

M2-Images

Intro-Synthèse d'images

J.C. Iehl

September 1, 2026

Introduction : synthèse d'images

qu'est ce que c'est ?

produire une image à partir d'une description des objets représentés,

analyse d'images :

retrouver les modèles des objets représentés dans une image.

Introduction : synthèse d'images

plusieurs formes de synthèses :

- ▶ 2D, 3D,
- ▶ réaliste,
- ▶ stylisée,
- ▶ "simplifiée".

2D



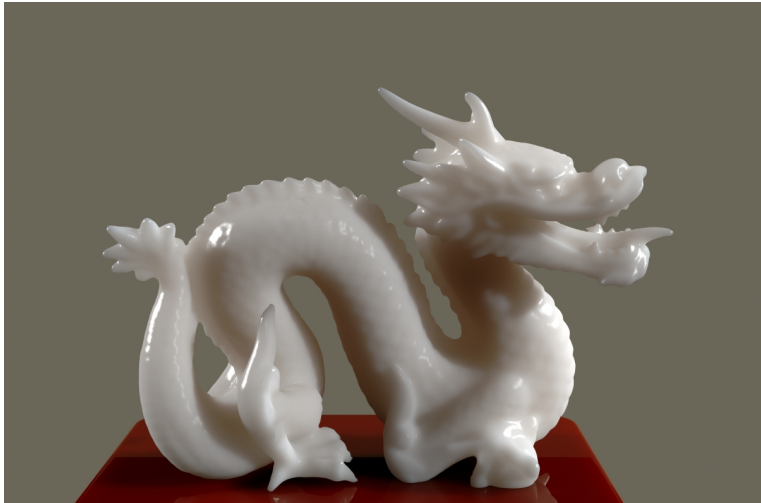
3D



3D



3D



3D



3D



3D



MadWorld, 2009

3D



Brave, Pixar 2012

3D



Monster U, Pixar 2013

3D réaliste



Gravity, 2013

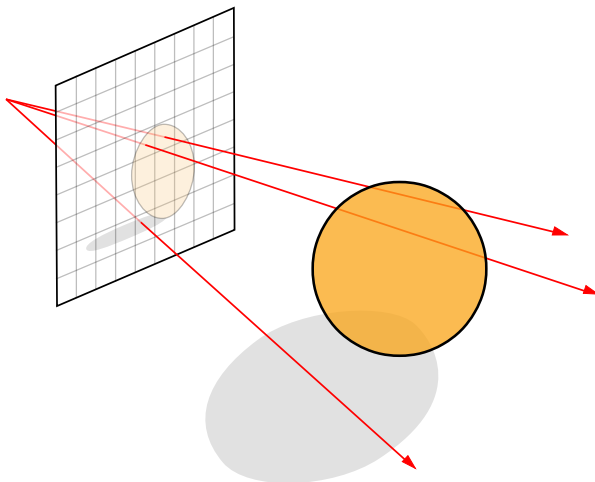
Introduction : produire une *image* ?

qu'est ce qu'une image ?

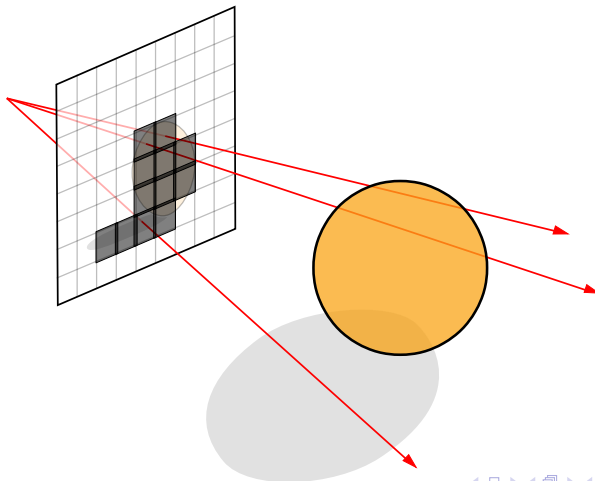
- ▶ un ensemble de pixels colorés,
- ▶ ou un ensemble de vecteurs (image vectorielle) ?

déterminer pour chaque pixel de l'image la couleur de l'objet représenté (de l'objet *visible* à travers le pixel ...).

Introduction : produire une *image* ?



Introduction : produire une *image* ?



