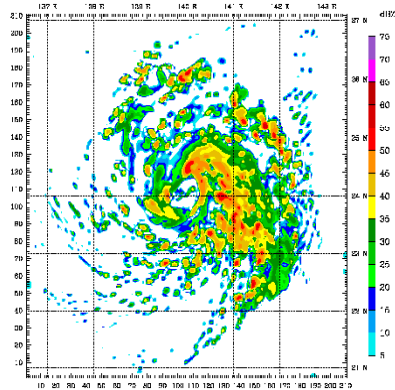
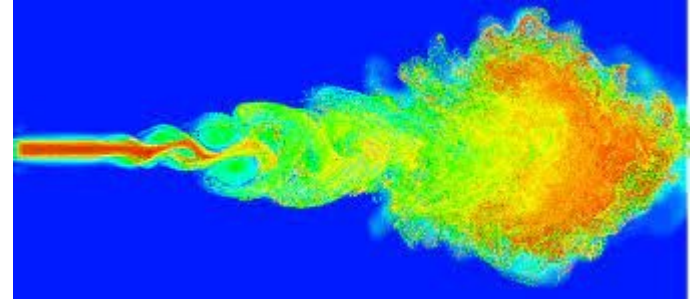
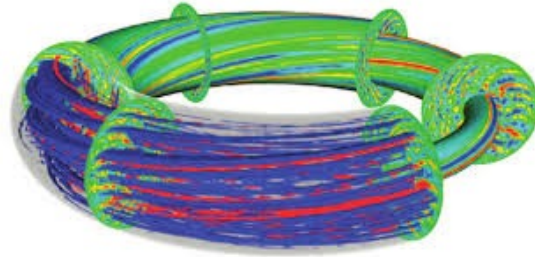
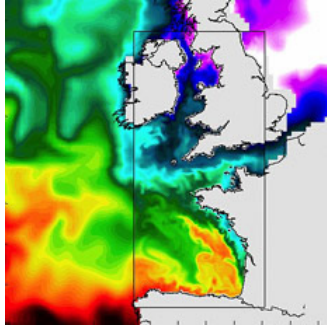
Domaines d'applications - Imagerie médicale



Domaines d'applications – Simulation biomécanique



Domaines d'applications - Visualisation scientifique



Domaines d'applications - Films d'animation



Pixar

Domaines d'applications - Jeux vidéo



Domaines d'applications – Effets spéciaux



Domaines d'applications – en bilan

Pas les mêmes besoins selon les domaines d'application

- Réalisme plus ou moins important

- Temps d'exécution plus ou moins important

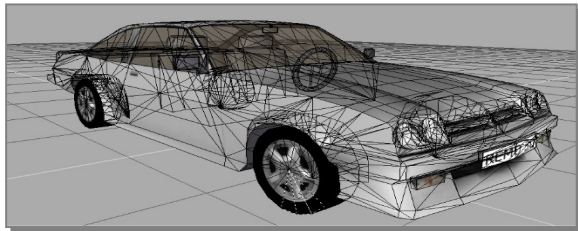
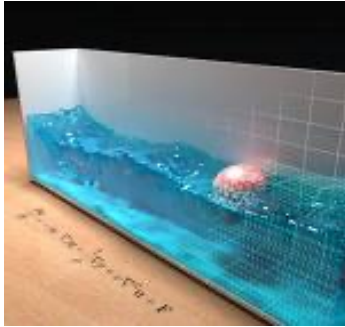
- Interactivité ou non avec les images créées

Pas les mêmes méthodes employées pour créer ces images

- Modèles et algorithmes différents

- Utilisation ou non du GPU, multi-cœurs

Comment créer des images virtuelles ?



Etapes de bases pour créer des images virtuelles animées

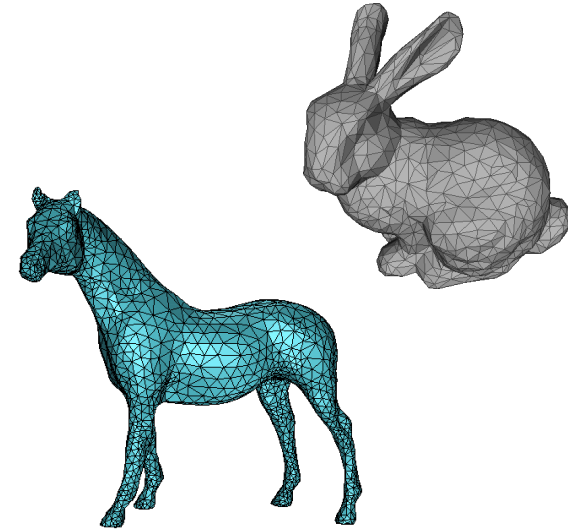
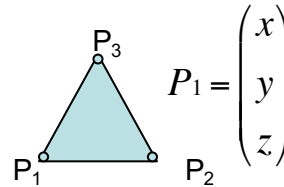
1. Création de la scène 3D

- Un ou plusieurs objets à positionner dans la scène
- Objet constitué de sommets / d'arêtes / de faces / de volumes élémentaires
- Chaque sommet est défini par sa position dans l'espace 3D : coordonnées (x,y,z)

Considérons le cas où les objets sont décrits par des triangles

Représentation de l'objet = maillage surfacique qui est défini par

- un ensemble de sommets
- un ensemble de faces (triangles)
- faces décrites par 3 sommets
- faces reliées ensemble par des arêtes communes



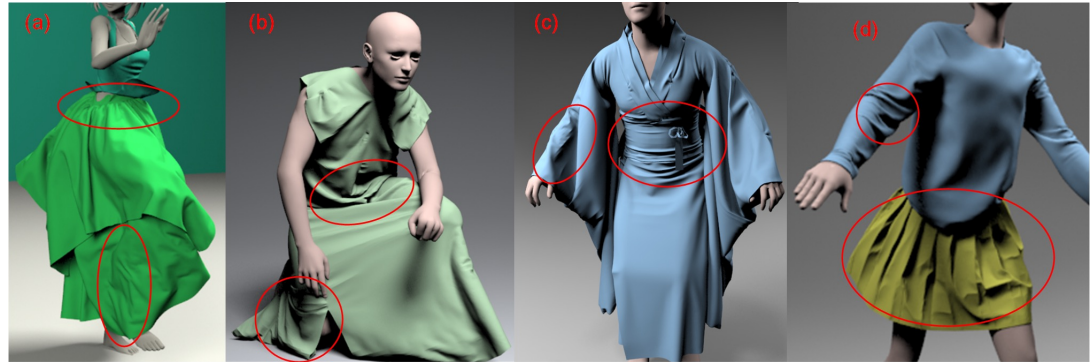
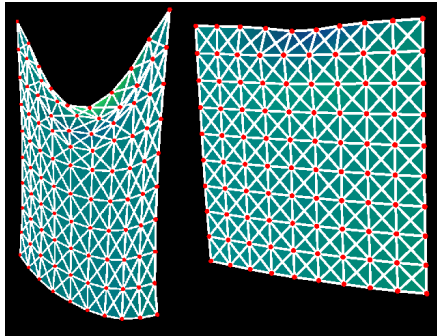
C'est l'étape de modélisation de la scène

Etapes de bases pour créer des images virtuelles animées

2. Les coordonnées (x,y,z) des sommets des objets peuvent changer au cours du temps

- Les objets sont en mouvement : translation, rotation
- Les objets peuvent se déformer :
 - Changement de coordonnées des sommets induit déformation des faces / volumes de l'objet

C'est l'étape d'animation / simulation des objets de la scène



Etapes de bases pour créer des images virtuelles animées

3. Informations supplémentaires pour afficher la scène 3D sur l'écran 2D

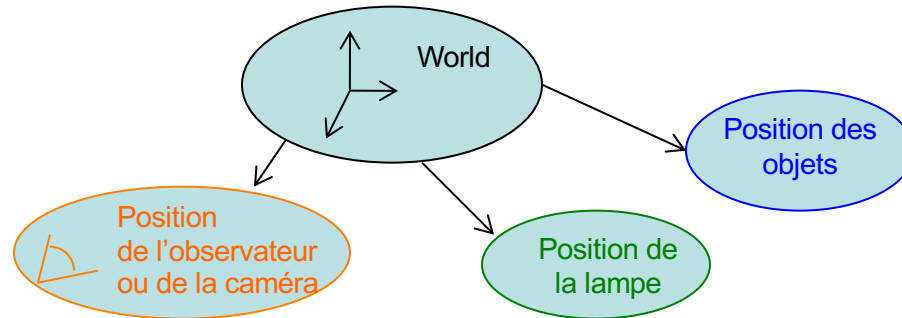
- Des primitives graphiques (couleur, propriétés matériaux) sont attachées aux sommets
- Des lumières doivent être positionnées dans la scène : position (x,y,z) des spots ayant une couleur
- Un observateur (ou une caméra) doit être positionné : position (x,y,z) de la caméra
- Qui s'ajoutent à la position et orientation des objets de la scène dans le repère monde (repère initial)

Ces informations permettront de faire l'affichage à l'écran de la scène qui est **l'étape de rendu**

Etapes de bases pour créer des images virtuelles animées

Au final, la scène 3D est constituée :

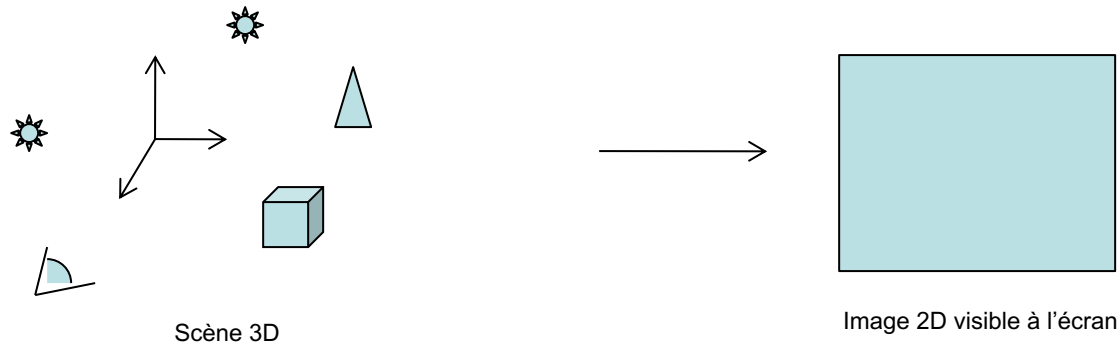
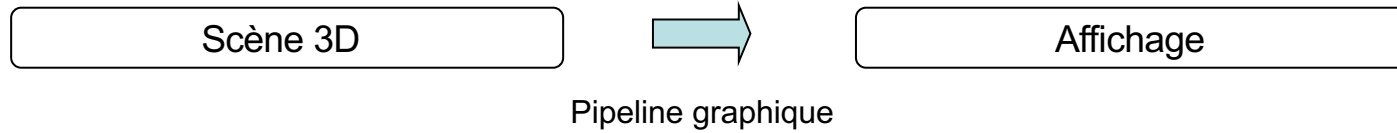
- De lumières ayant une position (x,y,z)
- D'une caméra ayant une position (x,y,z)
- D'objets positionnés / orientés dans le repère monde (repère initial)



La description de cet ensemble va permettre de générer l'image 2D à un instant donné
Pour cela les objets vont remplir une partie de l'image 2D à un instant donné

Etapes de bases pour créer des images virtuelles animées – Pipeline Graphique

Une étape importante : création de l'image finale



Création d'une image – au fait, c'est quoi une image en Informatique ?

- Rectangle (2D) : tableau 2D de pixels (= *picture element*)
 - nombre de lignes
 - nombre de colonnes
 - format des pixels (bit, niveaux de gris, niveaux de couleurs)
 - *compression éventuelle*



Continuous image



Digital image

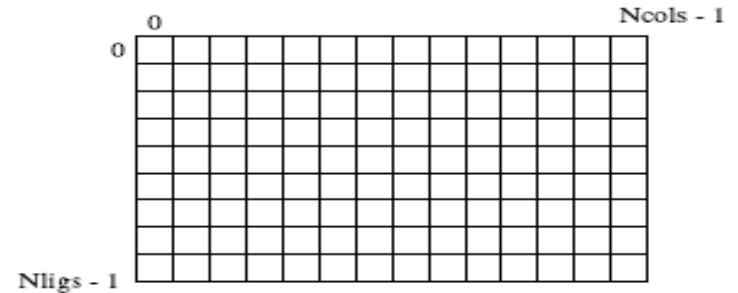
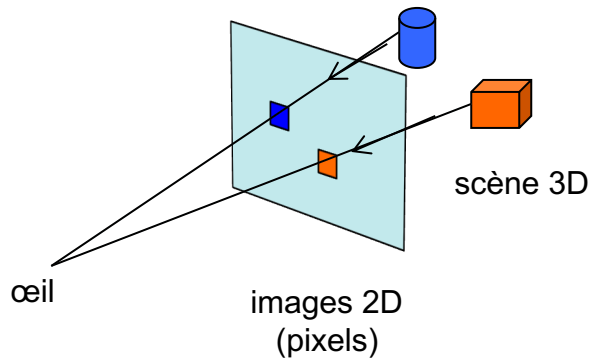


Image = ensemble de pixels

Etapes de bases pour créer des images virtuelles animées – Pipeline Graphique

Transformation à effectuer pour afficher la scène 3D sur un écran 2D
c'est-à-dire pour passer d'un espace 3D à un espace 2D

Plusieurs techniques possibles : cas simple du rendu projectif



Les objets sont projetés sur l'écran dans la direction de l'œil / de la caméra

Transformation des coordonnées 3D des modèles vers les coordonnées 2D des pixels

D'autres étapes encore à faire suite à cette projection

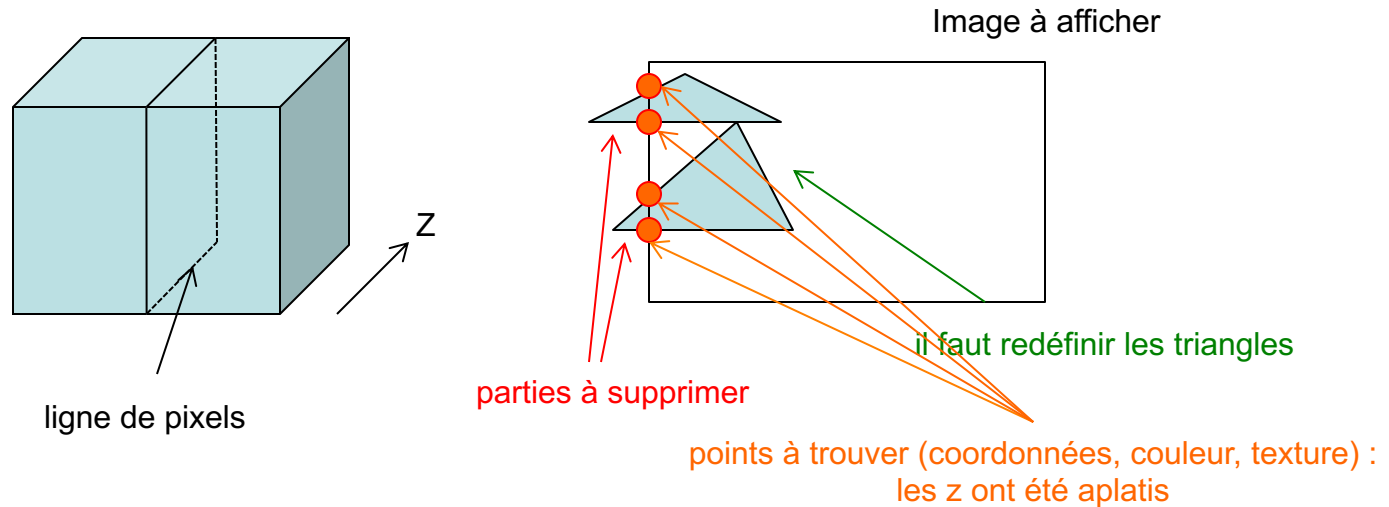
Etapes de bases pour créer des images virtuelles animées – Pipeline Graphique

Etapes du pipeline graphique (faites par la carte graphique) réalisées suite à cette projection

- *Clipping* des objets 3D selon la pyramide de vue
- Calcul de la couleur des pixels = illumination
- Résolution de l'occlusion avec la suppression des parties cachées = *Z-Buffer*

Etapes de bases pour créer des images virtuelles animées – Pipeline Graphique

Problème du *clipping* à gérer pour créer l'image 2D



Etapes de bases pour créer des images virtuelles animées – Pipeline Graphique

Calcul de la couleur de chacun des pixels de l'image 2D = illumination

La couleur du pixel est définie soit :

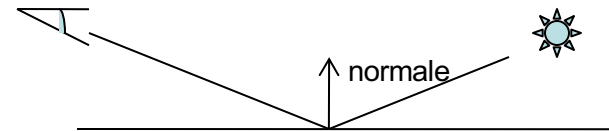
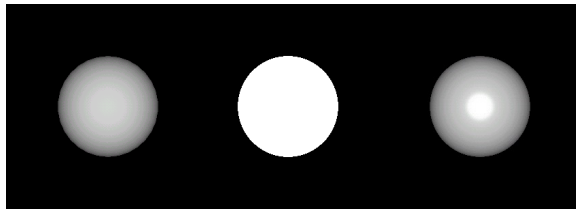
- de manière explicite
- ou selon un modèle de fond établi aux sommets des triangles,
- puis interpolation pour remplir les pixels à l'intérieur des triangles

Tient compte de la lumière pour donner un aspect plus réel

Tient compte du placement de lampes qui ont une couleur

Tient compte des couleurs des différents objets

Nécessité de calculer la normale aux différents sommets de la surface



Etapes de bases pour créer des images virtuelles animées – Pipeline Graphique

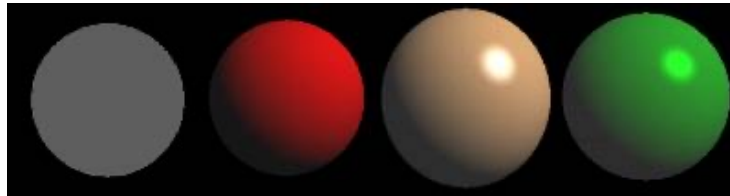
Pour tenir compte de la lumière - Il existe différents types de lumière

Composante ambiante : constante qui colore les pixels d'un objet par la même couleur quelque soit l'environnement lumineux.

Composante diffuse (différente pour chaque sommet) : permet de donner un effet 3D et lissé aux objets.

Composante spéculaire : correspond au léger reflet de la lumière sur les bords des objets.

Composante émissive : simulation de la lumière émise par un objet.



Composantes ambiante, diffuse, spéculaire et émissive

Etapes de bases pour créer des images virtuelles animées – Pipeline Graphique

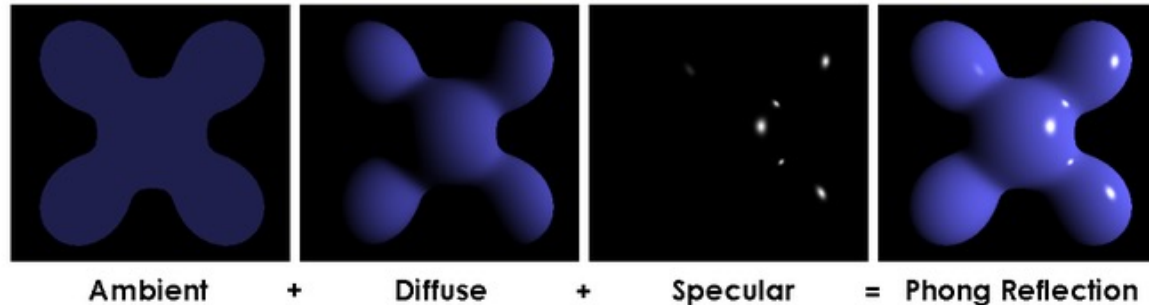
Modèle de Phong

L'illumination de Phong est un modèle local : il calcule l'intensité en chaque point

Il combine trois éléments :

- la lumière diffuse (modèle Lambertien)
- la lumière spéculaire
- et la lumière ambiante

Formule de couleur pour les différents sommets : $K_a L_a + K_d L_d \cos \theta + K_s L_s \cos \theta$
avec θ l'angle formé par la direction de l'œil et la source lumineuse



Problème de l'occlusion

Le problème apparaît quand un objet se trouve devant un autre

Il faut alors prendre en compte qu'un objet n'est pas toujours visible
selon le point de vue de l'observateur

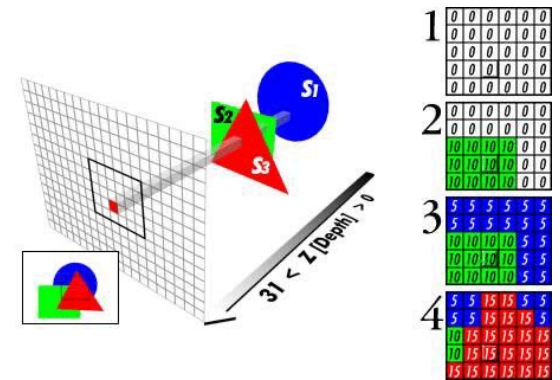
Ceci n'est pas pré-calculable car dépend du point de vue

Etapes de bases pour créer des images virtuelles animées – Pipeline Graphique

Savoir quel objet est visible sur l'image = *test de profondeur*

Le Z-Buffer permet de gérer le problème de la visibilité
→ déterminer quels éléments de la scène doivent être rendus,
c'est-à-dire lesquels sont cachés par d'autres
et dans quel ordre l'affichage des primitives doit se faire.

Z-Buffer = tableau à 2 dimensions (et Y) de la même taille que l'image
= chaque élément est un pixel de l'écran
= stocke les valeurs de profondeur des pixels (coordonnée Z)



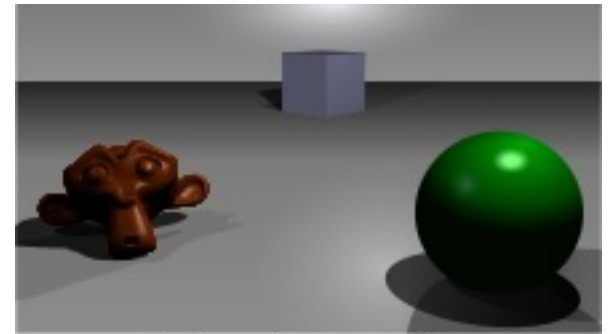
Etapes de bases pour créer des images virtuelles animées – Pipeline Graphique

Algorithme employé

Si un autre élément de la scène doit être affiché aux mêmes coordonnées (X,Y), la carte compare les deux profondeurs (Z), et n'affiche que le pixel le plus proche de la caméra

La valeur Z de ce pixel est ensuite placée dans le tampon de profondeur, remplaçant l'ancienne

Au final, l'image dessinée reproduit la perception de la profondeur habituelle et logique, l'objet le plus proche cachant les plus lointains.



A simple three-dimensional scene



Z-buffer representation

En résumé, les étapes de bases pour créer des images virtuelles animées

- 1- **Modélisation** : représentation mathématique des objets virtuels
- 2- **Simulation / animation** : déformation et mouvement des objets virtuels
- 3- **Visualisation / rendu** : affichage des objets virtuels - **pipeline graphique**

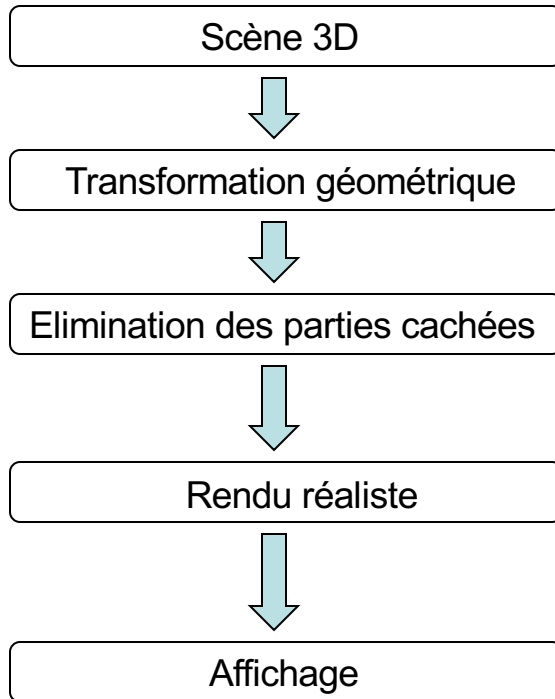


modélisation en 3D

rendu

Chaque étape correspond à un domaine d'expertise à part entière

Création d'images virtuelles en Informatique Graphique – Pipeline Graphique



- Modélisation géométrique
- Animation

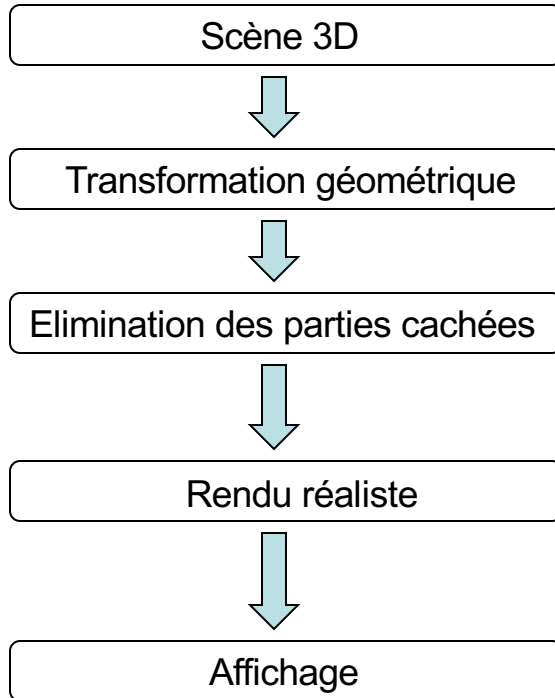
- Transformation
- Projection

- Z-Buffer

- Modèles de couleur
- Modèles d'illumination locale
- Textures
- Eclairage local (ombres, reflets) et global

- Ecran

Création d'images virtuelles en Informatique Graphique – Pipeline Graphique



- Modélisation géométrique
- Animation

- Utilisation de la librairie OpenGL :
 - Rendu (éclairage, etc.)
 - Affichage à l'écran (projection, éliminations des parties cachées, ...)

- Rendu volumique

Etapes de bases pour créer des images virtuelles animées

- 1- **Modélisation géométrique** : représentation mathématique des objets virtuels
- 2- **Simulation / animation** : déformation et mouvement des objets virtuels
- 3- **Visualisation / rendu** : affichage des objets virtuels

Modélisation géométrique

Intérêt

Permet de représenter un objet qui sera ensuite manipulé

- objets réels ou virtuels / inventés
- carrosserie de voiture, réacteur nucléaire, etc.

Permet de rendre observables certaines caractéristiques

- visite de réacteur nucléaire, etc.

Remplace les maquettes à taille réduite

- réduction des coûts

Utilisation

Affichage, animation, simulation physique, etc.

