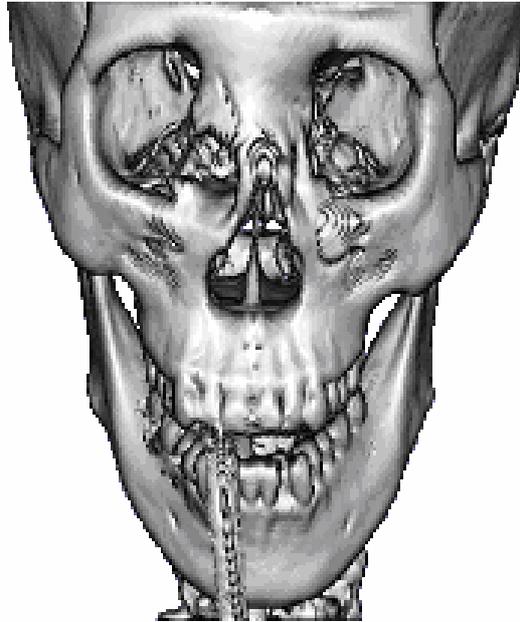


Maîtrise Option Images Etude de Synthèse.



Deuxième partie : Rendu Volumique



MICHEL GRAVE

Année 2003/2004

SOMMAIRE

I – Travail demandé	3
1. Objectif	3
2. Format des données a lire	3
3. Conversion des données en données volumiques	4
4. Méthode du lancer de rayons	4
5. Structure du programme	4
6. Description des fonctions à réaliser	5
➤ Initialiser graphique	5
➤ Charger volume	5
➤ Créer image	5
➤ Rendu	6
➤ Ray Trace	6
➤ Démonstration	6
II – Travail supplémentaire	8
1. Menu	8
2. Les différentes vues	9
3. Transparence	10

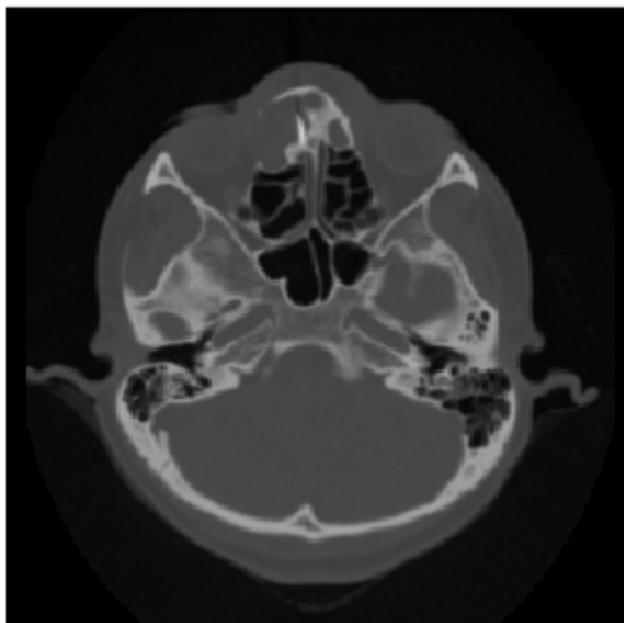
I – Travail demandé

1. Objectif

Visualiser une tête humaine par la technique de lancer de rayon.

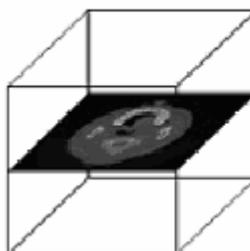
2. Format des données a lire

Les données fournies sont en fait issues d'un scanner cérébral. Cet examen consiste à mesurer les différences d'absorption d'un étroit faisceau de rayons X par les divers tissus qu'il traverse au moyen de détecteurs sensibles placés tout autour de l'appareil. La quantité de rayon X envoyé étant connue, il est possible de calculer la quantité de rayon absorbé par les structures anatomiques examinées. A l'aide d'un ordinateur on peut ainsi transcrire les informations obtenues sous forme d'images de coupes anatomiques. Cette production d'images en coupe porte le nom de TOMOGRAPHIE.



Exemple d'une coupe.