Introduction : produire une *image* ?

comment ?

- ▶ problème 1 : déterminer où se trouve l'objet (par rapport à la camera),
- ▶ problème 2 : déterminer l'ensemble de pixels (correspondant à la forme de l'objet),
- ▶ problème 3 : donner une couleur à chaque pixel.

Introduction : produire une *image* ?

2 organisations :

- ▶ pour chaque objet : déterminer l'ensemble de pixels, (que se passe-t-il lorsque plusieurs objets se "dessinent" sur le même pixel ?)
- ▶ pour chaque pixel : trouver l'objet visible,

trouver l'objet visible pour chaque pixel : trouver l'objet le plus *proche* de l'observateur.

Introduction : produire une *image* ?

remarques :

- ▶ que se passe-t-il lorsque plusieurs objets sont *visibles* dans un seul pixel ?
- ▶ que se passe-t-il lorsque l'objet visible est transparent ?

Introduction : description des objets

modèle de l'objet :

- ▶ de sa forme,
- ▶ de sa matière / de son aspect,
- ▶ de sa position,
- ▶ de "comment" il est éclairé ?

modèle d'énergie :

et des sources de lumières.

Introduction : couleur

rouge, vert, bleu ?

- ▶ perception ?
- ▶ physique ?

la couleur est une *perception* de l'énergie.

même une photo ne correspond pas à notre perception d'une scène.

Introduction : contenu du cours

ce cours présente :

- ▶ en première partie : la synthèse 3D temps réel, basée sur les fonctionnalités des cartes graphiques,
- ▶ en deuxième partie : la synthèse 3D réaliste, basée sur la physique et l'optique.

Introduction : découpage fonctionnel

modèles →

- ▶ 3D
- ▶ objet
- ▶ matière
- ▶ lumière
- ▶ observateur

calculs →

- ▶ passage 3D vers 2D,
- ▶ déterminer le point de l'objet visible à travers chaque pixel,
- ▶ déterminer l'énergie réfléchie par le point.

image 2D couleur

- ▶ déterminer la couleur du pixel.

la couleur de chaque pixel représente l'aspect du point de l'objet visible à travers le pixel.

Introduction : un peu de physique

lumière →

réflexion →

ou "interaction" avec la
matière d'un objet

énergie réfléchie →

couleur des objets
visibles par l'observateur

calculs dans le sens de propagation de la lumière.

comment limiter les calculs aux objets visibles par l'observateur ?

Introduction : visibilité

observateur →

- ▶ position et orientation de l'observateur
- ▶ position et forme des objets

déterminer les objets visibles

matières des objets visibles →

calculer l'énergie réfléchie vers l'observateur

sources de lumières →

visibles par les objets

lumière →

énergie associée à la source de lumière

ne faire le calcul que lorsque tous les éléments sont connus.

Modèles

choisir une représentation pour chaque élément manipulé pendant le calcul :

- ▶ modèle d'objets,
- ▶ modèle de source de lumières,
- ▶ représentation de l'énergie et des couleurs,
- ▶ modèle d'observateur.

comment placer et orienter les objets, l'observateur, etc. ?

Modèles d'objets

nombreuses représentations (et format de fichiers ...) :

- ▶ selon les besoins / métiers,
- ▶ pas de modèle complet,

le modèle dépend, en partie, de la méthode d'affichage (calcul de visibilité).

Modèles d'objets

représentation de la surface des objets :

- ▶ maillages (ensemble de triangles / quads),
- ▶ surfaces (carreau de Bézier, NURBS, surface de subdivision),
- ▶ scan 3D (position + orientation + couleur).

Modèles d'objets

représentation du volume des objets :

- ▶ primitives algébriques (plan, sphère, cone, tore, etc.),
- ▶ fonctions implicites,
- ▶ énumération spatiale (grille, octree),

