CPE5

Pipeline graphique 1 - fragmentation

J.C. Iehl

October 24, 2017

produire une *image* ?

directement :

- sans utiliser une librairie,
- afficher / dessiner plusieurs objets :
- trouver quel objet est visible pour chaque pixel,
- calculer la couleur du pixel pour représenter l'apparence de l'objet...

exemple

Brave, Pixar 2012

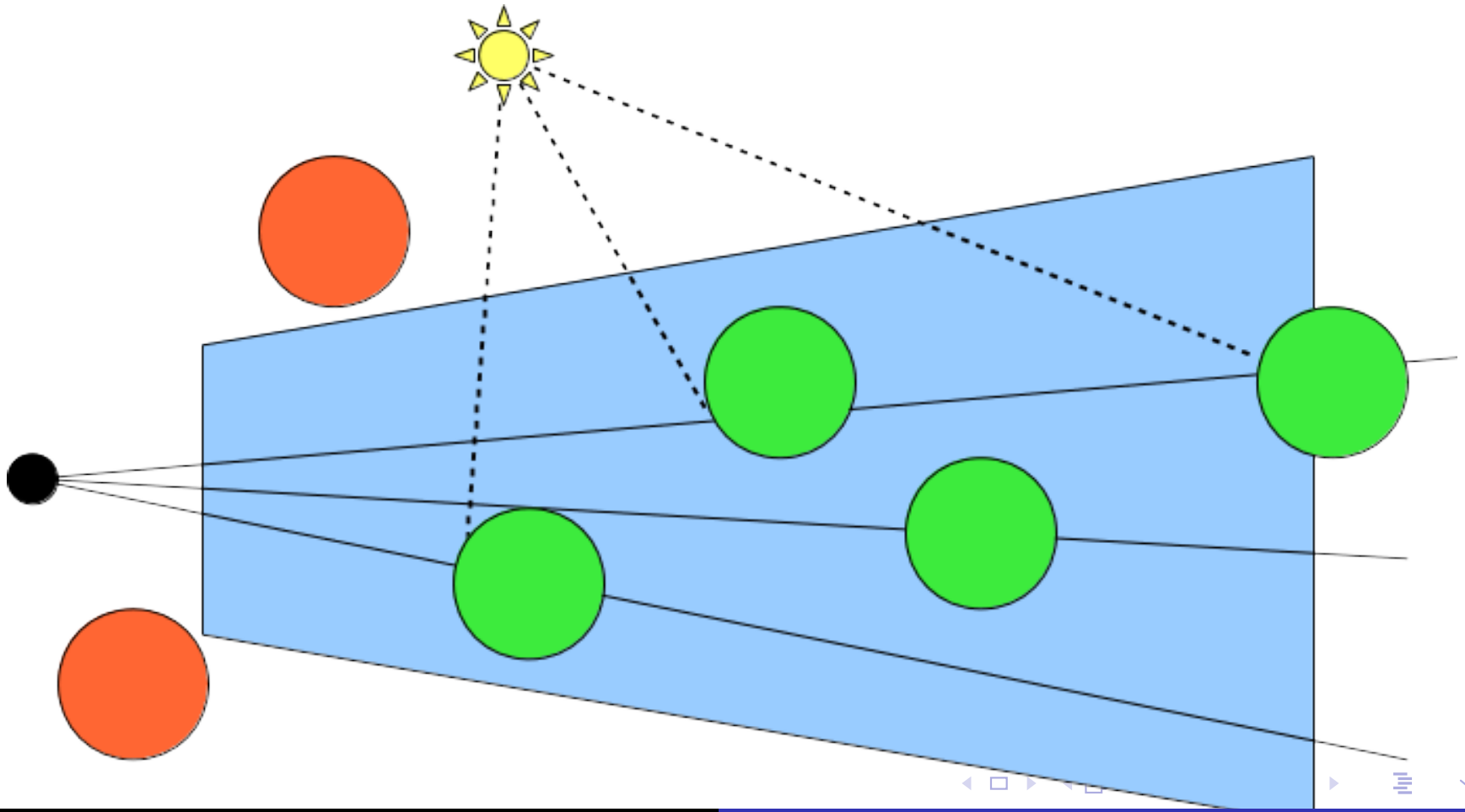
exemple

Monster U, Pixar 2013

exemple

Gravity, 2013

produire une *image* ?



produire une *image* ?

en dessinant plusieurs objets :

- 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,
- et calculer la couleur du pixel pour représenter l'apparence de l'objet...

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

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 ?
- que se passe-t-il lorsque l'objet est derrière la camera ?

décrire des objets

modèle d'un objet :

- de sa forme,
- de sa position (par rapport à la camera),
- de sa matière / de son apparence,
- de "comment" il est éclairé ?

décrire la forme d'un objet

cas simples :

- un point ?
- un segment ?
- un triangle ?
- une sphere ?

les cas simples dépendent de la méthode d'affichage...

