

Sujet d'examen

22 décembre 2005

Documents non autorisés.

Lisez attentivement les questions ; on vous conseil de commencer par les questions dont vous connaissez directement la réponse.

Parfois un schéma vaut mieux que vingt longues phrases. Si vous ne connaissez pas la réponse en détail, n'hésitez pas à donner votre intuition, cette dernière vaut toujours mieux que de ne rien répondre.

Les problèmes, et parfois les questions, sont quasi toujours indépendants. Ce n'est pas que la quantité qui compte, on apprécie des réponses claires et précises.

1 Espaces de couleurs

1.1 Questions

- 1.1.1 **Question :** Expliquez brièvement ce qu'est une couleur en utilisant et expliquant les mots "spectre" et "réflectance".
- 1.1.2 **Question :** Expliquez pourquoi deux spectres différents peuvent-être perçus comme la même couleur par un être humain (métamères).
- 1.1.3 **Question :** Décrivez la signification et l'avantage de chacun de ces espaces : RGB, LUV, HSV.
- 1.1.4 **Question :** Pourquoi la compression (diminuer la résolution d'un canal) d'une image en RGB se voit moins si elle est effectuée sur la canal B.
- 1.1.5 **Question :** Si on aimerait un rouge moins saturé par rapport à un rouge donné. Quel espace de couleurs proposeriez-vous et comment vous y prendriez-vous pour ce traitement ?

2 Raytracing

2.1 Questions

- 2.1.1 **Question :** Expliquez le principe de la caméra sténopée ("pinhole").
- 2.1.2 **Question :** Pourquoi, en réalité, les images créées avec des cameras basé sur ce principe ne sont-elles pas toujours "nette" ?
- 2.1.3 **Question :** Comment pourriez-vous simuler cet effet avec du raytracing ?
- 2.1.4 **Question :** Regardez l'image 1. La photo a été prise pendant une éclipse de soleil et n'a pas été manipulée par ordinateur. Comment expliquez-vous de telles ombres par terre ?



FIG. 1 – Deux photos prises pendant une éclipse.

3 Visibilité et transformations

3.1 Généralités

3.1.1 **Question :** Comment peut-on étendre un espace euclidien à un espace homogène ?

3.1.2 **Question :** Quels sont les nouveaux points de cet espace qui n'ont pas d'association avec des points de l'espace euclidien (mathématiquement, on supposant qu'on utilise l'association naturelle (canonique) entre les deux espaces, vue en cours.) ? Que représentent-ils (en Français) ?

La transformation perspective en OpenGL s'écrit sous forme matricielle :

$$\begin{pmatrix} \frac{f}{aspect} & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & \frac{zfar+znear}{znear-zfar} & \frac{2*zfar*znear}{znear-zfar} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

où f est le champ de vision, $aspect$ le ratio entre l'ouverture haut/bas et gauche/droite, $zfar$ (respectivement $znear$) la distance du centre optique au plan de clipping au loin (resp. proche).

La caméra regarde dans la direction des $-z$ (par $+z$!).

3.1.3 **Question :** Vérifiez que la rétraction d'objet avec la distance marche comme prévu (c-à-d que si le point $(X, Y, Z, 1)$ se projette en $(X', Y', Z', 1)$ alors le point $(X, Y, \alpha * Z, 1)$ se projette en $(X'/\alpha, Y'/\alpha, Z'/\alpha, 1)$ ($\alpha \neq 0$)).

3.1.4 **Question :** Vérifiez qu'un point $(X, Y, -zfar, 1)$ se projette en $(X'', Y'', 1, 1)$ et que $(X, Y, -znear, 1)$ se projette en $(X''', Y''', -1, 1)$.

3.1.5 **Question :** Pourquoi les points de coordonnées $(X, Y, 0, 1)$ sont-ils problématique ?.

3.1.6 **Question :** Expliquez ce qu'est un plan de clipping ("clipping plane") et à quoi il sert.

3.2 Travail sur les Z

Pour l'algorithme du Zbuffer (et uniquement pour lui, pas pour les questions suivantes), la composante Z calculée (avec la matrice précédemment fournie) est normalisée entre 0 et 1 par $f(x) = (x + 1)/2$.