opérations ensemblistes sur les volumes : union, différence, intersection

Modèles de matières

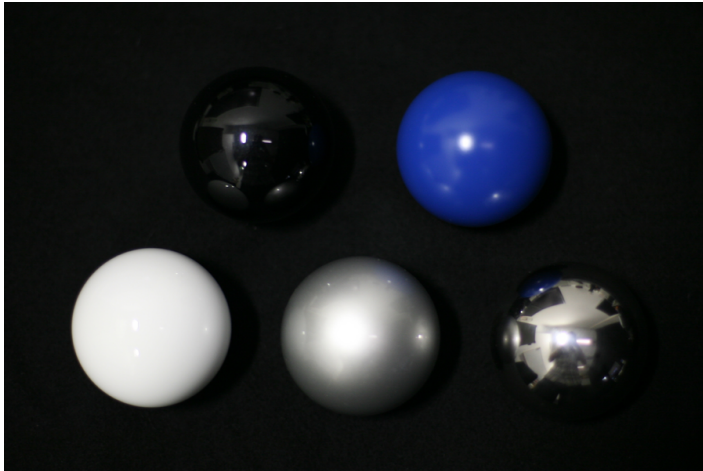
représenter l'aspect d'un objet indépendamment de sa forme :

- ▶ mat / diffus / lambertien : plâtre,
- ▶ spéculaire : miroir,
- ▶ réfléchissant : métal plus ou moins poli, bois verni, etc.
- ▶ transparent : eau, verre, etc.

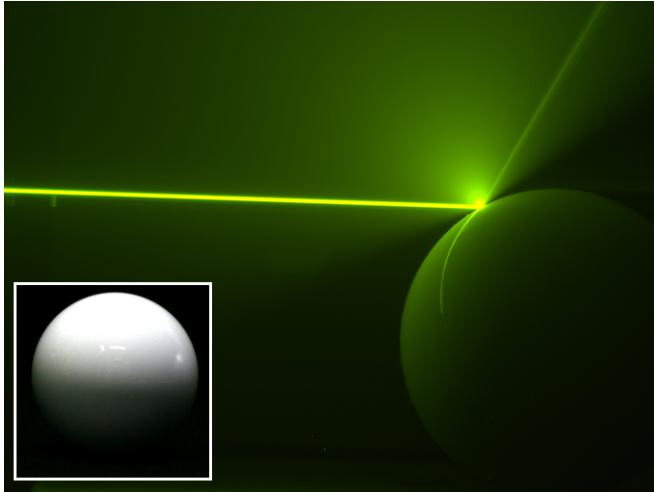
ce sont les cas simples, "idéaux", la plupart des matériaux ne correspondent pas à une seule catégorie,
notation : BRDF, fonction de réflectance bidirectionnelle.

Interactions lumière / matière

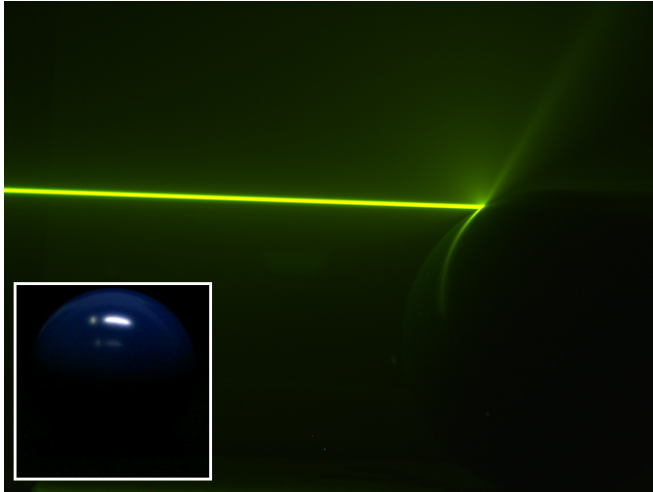
visualisation de la propagation de la lumière