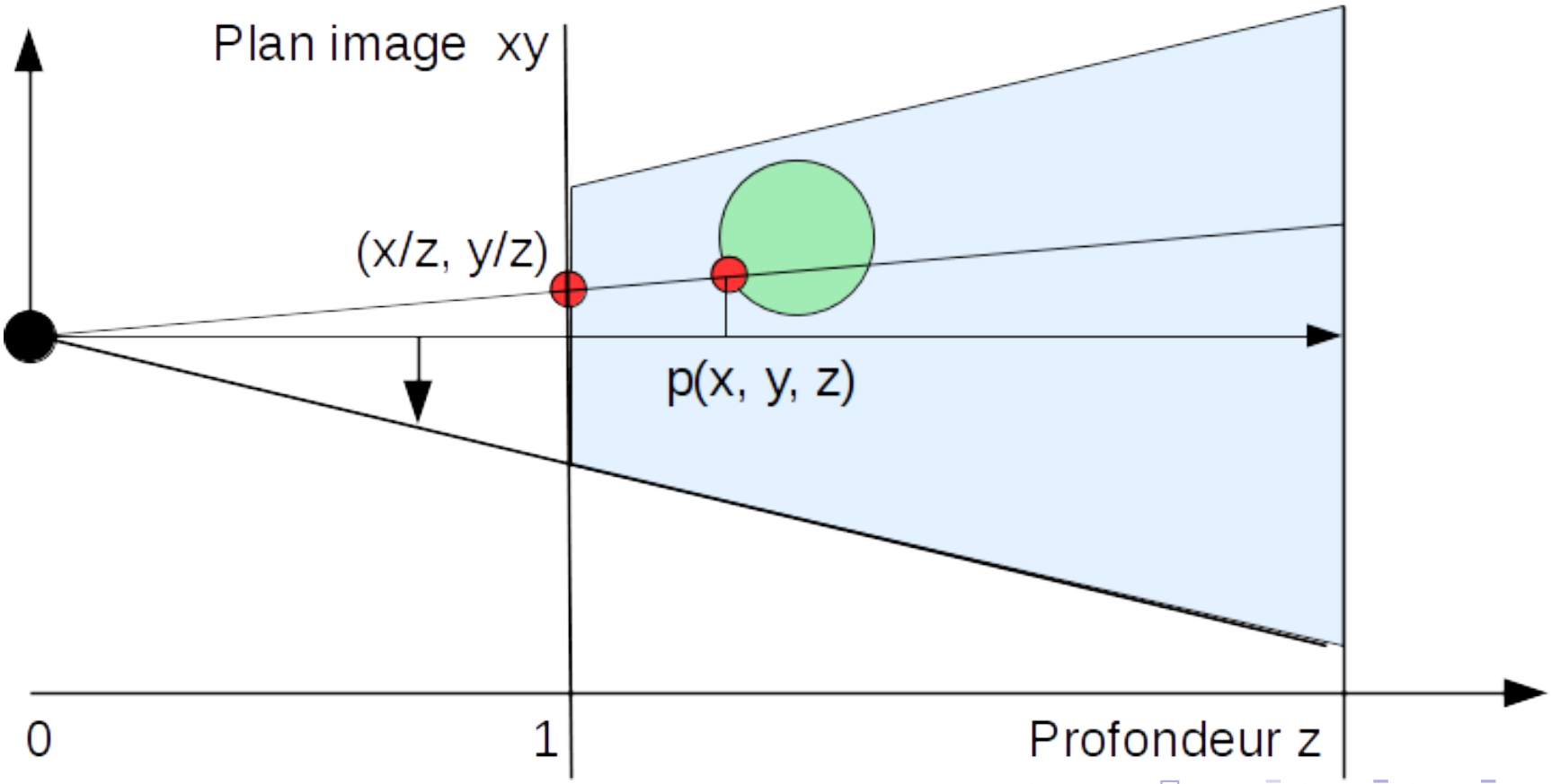
le plus simple :

un point :

- sur quel pixel de l'image se projette un point p ?
- dépend de la position et de l'orientation de la camera,
- de la projection,
- de la résolution de l'image...

dans quel repère ? comment projeter un point ?

projeter ?