Modélisation géométrique

Les modèles géométriques

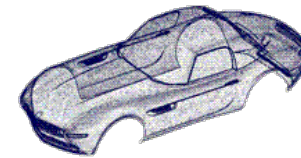
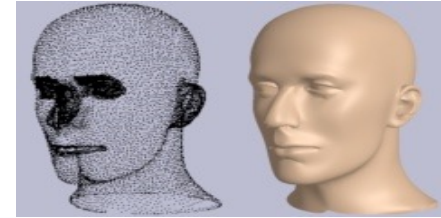
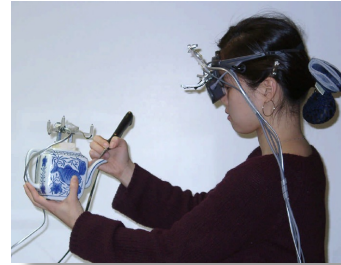
- modèles **non structurés**
 - points
 - soupes de polygones
- modèles **surfaiques**
 - maillage surfaique
 - surface paramétrique
 - surface de subdivision
 - surface implicite
- modèles **volumiques**
 - maillages volumiques
 - voxels, tétraèdres, etc.
- modèles **procéduraux**
 - fractales
 - système de particules
- modèles **à base d'images**
 - acquisition
 - rendu

Modélisation géométrique

Modèles non structurés - Nuages de points

Modèles obtenus

- par digitalisation manuelle ou par scanner
- par reconstruction à partir d'images
- par échantillonnage d'un autre modèle



➡ simple à afficher mais demande beaucoup de ressource mémoire

Modélisation géométrique

Modèles non structurés - Soupes de polygones

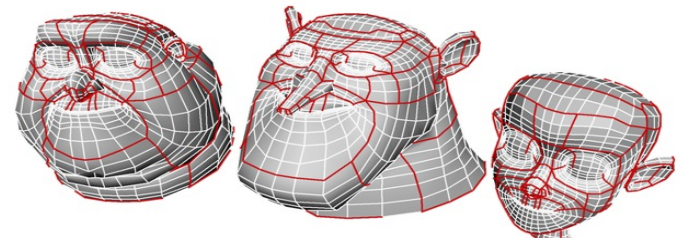
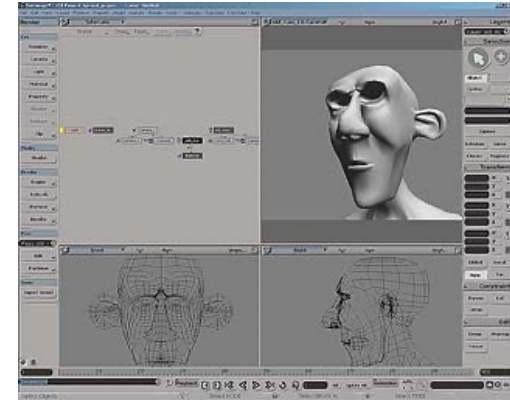
Ensemble non structuré de facettes

Avantages :

- représentation native de OpenGL
- nombreux logiciels d'édition

Inconvénients :

- opérations autre que l'affichage compliquées
- édition fastidieuse



Modélisation géométrique

Modèles surfaciques – Maillages surfaciques

Ensemble connecté de polygones

- triangles, quads, polygones convexes

Permet de représenter

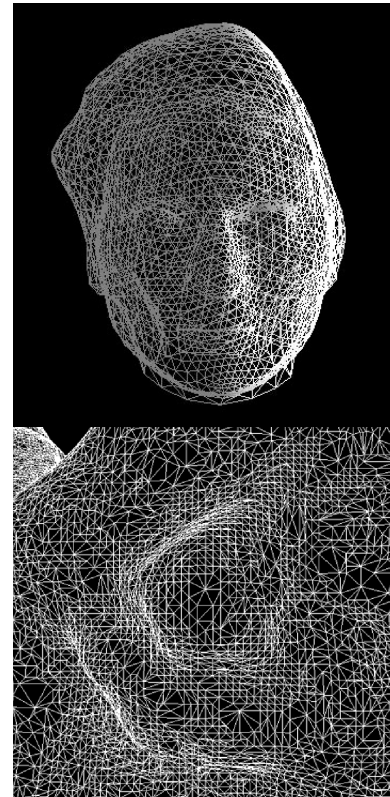
- la forme et la topologie de l'objet

Affichage facile avec librairie dédiée (OpenGL)

Permet d'effectuer facilement des calculs

- normales, courbures, simplification, etc.

➡ modèle souvent employé en animation



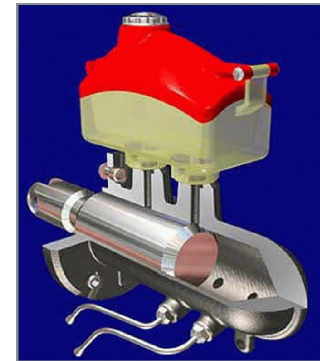
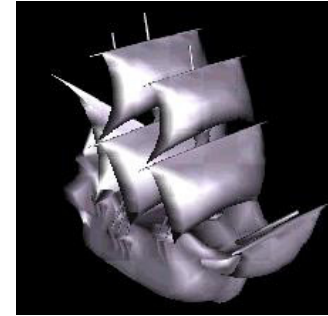
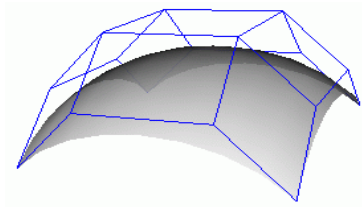
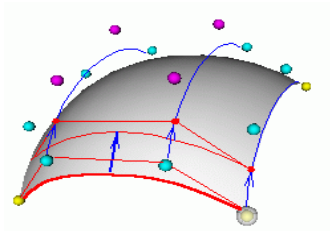
Modélisation géométrique

Modèles surfaciques – Surfaces paramétriques

Définition par des points de contrôle (splines)

Convertible en maillage pour effectuer le rendu

Edition aisée

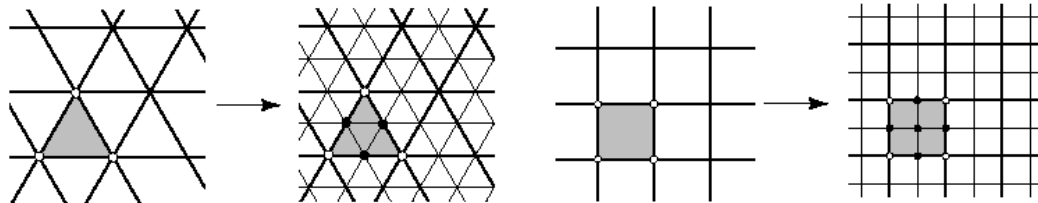


Modélisation géométrique

Modèles surfaciques – Surfaces de subdivision

Schéma de subdivision appliqué récursivement sur un maillage

- raffinement d'un ensemble de primitives géométriques en un ensemble plus dense
- création de nouveaux points et de nouvelles faces
- représentation multi-résolution naturelle d'une surface
- obtention d'une surface lisse

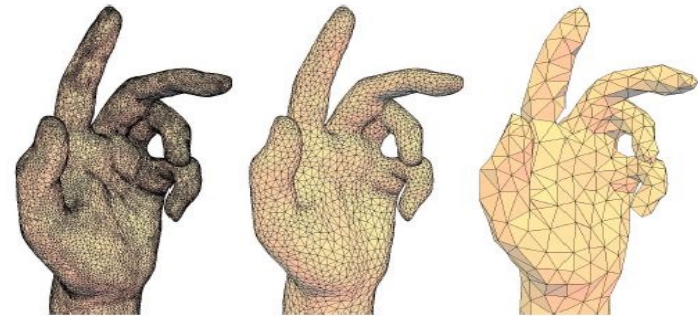
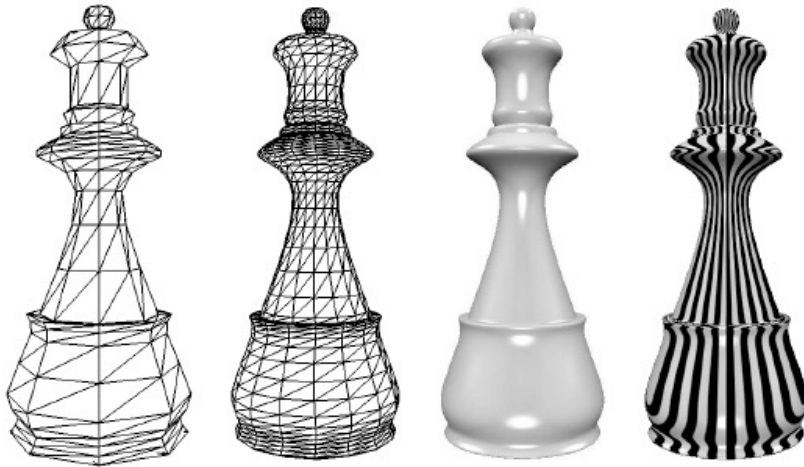


Modélisation géométrique

Modèles surfaciques – Surfaces de subdivision

Il existe de nombreux schémas de subdivision

→ les divisions de surface Catmull-Clark, Doo-Sabin, Loop, etc.



(a) 25,000 vertices. (b) 5,000 vertices. (c) 500 vertices.

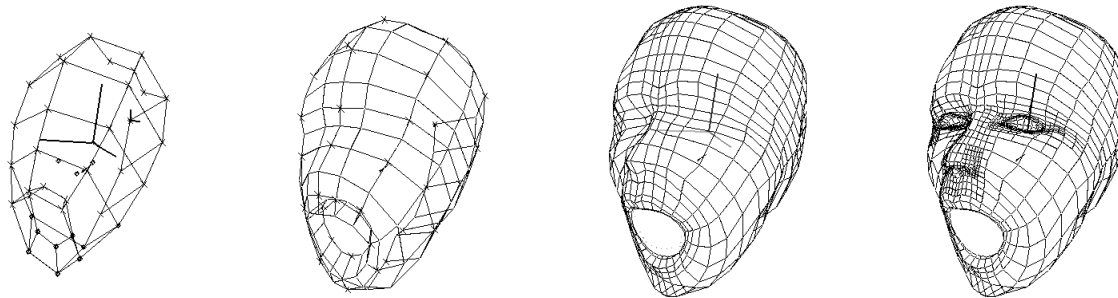
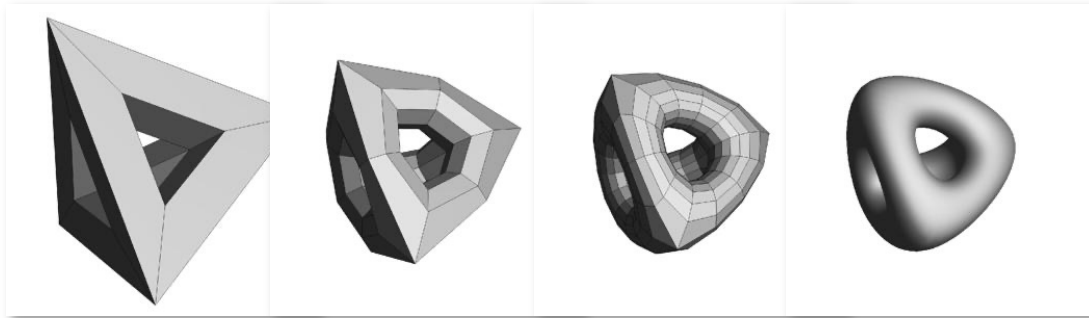
Modélisation géométrique

Modèles surfaciques – Surfaces de subdivision



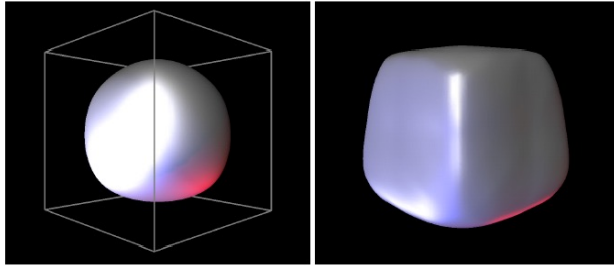
Modélisation géométrique

Modèles surfaciques – Surfaces de subdivision



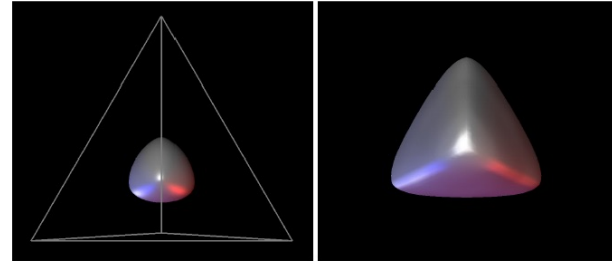
Modélisation géométrique

Modèles surfaciques – Surfaces de subdivision



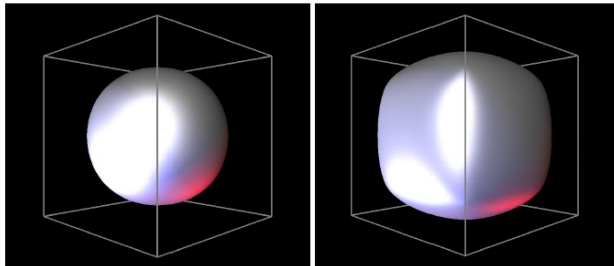
Loop

Butterfly



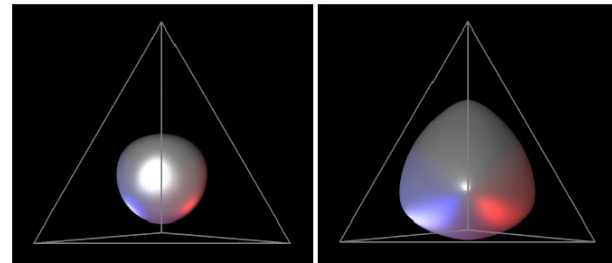
Loop

Butterfly



Catmull-Clark

Doo-Sabin



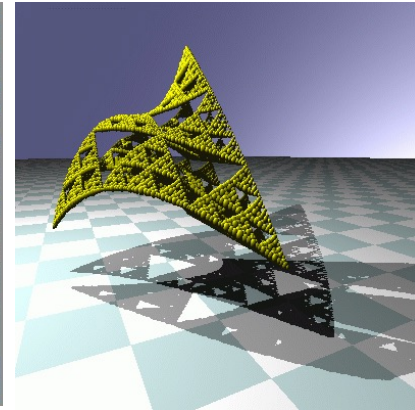
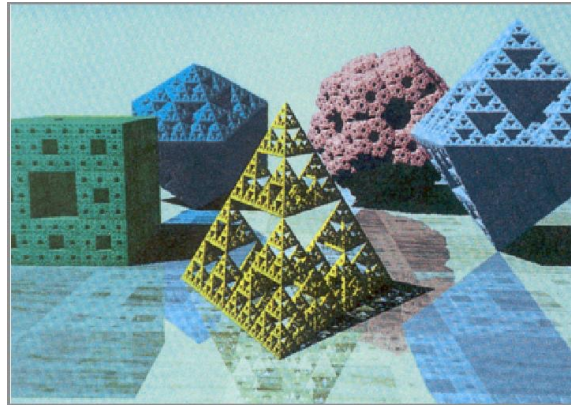
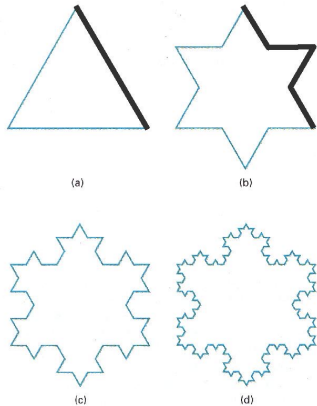
Catmull-Clark

Doo-Sabin

Modélisation géométrique

Modèles procéduraux - Fractales

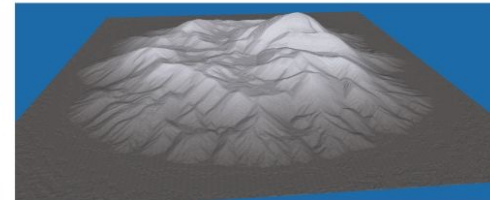
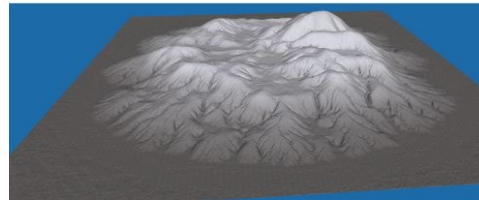
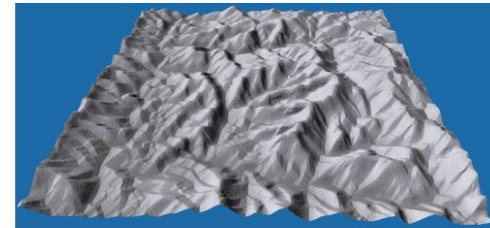
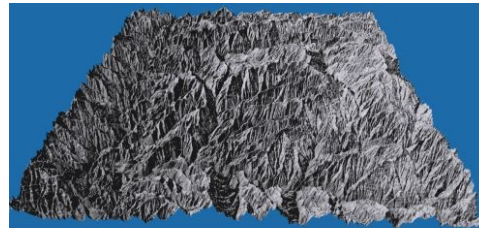
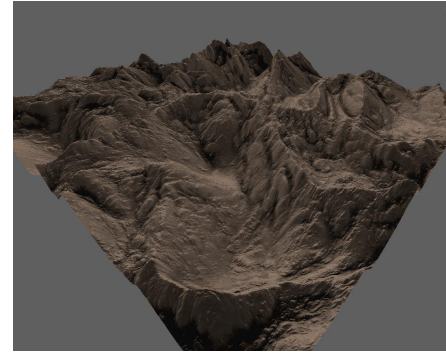
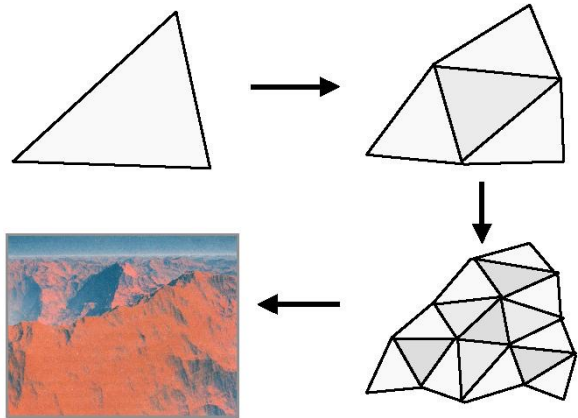
Utilisation de fractale pour élaborer des modèles



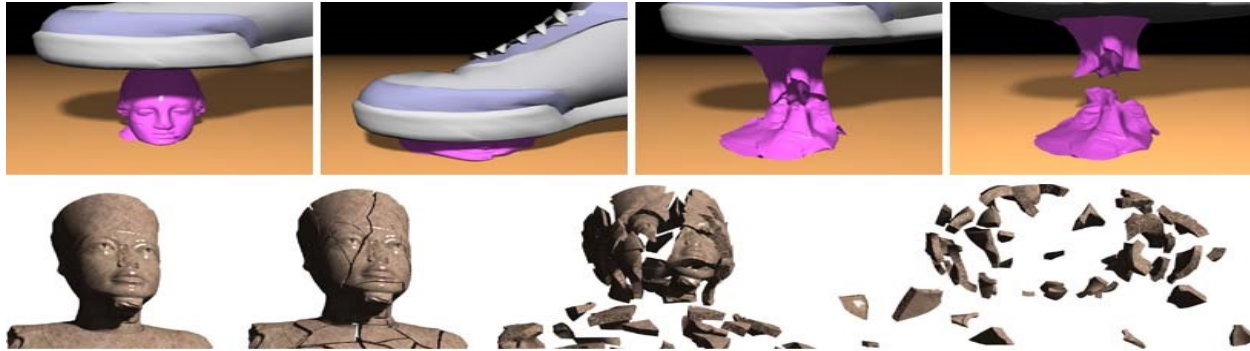
Modélisation géométrique

Modèles procéduraux – Fractales

Modèles souvent utilisés pour les terrains : générés au cours de l'exécution



Modélisation procédurale



Modélisation géométrique

Modèles surfaciques – Surfaces implicites

Surfaces implicites donnent une représentation volumique de l'objet

La surface n'est pas directement connue mais caractérisée indirectement comme l'ensemble des points de l'espace p solution d'une équation $f(p) = 0$

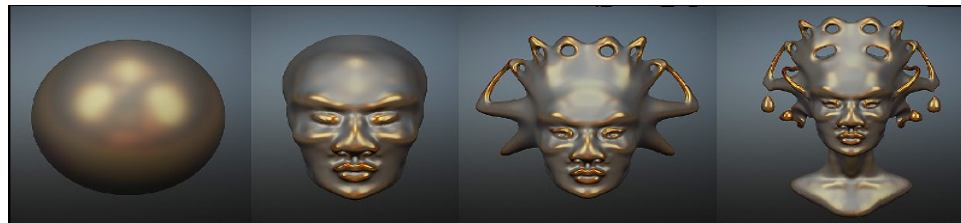
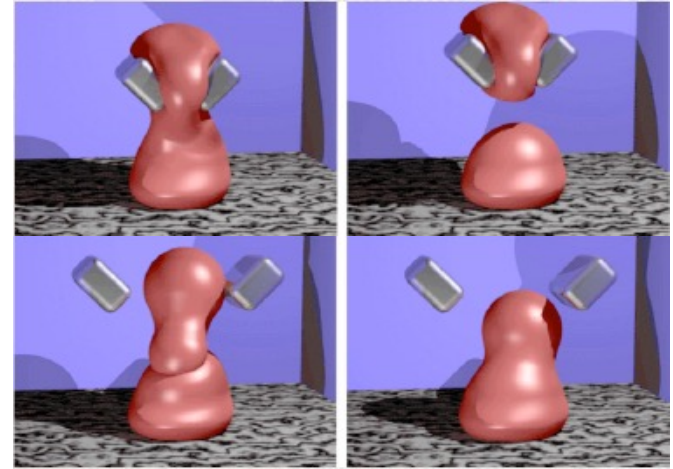
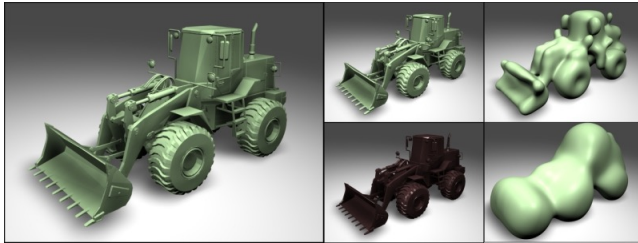
→ permet de représenter un équipotentiel

→ permet de déterminer si un point se trouve à l'intérieur ou l'extérieure du volume délimité par la surface (collisions, lancer de rayon)

Surfaces implicites adaptées pour les objets déformables avec changement de topologie ou de forme géométrique

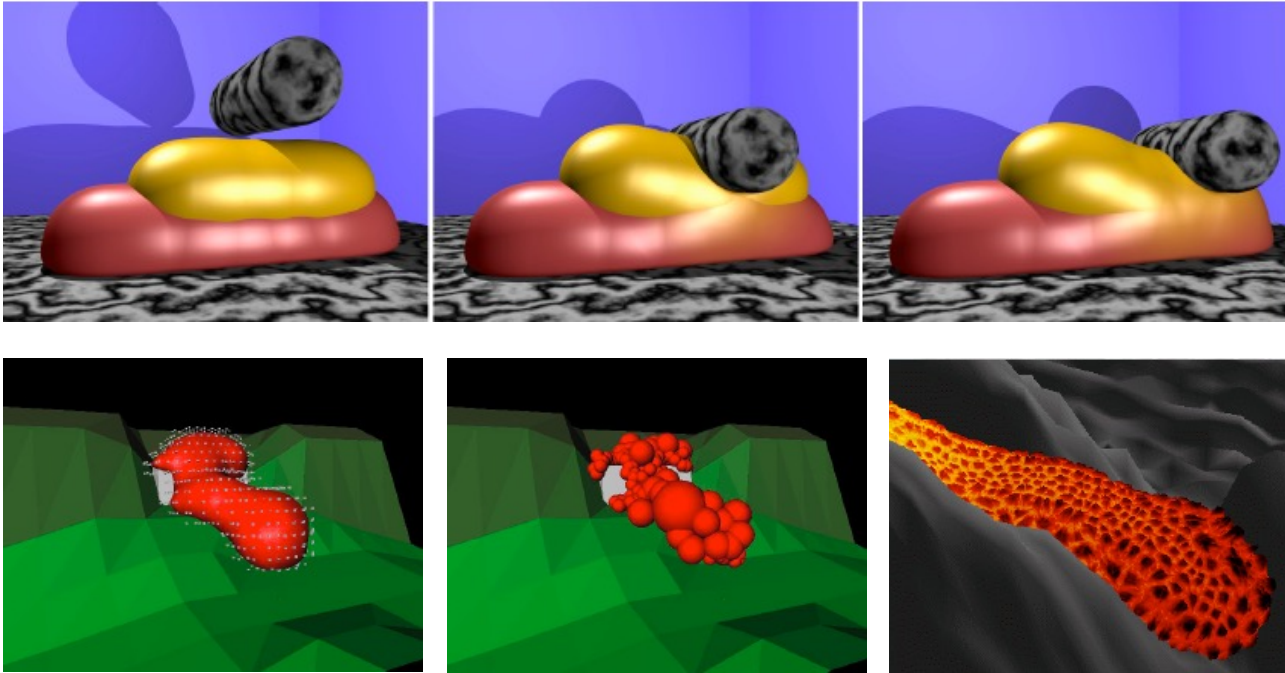
Modélisation géométrique

Modèles surfaciques – Surfaces implicites



Modélisation géométrique

Modèles surfaciques – Surfaces implicites



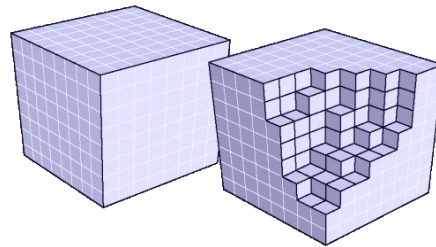
Modélisation géométrique

Modèles volumiques – Voxels

Discrétisation régulière de l'espace

2D → pixels

3D → voxels (élément de volume)

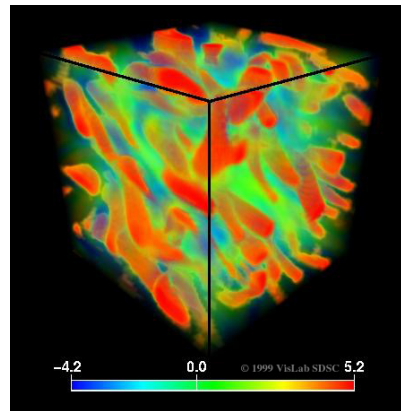


Une valeur est associée à chacun des voxels
mais pas de connexion entre les éléments de volumes

Modélisation géométrique

Modèles volumiques – Voxels

Valeurs obtenues par l'évaluation discrète d'une fonction issue d'une simulation numérique → valeurs de $f(x, y, z)$