Notre jeu de donnée est constitué de 94 coupes anatomiques (headsq.1 ... headsq.94)
Chacune de ces coupes est perpendiculaire à l'axe du corps.
Pour effectuer sur ces données un rendu, il est donc nécessaire de les convertir en données volumiques.



3. Conversion des données en données volumiques.

Chaque fichier contient donc $256*256$ valeurs de densité codées en niveau de gris sur 2 octets. (le premier octet représentant le poids fort et le second le poids faible)
Les valeurs sont cependant codées sur 12 bits c'est à dire qu'il faut éliminer les 4 bits de poids les plus élevés sur l'octet représentant le poids fort.
Chaque densité codée sur deux octets doit donc être convertie en un entier et insérer au tableau data. L'algorithme de conversion de données est le suivant :

```
    Pour chaque couche ( k )
      Ouverture du fichier
      Pour chaque ligne ( i )
        Pour chaque colonne ( j )
          Lecture du poids faible.
          Lecture du poids fort.
          Application d'un masque pour éliminer 4 bits de poids fort inutiles.
          Construction de l'entier représentant la densité.
          Insertion dans le volume de données.
        Fin
      Fin
    Fermeture du fichier
  Fin
```

4. Méthode du lancer de rayons

Cette méthode a pour principe de projeter des rayons imaginaires dans un volume de données. Suivant les valeurs rencontrées par ces rayons, les pixels correspondants se verront affecter une certaine valeur chromatique.

Dans notre cas et pour simplifier, tout rayon qui part de l'œil de l'observateur est perpendiculaire au volume. Ainsi la propagation du rayon se fait sur une seule dimension c'est à dire la dimension de profondeur k.

Cependant il faut connaître le point d'entrée du rayon dans le volume de données. Une conversion est donc nécessaire pour ramener l'écran aux proportions du volume. Une fois que l'on connaît ce point d'entrée dans la cellule, on connaît la valeur de la densité du point rechercher. Cette valeur est ensuite interpolée avec les densités voisines.

5. Structure du programme

- 3 fichiers + 94 fichiers de données.

- rv.c : Initialisations graphiques, fonction d'affichage, fonction de gestion de clavier et fonction main.
- rv.h: Structures de données.

Les structures de données sont les suivantes :

```

/* Donnees */

#define IMAX 256
#define JMAX 256
#define KMAX 256

EXTERN int data[IMAX][JMAX][KMAX];
EXTERN int dataOri[IMAX][JMAX][KMAX];
EXTERN int imax, jmax, kmax;

/* Image */
#define XMAX 1280
#define YMAX 1024

EXTERN GLubyte image[3*XMAX*YMAX];
EXTERN int xmax, ymax;

```

Les données sont stocker dans le tableau data qui est un tableau 3 dimensions (256 * 256 * 256).

- tp.c : Ce fichiers contient l'ensemble des fonctions décrites ci dessus. C'est a dire l'ensemble des fonctions a réaliser.

6. Description des fonctions à réaliser.

- Initialiser graphique

```
void initialiser_graphique(char *titre)
```

Fonction qui permet de créer la fenêtre de visualisation et d'initialiser les variables nécessaires a son fonctionnement.

- Charger volume

Cette fonction comporte deux parties :

```
int charger_volume()
void modifie_volume(int vue)
```

La première permet de charger le tableau de donnée en les récupérant à partir fichiers headsq.n (n étant un chiffre compris entre 1 et 94).

La deuxième permet de modifier le tableau de donnée afin que celui-ci nous permette de visualiser une autre vue du crâne.

- Créer image

```
void faire_image()
```

Cette fonction permet de lancer le rendu en fonction des différents paramètres que nous choisissons grâce au clavier ou à la souris. Ainsi grâce à cette fonction on peut choisir d'afficher le crâne en mode vue de coupe ou en mode visualisation 3D du crâne. De plus, en mode 3D on peut choisir, si on visualise la peau ou l'os pour chaque demi partie du crâne.

➤ Rendu

```
void rendu(int ia, int ib, int ja, int jb, int kmin, double sa, double sb, int Transparence)
```

La fonction rendu est le cœur du programme c'est elle qui lance les rayons pour chaque pixel de l'image que nous devons recréer.

De plus, c'est elle qui gère la transparence en lançant plusieurs fois le rayon.