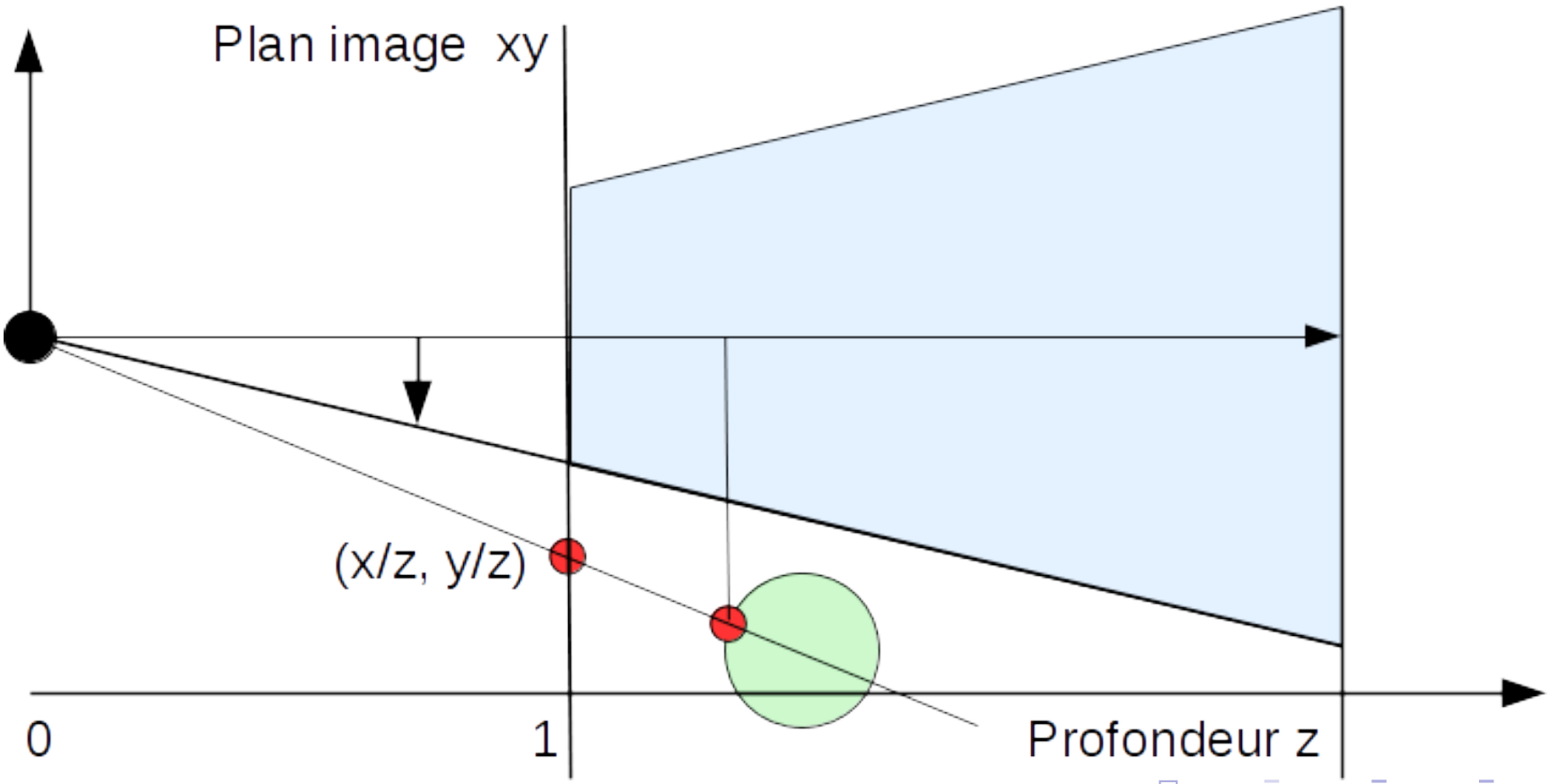
projeter ?

projeter $p(x, y, z)$:

- $(xp, yp) = (x/z, y/z)$,
- si le centre de projection est à l'origine du repère,
- sur quel pixel ?

par convention : le plan image est à $z = 1$

projeter ?



projeter ?

projection et image :

- un point se projette sur l'image si :
- $-1 < x/z < 1$,
- $-1 < y/z < 1$,
- coordonnées du pixel dans l'image $largeur \times hauteur$ pixels :
- $px = (x/z + 1) \times largeur/2$,
- $py = (y/z + 1) \times hauteur/2$.

on peut aussi définir un angle d'ouverture pour *zoomer* sur un objet... noté *fov* (field of view)

projeter ?

ensemble des points *visibles / observables* :

- un point se projette sur l'image si :
- $-1 < x/z < 1$,
- $-1 < y/z < 1$,
- les points associés à un pixel se trouvent dans le volume :
- $-z < x < z$,
- $-z < y < z$.

et pour les points derrière la camera ? ($z < 0$)

noté *frustum*.

et alors ?

bilan :

- dans le repère camera :
- projeter un point,
- sur un pixel de l'image...

changer la couleur du pixel...

les points c'est bien...

comment dessiner une forme ?