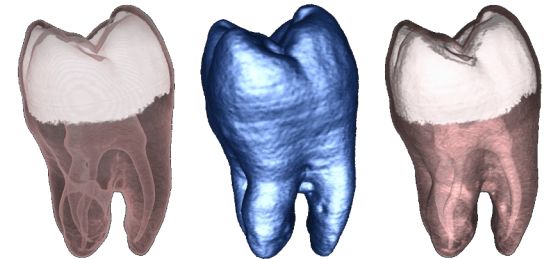
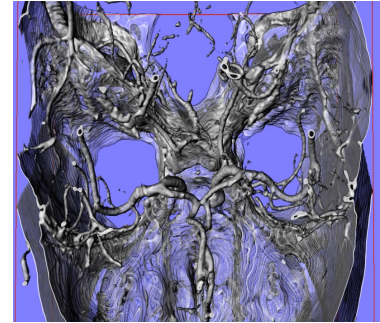
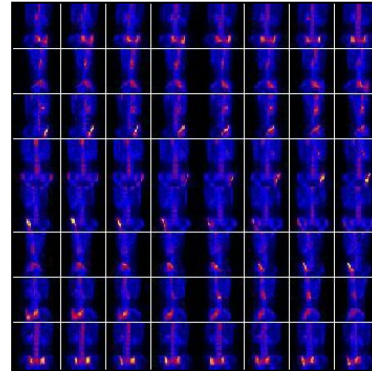
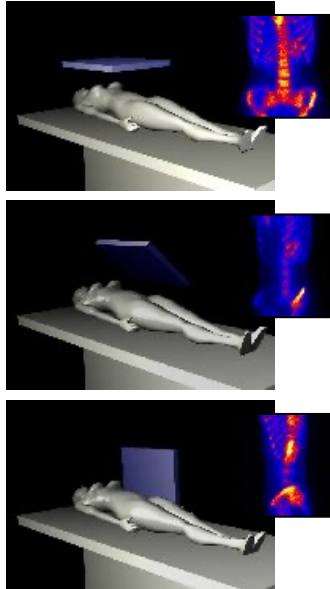


⇒ Rendu Volumique employé pour l'affichage des voxels

Modélisation géométrique

Modèles volumiques – Voxels

Valeurs obtenues à partir de coupes



Modélisation géométrique

Modèles volumiques – Maillages volumiques

Ensemble connecté de polyèdres

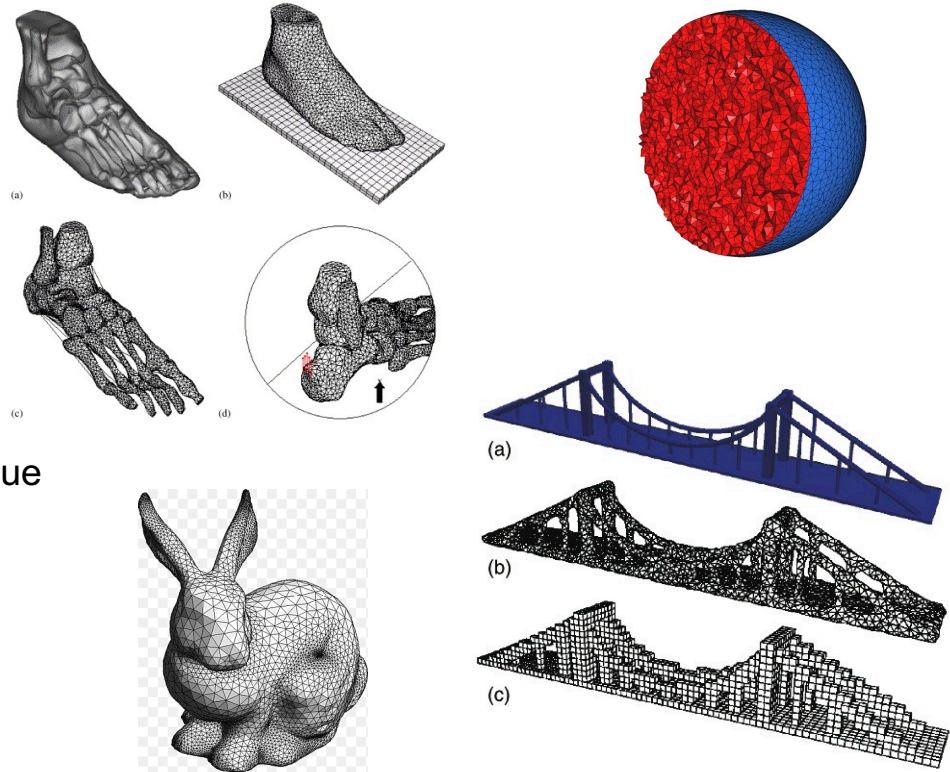
- tétraèdres, hexaèdres, prismes, etc.

Permet de représenter

- la forme et la topologie de l'objet

Affichage facile avec OpenGL

Obtention possible à partir d'un maillage surfacique



Modélisation géométrique

Au final : il existe de nombreux modèles qui sont adaptés à différents besoins

Dans le cadre de la réalisation **de simulations pour le médical**, modèles usuellement employés :

- **Maillages surfaciques et maillages volumiques**

Le modèle géométrique permet de décrire les caractéristiques des objets présents dans la scène 3D :

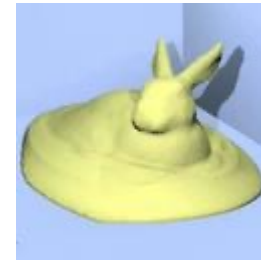
- De base : forme, topologie des objets
- Et plus encore : comportement physique, lumineux, propriétés du matériau, etc.

Etapes de bases pour créer des images virtuelles animées

1- **Modélisation** : représentation mathématique des objets virtuels

2- **Simulation / animation** : déformation et mouvement des objets virtuels

3- **Visualisation / rendu** : affichage des objets virtuels



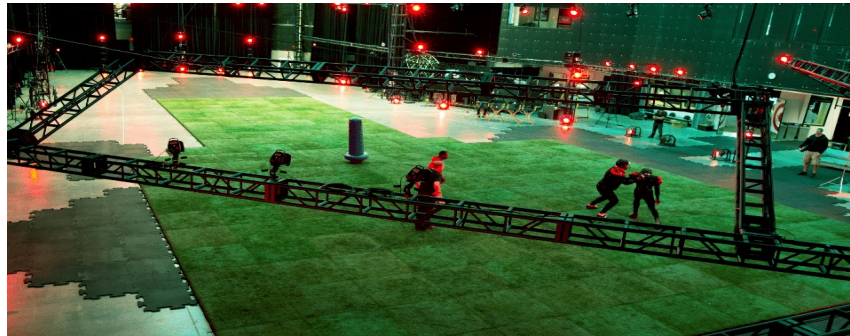
Animation d'objets 3D

Il existe différentes techniques d'animation en Informatique Graphique :

- Interpolation à partir de positions clés
 - cinématique directe et inverse
- Capture de mouvements
 - grande qualité mais spécifique
- Utilisation de modèles physiques
 - complexe mais permet une simulation automatique et réaliste

Animation de personnage

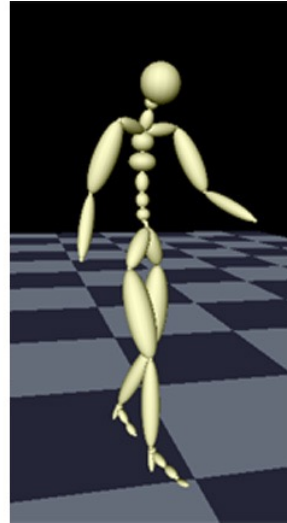
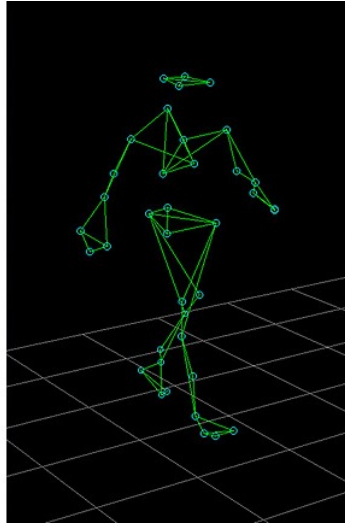
- *Motion Capture* (capture de mouvement) pour produire une animation



Animation de personnage

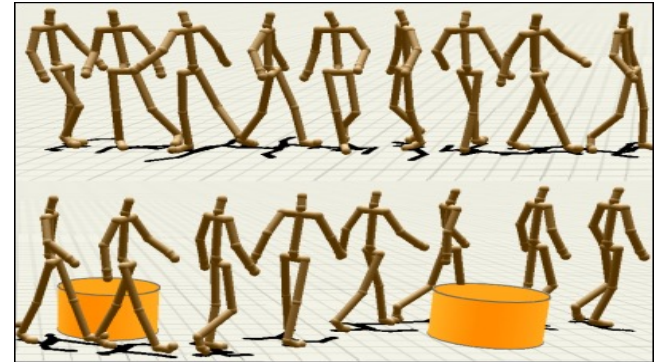
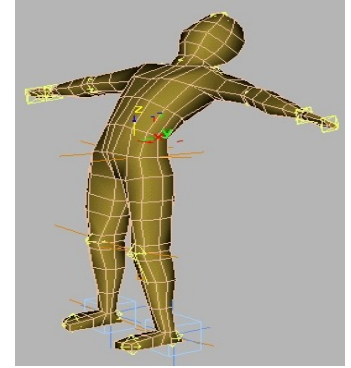
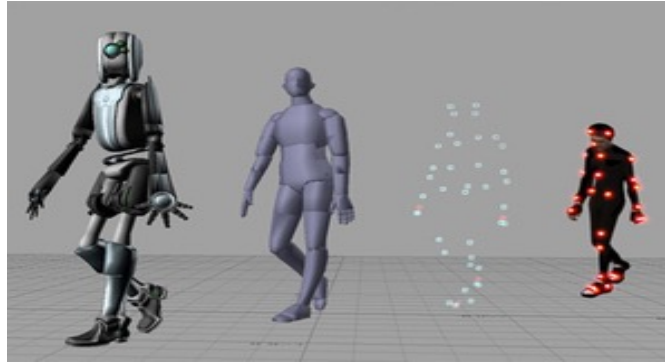
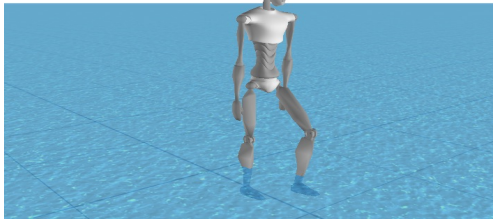
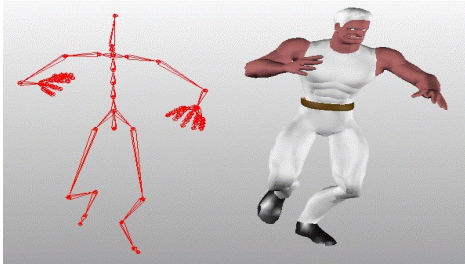
Les étapes de la capture de mouvement :

- Segmentation des marqueurs sur les images de chaque camera
- Reconstruction 3D de chaque marqueur
- Correspondance avec un squelette

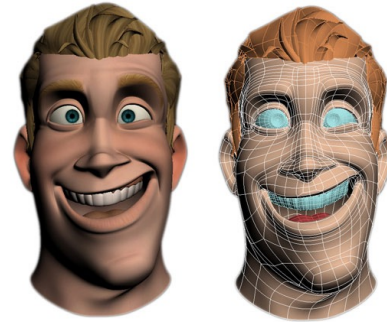


Animation de personnage

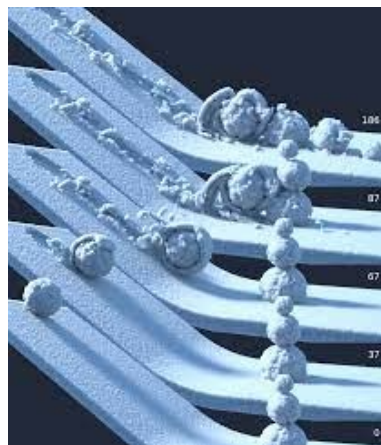
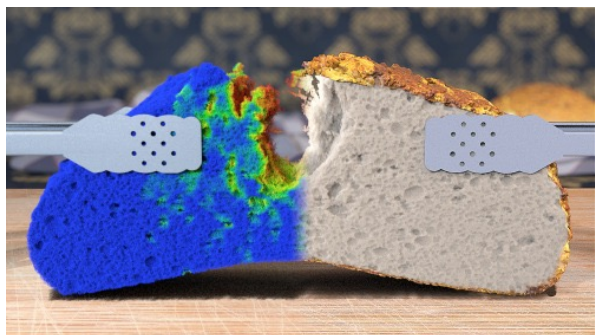
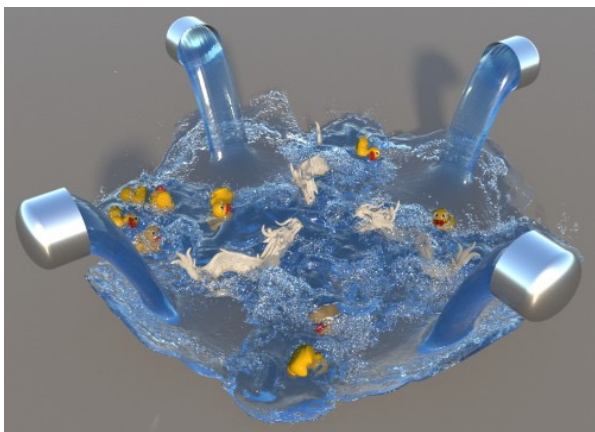
- Skinning
- Contrôle de mouvement



Animation de visage



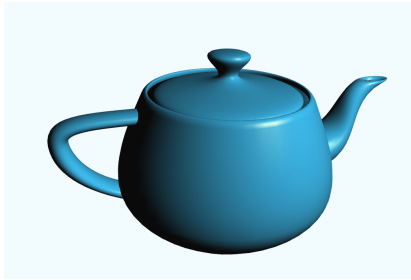
Animation par modèles physiques



Animation par modèles physiques

La modélisation est adaptée aux types d'objets à animer

Objets rigides

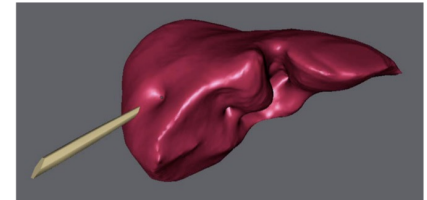
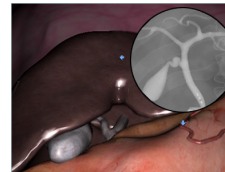


Objets déformables



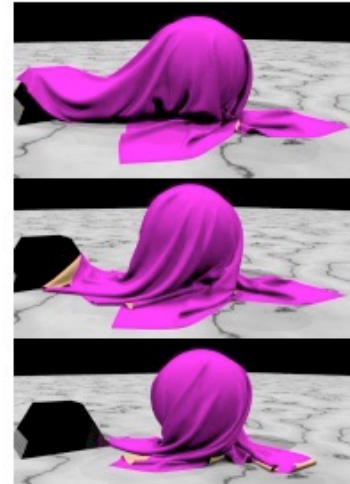
<https://youtu.be/InF6LoyJH2Q>

Simulation d'objets déformables à effectuer dans le cadre d'applications médicales



Animation par modèles physiques

Exemples de simulation d'objets déformables en Informatique Graphique



https://youtu.be/p5uhnSw8_Xw
<https://youtu.be/xvyGpBKevLM>

Animation par modèles physiques – Quelques vidéos

Simulation de fluide :

https://www.youtube.com/watch?v=8jD1bz4N3_0

<https://youtu.be/uRoH3HcftX8>

<https://youtu.be/chnS24QfgNY>

<https://youtu.be/Qve54Z71VYU>

<https://youtu.be/3yyLIG935zc>

<https://youtu.be/JI54WZtm0QE>

Simulation de sable :

<https://youtu.be/8iyvhGF9f7o>

<https://youtu.be/ZoZ0ZAzr6eg>

Simulation de fracture :

<https://youtu.be/INri-x2nK7o>

Simulation de neige :

<https://youtu.be/7Lbk9xJJRNU>

<https://youtu.be/1ES2Cmbvw5o>

<https://youtu.be/YQ7e06-MZec>

<https://youtu.be/V9tEXiQavNo>

Animation par modèles physiques

Il existe un certain nombre de modèles physiques :

- Méthode de résolution des Eléments Finis : basée sur la Mécanique des Milieux Continus
 - Simulation précise mais couteuse en temps de calcul
- Modèle discret des masses-ressorts : issue de l'informatique graphique
 - Simulation interactive mais approximation de la réalité
- Approche *Position Based Dynamics* : issue de l'informatique graphique
 - Basée sur la résolution de contraintes

Compromis précision et temps de calculs à faire

Animation par modèles physiques – Dynamique Newtonienne

Objet caractérisé par sa position, vitesse et accélération

Vitesse = dérivée par rapport au temps de la position : $v = x'$

Accélération = dérivée par rapport au temps de la vitesse : $a = v' = x''$

Position = intégration de la vitesse par rapport au temps

Vitesse = intégration de l'accélération par rapport au temps

Animation par modèles physiques – Dynamique Newtonienne

Objet également caractérisé par sa masse (en Kg) et par sa force (en N)

Accélération de l'objet proportionnelle à l'intensité de la force


Force d'un Newton = intensité de la force requise pour donner une accélération d'un mètre par seconde au carré à une masse d'un kilogramme

Animation par modèles physiques – Dynamique Newtonienne

Première loi de Newton

En l'absence de toute force, un objet au repos reste au repos

Si l'objet est en mouvement, et qu'aucune force extérieure ne lui est appliquée,
sa vitesse reste constante

 Mouvement d'un objet ne peut être modifié que par l'intervention d'une force

Animation par modèles physiques – Dynamique Newtonienne

Seconde loi de Newton : principe fondamentale de la dynamique

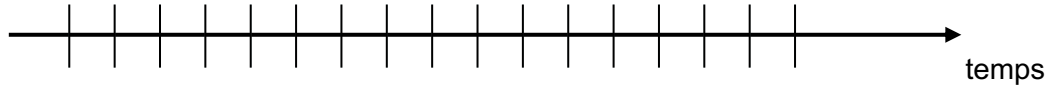
Soit un objet de masse constante m
accélération $\vec{a}(t)$
et force $\vec{F}(t)$

Equation du mouvement :

$$\vec{F}(t) = m\vec{a}(t) \Rightarrow \vec{a}(t) = \vec{F}(t)/m$$

Animation par modèles physiques – Dynamique Newtonienne

Temps continue est discrétisé



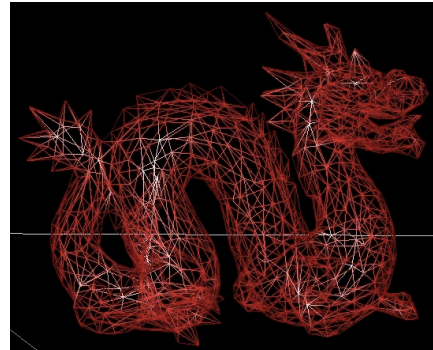
Etapes de calculs à effectuer à chaque pas de temps de la simulation = boucle de simulation

1. Calcul des forces exercées sur l'objets
 - forces dues au modèle physique employé
 - forces extérieures (gravité, etc.)
2. Calcul des accélérations (PFD) : somme des forces = masse x accélération
3. Intégration des accélérations pour obtenir les vitesses
4. Intégration des vitesses pour obtenir les positions

5. *Affichage de l'objet*

Animation par modèles physiques – Choix du modèle physique

Systèmes de particules



Animation par modèles physiques – Choix du modèle physique

Système masses-ressorts

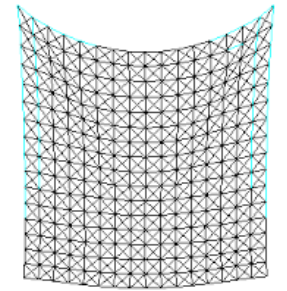
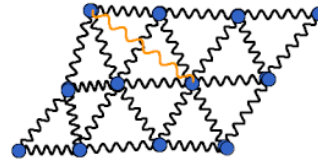
Matériel discrétisé en un ensemble de particules/masses connectées entre elles par des ressorts

Masse associée à chacune des particules

Vecteurs position, vitesse et accélération associée à chacune des particules

Les objets modélisés peuvent être :

- des courbes (cheveux, cordes, etc.) \rightarrow 1D
- des surfaces (textiles, surface de l'eau, etc.) \rightarrow 2D
- des volumes (objet volumique visqueux) \rightarrow 3D



Animation par modèles physiques – Choix du modèle physique

Formulation d'un système masses-ressorts

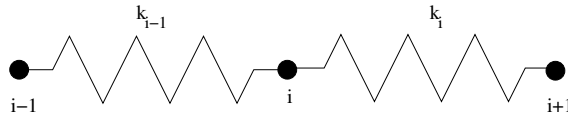
Système contient :

- p particules de masses m_i et de position x_i pour $1 \leq i \leq p$
- $(p-1)$ ressorts

Particule i est connectée par deux ressorts aux points $i-1$ et $i+1$

Ressort _{ij}

- relie les masses m_i et m_j
- avec une raideur $k_{ij} > 0$
- et une longueur au repos l_{ij}



Animation par modèles physiques – Choix du modèle physique

Equation du mouvement pour un système masses-ressorts

Soit \mathcal{A}_i = ensemble des indices j tels que m_i connectée à m_j

Soit F_i les forces externes appliquées à la masse i

Equation du mouvement de la particule i (PFD) :

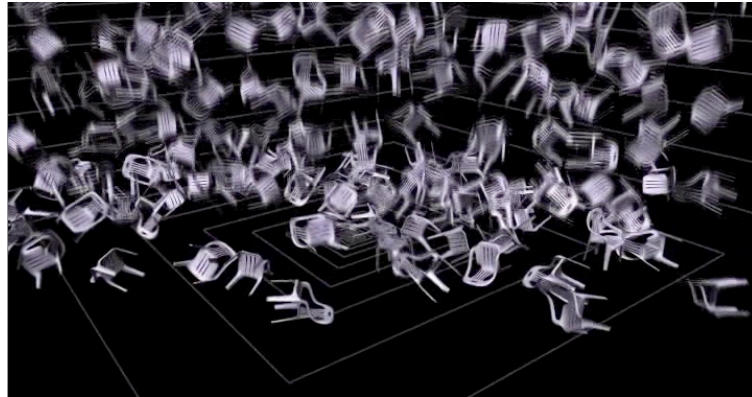
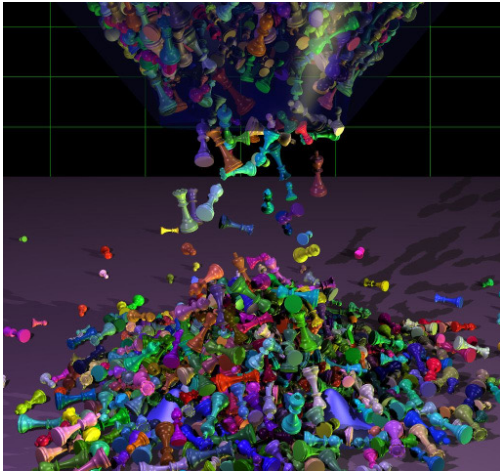
$m_i \ddot{x}_i =$ somme des forces exercées sur i

$$\Rightarrow m_i \ddot{x}_i = \sum_{j \in \mathcal{A}_i} k_{ij} (\|x_i - x_j\| - l_{ij}) + F_i$$

1. Obtention des accélérations des points au cours du temps
2. Intégration des accélérations pour obtenir les vitesses
3. Intégration des vitesses pour obtenir les positions

Animation par modèles physiques – Collisions

Gestion de plusieurs objets dans la scène

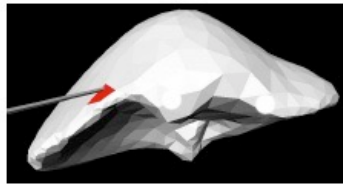


Souvent plusieurs objets en interaction dans la scène
→ collisions à traiter entre les objets en interaction

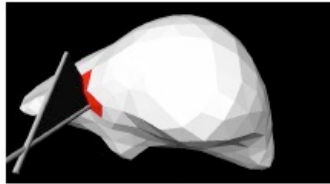
Animation par modèles physiques – Collisions

Exemple application pour le médical

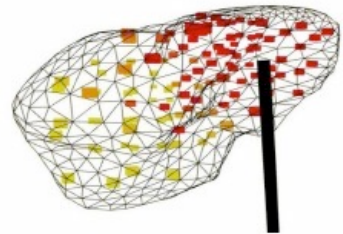
Collisions à traiter entre différents objets : entre organes ou entre organe et objet chirurgical



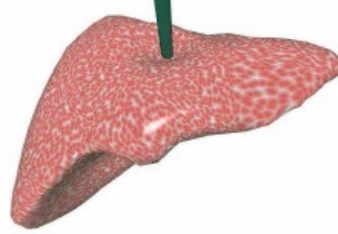
(i)



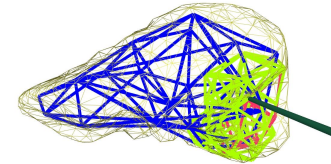
(ii)



(iii)



(iv)



Deux problèmes à résoudre :

- Détection des collisions dans la scène
- Traitement de ces collisions : réponse à donner

Animation par modèles physiques – Collisions

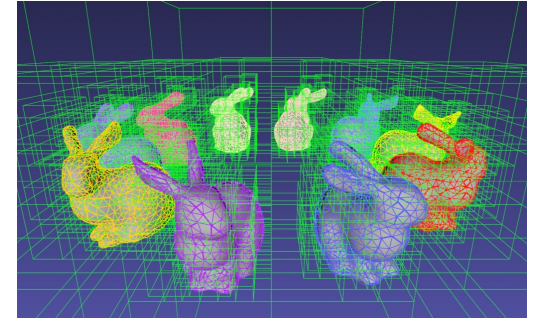
Détection des collisions

Plusieurs méthodes :

- Regarde s'il existe un plan partitionnant l'espace en 2 demi espaces
 - l'un contient l'objet A et l'autre l'objet B : pas de collision
- Calcul de la distance entre les objets
 - si elle est inférieure ou égale à 0, il y a collision

Utilisation de structure de données permettant d'optimiser la détection

- Objets sont encapsulés dans des boîtes
- Détection des collisions entre ces volumes englobants



Animation par modèles physiques – Collisions

Différents types de collisions possibles

- Contact en collision :
 - objets qui rebondissent l'un sur l'autre (force d'impulsion)
- Contact établi :
 - glissement ou frottement entre les objets (force de contact)

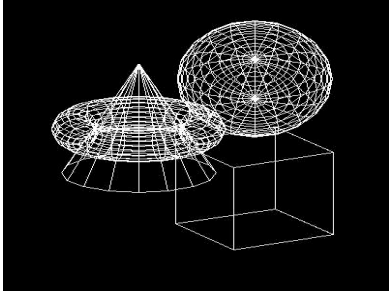


Modification de la vitesse des objets pour répondre aux collisions
Puis intégration de cette nouvelle vitesse dans la boucle d'animation

