

Notes de cours non autorisées. Calculatrices autorisées.

1) Tramage

- a) Expliquer brièvement l'utilité de cette technique.
b) On donne la matrice de Sierra (l'entrée entre parenthèses symbolise le pixel courant)

$$\frac{1}{4} \begin{pmatrix} 0 & 0 & 2 \\ 1 & 1 & 0 \end{pmatrix}$$

Utiliser celle ci pour effectuer le tramage de Floyd-Steinberg (par diffusion de l'erreur) d'une image dont les valeurs de luminosité de chaque pixel sont comprises entre 0 (noir) et 15 (blanc) et sont arrangées dans la matrice suivante :

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 8 & 8 & 8 & 8 \\ 8 & 8 & 8 & 8 \\ 15 & 15 & 15 & 15 \end{pmatrix}$$

Lors du tramage, on arrondira chaque valeur vers la valeur la plus proche parmi 0 ou 15, et on diffusera l'erreur commise vers les pixels voisins en utilisant la matrice de Sierra. Lorsqu'une opération arithmétique donne une valeur non entière, arrondir immédiatement à la valeur supérieure (3.25 → 4 , 3.5 → 4 , 3.75 → 4).

- c) Quel serait le résultat obtenu par simple arrondi ? Expliquer.

2) Transformations géométriques et coordonnées homogènes

Lors de la transformation des coordonnées utilisateur vers les coordonnées écran, un dispositif d'affichage concatène habituellement trois matrices utilisées pour transformer les coordonnées « utilisateur » des points de l'espace en coordonnées locales pour l'affichage:

$$M = M_s \cdot M_{proj_persp} \cdot M_v$$

La première de ces matrices (à droite) est une matrice de passage des coordonnées utilisateur vers des coordonnées universelles (liées à la caméra) par des translations et rotations, telles que l'on regarde dans le sens des z négatifs, que l'observateur soit situé à l'origine et que le midi (verticale) soit selon l'axe des y positifs. Cette matrice est sous la forme suivante :

$$M_v = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- a) Dans cette matrice, quelles sont les composantes qui définissent la partie « translation » de la transformation ?

Dans la suite, on considérera que l'observateur regarde dans la bonne direction, mais se trouve à une position telle que $x=1$, $y=2$ et $z=-1$ (valeurs exprimées en coordonnées utilisateur). Il convient donc d'effectuer une opération pour amener le repère dans les conditions prescrites. Quelle est la matrice M_v permettant cette opération ? La seconde matrice est une matrice dite perspective (en projection centrale) qui ramène les coordonnées précédentes vers le volume canonique $[-1,1]^3$. Elle a la forme suivante :

$$M_{proj_persp} = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{l+r}{l-r} & 0 \\ 0 & \frac{2n}{t-b} & \frac{b+t}{b-t} & 0 \\ 0 & 0 & \frac{f+n}{n-f} & \frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- b) A quoi correspondent les variables n, f, r, l, b, t dans cette matrice ?
- c) Comment est effectuée l'opération non linéaire qui permet d'effectuer la projection centrale proprement dite ?
- d) Prenons $n=-10$; $f=-110$; $l=-15$; $r=15$; $b=-10$; $t=10$. Construire la matrice et effectuer le changement de coordonnées pour un point **initial** situé en $(0,0,50)$. (attention, il faut d'abord appliquer M_v) Le point résultant est-il situé dans le volume canonique ?
- e) La dernière matrice appliquée permet, à partir des coordonnées canoniques, de projeter sur l'écran. Celui-ci est constitué de (n_x, n_y) pixels, centré à l'origine.

$$M_s = \begin{pmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x-1}{2} \\ 0 & -\frac{n_y}{2} & 0 & \frac{n_y-1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Calculer les coordonnées écran du point précédent (obtenu en d)) par transformation avec cette nouvelle matrice. On prendra $n_x = 1920$ et $n_y=1280$.