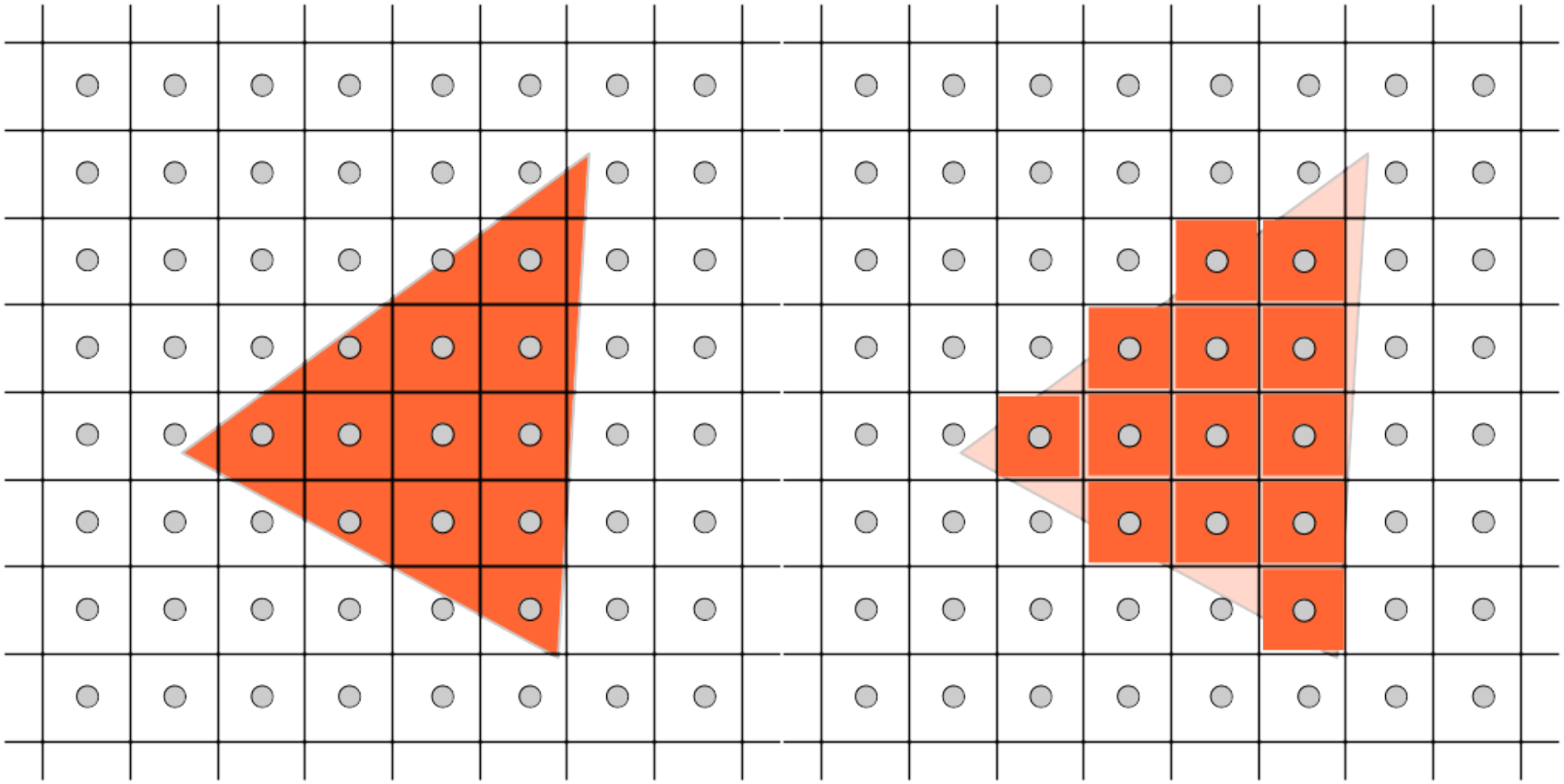
- un segment ?
- ensemble de points sur une droite 3d,
- projetter chaque point...
- un triangle ?
- ensemble de points sur un plan 3d...

dessiner un triangle ?

triangle :

- 3 sommets, a, b, c,
- dans le repere camera,
- et dans le volume visible par la camera.

dessiner un triangle



dessiner un triangle ?

algorithme :

- plusieurs idées, même principe algorithmique,
- diviser pour régner :
- si le problème est trop "complexe",
- le découper en sous problèmes,
- jusqu'à obtenir un (sous) problème dont la solution est connue...
- on sait : projetter un point sur un pixel.

diviser pour régner 1

découper l'image :

- facile de dessiner un rectangle de pixels,
- découper l'image en rectangles, soit vides, soit remplis par une partie du triangle...
- arreter lorsqu'un rectangle est vide,
- arreter lorsqu'un rectangle (partiellement plein) se réduit à un seul pixel,
- mais : déterminer s'il existe une intersection entre un rectangle et un triangle.

"A hidden surface algorithm for computer generated halftone pictures"
J. Warnock, 1969 !

diviser pour régner 2

découper le triangle :

- facile de dessiner un pixel,
- facile de dessiner un triangle plus petit qu'un pixel,
- découper un (gros) triangle en sous triangles.

indication : découper en 4 sous triangles, en calculant le milieu de chaque arete.

diviser pour régner 2

et ça marche ?

- plutot bien :
- c'est la technique utilisée par Pixar depuis 1980...
- et elle est utilisable avec des surfaces plus complexes qu'un triangle.