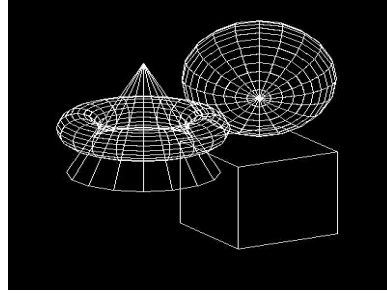
Etapes de bases pour créer des images virtuelles animées

- 1- **Modélisation géométrique** : représentation mathématique des objets virtuels
- 2- **Simulation / animation** : déformation et mouvement des objets virtuels
- 3- **Visualisation / rendu** : affichage des objets virtuels

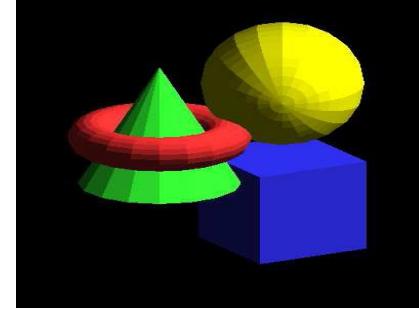
Rendu / affichage du modèle



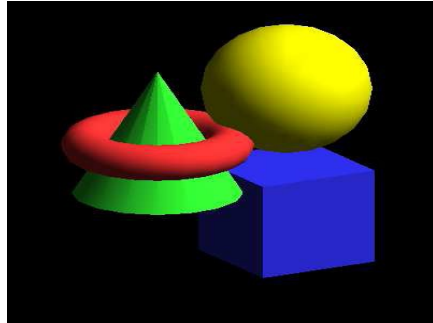
Fil de fer



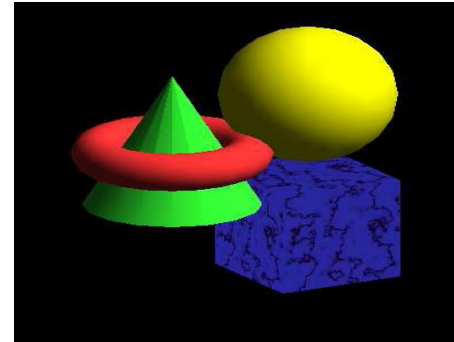
Faces cachées (objet)



Rendu Gouraud



Rendu Phong



Texture

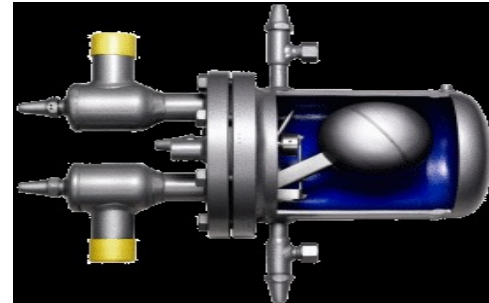
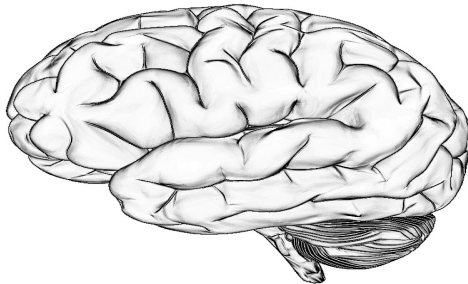
Rendu réaliste



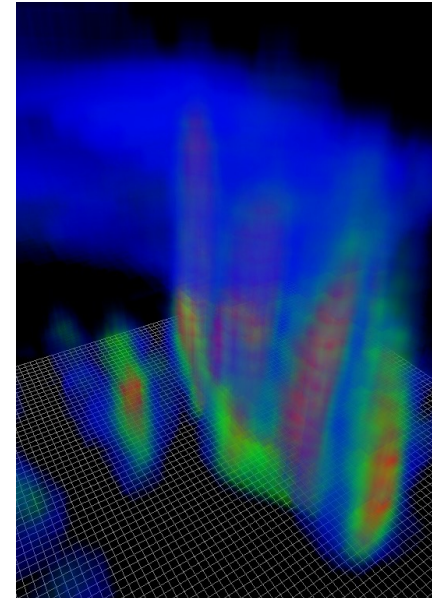
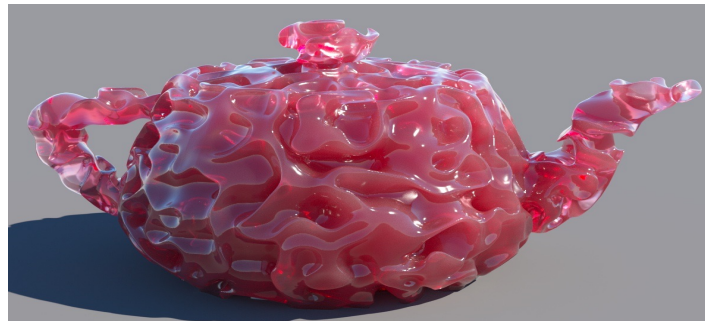
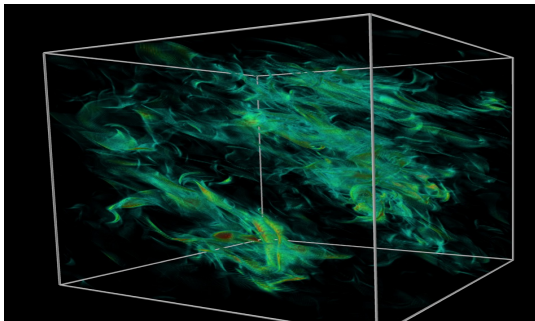
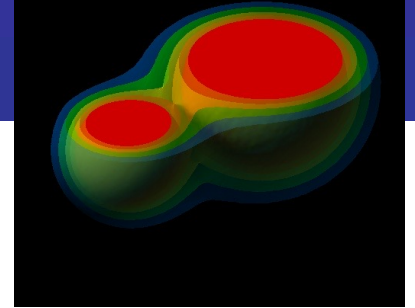
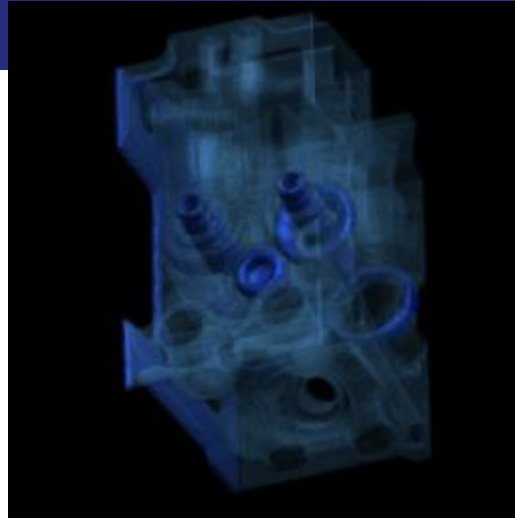
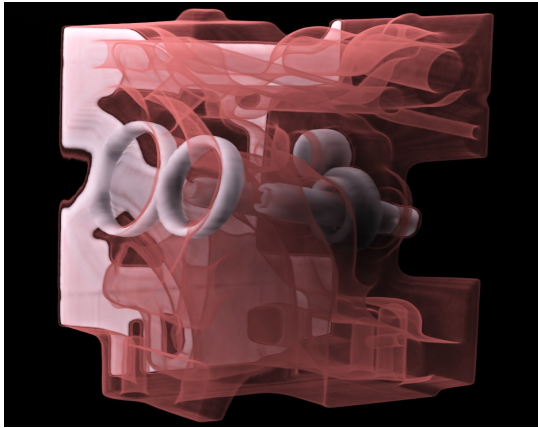
Rendu temps réel



Rendu non photo réaliste (NPR)



Rendu volumique

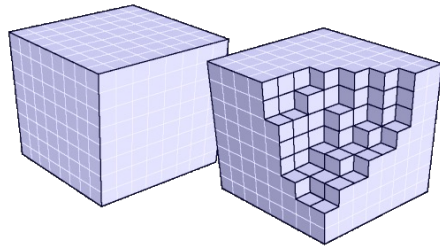


Rendu Volumique

Principe du rendu volumique

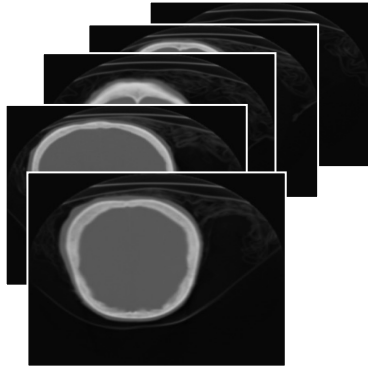
Le volume est vu en transparence grâce à l'emploi de voxels plus ou moins translucides

La juxtaposition des voxels transparents permet de voir l'intégralité du volume représentée par une image variant selon l'angle de visualisation

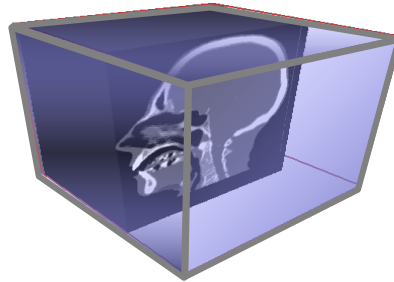


Rendu volumique pour le médical

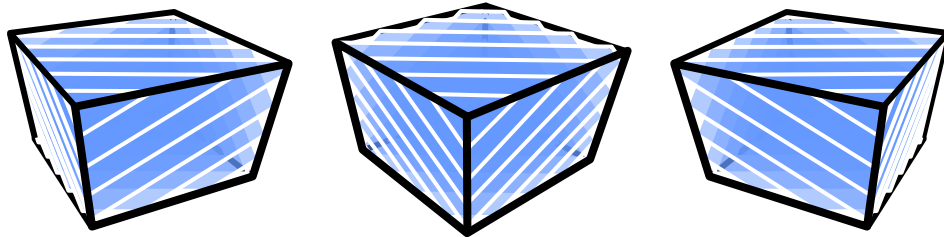
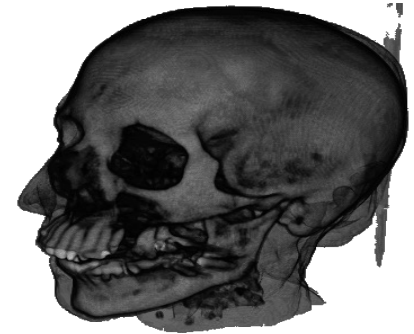
Données de départ



Assemblage

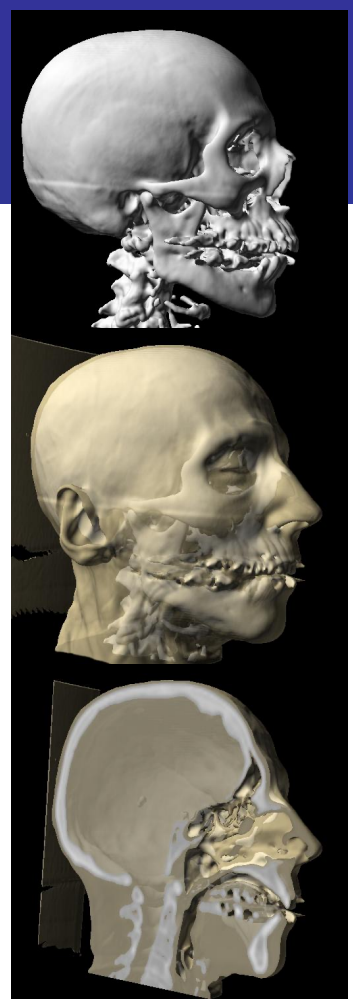
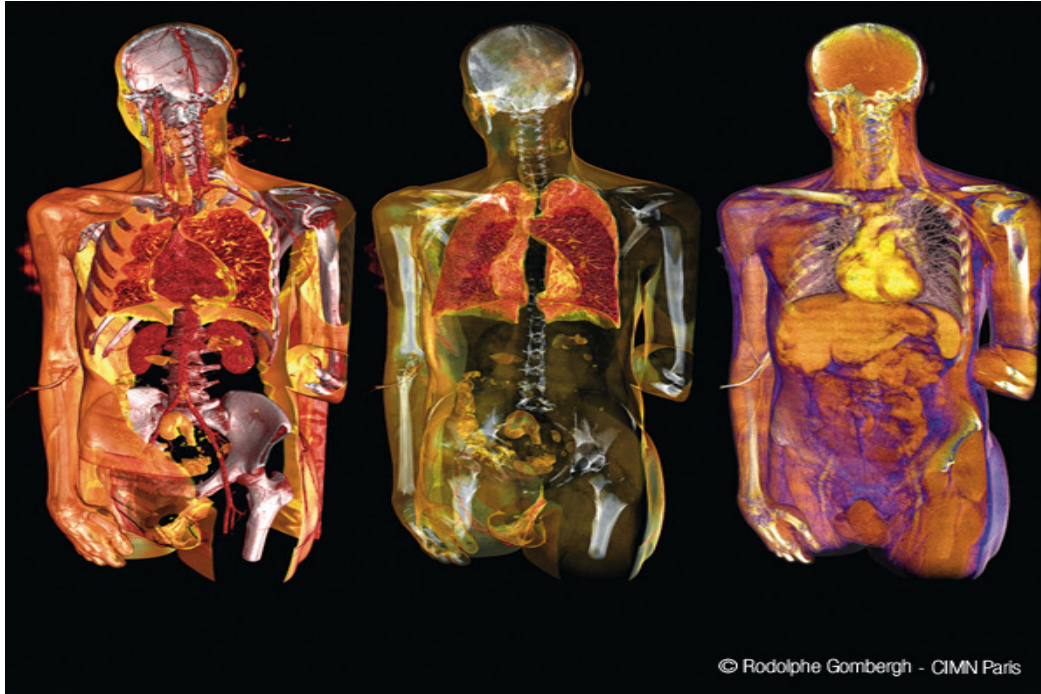


Rendu

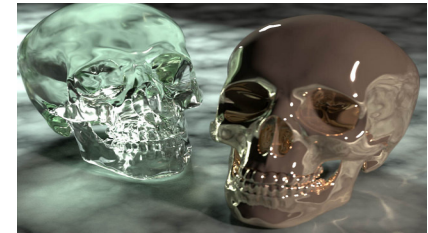
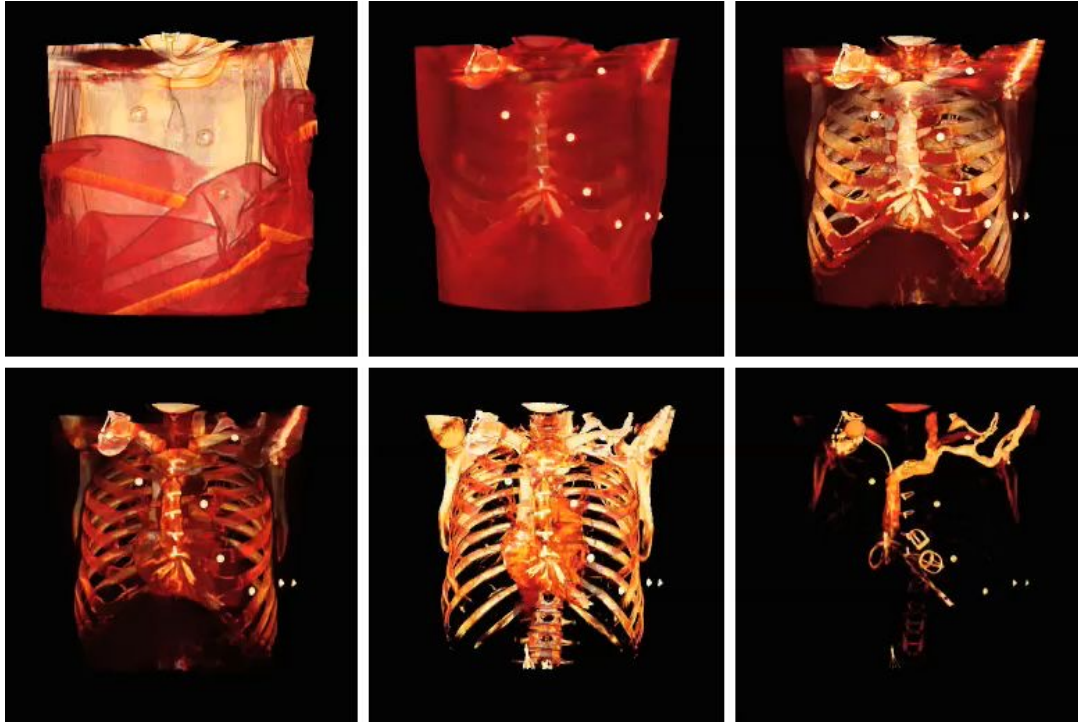


Tranches texturées
faisant face
à l'observateur

Rendu volumique pour le médical



Rendu volumique pour le médical



Etapes de bases pour créer des images virtuelles animées

- 1- **Modélisation géométrique** : représentation mathématique des objets virtuels
- 2- **Simulation / animation** : déformation et mouvement des objets virtuels
- 3- **Visualisation / rendu** : affichage des objets virtuels

Et ensuite, comment interagir avec l'image créée ?

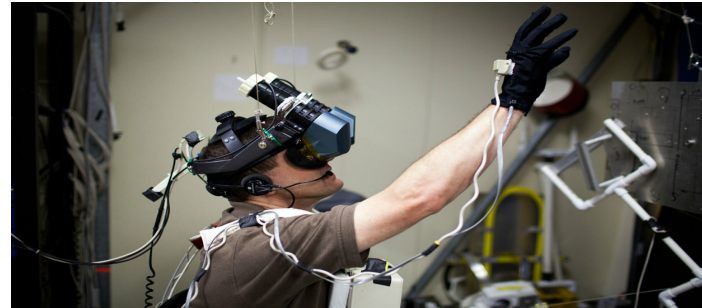


Comment interagir avec l'image créée ? c'est la Réalité Virtuelle

Image virtuelle + matériel de vision + robotique

Objectifs :

- Immerger l'utilisateur dans le monde virtuel
- Interagir avec le monde virtuel
- Reproduire les sensations du réel : vision, toucher, interaction



Objectif : améliorer l'immersion dans le monde virtuel

Ecran remplacé par un autre dispositif de vision

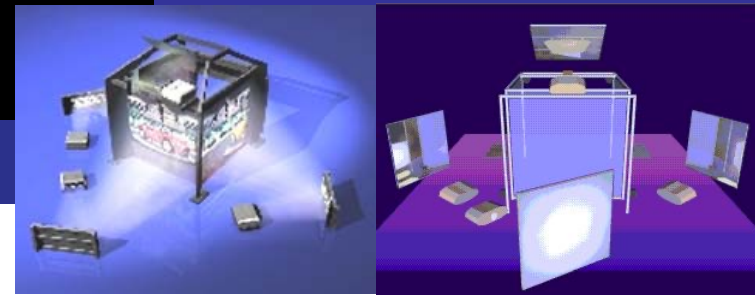
Mur de projections = association de plusieurs vidéo projecteurs

- Intérêts : haute résolution, collaboration, travail à l'échelle 1
- Mais...
 - Couverture incomplète du champ visuel
 - Coût élevé



Ecran remplacé par un autre dispositif de vision

- **Visio Cube ou CAVE** : immersion visuelle complète
 - Stéréoscopie masque structure cubique à l'utilisateur
 - Travail à l'échelle 1
 - Possibilité d'introduire des maquettes réelles



Ecran remplacé par un autre dispositif de vision

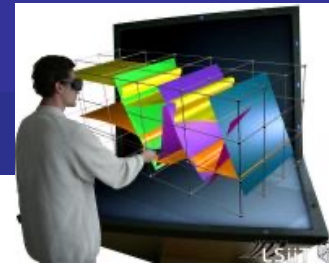
Visio plan ou Workbench

Premier workbench créé en 1993 au GMD (Allemagne)

Au lieu d'être immergé dans le monde virtuel, l'utilisateur le surplombe

- Couverture limitée du champ visuel
- Collaboration (limité en stéréoscopie)
- Position de travail approprié
(opération chirurgicale virtuelle)

Encombrement et coût raisonnable



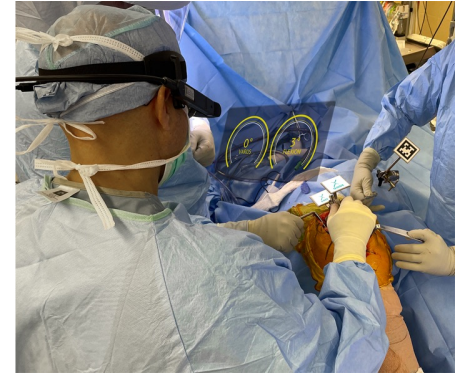
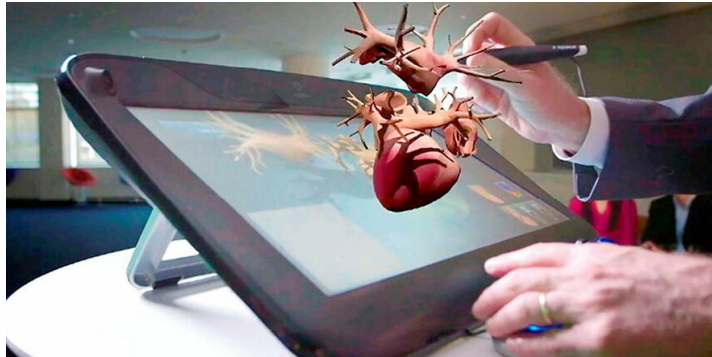
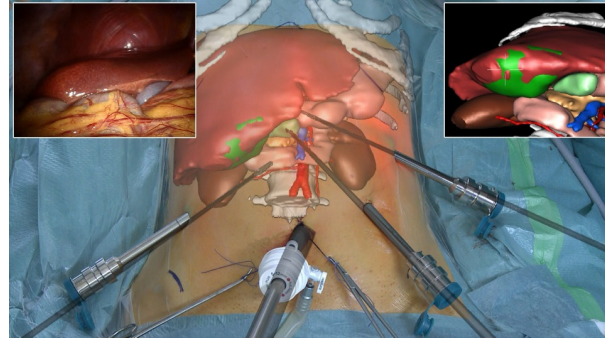
Ecran remplacé par un autre dispositif de vision

Casque de Réalité Virtuelle



Ecran remplacé par un autre dispositif de vision

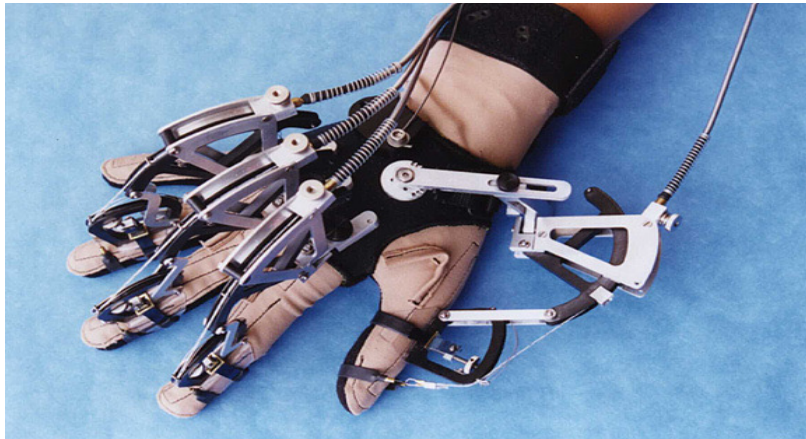
Et bien d'autres dispositifs...



Objectif : interagir en restituant la sensation du toucher

Souris-clavier remplacés par un autre dispositif d'interaction : interfaces haptiques

Gants haptiques : forces appliquées sur les doigts suite à la manipulation des objets présent dans l'environnement virtuel



Souris-clavier remplacés par un autre dispositif d'interaction : interfaces haptiques

Bras à retour d'efforts



Phantom Omni - Sensable



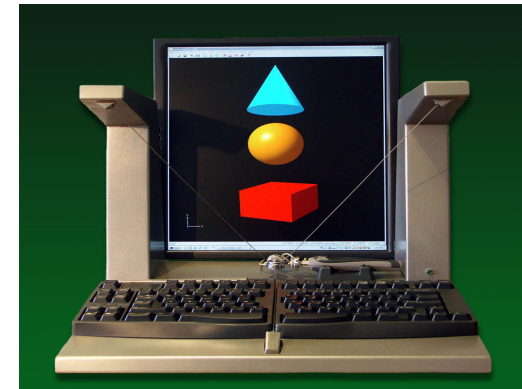
6 DOF Delta from Force Dimension



6 DOF Phantom Premium 1.5

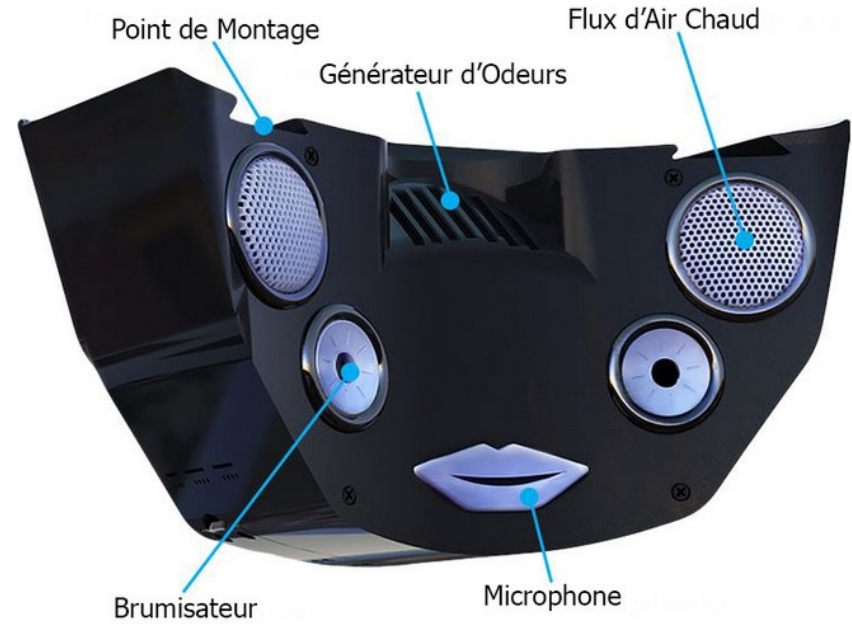


Falcom - Novint



Mantis - Mimic

Encore plus d'immersion : odeur



Domaines d'applications de la RV - Divertissement



Domaines d'applications de la RV - Santé

Traitement de différentes pathologies



Apprentissage



Domaines d'applications de la RV - Aviation



Simulateurs d'apprentissage pour pilotage avion

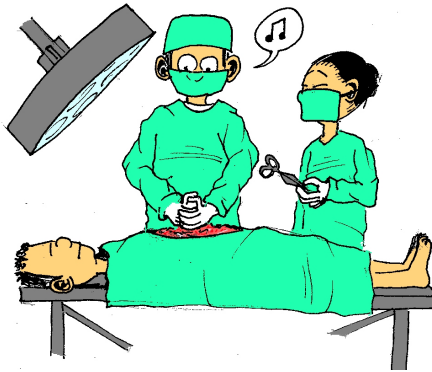


Apport de la Réalité Virtuelle dans le monde médical
Conception de simulateurs pour l'apprentissage de gestes médicaux

Simulateurs pour l'apprentissage de gestes médicaux - Motivation

Apprendre et acquérir la dextérité nécessaire aux gestes médicaux-chirurgicaux
sans risque pour le patient

« **Jamais la première fois sur le patient** »
Haute autorité de Santé (HAS), rapport 2012.



