Rendu volumique
CPE

5ETI IMI

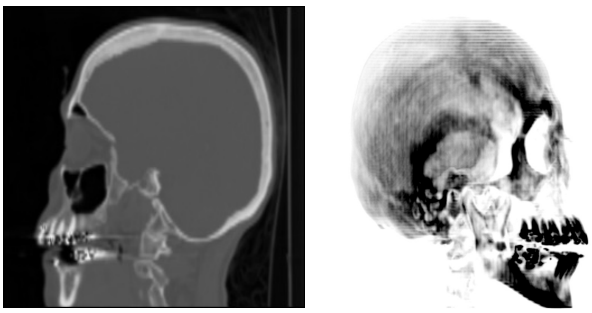

Figure 1: Gauche: Une coupe dans le jeu de données. Droite: Exemple de rendu volumique.

1 But

L'objectif de ce TP est de coder le rendu d'un volume de données scalaires par transparence. Nous traiterons directement les données brutes sans avoir recours à des bibliothèques externes de traitements.

2 Prise en main de l'environnement

2.1 Compilation

Question 1 *Compilez le projet et assurez-vous que celui-ci s'exécute. Assurez-vous que vous puissiez l'éditer depuis l'éditeur de votre choix (QtCreator ou autre). N'oubliez pas que vous pouvez configurer l'emplacement du lancement de votre exécutable dans vos IDE.*

2.2 Classe volume

La classe `volume` contient un jeu de donnée scalaire 3D (variable `vol`). Les données sont stockées en interne sous la forme d'un vecteur concaténé. Pour accéder à la donnée de coordonnées (x, y, z) entière, on utilisera l'appel `vol(x,y,z)`.

Question 2 *Observez la classe volume et les différentes méthodes et fonctions qui lui sont associées.*

2.3 Coupes

Question 3 *Dans la fonction* `main()`*, créez une image de taille $N_x \times N_y$, et stockez dans chaque pixel (k_x, k_y) de celle-ci le niveau de gris correspondant aux données volumiques situées aux coordonnées $(k_x, k_y, N_z/2)$.*
Sauvegardez cette image sur le disque dur et observez la, il s'agit d'une coupe dans les données.

Notez que vous disposez du jeu de données sous différent niveaux de résolutions. Les images de faibles tailles sont plus rapides à traiter dans le cas de tests.

Question 4 *Décommentez le reste du programme* `main()` *jusqu'à la Partie 2. Observez les sorties d'images dans le répertoire* `output/` *Observez les fonctions* `slice_x(),slice_y()` *et* `slice_z()` *qui réalisent ces coupes pour différentes orientations à la position choisie.*

Si l'on souhaite accéder à des coordonnées non entières par interpolation tri-linéaire, il est possible d'appeler la méthode `interpolate(x,y,z)`.

3 MIP

Soit la fonction `mip()` s'appliquant sur une classe `volume`. Cette fonction doit implémenter l'algorithme du *Maximum Intensity Projection* le long de l'axe x.

Question 5 *Créez et implémentez cette fonction. Observez l'image obtenue.*

Question 6 *Décommettez le code jusqu'à Partie 3 et observez l'application d'une rotation sur le jeu de données.*

Question 7 *Lancez le code python suivant* `python generate_animation.py` *qui vient convertir l'ensemble des images au format* `png` *dans le répertoire* `output_png/` *puis réalise un* gif *animé de votre résultat. Est-il facile de connaître le sens de rotation de l'image ? Pour une image arrêtée, est-il possible de savoir si il s'agit d'une vue avant ou arrière ?*

4 Ray casting

La fonction `ray_cast()` implémente l'algorithme du ray-casting suivant l'axe x. Dans un premier temps, on suppose que seul un modèle d'émission est considéré. Supposons de plus que chaque voxel associé à la donnée de valeur v émette une intensité v par unité de longueur. Chaque voxel de donnée est donc associé à une émission d'intensité égale à $v\,\mathrm{d}x$, avec $\mathrm{d}x = 1/(N_x - 1)$.

Question 8 *Implémentez l'algorithme de ce cas, et observez les images de sorties obtenues. Décommettez le reste de la fonction* `main()` *afin d'observer une animation.*

A l'inverse, supposez désormais un fonctionnement par atténuation. C'est à dire que l'intensité est atténuée en fonction des données rencontrées (on initialisera la valeur d'intensité à 1 en début de parcours). On supposera un coefficient d'atténuation égale à $5\,v$ par unité de longueur.

Question 9 *Modifiez l'algorithme précédent pour correspondre à ce cas, observez la sortie obtenue.*

Dans le cas général, il est possible de mélanger atténuation et émission à l'aide de fonctions plus complexes. La courbe d'atténuation/émission en fonction de la valeur de v est appelée fonction de transfert. De plus, il est possible de séparer les facteurs d'atténuation et d'émission dans les différentes composantes de couleurs (r, g, b).
Les fonctions du fichier `transfert_function` permettent de gérer les fonctions de transfert associées à l'émission et l'atténuation dans les différentes composantes de couleurs.

Question 10 *La fonction* `export_transfert_function()` *permet d'exporter la fonction de transfert dans un fichier de données. Appelez en ligne de commande le script* `script_plot.sh` *(il est peut être nécessaire de donner les droits d'exécution au fichier) qui vient afficher les fonctions de transferts sous forme d'images.*

Question 11 *Incorporez dans votre ray-casting l'utilisation de ces fonctions de transfert.*

Question 12 *En modifiant différents paramètres des fonctions de transfert, obtenez une visualisation volumique qui met en avant différents aspects. Vous pouvez éventuellement changer les données d'origine également.*

5 Travail supplémentaire

Question 13 *Mettez en place une composante d'illumination de type phong lors du calcul de l'intensité.*

Question 14 *Observez la librairie VTK, et mettez en place un exemple de visualisation volumique.*