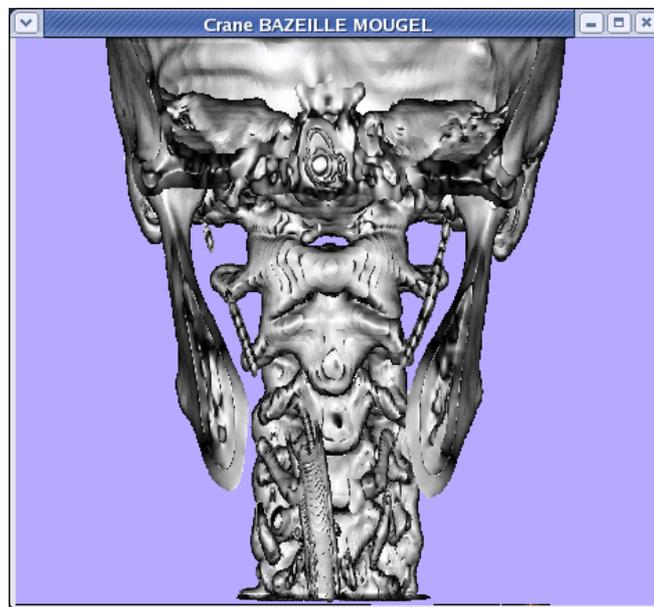
Enfin, si la variable Transparence est égale à 3 (valeur pour le mode coupe) elle effectue seulement une interpolation afin d'afficher l'image en coupe.

➤ Ray Trace

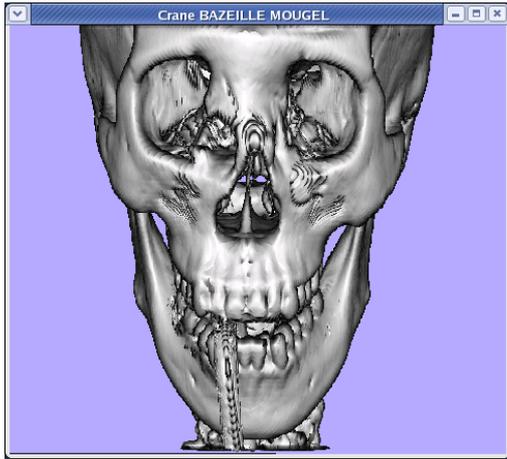
```
double ray_trace(const int k0, const int i, const int j, const double u, const double v, const double seuil)
```

Cette fonction permet de calculer la couleur du pixel grâce au seuil qui nous permet de savoir si on a touché le crâne ou la peau. Si c'est le cas on renvoie la couleur du point que l'on a touché en fonction de sa normale.