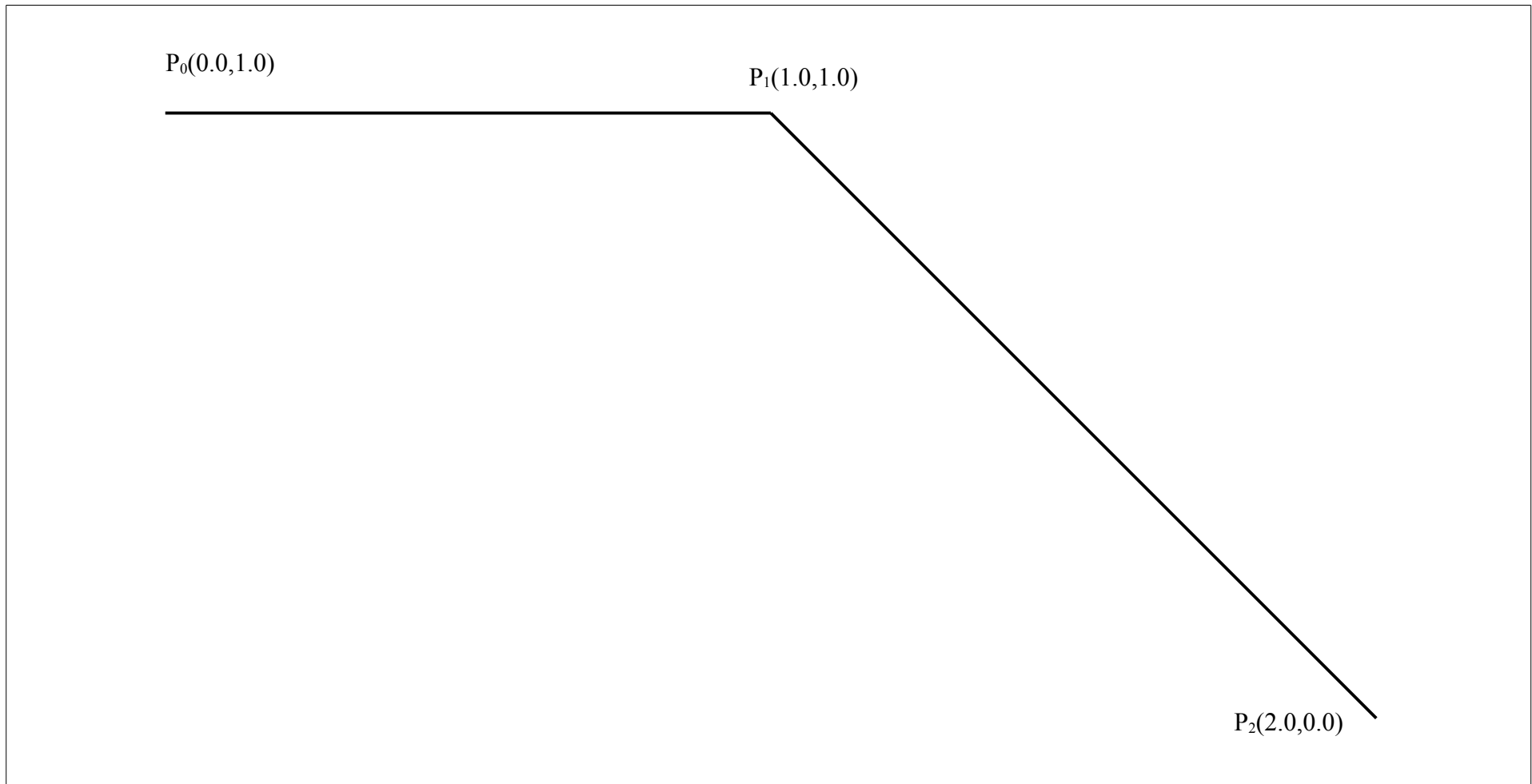
Question exclusivement pour les inscrits au cours de 30h:

- f) On souhaite utiliser la coordonnée z_c (après la division perspective) afin de pouvoir implémenter un z-buffer pour l'élimination des faces cachées. Pour cela, on souhaite pouvoir découper le domaine en tranches selon les z constants et assigner à chaque tranche un nombre entier positif k . On pose $k=0$ pour les points situés dans le plan $z=n$ et $k=2^8-1$ pour les points situés dans le plan $z=f$, 8 étant ici le nombre de bits utilisés pour encoder l'entier k . Quel est l'écart minimal Δz séparant deux plans parallèles au plan de projection pour lesquels deux points situés sur ces plans auront nécessairement des valeurs k différentes d'une unité au moins ? A quelle valeur minimum devrait-on fixer n dans la matrice de projection perspective pour être certain que $\Delta z < 0.1$, tous les autres paramètres restant fixés ?