"The Reyes rendering architecture"
R.L. Cook, L. Carpenter, E. Catmull, 1987

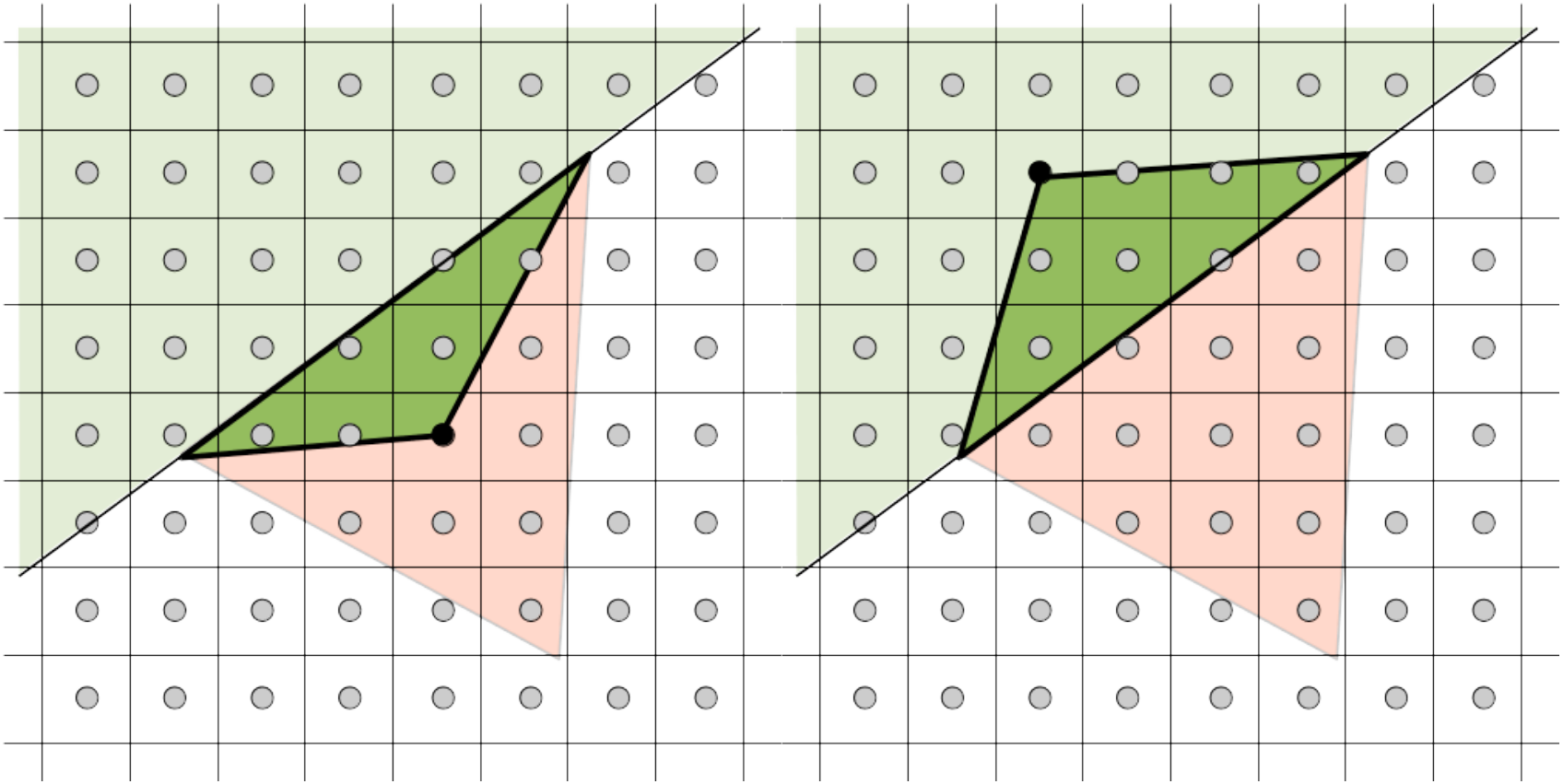
autre chose ?

plus simple :

- tester l'inclusion d'un point dans le triangle ?
- tester tous les pixels de l'image...
- comment ?
- si le point est du "bon" coté de chaque arete du triangle,
- il est à l'intérieur...

idée : orientation d'un triangle, aire algébrique (signée) d'un triangle, équation implicite de l'arete...

orientation d'un triangle / équation implicite de l'arete



orientation

si le pixel est du bon coté ?

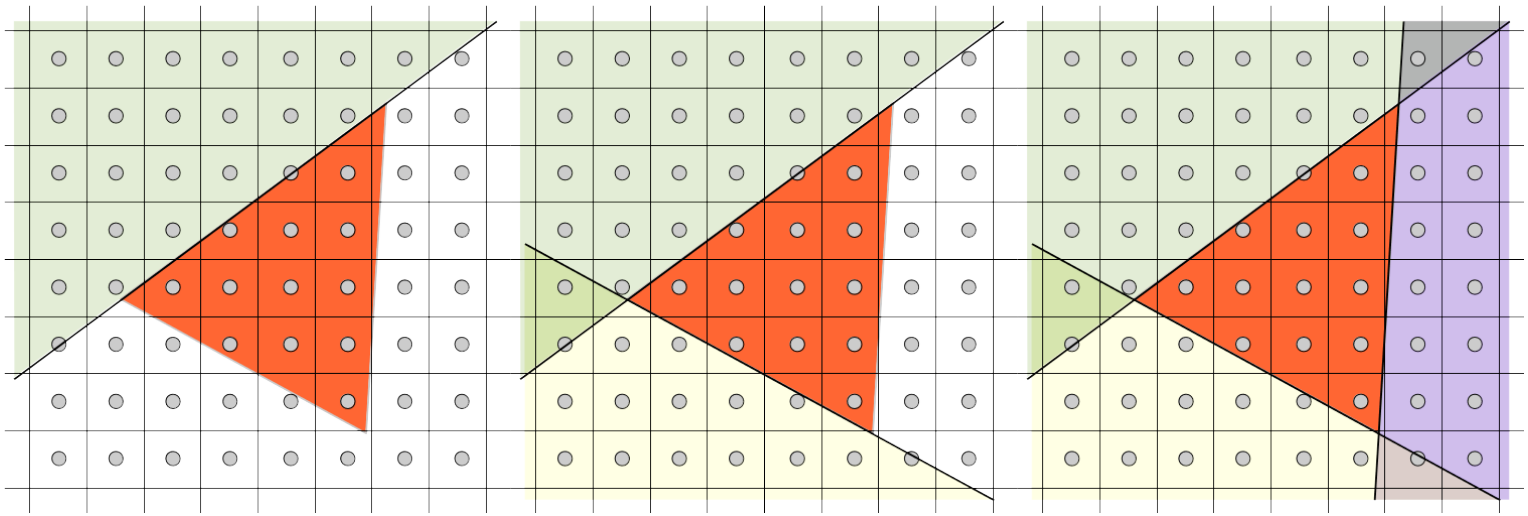
- un pixel et une arête forment un triangle,
- si ce triangle est bien orienté, le pixel est du bon coté...
- calculer l'aire algébrique (signée) du triangle, un coté est > 0 l'autre < 0,
- cf "déterminants, aires et volumes", wikipedia
- meme déduction avec l'équation implicite de la droite portant l'arete.