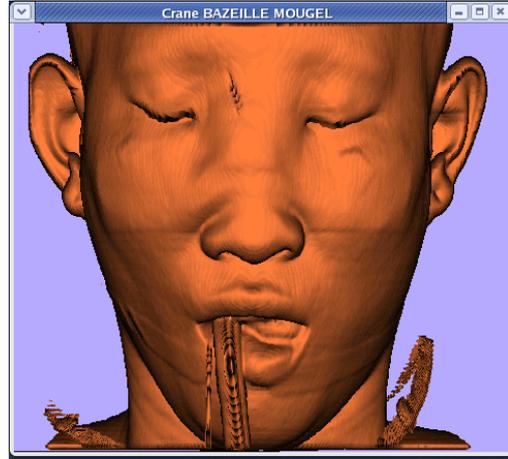
➤ Démonstration



Vue de face avec le plan avancé à 120

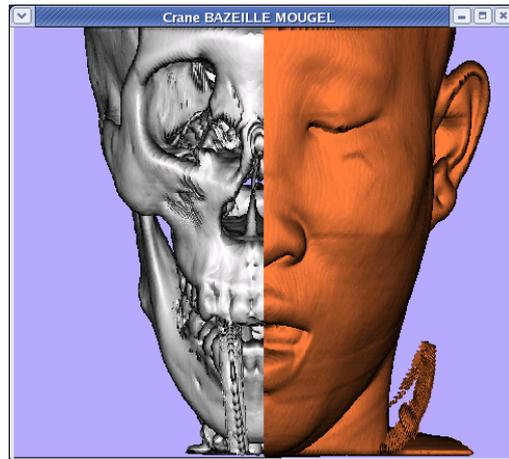
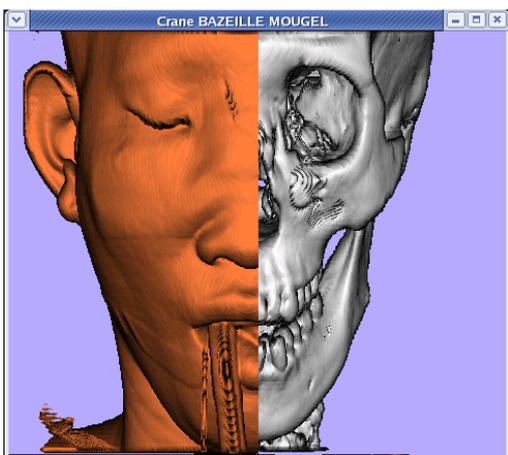


OS



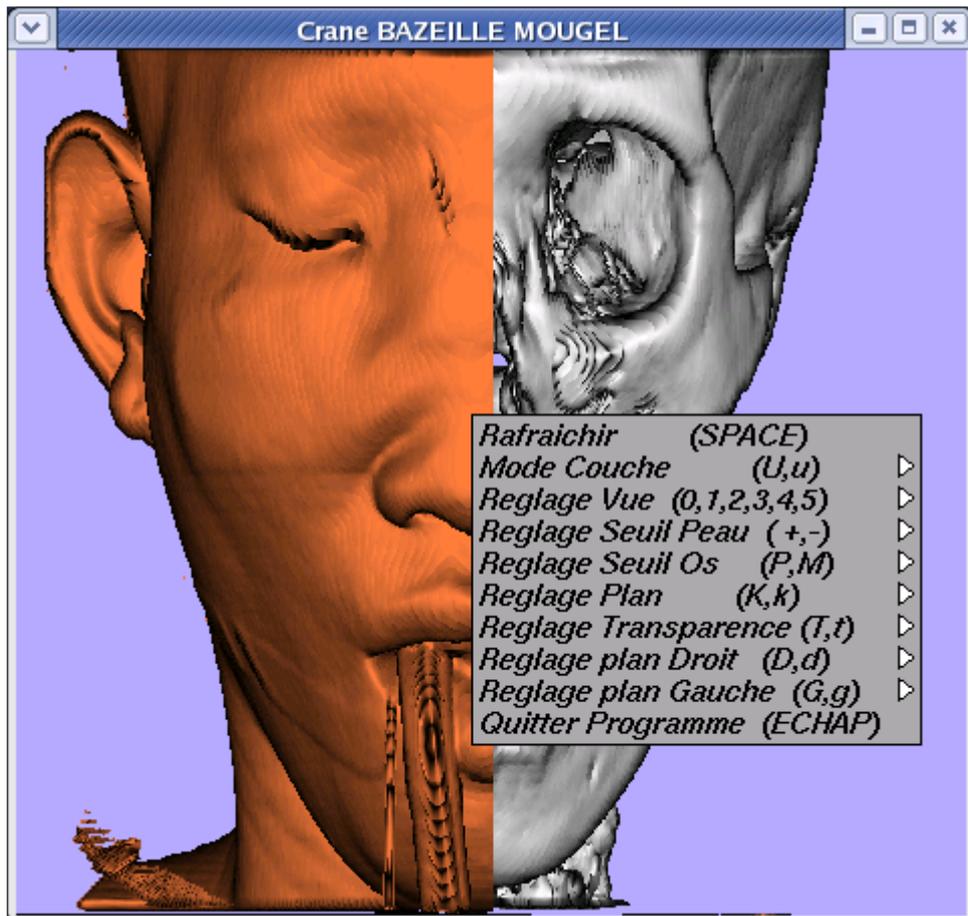
PEAU

Pour les deux images suivantes, la fenêtre d'affichage a été coupée en deux parties. Deux lancers de rayons sont effectués, un avec la valeur de seuil égale à la densité de la peau, l'autre égale à la densité de la l'os.