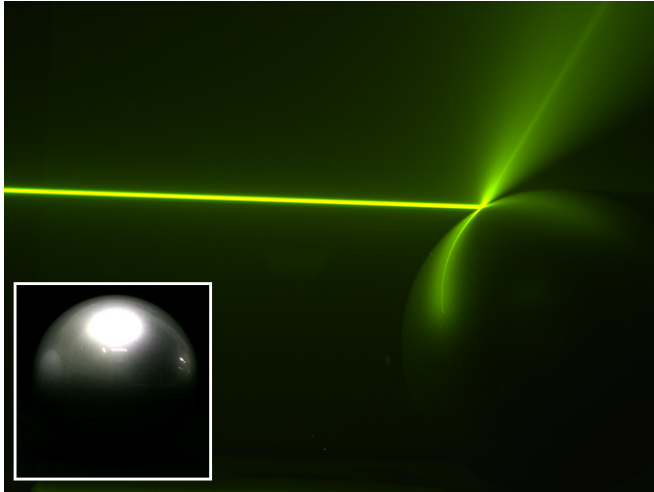
Interactions lumière / matière



Interactions lumière / matière



Interactions lumière / matière



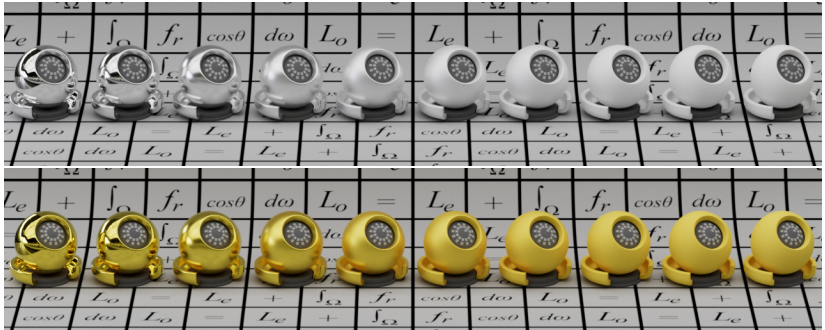
Interactions lumière / matière



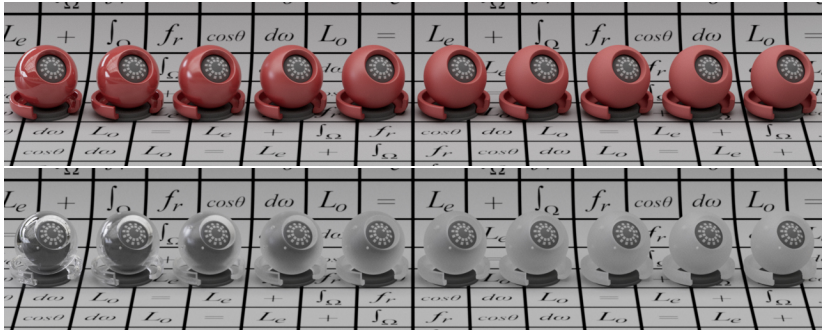
Interactions lumière / matière



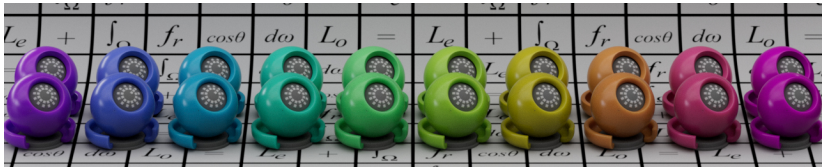
Interactions lumière / matière



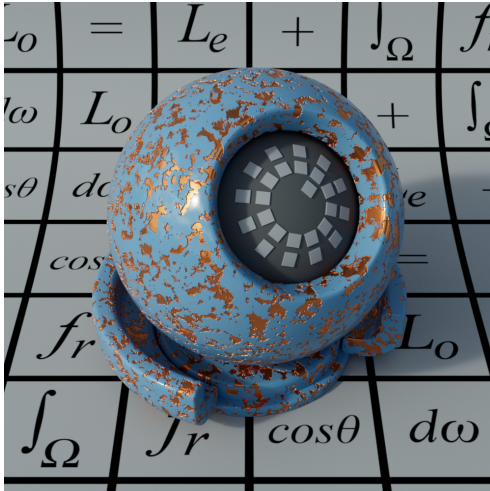
Interactions lumière / matière



Interactions lumière / matière



Interactions lumière / matière



Modèles de sources (de lumière)

un objet dont la surface émet de l'énergie / de la lumière.

forme géométrique + fonction d'émission :

- ▶ uniforme, directionnelle,
- ▶ ou mesurée.

le plus rapide : un point avec une émission uniforme,
le plus courant : un quad avec une émission uniforme (sur une seule face).