si le pixel est du même coté des 3 arêtes :

il est à l'intérieur du triangle.

les 3 aires ont le même signe que le triangle, en fonction de son orientation

3 aretes...



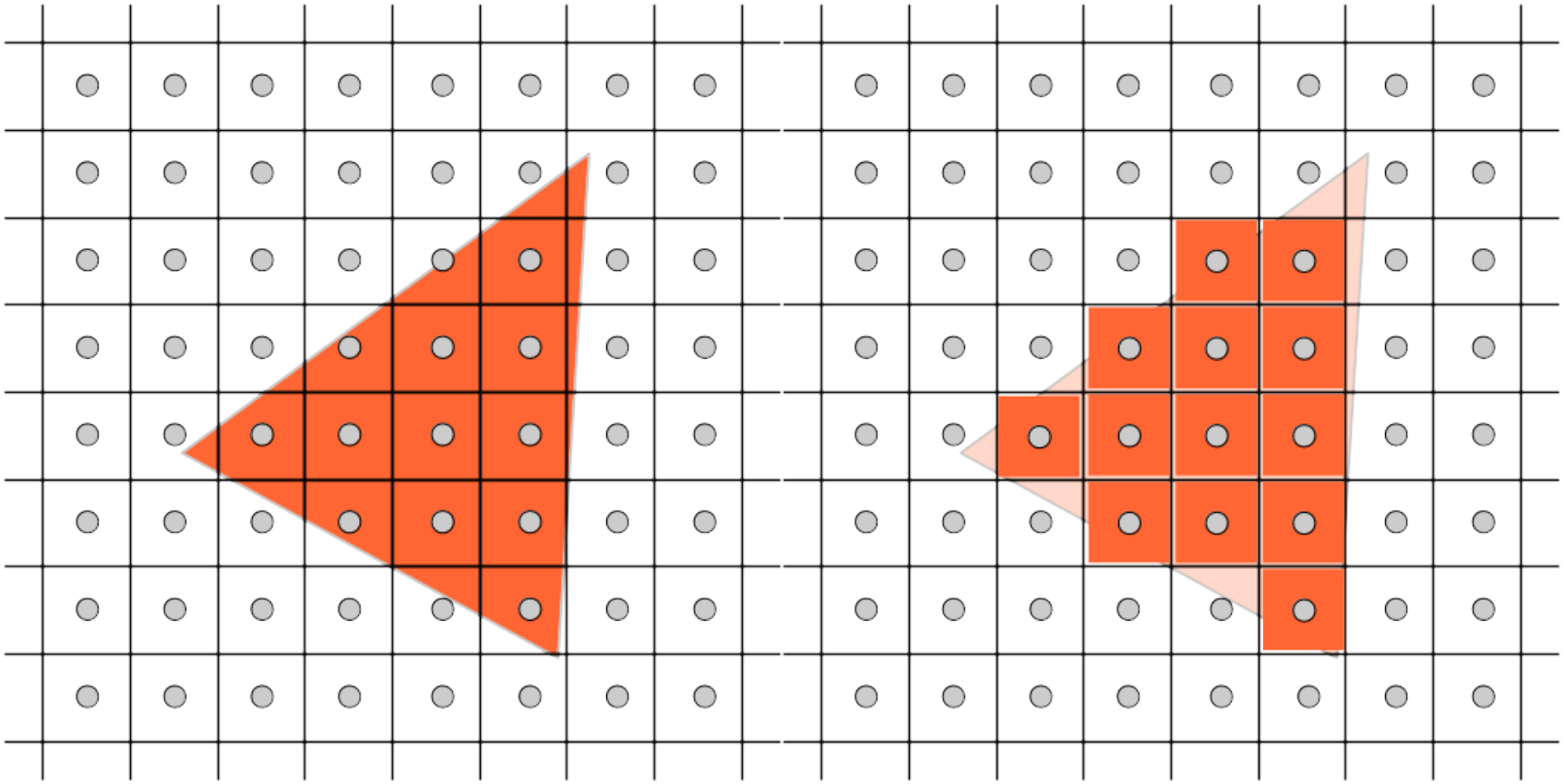
orientation

et ça marche ?