- 3.2.1 **Question :** Décrivez l'algorithme du Zbuffer. Expliquez à quoi il sert en pratique.
- 3.2.2 **Question :** Où se trouve le point $(0, 0, \frac{-z_{near}-z_{far}}{2})$ dans l'espace par rapport aux plans de clipping ? Est-ce la valeur est 0 (au milieu entre -1 et 1) ? En quoi c'est surprenant ?
- 3.2.3 **Question :** Soit $g : [-1, +1] \rightarrow [-1, +1]$ une fonction croissante. Montrez que l'algorithme du Zbuffer fonctionne toujours si l'on stocke $g(Z)$ au lieu de Z .
- 3.2.4 **Question :** Montrez que la valeur en Z obtenue après projection donne plus de précision pour les points proches du plan *near* (première étape : calculer la fonction qui donne le Z' projeté en fonction du Z de l'espace monde situé dans l'intervalle $[-z_{far}, -z_{near}]$).
- 3.2.5 **Question :** A quoi cela pourrait-il servir en pratique ?

4 Ombres

4.1 Application

Un architecte aimerait avoir une représentation très exacte de l'éclairage d'un bâtiment statique. Néanmoins, il aimerait avoir la possibilité de bouger à l'intérieur en temps réel.

- 4.1.1 **Question :** Quelle approche lui proposeriez-vous ?

Dans un jeu vidéo, les artistes souhaitent utiliser 10^6 polygones dans une scène dynamique.

- 4.1.2 **Question :** Pour quelles raisons la technique précédente n'est pas utilisable dans un tel contexte ?
- 4.1.3 **Question :** Pour les ombre de la scène, quelle technique utiliseriez-vous. Pourquoi ?
- 4.1.4 **Question :** Même question pour une scène moins complexe et une scène statique.

4.2 Shadow mapping

- 4.2.1 **Question :** Expliquez l'algorithme du shadow mapping.
- 4.2.2 **Question :** Pourquoi a-t-on besoin d'une valeur ϵ (*epsilon*) dans cette approche ? Quels sont les problèmes pour un epsilon trop petit/trop grand ?
- 4.2.3 **Question :** Quels sont les artefacts rencontrés avec cette approche ?
- 4.2.4 **Question :** A quoi sert d'augmenter la résolution du *depth map* ?
- 4.2.5 **Question :** A quoi sert d'évaluer la visibilité pour plusieurs pixels autour de la position associée dans la *depth map* ?

Imaginons un algorithme de shadow mapping modifié. Au lieu de stocker la distance, on stocke l'index du triangle le plus proche (il y a un index unique pour chaque triangle).

- 4.2.6 **Question :** A quoi cela pourrait-il servir ? Avez-vous une idée pourquoi cette approche n'est pas vraiment utilisé ?
- 4.2.7 **Question :** Sur une carte graphique avec 4 canaux à 8 bits, combien d'indices peut-on encoder ?
- 4.2.8 **Question :** Avez-vous une idée ce qui pourrait coûter cher dans cette approche ?

5 Textures

5.1 Généralités

5.1.1 Question : Expliquez à quoi peuvent servir les textures et ses limitations.

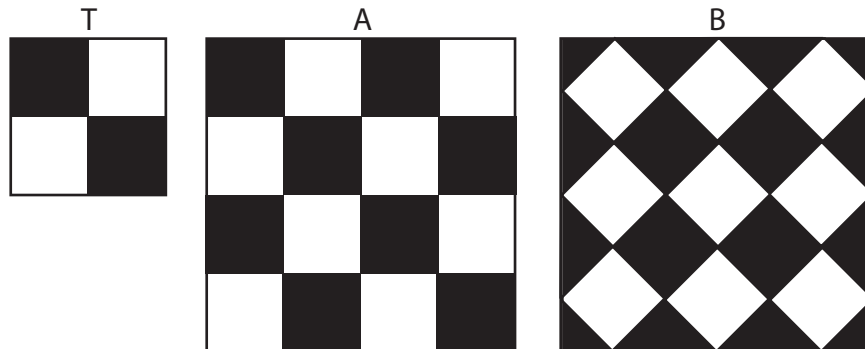


FIG. 2 – Textures

5.1.2 Question : Trouvez les coordonnées de texture pour la texture T qu'il faut spécifier aux coins du quadrilatère A sur la figure 2 pour obtenir le texturage résultant. Quel mode faut-il choisir pour le comportement aux bords (GL_REPEAT, GL_CLAMP)? Expliquez le fonctionnement de ces deux modes.