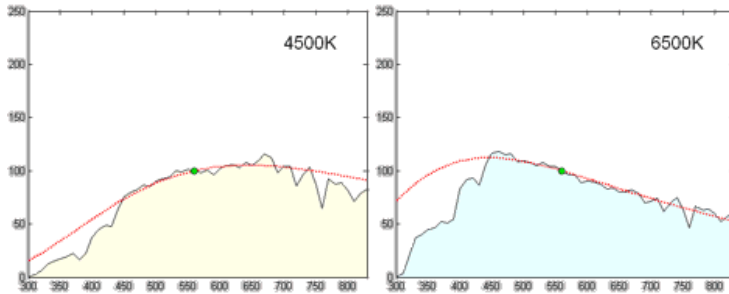
Représentation de l'énergie / couleur

l'oeil est sensible à un ensemble de longueur d'ondes : le domaine visible.

plusieurs types de cellules sur la rétine :

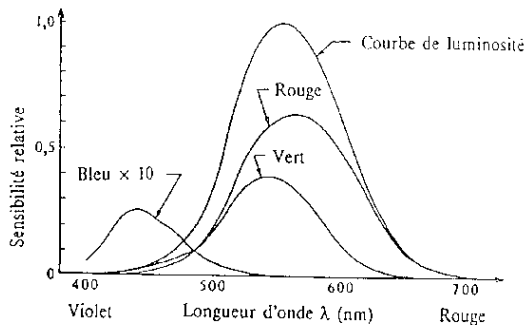
- ▶ longueurs d'ondes courtes : perception du bleu,
- ▶ moyennes : vert,
- ▶ longues : rouge.

Représentation de l'énergie / couleur



émission d'une source de lumière sur le domaine visible.

Représentation de l'énergie / couleur



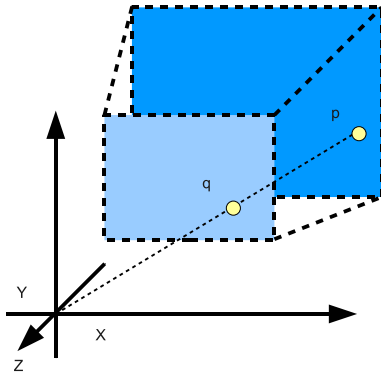
le plus rapide : énergie / couleur représentée par un triplet R, V, B.

Modèles d'observateur

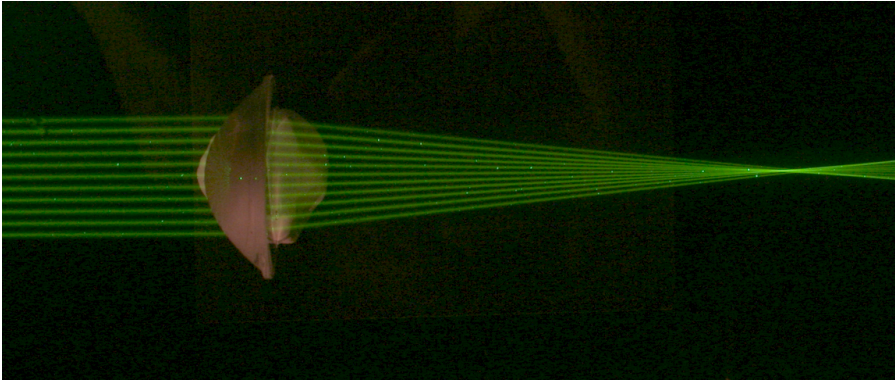
projection idéale : un point,
mais un objectif optique réel est plus complexe (plan focal,
ouverture, ...)

Modèle d'observateur : pyramide de vision

observateur placé et orienté dans la scène :



Modèle d'observateur : optique



Modèle d'observateur : optique



Description de scène

(et composition de transformations)

placer et orienter les objets dans la scène :

- ▶ translation,
- ▶ rotation,
- ▶ changement d'échelle ?

placer et orienter l'observateur dans la scène :

- ▶ translation,
- ▶ rotation,
- ▶ projection ?

ou sont les objets par rapport à l'observateur ?

Représentation des positions et des orientations

(et des "projections")

une représentation pour exprimer :

- ▶ une translation,
- ▶ une rotation,
- ▶ une "projection",
- ▶ un changement de repère.

et tout combiner / composer ensemble ?

Transformations affines et espace homogène

toutes les transformations citées se représentent sous forme d'une matrice ... sauf la translation.

idée

comment représenter une translation avec une matrice ?

Transformations affines et espace homogène

matrices 4×4 :

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

Points homogènes

point homogène :

$$p_h = \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix}$$

point réel associé :

$$p = p_h / w = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Vecteurs homogènes

$$v = \begin{bmatrix} x \\ y \\ z \\ w \equiv 0 \end{bmatrix}$$

un vecteur ne subit pas de "translation".