LEMI Dessins de Presse
et autres Galipotes

Simulateurs pour l'apprentissage de gestes médicaux - Motivation

Intérêt :

- Apprentissage sans risque pour le patient
- Entraînement avant une opération à risques (patient spécifique)
- Multiplier et cibler les situations rencontrées :
 - Gestes / pathologies / situations morphologies : usuels et rares
- Améliorer la connaissance et le raisonnement du geste
 - Pour mieux l'acquérir, le comprendre, se l'approprier

Plusieurs étapes à résoudre pour concevoir une telle application :

- Modèles géométriques
- Rendu moteur (animation et affichage)
- Interfaces haptiques
- Validation

Simulateurs pour l'apprentissage de gestes médicaux - Enjeux

Fournir à l'équipe médicale une interface la plus :

complète possible ; simple d'utilisation ; proche possible du terrain
Pour assurer l'acceptabilité de l'outil par le corps médical

Avoir un coût compatible avec utilisation en milieu hospitalier

Assurer la portabilité du dispositif proposé
Implantation de salles complètes ou simulateur déplacé régulièrement

Assurer le réalisme du simulateur pour immerger équipe médicale

Réalisme doit être en accord avec son apport pour la formation

Elaborer des scénarios pertinents et progressifs pour l'apprentissage

Evaluer l'apport pour l'apprentissage :

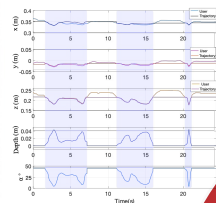
Accélération de l'apprentissage ? / Qualité de l'apprentissage ?

Simulateur doit permettre d'apprendre le « vrai geste »

« Il faut apprendre à opérer un vrai patient et non pas à opérer sur le simulateur. »

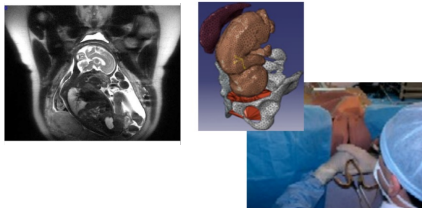
Etapes de conception d'un simulateur d'apprentissage basé sur la RV

Logiciel didactique



- 1- Analyser, comprendre le geste et son apprentissage
- 2 - Définir le cahier des charges du simulateur
- 3 - Elaborer des scénarios pertinents pour l'apprentissage
- 4 - Evaluer le simulateur

Données médicales, expertise & validation



Valider les composants du simulateur

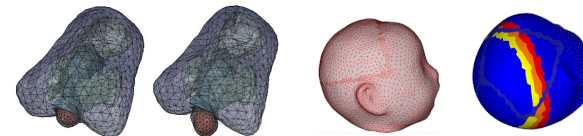
Didactique

Biomécanique
et Informatique
graphique

Clinique

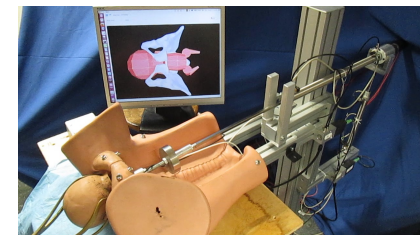
Robotique

Simulation numérique



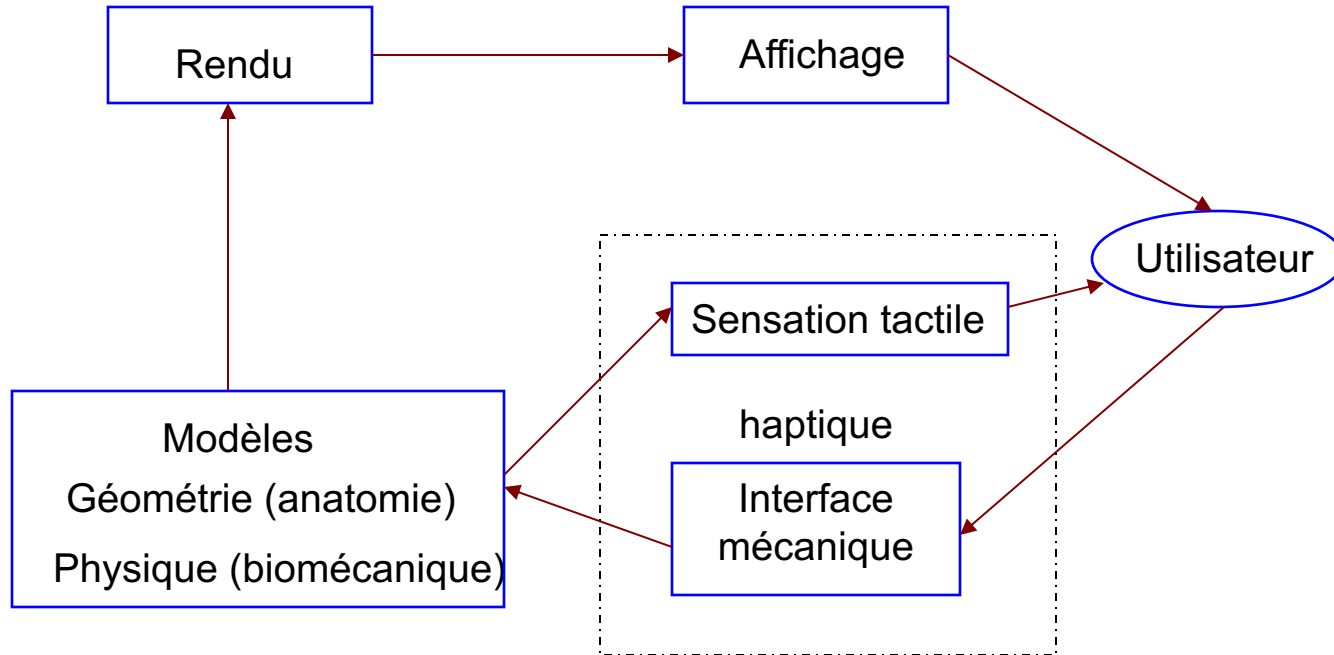
Simuler et visualiser le comportement
des organes en temps réel

Système haptique

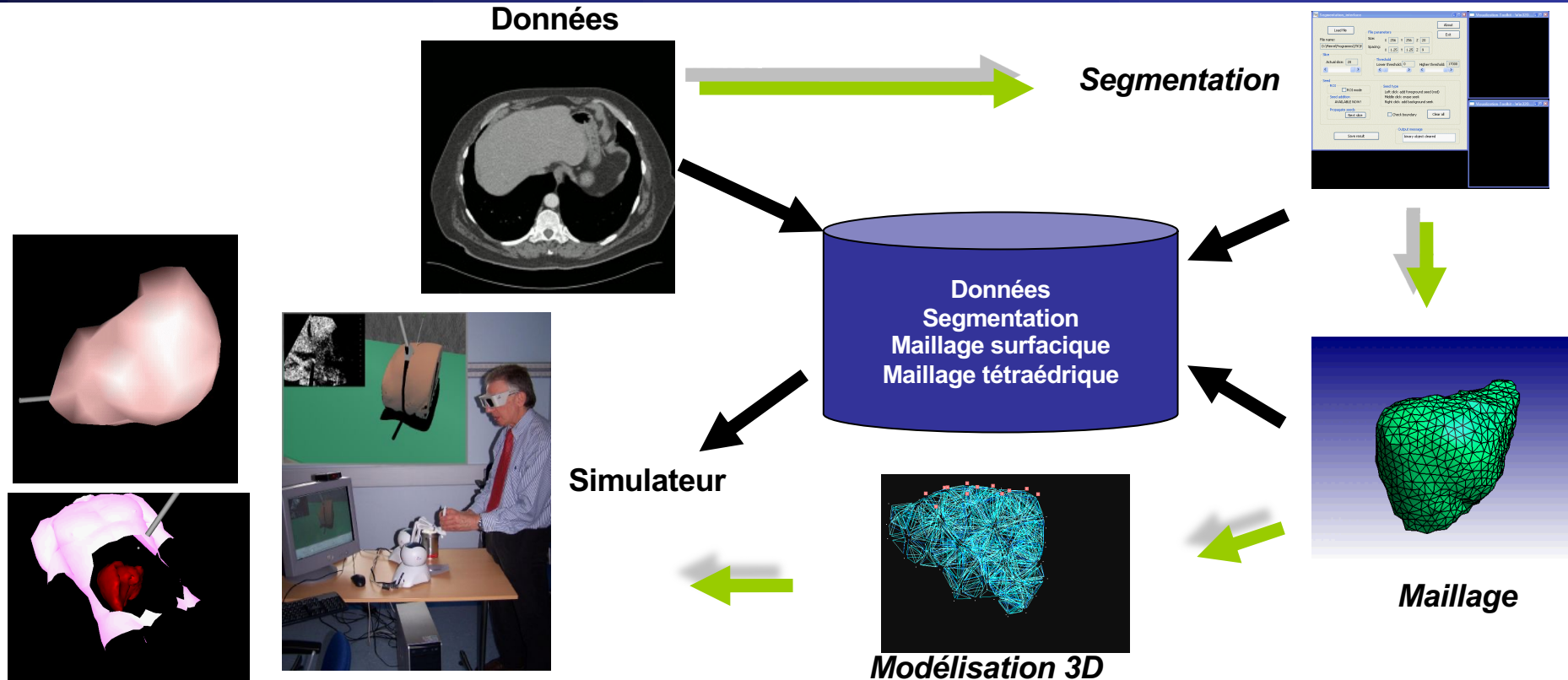


Reproduire sensations tactiles du geste

Architecture du système de Réalité Virtuelle



Architecture du système de Réalité Virtuelle - Simulateur de biopsies de foies



Etapes de base de la conception d'un simulateur médical

Récupération de données de patient de type IRM, scanners, etc.
Segmentation de ces données par un médecin expert

1- Modélisation

Traitement des données pour obtenir une structure 3D

2- Simulation / animation des objets 3D

3- Visualisation / rendu / affichage des objets simulés

4- Couplage avec des interfaces haptiques (Réalité Virtuelle)

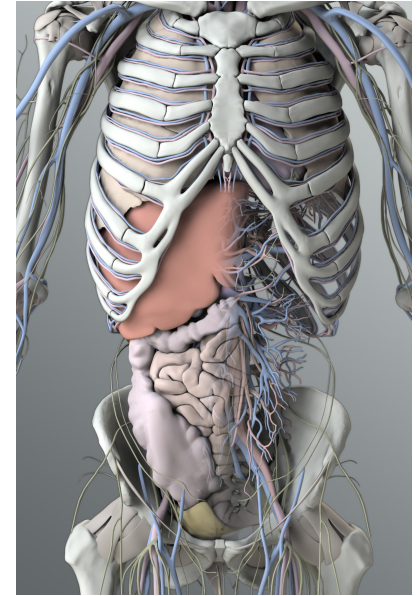
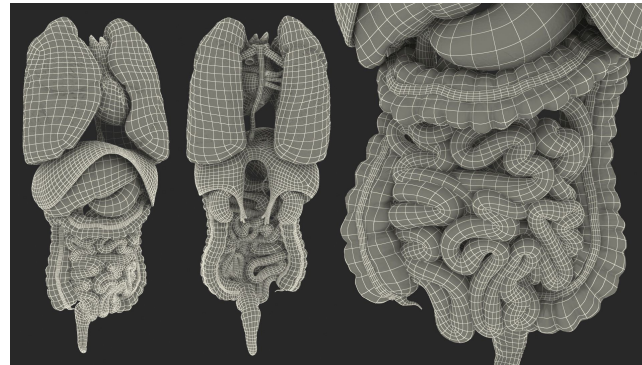
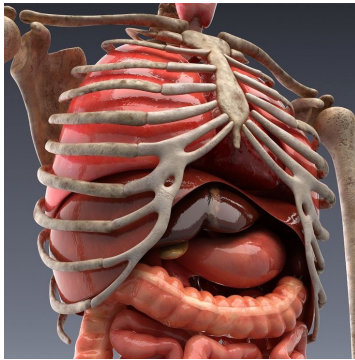
Simulateurs médicaux - Modèles géométriques

Objectif

Modélisation géométrique utilisée pour reproduire l'anatomie

Modèles géométriques issus des données scanner (CT Scan, ...)

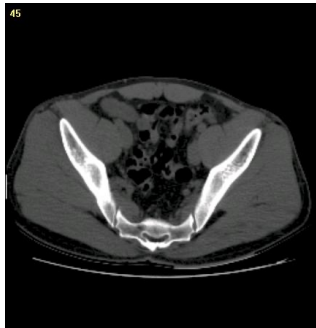
Données peuvent être spécifiques à un patient



Simulateurs médicaux - Modèles géométriques

Etapes pour passer des données du patient au modèle 3D

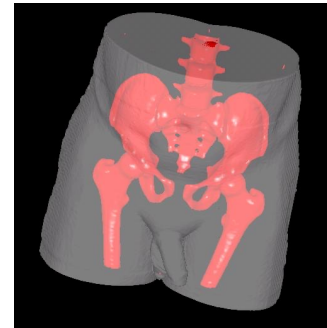
1. Récupération des données du patient
 - scanner, IRM, etc.
2. Segmentation des données par un médecin expert
3. Génération du modèle 3D



Données du patient :
CT images



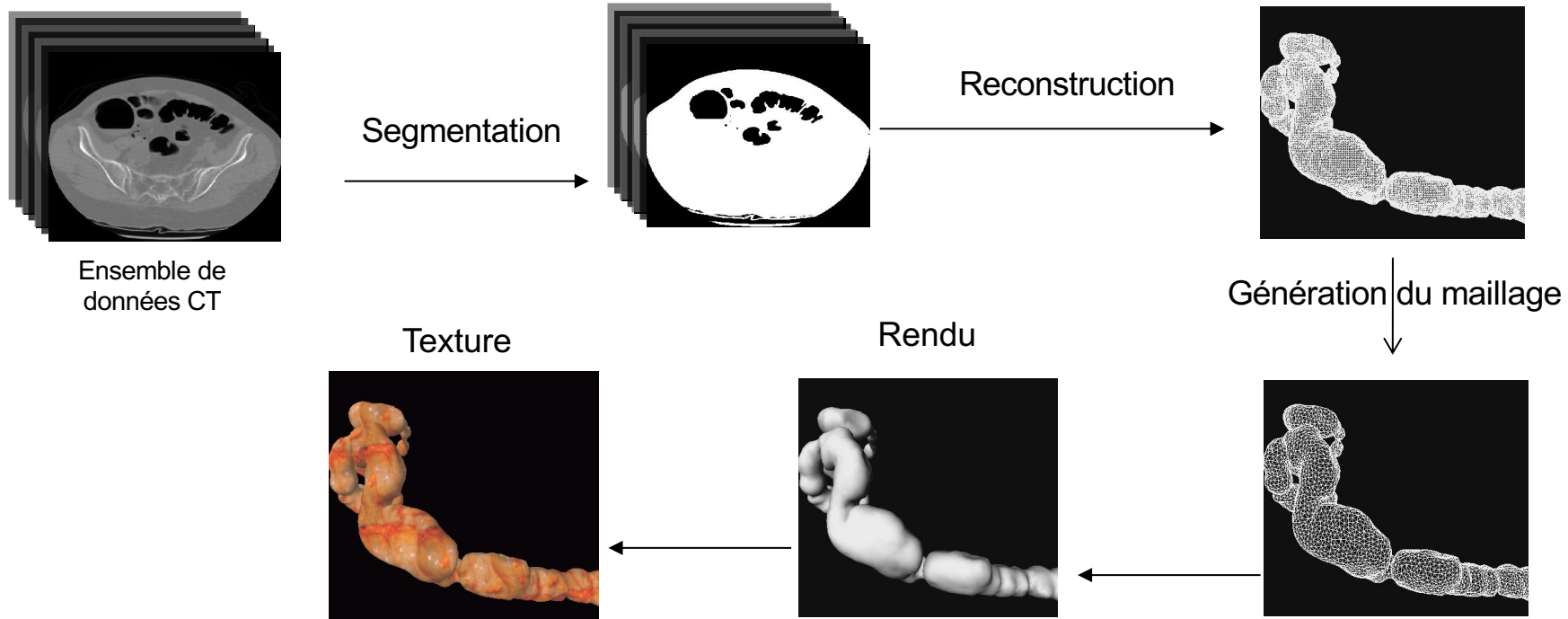
Segmentation



Maillage

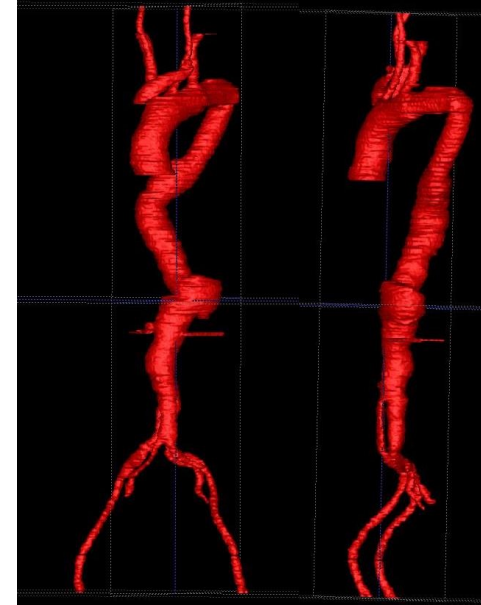
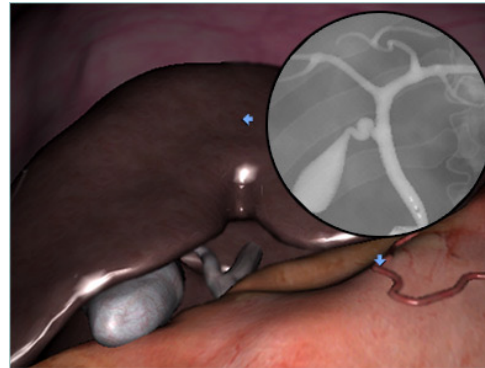
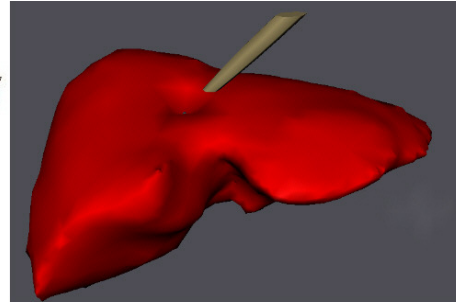
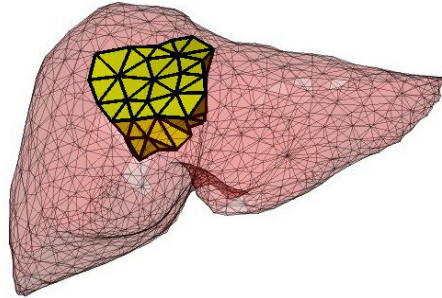
Simulateurs médicaux - Modèles géométriques

Etapes pour passer des données du patient au modèle 3D



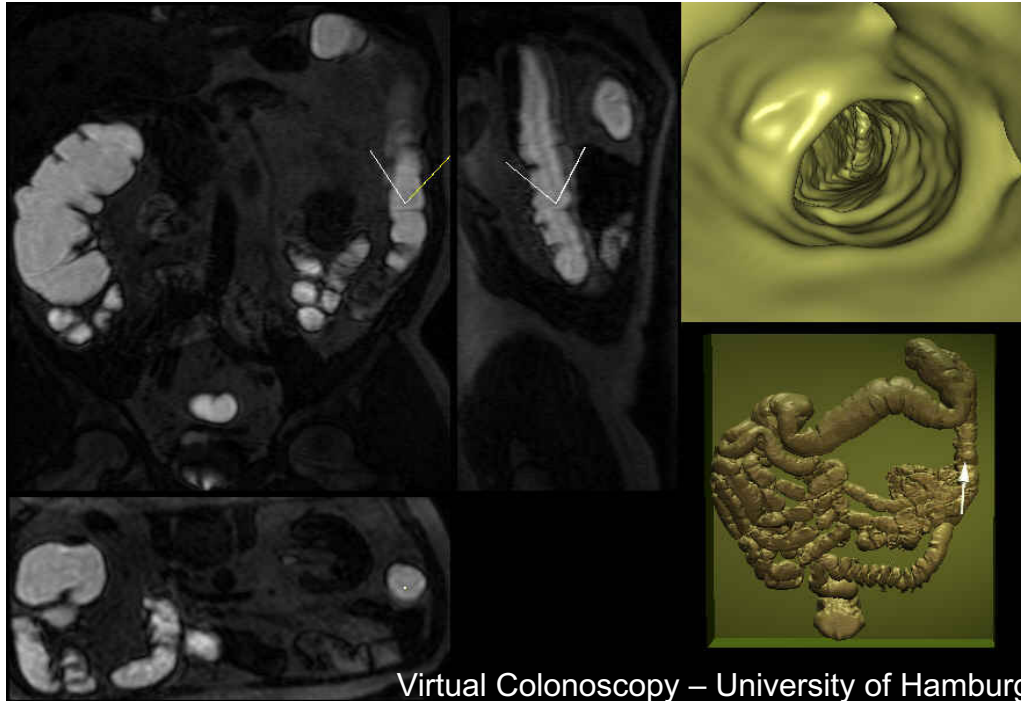
Simulateurs médicaux - Modèles géométriques

Quelques exemples

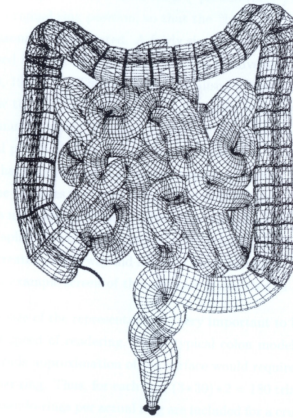


Simulateurs médicaux - Modèles géométriques

Quelques exemples



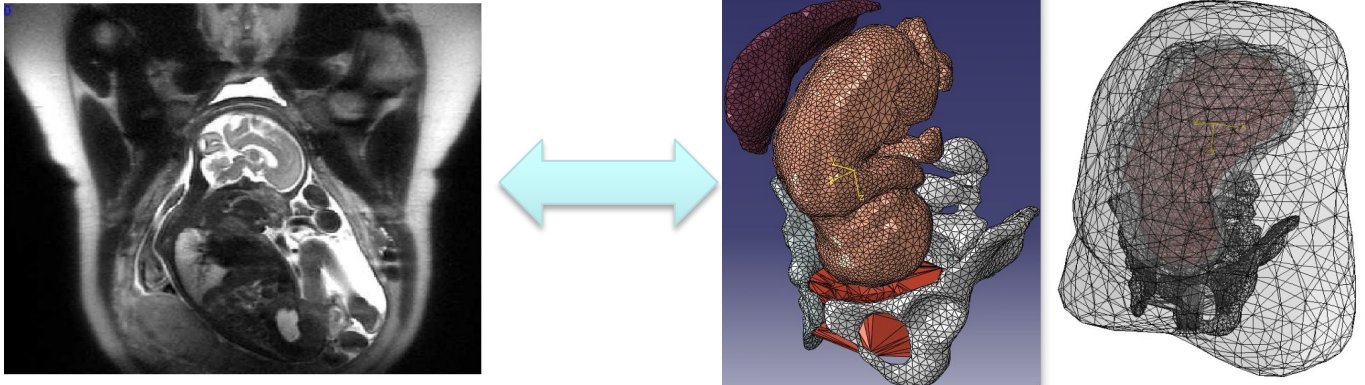
Virtual Colonoscopy – University of Hamburg



Modèle sphérique du colon

Simulateurs médicaux - Modèles géométriques

Quelques exemples

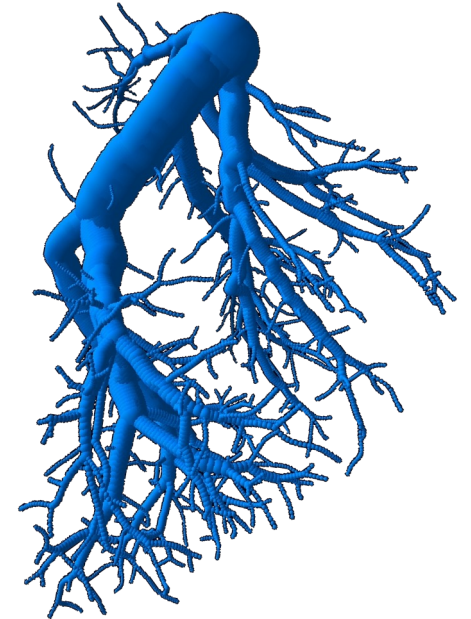


Modèle des organes impliqués dans un accouchement

Simulateurs médicaux - Modèles géométriques

Quelques exemples

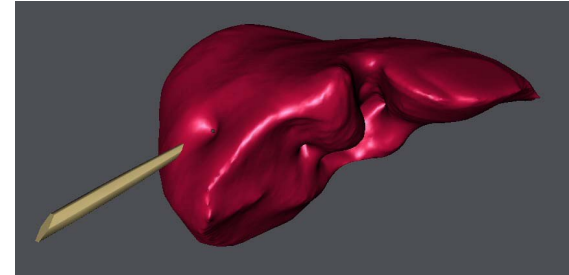
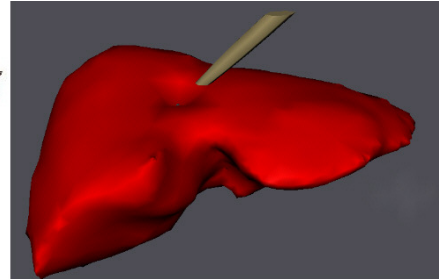
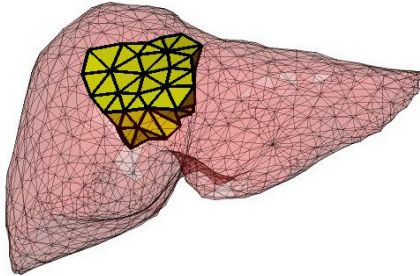
Projet Merlion – Données issues de l'IRCAD (Strasbourg)
Réseaux vasculaires



Simulateurs médicaux – Simulation par modèles physiques

Simulation physique

Reproduire le comportement des organes en interaction



Caractéristiques des objets déformables

Caractéristique d'un objet déformable = **élasticité**

→ objet retourne à sa forme initiale après une déformation

Objet déformable caractérisé par le rapport entre
la **contrainte** (*stress*) et la **déformation** (*strain*)

Contrainte = intensité de la force appliquée divisée par l'aire de la surface
sur laquelle la force est exercée

