Modélisation par surfaces implicites

Sujet proposé par Dominique Rossin

rossin@liafa.jussieu.fr

Le but de ce projet est de réaliser une plateforme capable de représenter des surfaces implicites. On se donne une fonction $f(x, y, z)$ et on cherche à trouver l'isosurface $f(x, y, z) = isovaleur$ avec $isovaleur \in \mathbb{R}$. Le projet se découpe donc en quatre grands axes:

1. Écrire un interpréteur pour le langage associé.
2. Trouver une hiérarchie d'objets permettant de représenter les fonctions.
3. Résoudre $f(x, y, z) = c$ et en déduire une triangulation de la surface.
4. Dessiner cete surface. Cette partie ne sera pas à programmer. En effet, nous utiliserons `geomview` qui permet de faire cette visualisation à partir d'un fichier de données.

La modélisation d'objets par des surfaces implicites offre de très nombreux avantages par rapport aux représentation classique des surfaces par assemblage de formes géométriques simples. En effet, on a un controle sur la continuité et la dérivabilité des surfaces obtenues. Ainsi, il est facile de réaliser la jonction de deux objets de manière à ce que cette jonction ne se fasse pas par une arête vive mais par une surface lisse.

1 Les fonctions

Les fonctions que nous allons employer sont des "assemblages" de fonctions de base. Les fonctions de base seront du type $f(P) = g(d(P, O))$ où P est un point de l'espace, O un objet géométrique (point, carré, rectangle, disque, cube ...), $d(P, O)$ la distance du point P à l'objet O et $g(d)$ une fonction potentielle à une variable.

1.1 Fonction potentielle

Nous n'utiliserons qu'un seul type de fonction potentielle dans ce projet à savoir:

$$\begin{cases} g_R(d) = \frac{1}{R^8}(d^2 - R^2)^4 \text{ Si } d < R \\ g_R(d) = 0 \text{ sinon} \end{cases}$$

On remarquera qu'une telle fonction est $\mathcal{C}^1$. L'intéret de prendre une fonction de ce type plutôt qu'une gaussienne est la rapidité de calcul. En effet, pour des points éloignés la fonction est nulle et pour les autres il ne faut que deux multiplications et une soustraction pour calculer la valeur en un point.

1.2 Fonctions de base

Les fonctions de base dépendent de deux données:

- Une fonction potentielle g_R
- Un objet géométrique O

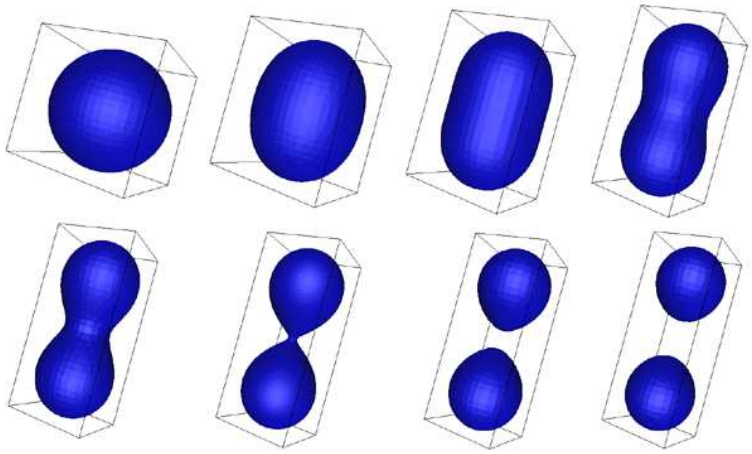
Les objets considérés seront le point, le segment de droite, le carre, le cercle, la boite, la sphere, ... Tous les objets considérés seront représentés centrés en $(0,0,0)$ et seront dotés d'une couleur.

1.3 Fonction de composition

Nous allons maintenant composer les fonctions de base afin d'obtenir des objets plus complexes.

1.3.1 Addition

L'addition de deux fonctions permet de réaliser des blobs. Par exemple si l'on considère la somme de deux fonctions de base dépendant de la distance à un point, suivant les positions respectives des points on peut obtenir les différentes figures suivantes:



1.3.2 Maximum et minimum

Le maximum entre deux fonctions permet de faire une union des surfaces représentant chacun des objets.

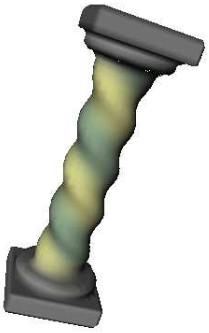
Le minimum permet d'extraire l'intersection des deux volumes.