Transformation affine et projection

projection orthographique sur le plan $z = 0$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

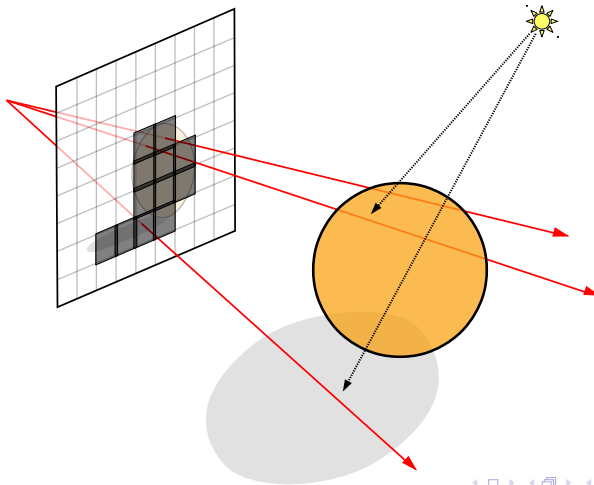
mais : on ne peut pas inverser cette matrice.

Transformation affine et projection

utiliser une transformation affine qui n'est pas une projection mais qui fournit un résultat équivalent ...

les détails dans le cm sur les transformations.

Pipeline graphique



Pipeline graphique

exercice:

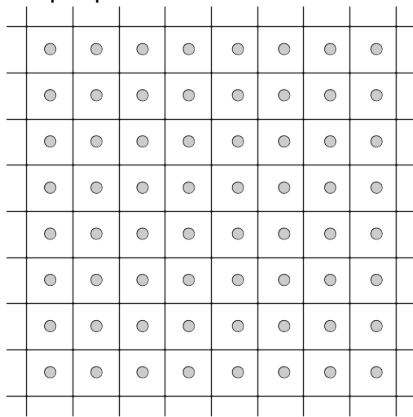
- ▶ proposez au moins 3 solutions pour dessiner un triangle ou un quad :
- ▶ les coordonnées des sommets sont connues dans le repère de la scène,
- ▶ les pixels dans lesquels se projettent les sommets sont connus.
- ▶ (les transformations sont connues...)

exercice:

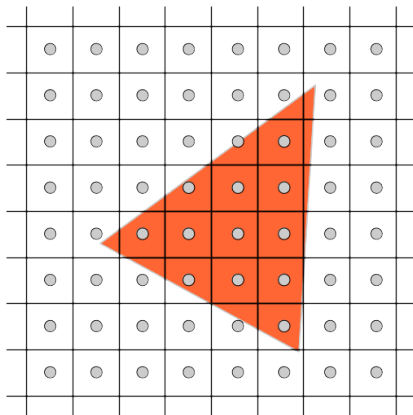
- ▶ même question avec 2 objets ou plus.

Quelques indices pour démarrer...

une image est un ensemble de pixels, on choisit un point à "l'intérieur" de chaque pixel :



Quelques indices pour démarrer...



Quelques indices pour démarrer...

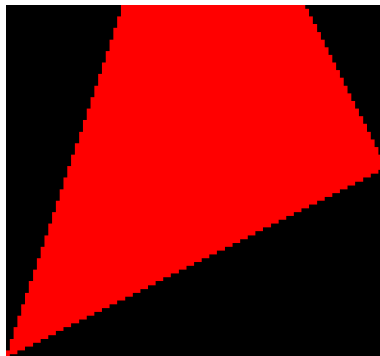
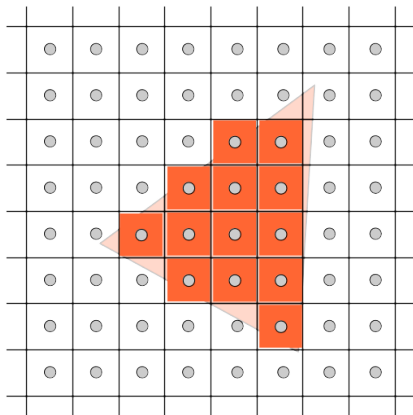
on sait :

- ▶ recalculer les coordonnées des sommets du triangle dans un repère quelconque,
- ▶ changer la couleur d'un pixel de l'image,
- ▶ découper un triangle, en plusieurs triangles plus petits...

et alors ?

- ▶ ??

Quelques indices pour démarrer...