II – Travail supplémentaire

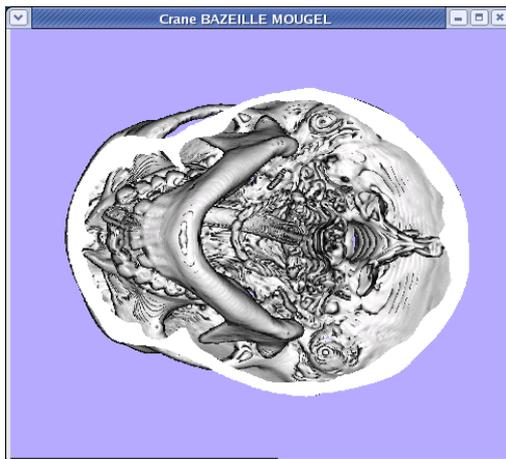
1. Menu



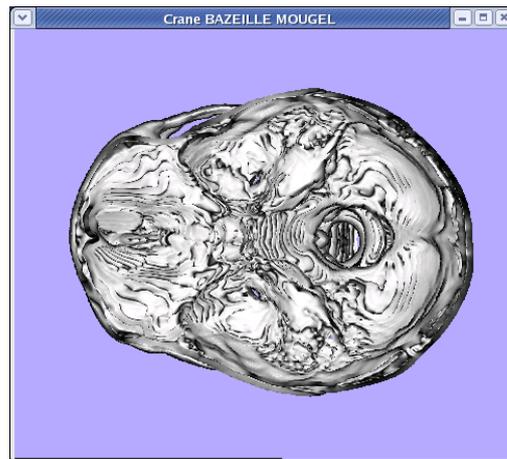
Fenêtre en taille réelle.

Un clic droit de la souris permet d'ouvrir ce menu et de sélectionner « les options ». Il permet aussi de connaître les raccourcis clavier pour chaque réglage. Toutes les fonctionnalités sont disponibles à partir de la souris ou du clavier.

