

Algorithmique et Programmation 1

février 2015

Projet : découper des tranches dans un objet 3D

But du projet

Pour pouvoir manipuler des objets 3D à l'aide d'ordinateurs, on doit pouvoir les coder dans des fichiers. Il existe plusieurs façons de coder les objets tri-dimensionnels. Le but de ce projet est d'apprendre à décoder un de ces formats, puis de découper cet objet en tranches, soit simplement pour en obtenir une représentation sous forme de courbes de niveau, soit pour découper physiquement ces tranches dans du carton, et obtenir ainsi une version 3D de l'objet ¹.

les différentes étapes

Le projet peut se décomposer en plusieurs parties :

- Ouvrir un fichier de description 3D et en extraire les informations.
- Couper l'objet 3D par une famille de plans et calculer les courbes de niveau ainsi obtenues.
- Créer une image pour chaque courbe de niveau calculée à l'étape précédente.

Examinons chacune de ces étapes séparément.

Lire un fichier de description 3D

Le format que nous choisissons d'étudier est le format STL (pour STéreoLithographie). Dans un fichier STL, un objet en trois dimensions est découpé en triangles (voir figure 1. Un objet peut contenir parfois plusieurs millions de triangles, les données sont donc codées en binaire dans le fichier afin de prendre moins de place. La structure du fichier est la suivante :

- Les 80 premiers octets sont un commentaire.
- Les 4 octets suivants forment un entier sur 32 bits qui représente le nombre de triangles présents dans le fichier.
- Ensuite, pour chaque triangle, on a une description sur 50 octets qui se décompose comme suit :

1. Voir par exemple : <http://geekologie.com/2015/01/build-it-yourself-laser-cut-cardboard-hu.php> ou <http://www.instructables.com/id/How-to-Slice-Up-a-T-Rex-in-123D-Make>

- 3 fois 4 octets, chaque paquet de 4 octets représentant un flottant : les coordonnées (x,y,z) de la direction normale au triangle : cette information est importante si on veut un rendu réaliste de l'objet (elle conditionne la façon dont le triangle reflète les rayons lumineux), mais est inutile pour nous dans le cadre de ce projet.
 - 3 paquets de 3 fois 4 octets, chaque groupe de 4 octets représentant un flottant, les coordonnées (x,y,z) de chacun des sommets du triangle.
 - Deux octets représentant un octet de contrôle (inutile dans le cadre de ce projet).
- Quelques indications pour lire un fichier :
- La méthode `read(n)` des fichiers permet de lire et de stocker n octets.
 - La méthode `int.from_bytes()` peut vous aider à transformer un tableau d'octets en un entier.
 - La méthode `struct.unpack()` du module `struct` peut vous aider à transformer une suite d'octets en un flottant. Attention, dans un fichier STL, les flottants sont codés en *little-endian*.

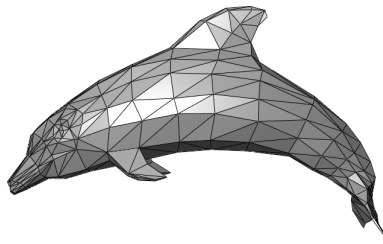


FIGURE 1 – Un objet 3D défini par des triangles (image Chrschn via Wikimedia Commons)

Couper un objet avec un plan

Une fois que vous avez lu et mémorisé tous les triangles de la figure, il va falloir, pour un plan donné, parcourir la liste de triangles pour voir ceux qui seront coupés par ce plan, et calculer la trace de cette coupure. l'ensemble de ces traces constituera une courbe de niveau inscrite sur ce plan.

Représenter un plan

Dans un espace à trois dimensions, un plan est représenté par l'équation :

$$ax + by + cz + d = 0$$

C'est le plan perpendiculaire au vecteur (a, b, c) situé à la distance $-d$ de l'origine. Pour simplifier, nous ne nous intéresserons qu'aux plans parallèles aux axes Ox , Oy , Oz dont les équations se simplifient en :

- $x + d = 0$: les plans perpendiculaires à l'axe Ox passant par $x = -d$
- $y + d = 0$: les plans perpendiculaires à l'axe Oy passant par $y = -d$
- $z + d = 0$: les plans perpendiculaires à l'axe Oz passant par $z = -d$

Comment savoir si un triangle est coupé par un plan connu ?