⇒ contrainte = force / surface

Simulateurs médicaux – Simulation par modèles physiques - MMC

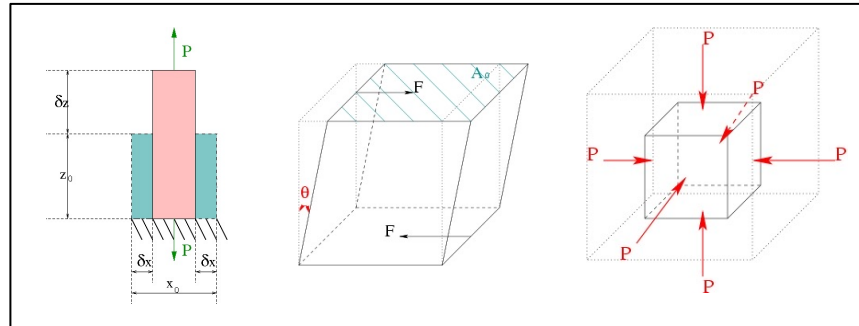
Principaux paramètres d'élasticité qui caractérisent les objets déformables

Différents types de déformation correspondant à différents paramètres

Elongation → module de Young, coefficient de Poisson

Cisaillement → module de cisaillement (Coulomb)

Compression → module de compressibilité (Bulk)

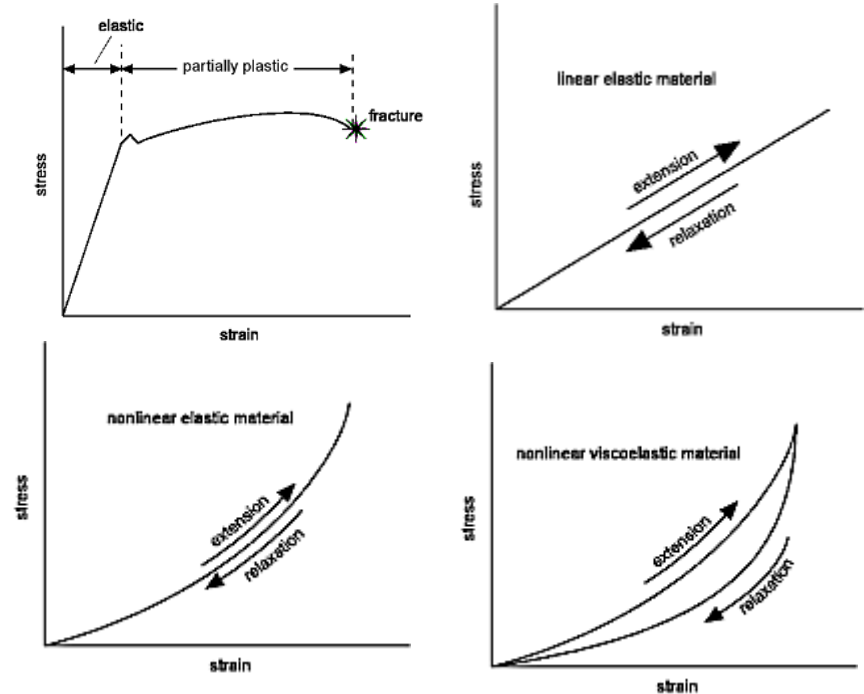
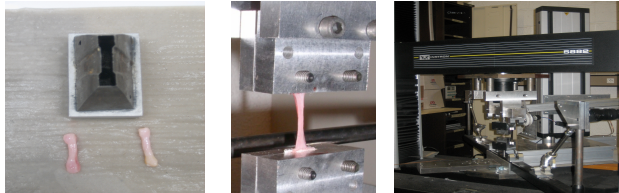


Simulateurs médicaux – Simulation par modèles physiques - MMC

Autres caractéristiques des objets déformables

Lois de comportement :

- linéaire
- plastique
- non-linéaire
- non-linéaire visco



Loi d'élasticité linéaire ou de Hooke :

contrainte (σ) = module de Young (E) x déformation (ϵ)

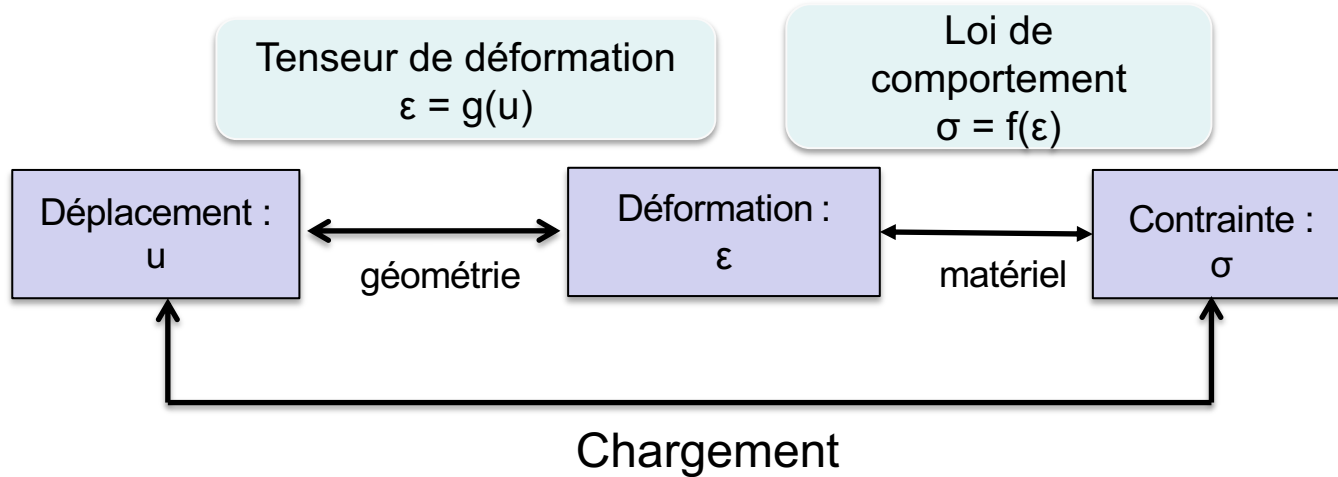
Simulateurs médicaux – Simulation par modèles physiques - MMC

Déformation de l'objet quantifiée par un **tenseur de déformations**

→ décrit l'état de déformation local résultant de contraintes (efforts internes)

→ matrice des allongements (termes diagonaux)
et des cisaillements (termes non diagonaux) subis par l'objet

→ tenseur des déformations (ϵ) est relié au champ de contrainte (σ)
par la **loi de comportement de l'objet** (Hooke, Néo-Hooke, Saint Venant Kirchoff, Yeoh, etc.)



Relations de la Mécanique des Milieux Continus

Tenseur de déformations de Green-Lagrange

Soit u le déplacement d'un point de l'objet par rapport à sa position d'origine

$$\varepsilon_{ij} = \frac{1}{2} (\vec{\nabla}u + [\vec{\nabla}u]^T + [\vec{\nabla}u]^T \cdot [\vec{\nabla}u])$$

Tenseur de déformations de Cauchy pour les petites déformations

$$\varepsilon_{ij} = \frac{1}{2} (\vec{\nabla}u + [\vec{\nabla}u]^T)$$

avec le gradient de u défini par :

$$\vec{\nabla}u = \begin{pmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} & \frac{\partial u_x}{\partial z} \\ \frac{\partial u_y}{\partial x} & \frac{\partial u_y}{\partial y} & \frac{\partial u_y}{\partial z} \\ \frac{\partial u_z}{\partial x} & \frac{\partial u_z}{\partial y} & \frac{\partial u_z}{\partial z} \end{pmatrix}$$

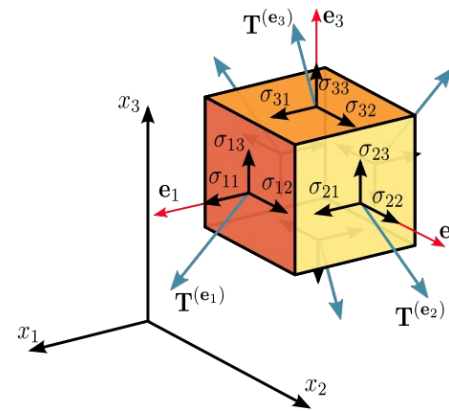
Tenseur de contraintes σ

- Définit la contrainte en un point à l'intérieur du matériau

$$\sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}$$

avec la traction $T = \sigma n dS$

normale \nearrow surface



Simulateurs médicaux – Simulation par modèles physiques - MMC

Loi de comportement élastique linéaire (petite déformation / isotropique)

– **Loi de Hooke** $[\sigma] = 2\mu[\varepsilon] + \lambda Tr([\varepsilon])[I]$

Avec les coefficients de Lamé $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ $\mu = \frac{E}{2(1+\nu)}$

E : module de Young – rigidité de l'objet (en Pa)

ν : coefficient de Poisson - compressibilité de l'objet (< 0.5)

$$[\sigma] = \frac{E}{(1+\nu)} \left([\varepsilon] + \frac{\nu}{(1-2\nu)} Tr([\varepsilon])[I] \right)$$

Loi de comportement pour élasticité non-linéaire

- Difficile de trouver la relation entre contrainte et déformation

$$\sigma = \frac{\partial W(\varepsilon)}{\partial \varepsilon} \leftarrow \text{Energie de déformation}$$

Simulateurs médicaux – Simulation par modèles physiques - MMC

Loi fondamentale de la dynamique s'écrit alors sous la forme

$$\operatorname{div}([\sigma]) + f_{\text{ext}} = \rho \ddot{u}$$

ρ masse volumique

f_{ext} forces volumiques appliquées au point

$\operatorname{div}(\sigma)$ les forces surfaciques

- résolution de l'équation (système différentielle du second ordre) donne le déplacement u au cours du temps
- obtention du mouvement de l'objet au cours du temps

Simulateurs médicaux – Simulation par modèles physiques - MMC



Simulateurs médicaux – Simulation par modèles physiques

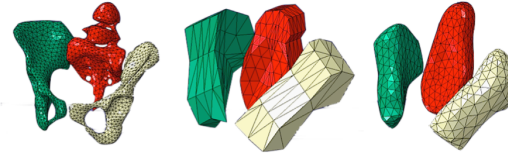
Intégration de la simulation dans un simulateur d'apprentissage de gestes médicaux
Simulation doit être en temps réel

Un certain nombre de simplifications à faire pour arriver au temps réel

- Simplification des modèles géométriques des organes
- Lois de comportement simples
- Modèles physiques
- Conditions aux limites
- Modèle de collision
- Parallélisation des algorithmes de simulation

Simulateurs médicaux – Simulation physique de la descente d'un fœtus

- Simplification of the meshes (less nodes)



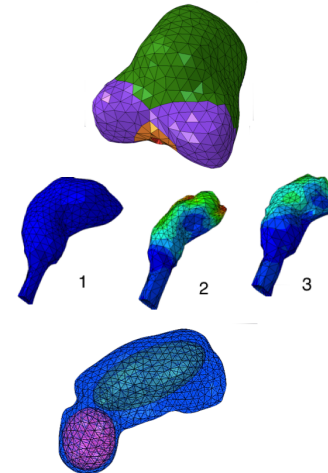
- Simplification of the constitutive law of organs

- **Pelvis:** Hooke - $E = 23 \text{ Mpa}$, $\nu = 0.3$, $\rho = 1\,000 \text{ kg/m}^3$

- **Abdomen:** Néo-Hooke - $C10 = 5\text{kPa}$, $\rho = 2\,500 \text{ kg/m}^3$

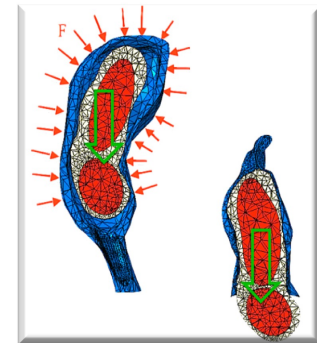
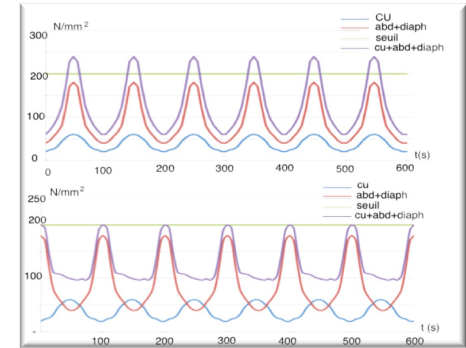
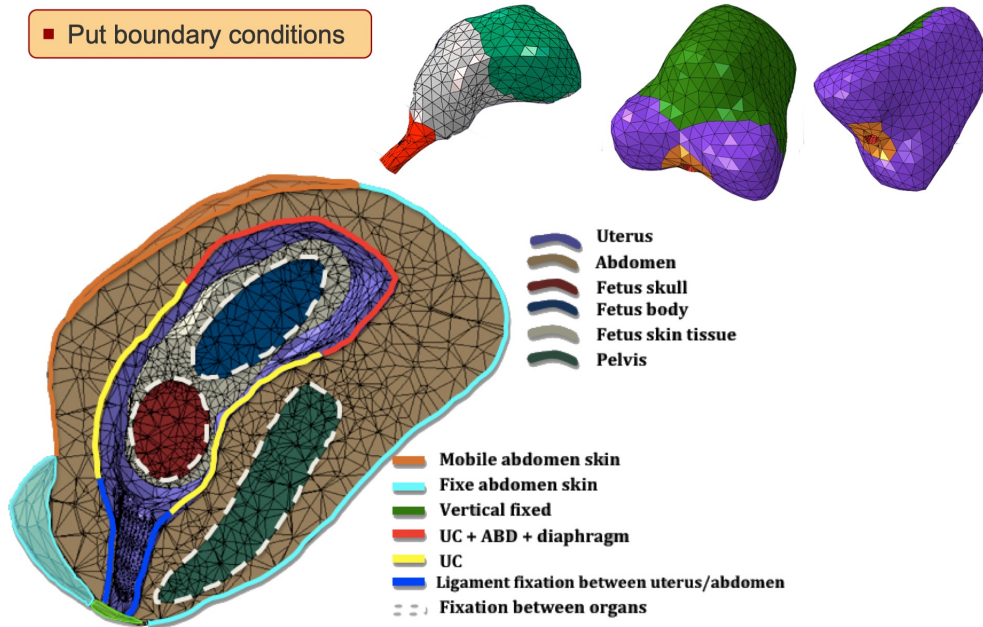
- **Uterus:** Néo-Hooke - $C10 = 30\text{kPa}$, $\rho = 950 \text{ kg/m}^3$

- **Fetus:** Néo-Hooke
 - Skin: $C10 = 130 \text{ kPa}$, $\rho = 400 \text{ kg/m}^3$
 - Skull: $C10 = 75 \text{ kPa}$, $\rho = 950 \text{ kg/m}^3$
 - Body: $C10 = 70 \text{ kPa}$, $\rho = 950 \text{ kg/m}^3$



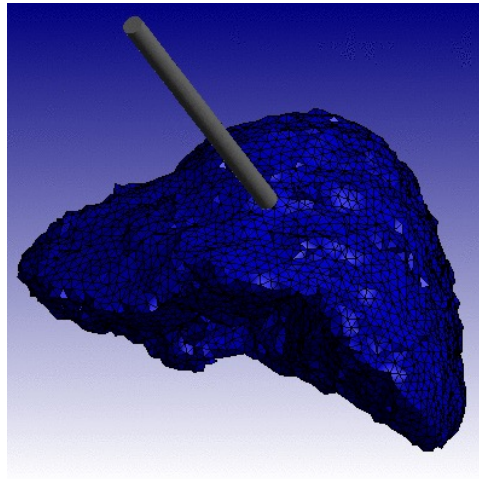
Simulateurs médicaux – Simulation physique de la descente d'un fœtus

Put boundary conditions

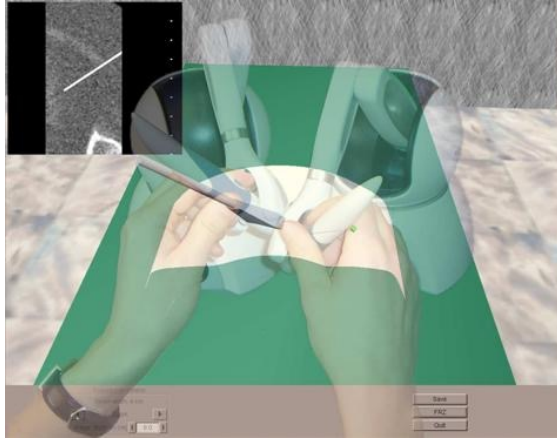


Simulateurs médicaux – Simulation physique

Interaction : gestion des collisions



Simulateurs médicaux - Interfaces haptiques



Bras Phantom pour la biopsie



VSP (Voxmap Pointshell)
pour la réalisation d'un
sondage

Simulateurs médicaux - Interfaces haptiques

Apprentissage pour la chirurgie dentaire
(DIGISENS)



Simulateurs chirurgicaux - Validation

Application validée par un médecin expert :

- Reproduction des sensations réelles ?

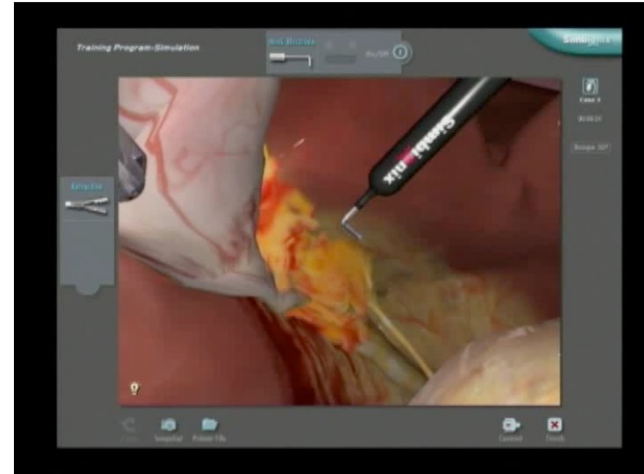
Application validée par un didacticien :

- Mise en place de scénarios pertinents pour l'apprentissage du geste
- Mais il ne faut pas que reproduire le geste mais se l'approprier
 - Outil (action) → Instrument (perception de pourquoi on fait le geste)
- Evaluation de cet apprentissage
 - Comparaison avec un apprentissage standard
 - « Bon sur simulateur » équivalent à « bon chirurgien » ?

Simulateurs chirurgicaux – Simulateurs commerciaux



Lapsim Laparoscopic
Simulator

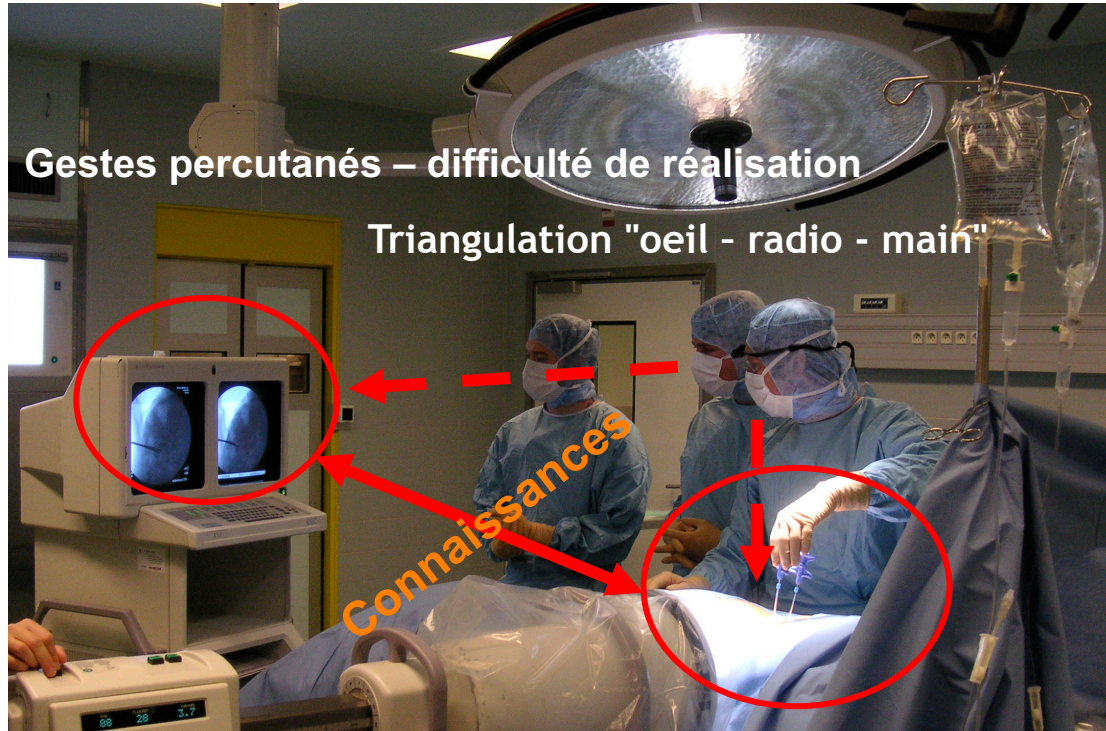


Simbionix Lap Mentor

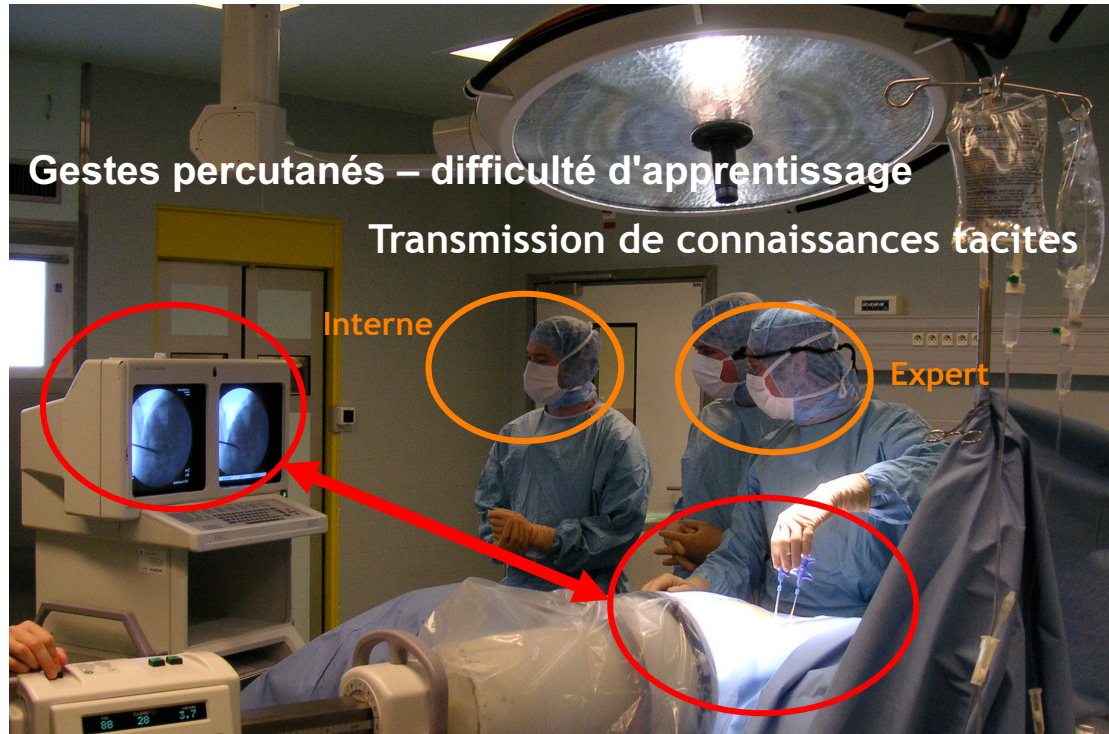
<https://www.youtube.com/watch?v=R-YUq2w6Z-A&t=37s>

<https://www.3dsystems.com/video/simbionix-lap-mentor-lobectomy>

Vissage Sacro-Iliaque – Projet TELEOS

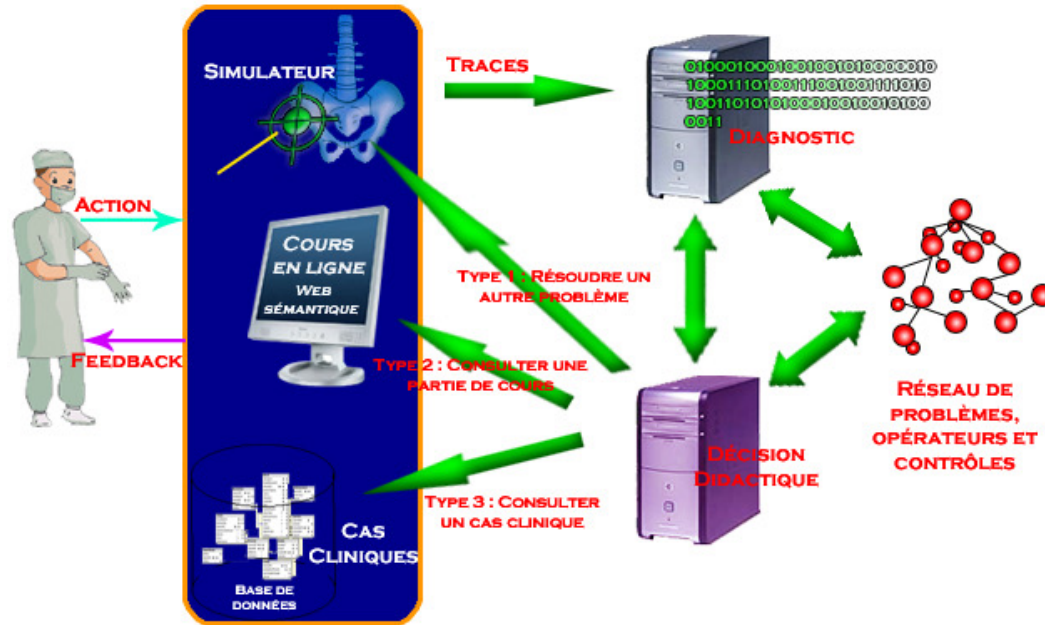


Vissage Sacro-Iliaque – Projet TELEOS

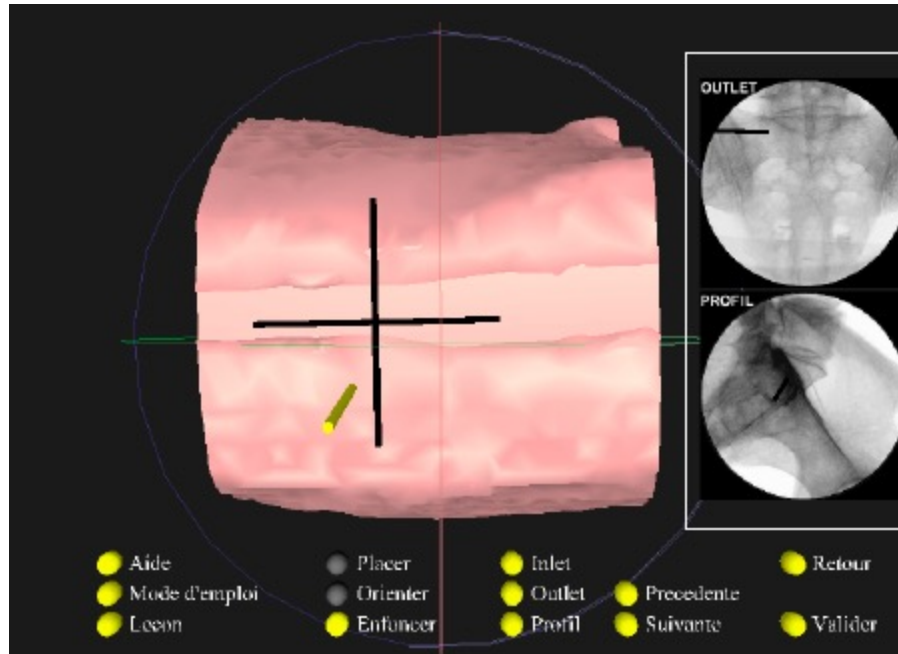


Vissage Sacro-Iliaque – Projet TELEOS

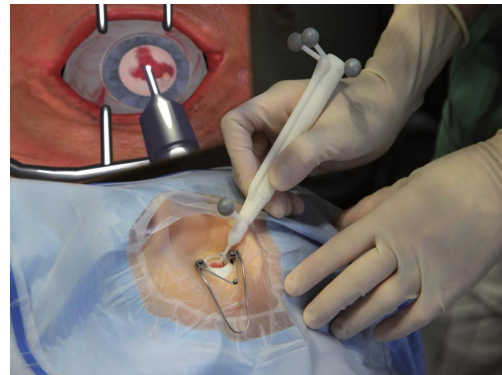
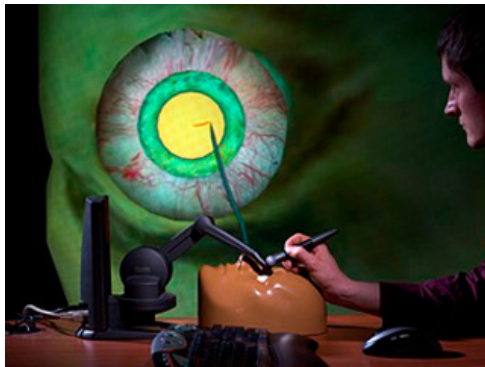
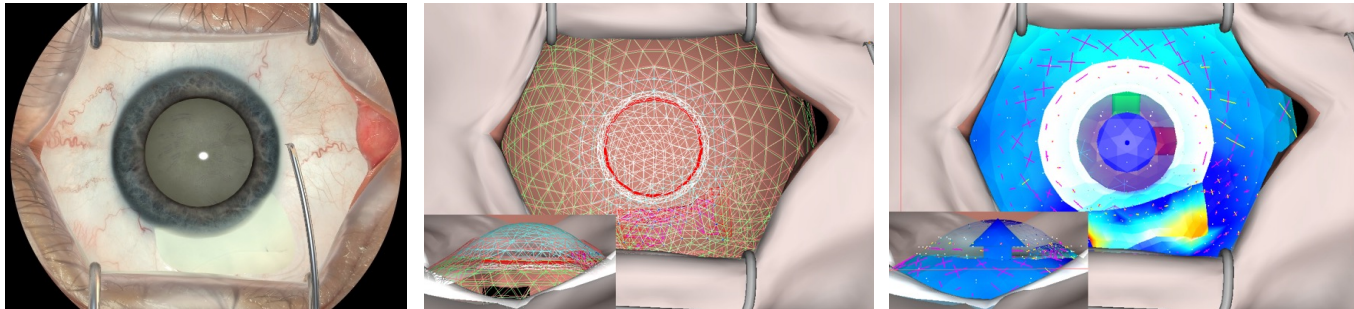
ARCHITECTURE