3) Représentation de courbes

On s'intéresse ici à une courbe de Bézier C dans le plan, et dont le polygone de contrôle est donné sur la page suivante :

- a) Quel est le degré polynomial de la courbe ?
- b) Calculer numériquement les coordonnées du point G correspondant à $t=1/2$ (on peut s'aider de l'algorithme de De Casteljau sous forme de construction graphique – mais le résultat analytique est demandé). Quels sont les points correspondants à $t=0.0$ et $t=1.0$?
- c) À l'aide des points intermédiaires calculés à l'aide de l'algorithme de De Casteljau, donner les coordonnées des points de contrôles des deux courbes (confondues avec la courbe de départ) formées par le découpage de la courbe C par le point G : La courbe P_0 - G et la courbe G - P_2 . Positionner ces points sur le graphique.
- d) On construit en fait à partir des 3 points de contrôles initiaux $P_{2i}P_{2i+1}P_{2i+2}$ (ici $i=0$) un nouveau polygone de contrôle formé ici des 5 points de contrôle : $P_{4i}^{(1)}P_{4i+1}^{(1)}P_{4i+2}^{(1)}P_{4i+3}^{(1)}P_{4i+4}^{(1)}$ qui sont les points calculés à la question précédente. Exprimer les coordonnées de chacun de ces points en fonction des coordonnées des P_j initiaux.
- e) Renouveler cette opération (sur le graphique) et numéroter les points $P^{(2)}$ du nouveau polygone de contrôle (il y en a 9). Ici, les points de départ sont les $P^{(1)}$ calculés précédemment (il y en a 5 donc cette fois-ci i peut valoir 0 ou 1). En répétant l'opération, que peut-on dire de la séquence de polygones de contrôles $P^{(k)}$ lorsque $k \rightarrow \infty$?
- f) Cette procédure a été adaptée aux surfaces, elle permet de définir des surfaces dont les propriétés sont similaires aux propriétés des surfaces B-Splines d'ordre peu élevé. Comment se nomment ces surfaces dans le cas général ?



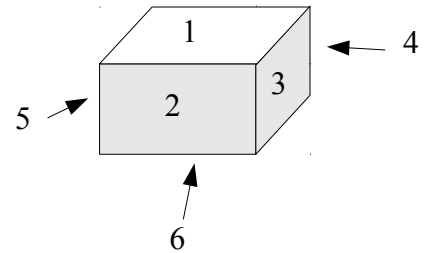
4) Textures

On souhaite faire apparaître un léger relief sur une surface lisse (par exemple un plan ou une sphère). Décrire trois méthodes possibles et expliquer les avantages ou inconvénients de celles ci.

5) Radiosité - Équation de rendu

Soit un cube (6 faces) formé d'une surface émettrice de lumière ($E_1=1, R_1=2/3$) et des autres faces toutes identiques ($E_n=0, R_n=2/3$). Les équations de la radiosité se mettent sous la forme suivante :

$$\begin{pmatrix} 1-R_1F_{11} & -R_1F_{12} & \cdots & -R_1F_{1n} \\ -R_2F_{21} & 1-R_2F_{22} & \cdots & -R_2F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -R_nF_{n1} & -R_nF_{n2} & \cdots & 1-R_nF_{nn} \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix}$$



Les dimensions du cube sont $2 \times 2 \times 2$.

a) On demande ici de calculer la matrice du système linéaire et de résoudre (si possible)

Les facteurs de forme sont calculés sur chaque face en utilisant la formule vue en cours :

$$T_{ij} = \int_{A_i} \int_{A_j} \frac{\cos \theta \cos \theta'}{\pi \|x - x'\|^2} V(x, x') dA dA' \quad \text{et} \quad T_{ij} = A_i F_{ij}$$

$V(x, x')$ est le facteur de vue : il vaut 1 si les points x et x' sont mutuellement visibles, 0 autrement.

On calculera de façon approchée avec un seul point d'intégration situé au centre des faces :

$$\int_T f(x, y) dx dy \approx A f(x_c, y_c)$$

b) Après la résolution, la solution obtenue est évidemment constante pour chaque face du cube. La réalité est autre : proche de la face émettrice de lumière, l'intensité émise est plus forte.

Comment peut-on faire pour améliorer le réalisme de la simulation ?

6) Géométrie algorithmique (Question exclusivement pour les personnes ayant suivi le cours de 30h)

- a) Construire la structure de données BSP-Tree – Binary Space Partition Tree (Arbre de partition binaire) correspondant à l'exemple donné ci après, et insérer les segments dans l'ordre indiqué. On donnera une représentation graphique de cet arbre sur la partie droite de la feuille.
- b) Décrire le parcours dans l'arbre BSP de façon à afficher les segments dans l'ordre de distance décroissante, par rapport à un point d'observation O donné.