Pour un point (x, y, z) , la valeur de $ax + by + cz + d$ (où a, b, c, d sont les caractéristiques du plan, nous renseigne sur la distance entre le plan et le point. Le signe de cette quantité nous permet aussi

de savoir de quel côté du plan se trouve le point.

Donc, si pour deux points $A(x_A, y_A, z_A)$ et $B(x_B, y_B, z_B)$, les quantités $ax_A + by_A + cz_A + d$ et $ax_B + by_B + cz_B + d$ sont de signe différents, alors le segment qui relie ces deux points traverse le plan. trouver le point de ce segment qui appartient aussi au plan ne devrait pas être bien compliqué, surtout si les plans sont perpendiculaires aux axes...

Quand un triangle est coupé par un plan, quelle est sa trace sur ce plan ?

Quand un triangle est coupé par un plan, alors un de ses sommets (A) est d'un côté du plan, et les deux autres (B et C) sont de l'autre côté. La trace du triangle coupé sur le plan est le segment qui relie les deux points où le plan coupe les segments AB et AC ...

Pour un plan de coupe donné, l'ensemble des triangles coupés par ce plan produisent une liste de segments qui constituent la trace de la découpe de l'objet complet par ce plan.

Dessiner des courbes de niveau

On découpera l'objet en N plans parallèles (I.e. ne différant dans leur équation que par la valeur du coefficient d) : à vous de voir quelles valeurs donner à ce coefficient d pour obtenir N tranches intéressantes. Pensez à la façon dont vous coupez un pain...

On devra pouvoir faire les découpes selon les trois axes Ox, Oy, Oz .

Vous pouvez choisir de dessiner toutes les courbes de niveau sur la même image, ou bien produire une image pour chaque courbe de niveau. Cette deuxième solution vous permet d'avoir un modèle pour ensuite découper chacune des tranches dans du carton épais, et construire un modèle 3D réel. Vous pouvez chercher quel module Python vous semble le mieux adapté à la production des images. A priori, le module Tkinter, inclus dans la distribution de base, devrait suffire.

Améliorations possibles

- Calculer des courbes de niveau et des encoches dans deux dimensions perpendiculaires, pour obtenir un *sliceform*, comme sur la figure 2
- Permettre l'utilisation de plans quelconques.



FIGURE 2 – Quelques exemples de sliceforms (auteur et photographe : Maria Garcia Monera)

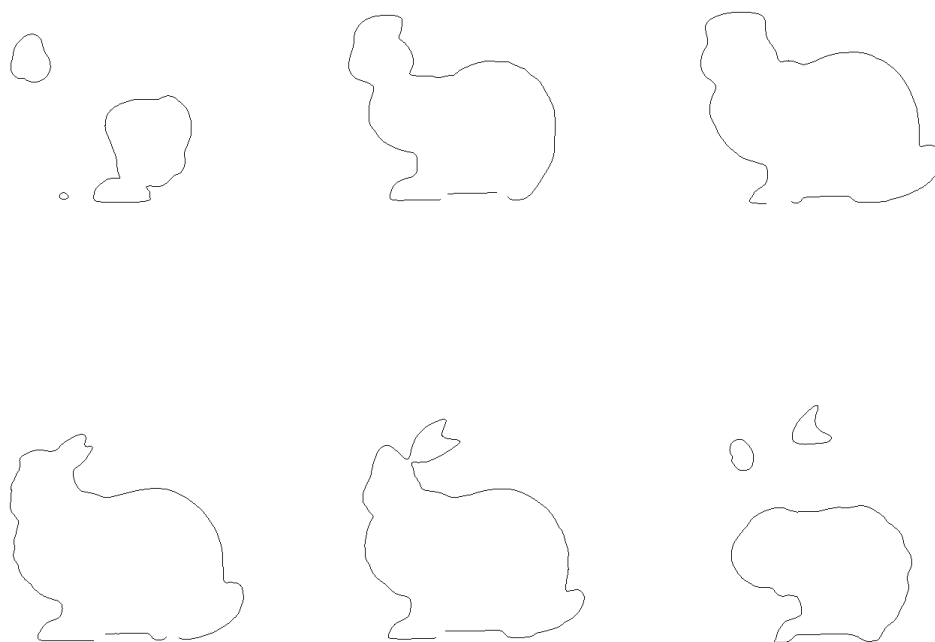


FIGURE 3 – Tranches de lapin