Vissage Sacro-Iliaque – Projet TELEOS



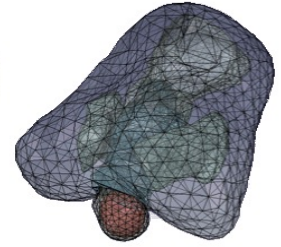
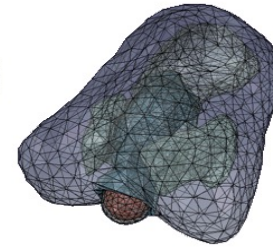
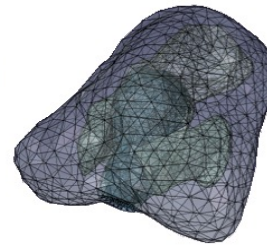
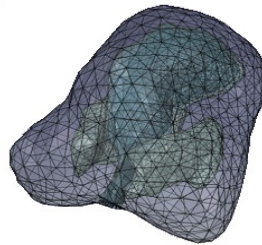
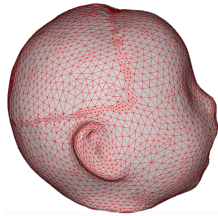
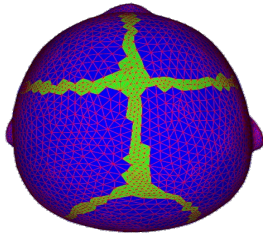
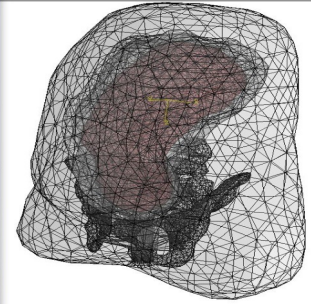
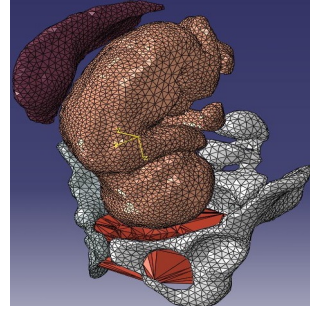
Simulateurs de gestes chirurgicaux – Traitement de la cataracte



INRIA Lille
HelpMeSee

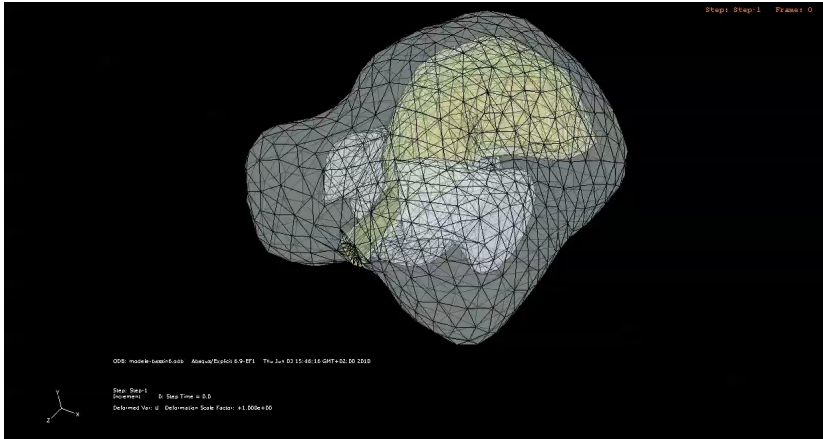
<https://www.youtube.com/watch?v=-UJYWlaTZr0>

Simulateurs de gestes chirurgicaux – Usage des forceps pour extraction fœtus

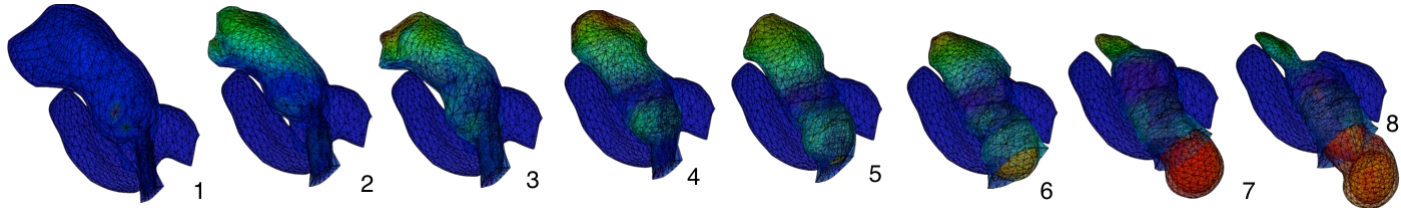
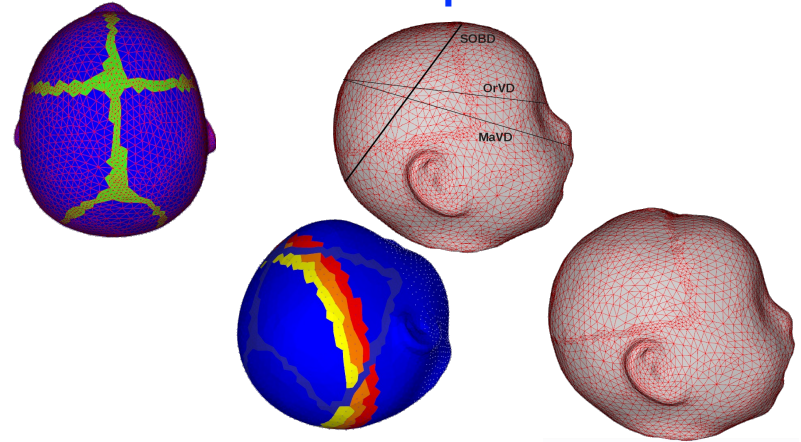


Apprentissage des gestes d'un accouchement par forceps

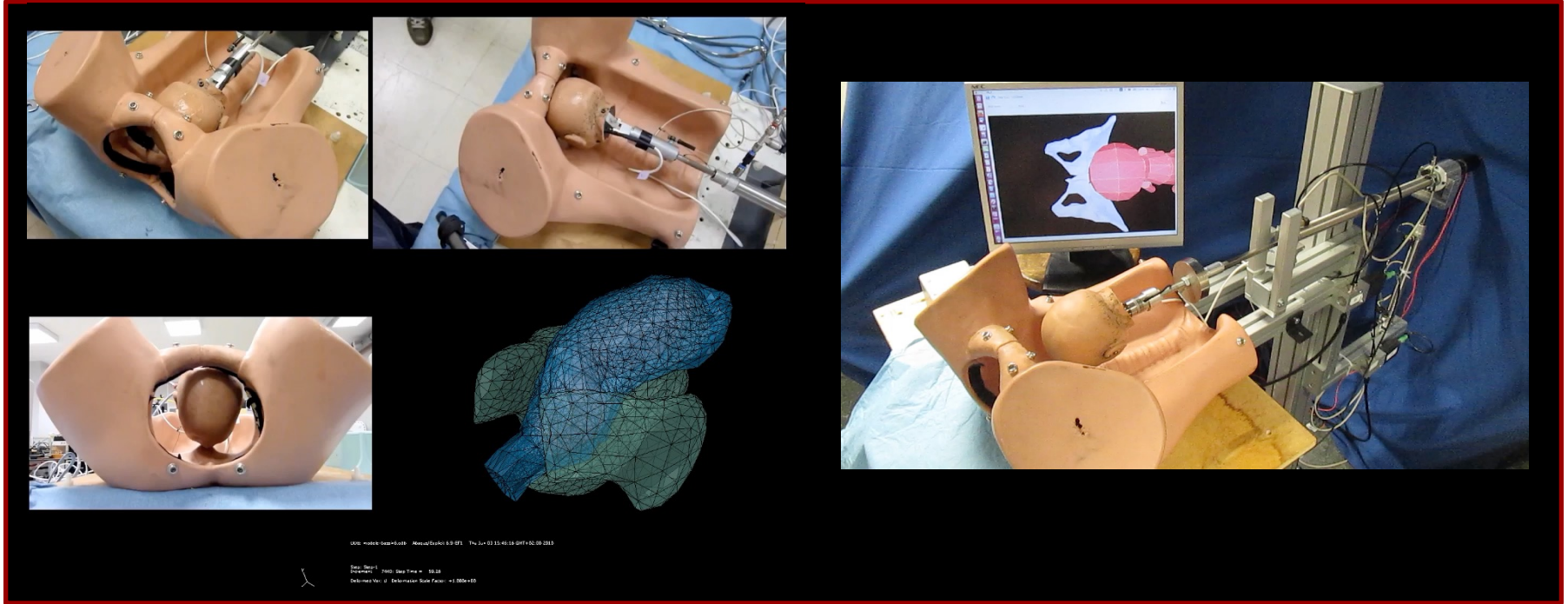
Simulation de la descente du fœtus sans trajectoire imposée



Déformation de la tête foetale due aux pressions intra-utérines et forceps



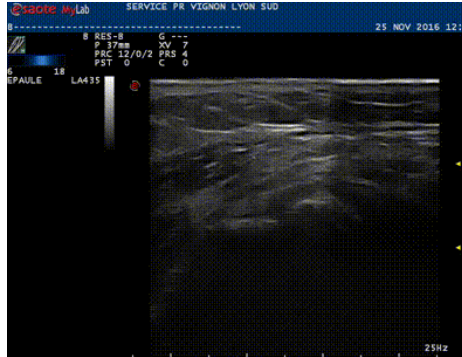
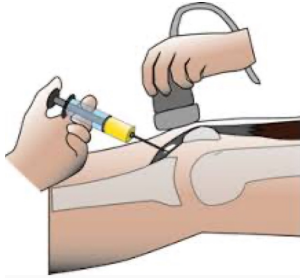
Apprentissage des gestes d'un accouchement par forceps



Couplage : simulation numérique & dispositif haptique

Apprentissage du geste de la ponction guidée par échographie

Difficulté de l'apprentissage d'un geste bi-manuelle

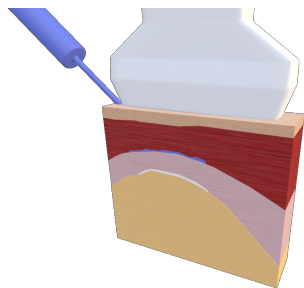


Enjeux du simulateur :

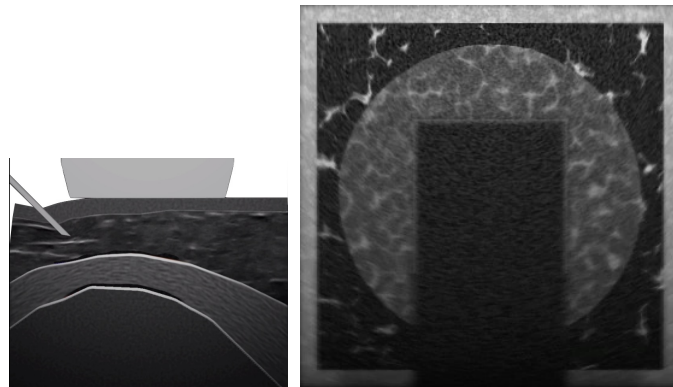
- Reproduire le comportement de l'aiguille (pénétration dans les organes)
- Reproduire le comportement de la sonde et de son rendu
- Reproduire les sensations tactiles durant le geste de ponction
- Evaluer le geste

Apprentissage du geste de la ponction guidée par échographie

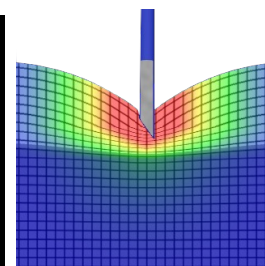
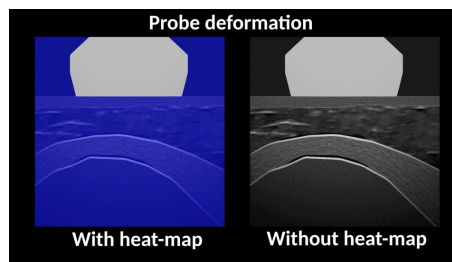
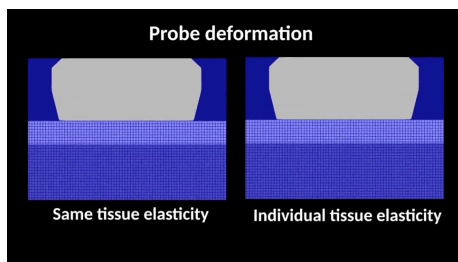
Modèles 3D



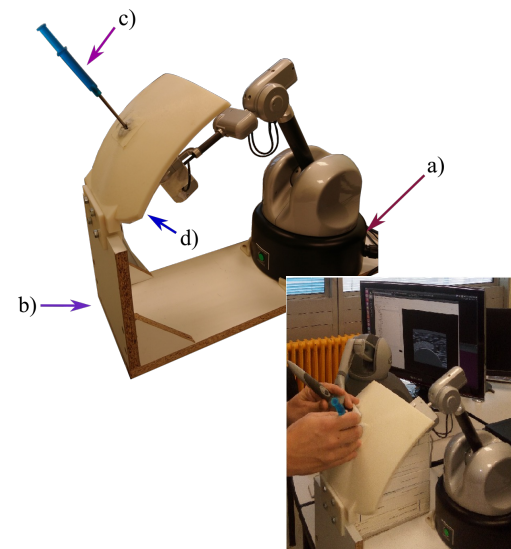
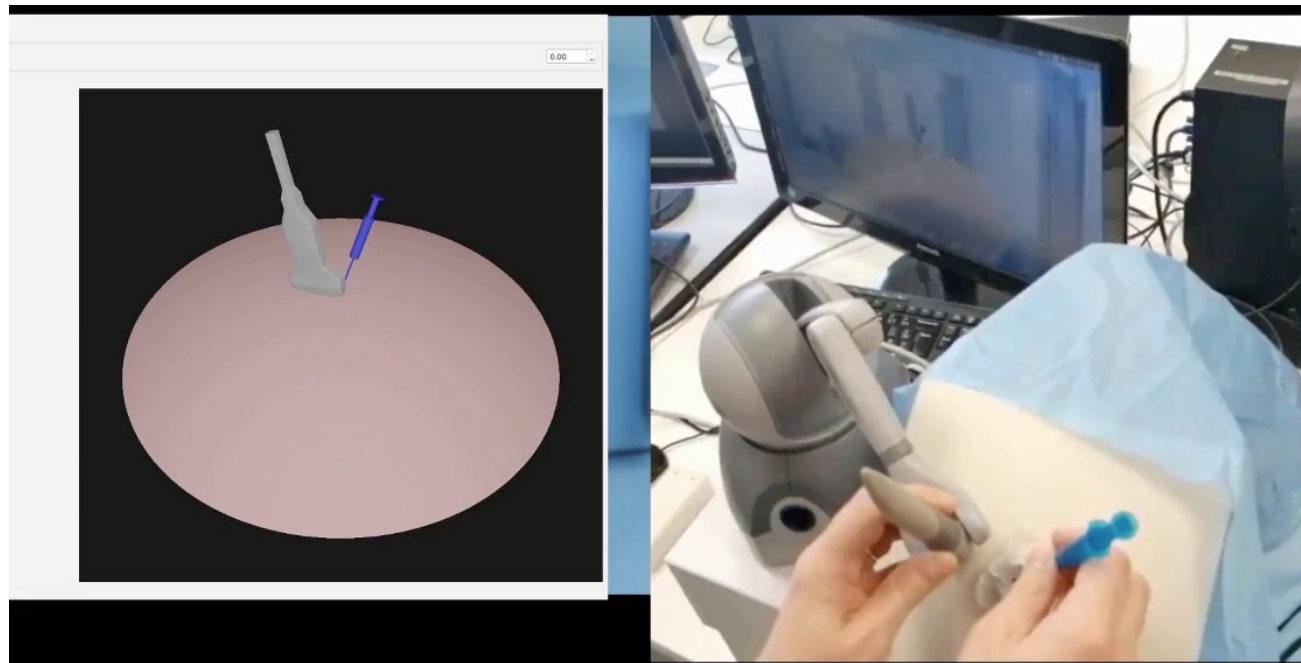
Rendu de type échographie



Déformations : aiguille et sonde



Apprentissage du geste de la ponction guidée par échographie



Couplage : simulation numérique & dispositif haptique

Un long processus de développement...

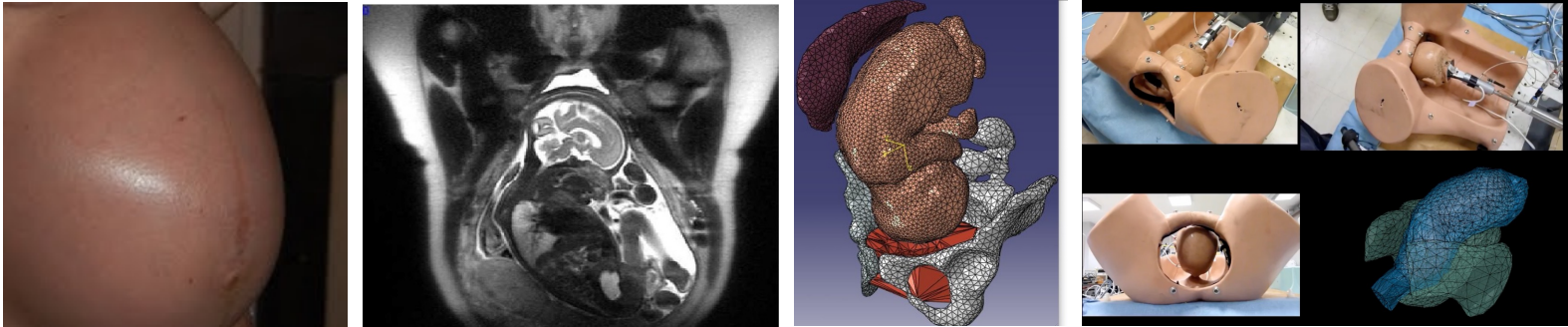
Pour obtenir un simulateur améliorant l'apprentissage des gestes

- **Première validation :**
 - Simulation & haptique
- **Seconde validation :**
 - Intégration des scénarii pertinents pour l'apprentissage
 - Le simulateur doit pouvoir reproduire ces scénarii
 - Gestion de situations usuelles & rares : morphologie, pathologie, geste à réaliser
- **Troisième validation :**
 - Campagne de tests avec les praticiens
 - Evaluation de l'apport du simulateur pour l'apprentissage

Allons plus loin : vers des simulateurs patient spécifique

Encore plus de verrous scientifiques et techniques à lever pour la simulation :

- Récupérer les **données** directement **auprès du patient** (IRM, paramètres biomécaniques)
- Nécessiter d'**automatiser** tout le pipeline permettant de **générer les modèles 3D**
- Nécessiter d'avoir des **simulations plus précises**
- Avec une **validation robuste!**



En bilan

- **Collaboration multidisciplinaire indispensable** pour réaliser de tels simulateurs
- **Travail avec le corps médical** tout au long de la conception

- **Difficultés et limites :**
 - Simulateur doit permettre de reproduire et apprendre le vrai geste
 - Validation complexe (simulation & haptique)

- **Intérêts escomptés :**
 - Accélération de l'apprentissage
 - Amélioration des connaissances au niveau physiologique
 - Mise en place de nouvelles méthodes d'évaluation du geste
 - Mise en place de nouvelles techniques / nouveaux gestes

Conclusion

Informatique Graphique, RV et applications médicales

Informatique Graphique

- Processus complet de création d'images virtuelles animées
 - Modélisation, animation, rendu

Réalité Virtuelle

- Outils pour ajouter du réalisme à la scène virtuelle

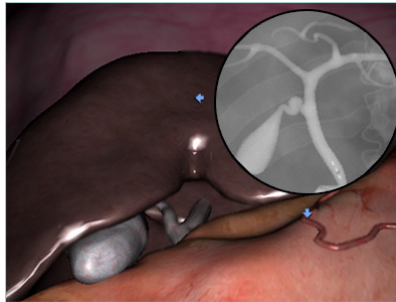
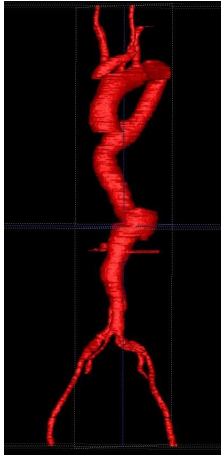
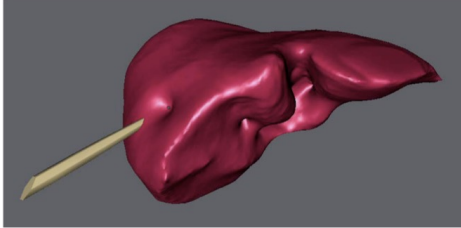
Utilisation de ces technologies pour le monde médicale

- Apprentissage de gestes médicaux-chirurgicaux
- Compréhension des comportements
- Entraînement avant opérations délicates
- Tester de nouvelles méthodes / gestes ?