Pipeline graphique

plusieurs traitements nécessaires :

- ▶ visibilité, trouver l'objet visible pour un pixel,
- ▶ trouver les sources de lumières visibles par le point,
- ▶ calculer l'énergie réfléchie,
- ▶ déterminer la couleur associée à l'énergie réfléchie,
- ▶ "insérer" la couleur dans le pixel de l'image.

Pipeline graphique

3 organisations des traitements : pipeline, chaîne de traitements.

Pipelines graphiques :

- ▶ lancer de rayons,
- ▶ REYES (Renderman),
- ▶ rasterisation (OpenGL / DirectX).

Pipeline graphique : lancer de rayons

visibilité :

- ▶ calculer le rayon associé à un pixel,
- ▶ calculer les intersections du rayon avec tous les objets de la scène,
- ▶ conserver la plus proche de l'observateur.

+ étapes suivantes du pipeline.

toute la scène doit être en mémoire.

Pipeline graphique : rasterisation

visibilité :

- ▶ pour chaque objet :
- ▶ pour chaque primitive de la surface de l'objet :
- ▶ déterminer les pixels sur lesquels se projette la primitive,
- ▶ + étapes suivantes du pipeline.

ne conserver que la couleur de la primitive la plus proche de l'observateur ?

Pipeline graphique : rasterisation

Z-buffer :

image de profondeur pour conserver le point de l'objet le plus proche de l'observateur.

les objets sont dessinés un par un, dans un ordre quelconque, mais l'image et le Z-buffer conservent la couleur et la profondeur du point (de l'objet) le plus proche (vu à travers le pixel).

Pipeline graphique : REYES

visibilité :

- ▶ pour chaque objet :
- ▶ déterminer les portions (*a priori*) visibles de l'objet,
- ▶ découper ces portions en éléments plus petits qu'un pixel,
- ▶ + étapes suivantes du pipeline sur chaque élément.

+ Z-Buffer

un seul objet traité à la fois (cf. rasterisation).

Pipeline graphique : et alors ?

lancer de rayons :

- ▶ toute la scène est en mémoire,
- ▶ facile de déterminer si 2 points quelconques sont visibles,
- ▶ calculs effets globaux (très) simples (ombres, etc).

mais : stocker toute la scène.

REYES / rasterisation :

- ▶ traitement d'un seul objet à la fois,
- ▶ visibilité associée à l'observateur, pas d'effets globaux simples.

mais : traiter un seul objet, permet de calculer des scènes très complexes, cf. Pixar / La-Haut / Brave

Introduction
Découpage fonctionnel
Modèles
Description de scène
Algorithme général

pipeline: lancer de rayons
pipeline: rasterization
pipeline: REYES
et alors ?
reste du pipeline

exemple



Up, Pixar 2009

exemple



exemple



Brave, Pixar 2012

exemple



Toy Story 4, Pixar 2019

Introduction

Découpage fonctionnel

Modèles

Description de scène

Algorithme général

pipeline: lancer de rayons

pipeline: rasterization

pipeline: REYES

et alors ?

reste du pipeline

exemple



Lightyear, Pixar 2022

quelques années plus tard...

en 2022, comment sont fabriqués les films d'animations ? cf
documentaire disney / wish

plus de REYES, tout est fait en lancer de rayons... malgré la taille
des scènes... par exemple, **l'île et les nuages de moana**

en re-organisant les calculs, par exemple : "**Sorted Deferred
Shading for Production Path Tracing**" 2013

quelques années plus tard...



Battle Angel Alita, 2019 **VFX**

+ reste du pipeline ?

finir de construire l'image ?

- ▶ transparence ?
- ▶ plusieurs objets dans le même pixel ?

cf. cours traitement du signal : filtrage et reconstruction d'une fonction échantillonnée.

+ reste du pipeline ?

intuition :

- ▶ dessiner un point à l'intérieur d'un pixel ?
- ▶ dessiner 2 points (à l'intérieur d'un pixel) ?
- ▶ dessiner un segment (à l'intérieur d'un pixel) ?
- ▶ dessiner une partie d'un triangle (à l'intérieur d'un pixel) ?
- ▶ dessiner 2 triangles adjacents (1 arête commune) ?
- ▶ dessiner 2 triangles quelconques ?
- ▶ dessiner 2 triangles transparents quelconques ?

le filtre le plus simple est une moyenne...