- plutot bien :
- c'est la solution utilisée par les cartes graphiques...
- c'est simple, brutal, et parallélisable...

description complète d'un système :
"Incremental and Hierarchical Hilbert Order Edge Equation Polygon Rasterization"
M. McCool, C. Wales, K. Moule, 2001

et ça marche ?



pour les curieux :

il existe d'autes solutions :

- "Forward rasterization"
  V. Popescu, P. Rosen, 2006
- "3D rasterization"
  T. Davidovic, T. Engelhardt, I. Georgiev, P. Slusallek, C. Dachsbacher, 2012

et ça marche ?

il reste...

- quelques détails à régler :
- et si plusieurs triangles se dessinent sur le meme pixel ?
- quel point conserver ?
- et en 3d ? si une partie d'un triangle est derrière la camera ?

Zbuffer

plusieurs triangles :

- se dessinent sur le meme pixel,
- lequel garder ?
  rappel : l'image représente ce que voit la camera
- garder le plus proche...
- quelle est la profondeur du triangle pour un pixel ?
  rappel : un point (x, y, z) se projette en $(x/z, y/z)$
- il suffit de stocker aussi z, la profondeur du point,
- dans une 2ième "image" : le Zbuffer,
- et de comparer le z du point à la profondeur du zbuffer.

le repère image est un *cube / 3d* !

Zbuffer

oui, mais...

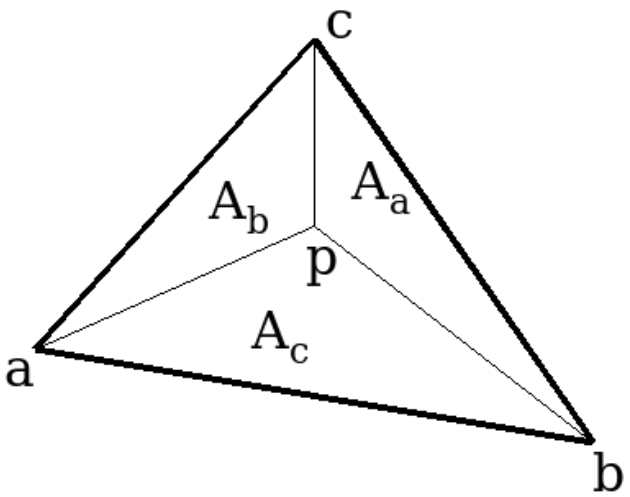
- on ne connait que les coordonnées des sommets du triangle,
- comment calculer la profondeur, z du pixel / point ?
- interpoler la profondeur des sommets...
- comment ?

retour au test d'inclusion d'un point dans un triangle...

Zbuffer et interpolation

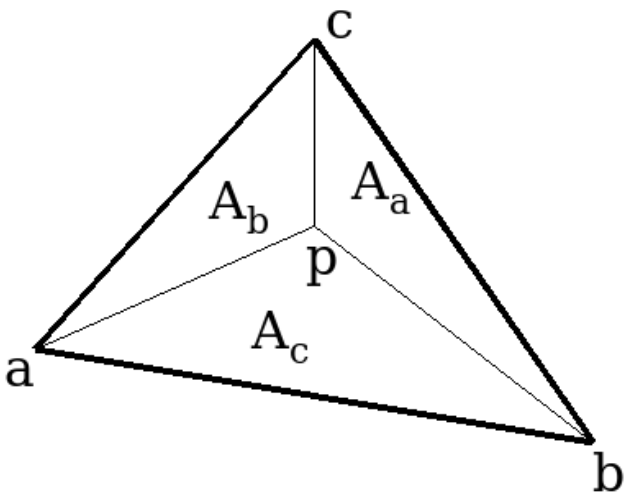
le pixel est inclus dans le triangle :

- le test d'inclusion calcule les 3 triangles : pab, pbc, pca
- et leurs aires,
- ces 3 aires définissent les coordonnées barycentriques de p.



Zbuffer et interpolation

- $A_a + A_b + A_c = A$
- $\alpha = A_a/A$, $\beta = A_b/A$, $\gamma = A_c/A$,
- $\alpha + \beta + \gamma = 1$,
- la position de p peut etre calculée par interpolation :
- $p(\alpha, \beta) = \alpha a + \beta b + \gamma c$ avec $\gamma = 1 - \alpha - \beta$.



Zbuffer et interpolation

interpolation barycentrique :

- permet d'interpoler n'importe quelle propriété des sommets du triangle,
- position, profondeur,
- couleur, normale, texture, etc...