5.1.3 Question : Comment faire pour B ? (Un dessin est conseillé.)

5.1.4 Question : Quel problème va se produire au moment où l'objet s'éloigne et plusieurs pixels de la texture ne se projettent un seul pixel ?

5.1.5 Question : Décrivez la technique du *MipMapping* et comment elle peut aider à corriger ce problème.

5.2 Area summed tables

Dans la suite on s'intéresse à la technique de l'area summed table (table d'air sommée). Figure 3 explique la technique. À partir d'une texture 2D T donnée, une nouvelle texture A est créée. L'idée est de ne pas stocker directement une couleur dans la texture A, mais plutôt la somme de toutes les valeurs de la texture T qui se trouvent en bas et à gauche de la position du pixel.

5.2.1 Question : Pour faire de A une area summed table de T, quelle valeur faudrait-il attribuer au pixel X ?

5.2.2 Question : Si l'on est intéressé par la somme des valeurs dans un rectangle arbitraire de T, comment pourrait-on calculer cette somme en n'utilisant que 4 valeurs de la texture A ?

5.2.3 Question : À quoi cette technique de l'area summed table pourrait servir si l'on aimerait éviter les effets d'aliasing des textures ?

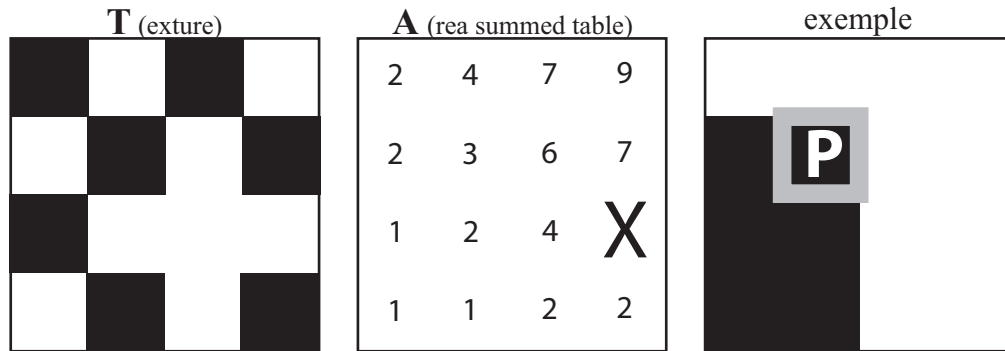


FIG. 3 – Pour une texture T donnée (gauche - les cases blanche ont une valeur de 1, les cases noires de 0) une texture A (milieu) est calculée (la valeur X reste à trouver). La somme de tous les pixels de la texture au-dessous et à gauche de la position courante est stockée. Exemple : pour le pixel P toutes les pixels noirs sont sommés.

6 Parties cachées et silhouettes

6.1 Silhouettes

On s'intéresse au problème de rendu non photo-réaliste. Pour ceci, on souhaite détecter les silhouettes d'un maillage triangulaire.

6.1.1 Question : Décrivez comment on peut définir une arête de silhouette.

6.1.2 Question : Comment calcule-t-on la normale d'un triangle étant donné ses trois sommets ?

6.1.3 Question : Pourquoi n'est-il pas nécessaire de traiter les arêtes adjacentes à deux triangles non orientés vers l'observateur ?

6.1.4 Question : Expliquez ce qu'est le back-face culling.

6.1.5 Question : Comment déterminer si un triangle est orienté vers l'observateur pour une caméra orthogonale/perspective ? Justifiez votre approche.

6.1.6 Question : Comment peut-on (à partir de cette information) détecter les silhouettes d'un maillage ?

Pour accélérer le calcul, on aimerait appliquer une astuce simple, le frustum culling.

6.1.7 Question : Qu'est-ce que le frustum culling ?

6.1.8 Question : Comment peut-on déterminer si une arête se trouve partiellement à l'intérieur du frustum de vue (étant donnés les plans haut, bas, gauche, droite, near et far) ?