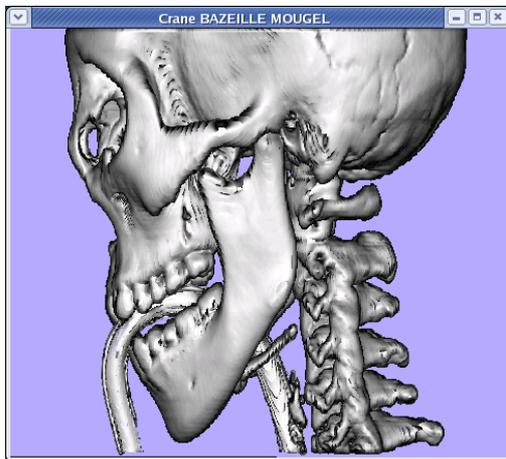
2. Les différentes vues



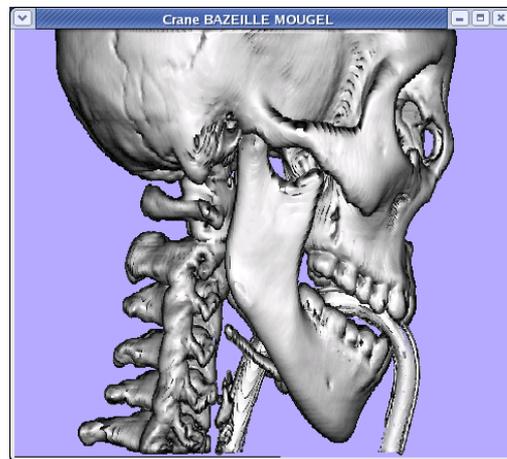
Vue de dessous



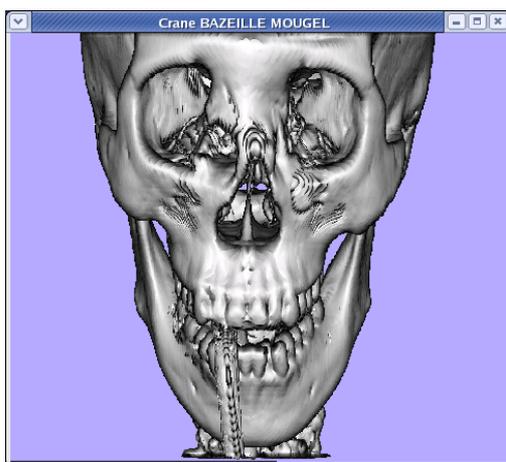
Vue de dessus



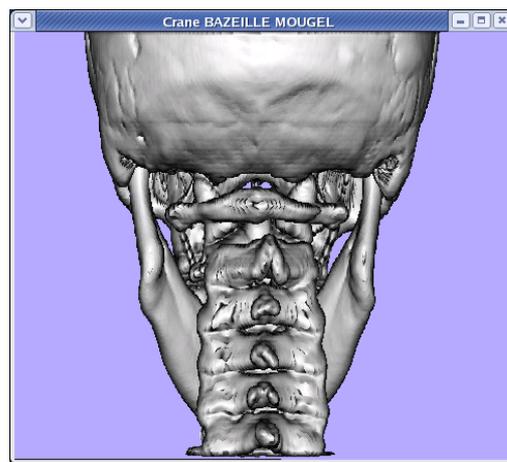
Vue de gauche



Vue de droite



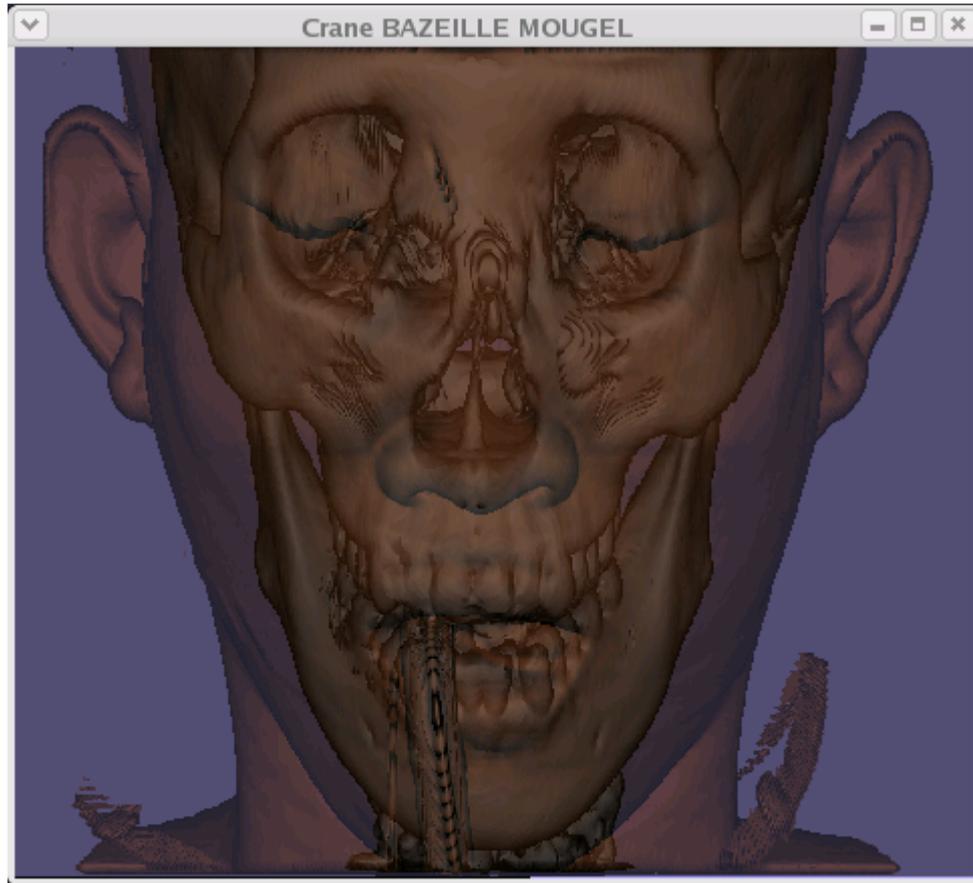
Vue de face



Vue de dos

Toutes les vues peuvent avoir les différents modes pour la demi coupe comme le choix de la peau, de l'os. De plus, le déplacement du plan de vue et la transparence sont disponibles pour l'ensemble de la vue.

3. Transparence



Visualisation a la fois de l'os de la peau sur la même image par transparence.