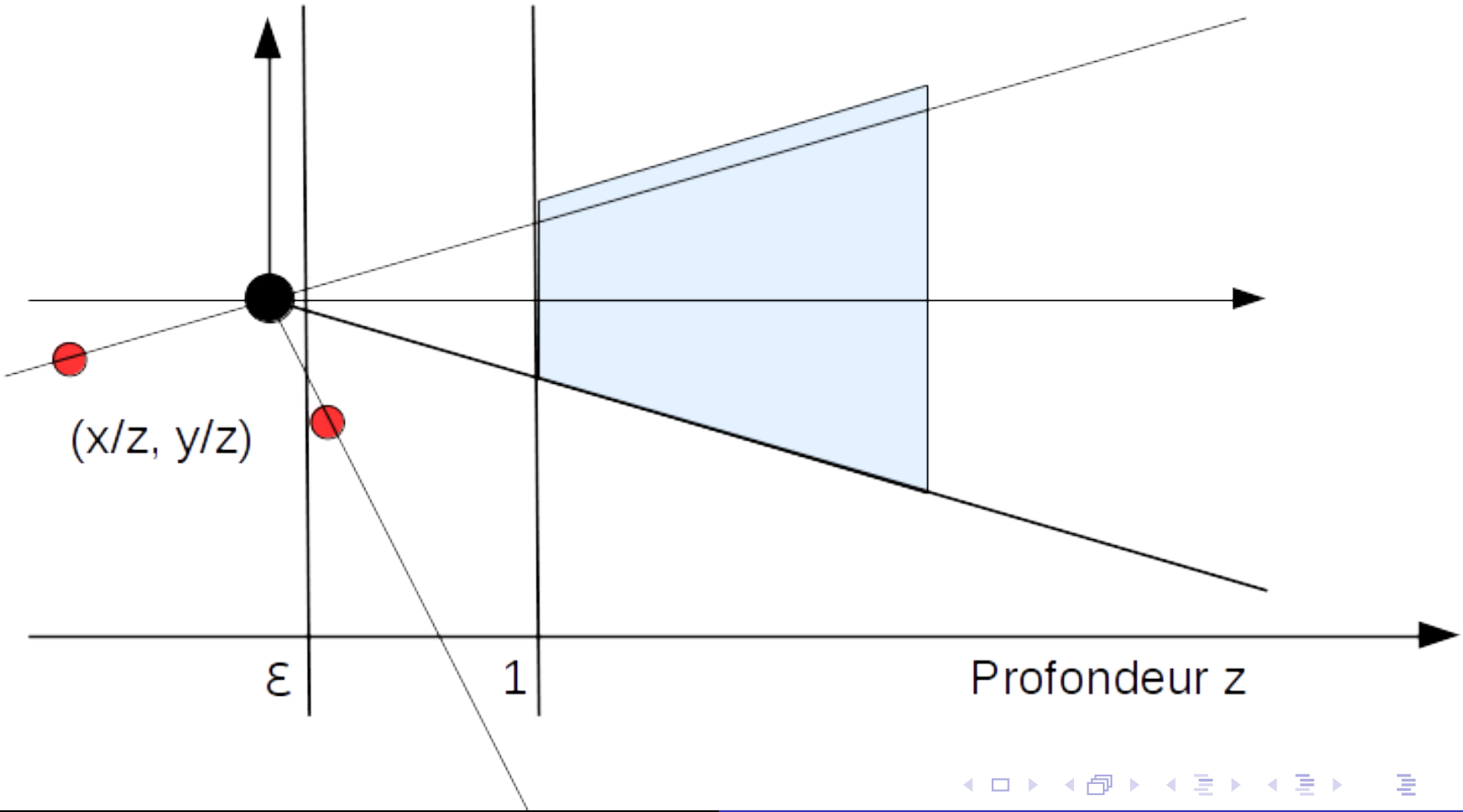
et en 3d ?

$1/z$ n'est pas toujours représentable...

- un point derrière la camera ($z < 0$) se projete aussi sur le plan image...
- un point trop proche de la camera ($0 < z < \epsilon$) se projete sur le plan image, mais n'est pas représentable numériquement avec des float ou des double.

tester avant de projeter...

et en 3d ?



et en 3d ?

triangle non représentable :

- si un sommet du triangle n'est pas projetable,
- découper le triangle pour ne garder que la partie projetable,
- mais ce n'est plus un triangle...

si les 3 sommets sont non projetables, rien à dessiner...

retour au problème

produire une image :

- en dessinant des triangles,
- *trianguler* la *surface* des objets (*attention* à l'orientation),
- pour chaque fragment, calculer la couleur du pixel,
- ne garder que le plus proche de la camera,
  (pour les objets opaques...)

- placer et orienter les objets dans la scène,
- placer et orienter la camera dans la scène,
- transformer les sommets des triangles dans le repère de la caméra,
- dessiner chaque triangle...

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 et la projection.

idée

comment représenter une translation avec une matrice ?

espace homogène et 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

$$p_h = w \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix}$$

$$p_h = \begin{bmatrix} x \\ y \\ z \\ w \neq 0 \end{bmatrix}$$

on retrouve le point réel associé au point homogène en divisant par w :

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

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" perspective sur le plan $z = d$

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

$+$ retrouver le point réel associé $= \begin{bmatrix} d\frac{x}{z} \\ d\frac{y}{z} \\ d \\ 1 \end{bmatrix}$

composition de transformations

toutes les transformations se représentent sous forme de matrices.

plusieurs repères :

$$\text{objet} \xrightarrow[M]{} \text{scène} \xrightarrow[V]{} \text{observateur} \xrightarrow[P]{} \text{projection} \xrightarrow[I]{} \text{image}$$

déterminer directement les coordonnées d'un point de l'objet dans le repère projectif : $q = P(V(Mp))$
transformation globale : $q = Tp$ avec $T = P \cdot V \cdot M$

passer d'un repère à l'autre avec l'inverse de la transformation :
$p = T^{-1}q$