Informatique Graphique, Réalité Virtuelle et Applications médicales

Florence Zara

Université Claude Bernard Lyon 1

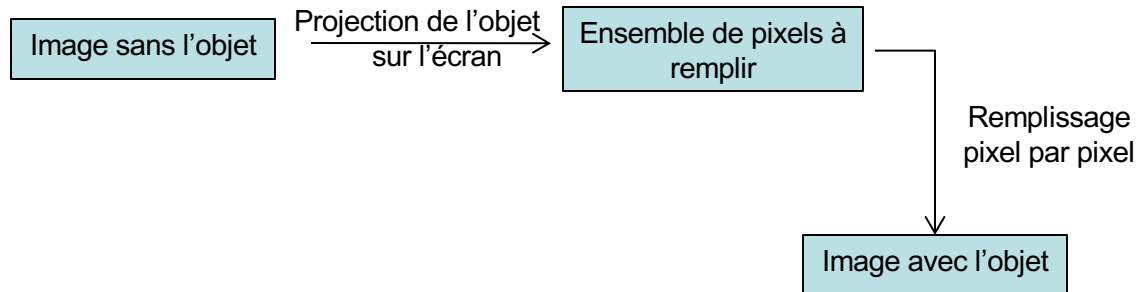


Concepts de base de la librairie OpenGL

3 – Rendu de la scène

Le rendu s'effectue en deux étapes :

1. Placement des objets dans l'écran
2. Remplissage de l'objet pixel par pixel
 - Algorithme de profondeur employé pour savoir si c'est l'objet considéré qui doit remplir le pixel ou un autre



Transformations géométriques

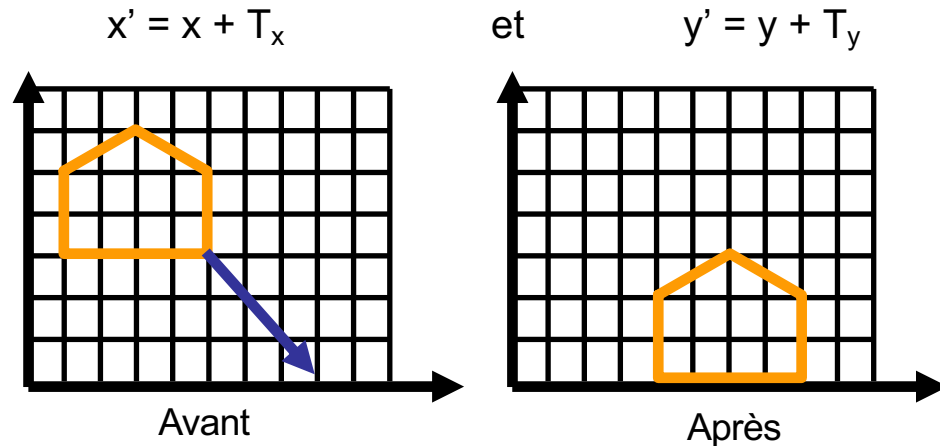
Transformations géométriques en 2D : translation, rotation, changement d'échelles

Introduction des coordonnées homogènes

Transformations géométriques en 2D et 3D en utilisant les coordonnées homogènes

Transformations géométriques

Translation d'un point 2D



Notation : $P' = P + T$ (somme vectorielle)

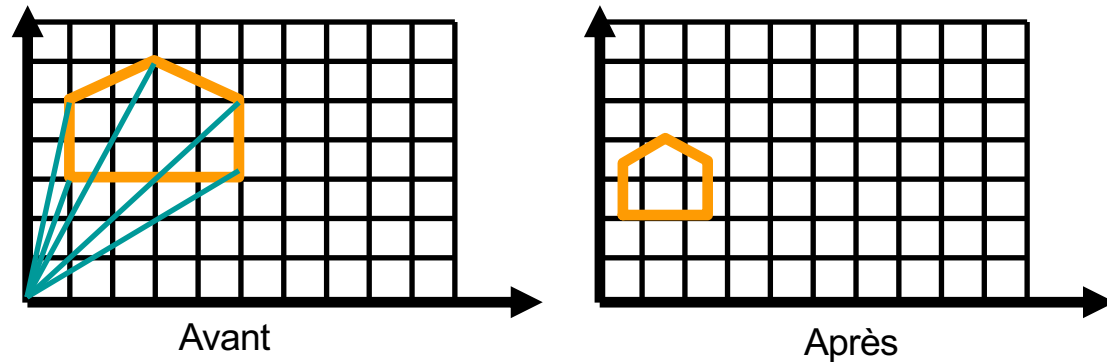
avec $P = \begin{pmatrix} x \\ y \end{pmatrix}$ et $T = \begin{pmatrix} T_x \\ T_y \end{pmatrix}$

Transformations géométriques

Changement d'échelle d'un point 2D

$$x' = S_x x \quad \text{et}$$

$$y' = S_y y$$



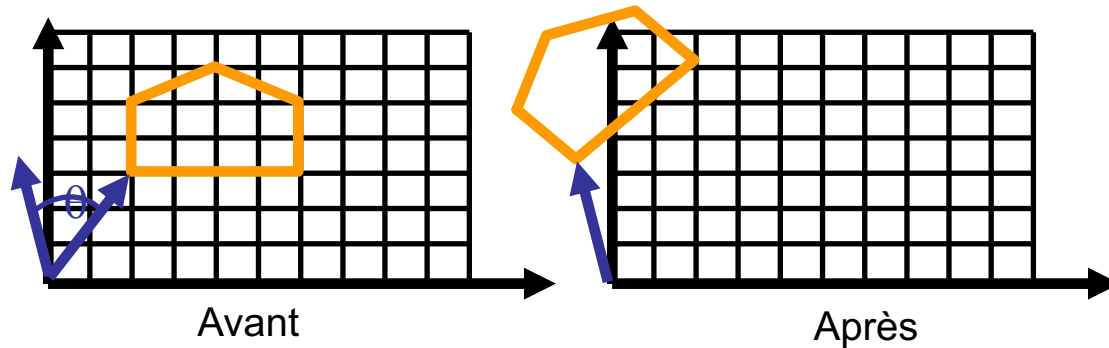
Notation : $P' = S P$ (multiplication matricielle)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Transformations géométriques

Rotation d'un point 2D

$$x' = \cos\theta x - \sin\theta y \quad \text{et} \quad y' = \sin\theta x + \cos\theta y$$



Notation : $P' = R P$ (multiplication matricielle)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Transformations géométriques

La notation pour ces transformations est simple et concise

Mais elle n'est pas unifiée :

- addition ou multiplication
- comment faire pour concaténer plusieurs transformations ?

On souhaite obtenir une notation unique

- qui permette de noter facilement les combinaisons de transformations

 Utilisation de **coordonnées dites homogènes**

Transformations géométriques

Coordonnées homogènes employées en Image, Vision et Robotique

On ajoute une troisième coordonnée notée w

Un point 2D devient un vecteur à 3 coordonnées : $\begin{pmatrix} x \\ y \\ w \end{pmatrix}$

Deux points sont alors égaux ssi :

$$x' / w' = x / w \quad \text{et} \quad y' / w' = y / w$$

Si $w=0$, on a un point à l'infini (utile pour les projections)

Transformations géométriques

Notation de la translation en coordonnées homogènes en 2D :

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix} \Rightarrow \begin{cases} x' = x + wT_x \\ y' = y + wT_y \\ w' = w \end{cases}$$

$$\Rightarrow \begin{cases} \frac{x'}{w'} = \frac{x}{w} + T_x \\ \frac{y'}{w'} = \frac{y}{w} + T_y \end{cases}$$

Transformations géométriques

Notation du changement d'échelle en coordonnées homogènes en 2D :

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix} \Rightarrow \begin{cases} x' = xS_x \\ y' = yS_y \\ w' = w \end{cases}$$

$$\Rightarrow \begin{cases} \frac{x'}{w'} = \frac{x}{w} S_x \\ \frac{y'}{w'} = \frac{y}{w} S_y \end{cases}$$

Transformations géométriques

Notation de la rotation en coordonnées homogènes en 2D :

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix} \quad \Rightarrow \quad \begin{cases} x' = \cos \theta x - \sin \theta y \\ y' = \sin \theta x + \cos \theta y \\ w' = w \end{cases}$$

$$\Rightarrow \begin{cases} \frac{x'}{w'} = \cos \theta \frac{x}{w} - \sin \theta \frac{y}{w} \\ \frac{y'}{w'} = \sin \theta \frac{x}{w} + \cos \theta \frac{y}{w} \end{cases}$$

Transformations géométriques


Notation de composition des transformations :

Il suffit de multiplier les matrices entre-elles :
composition d'une rotation et d'une translation :

$$M = R T$$

Exemple d'une rotation autour d'un point Q

- Translater Q à l'origine : T_Q
- Rotation autour de l'origine : R_Θ
- Translation en retour vers Q = $-T_Q$

 $P' = (-T_Q) R_\Theta T_Q P$

Transformations géométriques

Notation en coordonnées homogènes en 3D :

Introduction d'une **quatrième coordonnée notée w**

Un point 3D devient un vecteur à 4 coordonnées :

$$\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$$

Deux points sont alors égaux ssi :

$$x' / w' = x / w \quad , \quad y' / w' = y / w \quad \text{et} \quad z' / w' = z / w$$

Toutes les matrices de transformations sont de taille 4 X 4 : $M_{4 \times 4}$

Transformation inverse obtenue en utilisant matrice $M^{-1}_{4 \times 4}$

Transformations géométriques

Passage en coordonnées homogènes :

Coordonnées cartésiennes

Coordonnées homogènes

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$



$$\begin{pmatrix} x'/w \\ y'/w \\ z'/w \\ w \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$$

w est le facteur
d'échelle

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$



$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

cas avec w=1

Transformations géométriques

Notation de la translation en coordonnées homogènes en 3D :

$$\text{Matrice de la translation } T(T_x, T_y, T_z) : \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow \begin{cases} x' = x + wT_x \\ y' = y + wT_y \\ z' = z + wT_z \\ w' = w \end{cases}$$

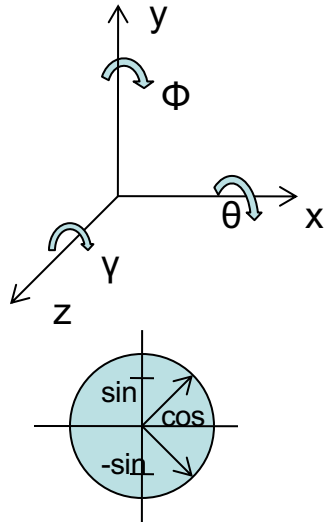
Notation du changement d'échelle en 3D :

$$\text{Matrice du changement d'échelle } S(S_x, S_y, S_z) : \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow \begin{cases} x' = xS_x \\ y' = yS_y \\ z' = zS_z \\ w' = w \end{cases}$$

Transformations géométriques

Rotation en coordonnées homogènes en 3D :

Rotation dépend d'un axe et d'un angle



Rotation autour de l'axe X :
(coordonnée en x non modifiée)

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation autour de l'axe Y :
(coordonnée en y non modifiée)

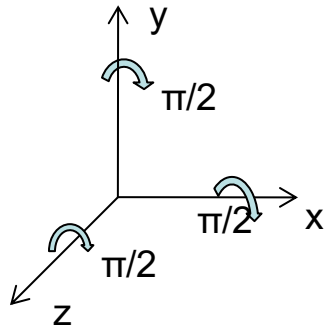
$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation autour de l'axe Z :
(coordonnée en z non modifiée)

$$R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Transformations géométriques

Rotation en coordonnées homogènes en 3D de l'angle $\pi/2$:



Rotation autour de l'axe X :

(coordonnée en x non modifiée
coordonnée en y changée en z
coordonnée en z changée en -y)

$$R_x\left(\frac{\pi}{2}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation autour de l'axe Y :

(coordonnée en y non modifiée
coordonnée en z changée en x
coordonnée en x changée en -z)

$$R_y\left(\frac{\pi}{2}\right) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation autour de l'axe Z :

(coordonnée en z non modifiée
coordonnée en x changée en y
coordonnée en y changée en -x)

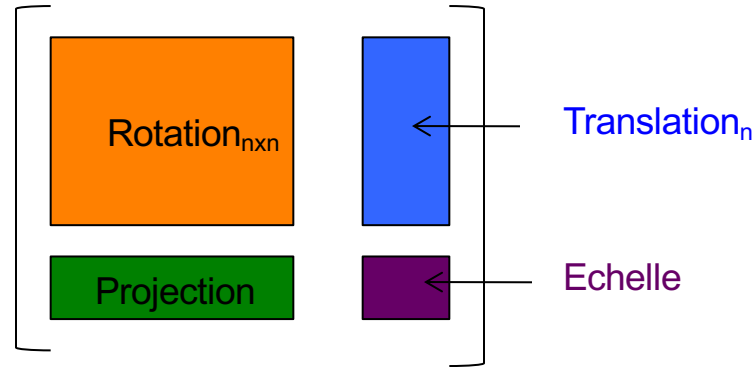
$$R_z\left(\frac{\pi}{2}\right) = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Transformations géométriques

Matrice de transformation générale en nD :

Utilisation des coordonnées homogènes

Matrice de taille $(n+1) \times (n+1)$



Attention :

La multiplication de matrices n'est pas commutative

L'ordre des transformations est donc important

Rotation puis translation \neq translation puis rotation

Projection de la scène 3D pour obtenir l'image 2D

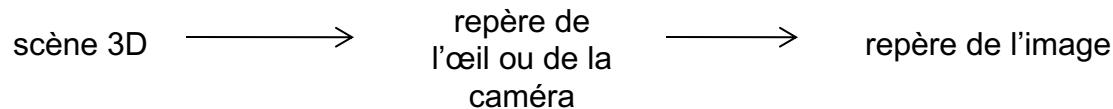
Projection de la scène 3D :

La scène est représentée par rapport à un observateur virtuel

- Point de vue : position de l'observateur
- Direction de visée : direction vers laquelle est tournée l'observateur
- Direction en haut : verticale pour l'observateur

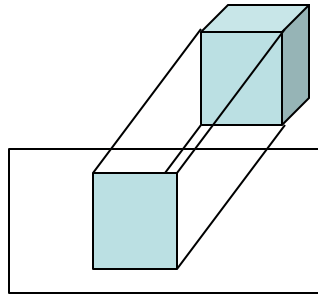
La projection de la scène 3D sur l'image se fait en deux étapes :

1. On doit ramener tous les points de la scène dans le repère de l'œil (translation et changement d'échelle)
2. Puis, projection du repère de l'œil vers le repère de l'image

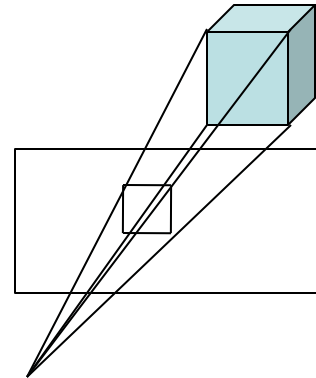


Projection de la scène 3D pour obtenir l'image 2D

Différents types de projection possibles :



Projection parallèle



Projection perspective

Projection de la scène 3D pour obtenir l'image 2D

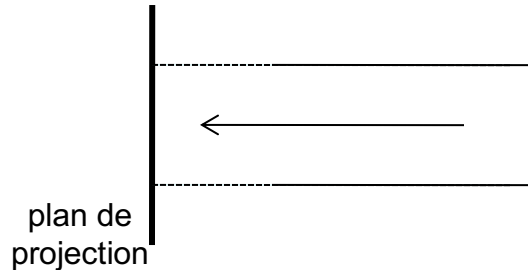
Projection parallèle

Projection parallèle sur le plan $z=0$

Direction de projection est $(0, 0, 1)$

⇒ $x' = x, y' = y, z'=0$ et $w'=w$

$$P = \begin{pmatrix} x \\ y \\ z \\ w=1 \end{pmatrix} \Rightarrow P' = \begin{pmatrix} x'=x \\ y'=y \\ z'=0 \\ w'=1 \end{pmatrix} \Rightarrow \text{matrice de projection : } \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Projection de la scène 3D pour obtenir l'image 2D

Problème de la projection parallèle

Pas de rétrécissement des objets dans le lointain

➡ pas possible de rendre ce type d'images

➡ **manque de réalisme**



Projection de la scène 3D pour obtenir l'image 2D

Projection perspective

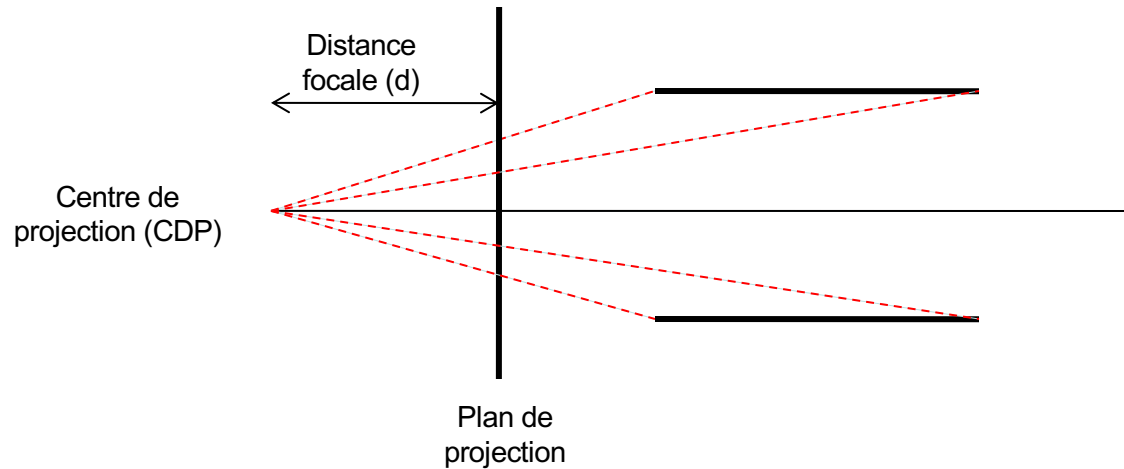
Intérêt : les objets lointains sont plus petits

Deux droites parallèles se rejoignent en un point appelé *point de fuite*

Plusieurs types de projection : un, deux ou trois points de fuite

Projection de la scène 3D pour obtenir l'image 2D

Projection perspective – Notion de distance focale

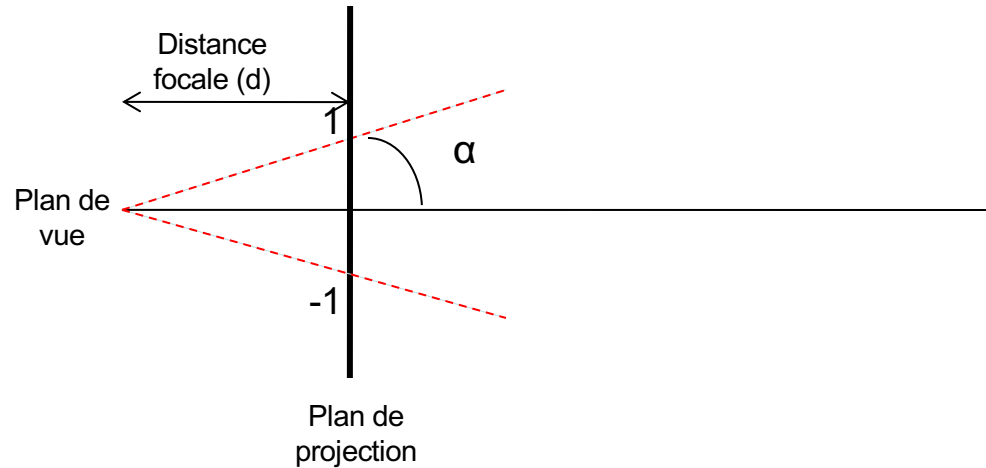


Projection de la scène 3D pour obtenir l'image 2D

Projection perspective - Notion d'ouverture de vue

Angle qui exprime la largeur du champ visuel

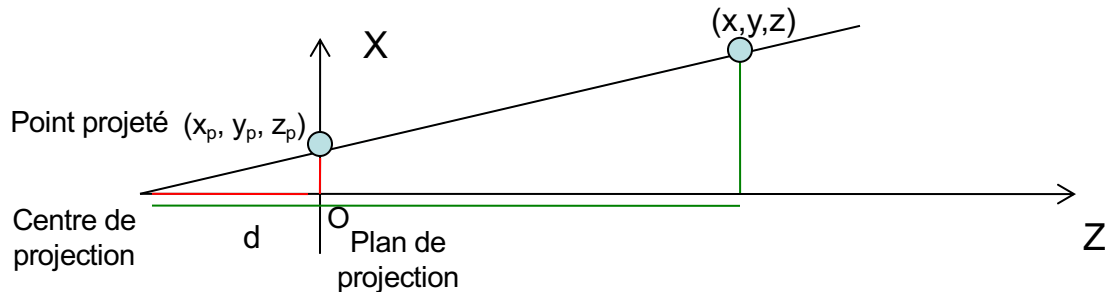
Relation avec la distance focale : $\tan \alpha = \frac{1}{d}$



Projection de la scène 3D pour obtenir l'image 2D

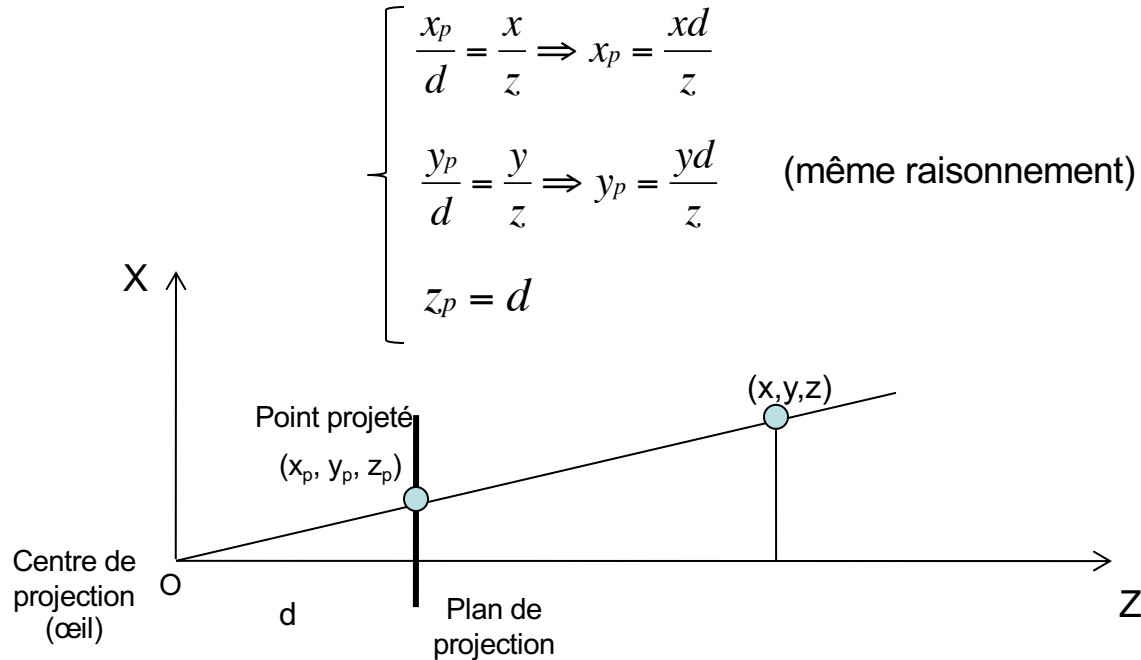
Equations de perspectives :

$$\left\{ \begin{array}{l} \frac{x_p}{d} = \frac{x}{d+z} \Rightarrow x_p = \frac{xd}{d+z} \\ \frac{y_p}{d} = \frac{y}{d+z} \Rightarrow y_p = \frac{yd}{d+z} \quad (\text{m\^eme raisonnement}) \\ z_p = 0 \end{array} \right.$$



Projection de la scène 3D pour obtenir l'image 2D

Equations de perspectives simplifiées en plaçant l'origine / l'œil au point de projection :



Projection de la scène 3D pour obtenir l'image 2D

Equations de perspectives en coordonnées homogènes (cas simplifié) :

$$\left\{ \begin{array}{l} x_p = \frac{xd}{z} = \frac{x}{w} \\ y_p = \frac{yd}{z} = \frac{y}{w} \\ z_p = d = \frac{z}{w} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} wx_p = x \\ wy_p = y \\ wz_p = z \\ w = \frac{z}{d} \end{array} \right. \Rightarrow \begin{pmatrix} wx_p \\ wy_p \\ wz_p \\ w \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

→ Matrice de passage $M_{I \leftarrow C}$:

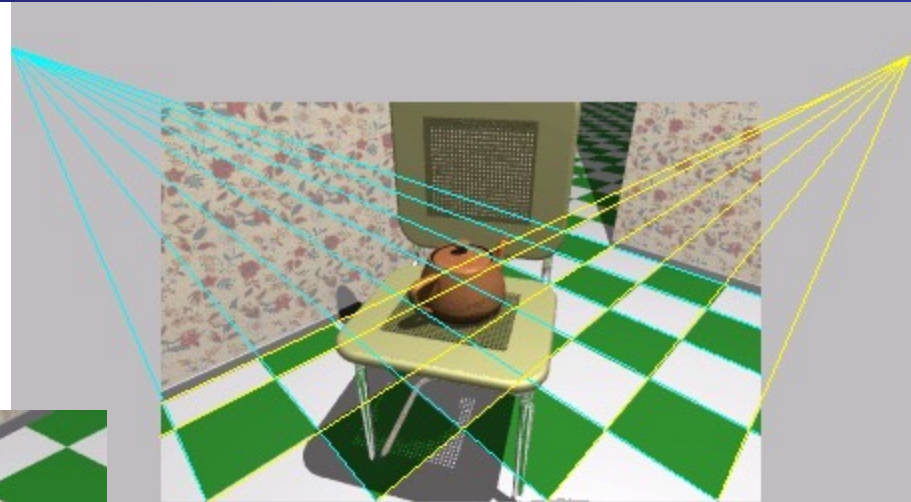
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix}$$

Cette matrice permet de passer du repère de l'œil / de la caméra au repère de l'image

Projection de la scène 3D pour obtenir l'image 2D

Comparaison des
deux types de
projection

Projection parallèle



Projection perspective

Projection de la scène 3D pour obtenir l'image 2D

Récapitulatif pour la projection perspective

Projection est définie par une position de l'œil (ou de la caméra)
notée $E = (E_x, E_y, E_z)$

Coordonnées définies dans le repère absolu (repère du monde virtuel) dans lequel sont définis les objets 3D de la scène

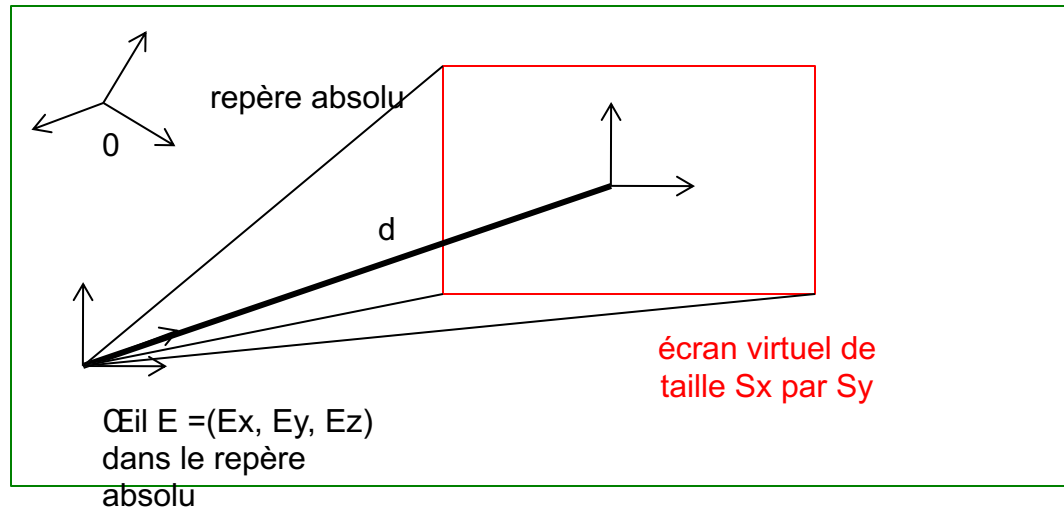
Repère local placé sur cette position définissant la direction de l'observateur

Définition de l'écran virtuel sur lequel on projettera les points 3D de taille S_x et S_y

Cet écran se trouve à une distance d de l'œil / de la caméra

Projection de la scène 3D pour obtenir l'image 2D

Récapitulatif pour la projection perspective



monde virtuel

Projection de la scène 3D pour obtenir l'image 2D

Récapitulatif pour la projection perspective

Pour pouvoir appliquer la projection perspective

1. Il faut tout d'abord ramener le repère vers l'œil (translation et changement de repère)
2. Puis il faut appliquer la perspective
3. Et enfin, il faut remettre à l'échelle de l'écran (S_x , S_y)



Définition des matrices de transformation de ces différentes étapes
(attention à l'ordre de ces transformations)

Projection de la scène 3D pour obtenir l'image 2D

Matrices de transformation pour la projection perspective

$$S = \begin{pmatrix} Sx/2 & 0 & 0 & 0 \\ 0 & Sy/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Mise à l'échelle
de l'écran

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix}$$

Perspective

$$M = \begin{pmatrix} (M_{oeil})^{-1} & & & 0 \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Changement de
base

$$D = \begin{pmatrix} 1 & 0 & 0 & -Ex \\ 0 & 1 & 0 & -Ey \\ 0 & 0 & 1 & -Ez \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Changement de
repère : translation
vers l'œil qui est
l'origine

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = SPMD \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$$

Point dans l'image

Point dans le monde virtuel