1.4 Transformations

Une transformation est une déformation locale $h(x,y,z)$ de l'espace que l'on applique au point M avant de calculer le potentiel. Nous utiliserons les transformations suivantes:

- Translations : $h(x,y,z) = (x - x0, y - y0, z - z0)$
- Rotations autour des axes $(Ox), (Oy), (Oz)$.
- Warp : Transformation hélicoïdale autour de (Oz)

En combinant deux boites, deux disques, deux segments et un warp on peut obtenir la figure suivante:



1.5 Représentation de la surface

Partant d'une fonction $f(x, y, z)$ nous allons représenter la surface sous forme de petits triangles. Pour ce faire, on imagine que l'espace est subdivisé en cubes de petite taille puis sur chacun de ces cubes, on regarde si la fonction f-isovaleur s'annule. Si tel est le cas, on construit le ou les triangles correpondants.

De manière plus précise, on calculera la valeur de la fonction $f - isovaleur$ aux sommets du cube. Si ces différentes valeurs ne sont pas toutes de même signe alors on dira que la surface traverse le cube. En considérant maintenant une arête reliant deux sommets de potentiels de signe différents, on approxime le zéro de la fonction par interpolation. Ensuite il reste à considérer l'ensemble de ces points d'intersection et d'approximer la surface par des triangles à l'intérieur du cube. Notez que si on se limite aux fonctions somme et à la transformation locale de l'espace les fonctions considérées sont C^3 et donc il n'y pas de points singuliers.

On écrira les données dans un fichier "sortie.off" sous la forme:

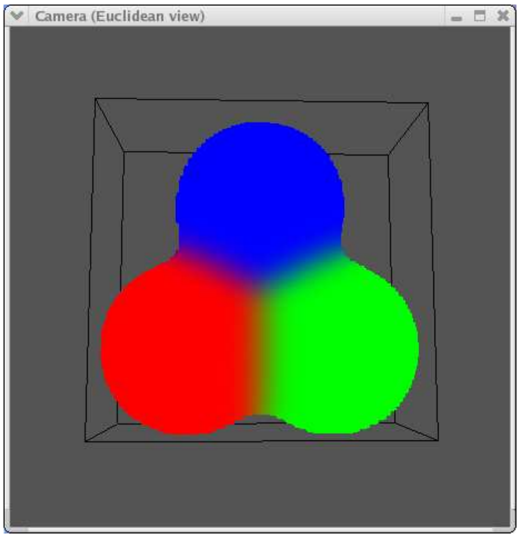
```
COFF
nSommets nFaces 0
x0 y0 z0 r0 g0 b0 1.0    ## Coordonnes du point 0 suivi de sa couleur
                         ## (r=red,g=green,b=blue)
x1 y1 z1 r1 g1 b1 1.0    ## 0.0 <= r,g,b <= 1.0
...
x(n-1) y(n-1) z(n-1) r(n-1) g(n-1) b(n-1) 1.0
3 p1 p2 p3               ## 3 pour triangle suivi des numeros des
                         ## 3 points du triangle
                         ## dans l'ordre trigonometrique
```

Puis on pourra visualiser le résultat à l'aide de *geomview sortie.off*

2 Langage

Nous proposons d'utiliser les instructions suivantes :

- $pR(0.3)$ désignera la fonction potentielle pour $R = 0.3$
- $dPoint(potentiel, r, g, b)$: Point situé en $(0, 0, 0)$ et de fonction potentielle `potentiel` et de couleur r,g,b



- $dCube(potentiel, taillex, tailley, taillez, r, g, b)$: cube de taille taillex,tailley,taillez
- $dSegment(potentiel, taille, r, g, b)$: segment centré en $(0, 0, 0)$ sur l'axe (Oz)
- $dDisque(potentiel, rayon, r, g, b)$
- $fMax(fonction_1, fonction_2)$: Fonction définie comme le maximum des fonctions 1 et 2.
- $fPlus(fonction1, fonction2)$
- $tTranslation(fonction, x, y, z)$ translation de l'espace de vecteur $(x.y, z)$
- $tRotationX(fonction, angle)$ les angles sont donnés en degré.
- $tRotationY(fonction, angle)$
- $tRotationZ(fonction, angle)$
- $tWarping(fonction, longueur)$: longueur d'un tour d'hélice.

Ainsi, l'appel à
`fPlus(fPlus(dPoint(pR(10),1,0,0),tTranslation(dPoint(pR(10),0,1,0),10,0,0)), tTranslation(dPoint(pR(10),0,0,1),5,10,0))` produira:

3 Extensions

Une attention particulière sera portée sur la structuration de votre programme et la facilité à rajouter :

- Des fonctions potentielles.
- D'autres primitives géométriques.
- D'autres algorithmes de rendu.

Comme alternative pour le rendu, on pourra utiliser la technique dite du MarchingCube. L'idée est de suivre le contour de la surface plutot que de parcourir l'ensemble des cubes. Pour cela, on cherche un premier cube intersectant la surface, puis selon l'endroit ou la surface l'intersecte on en déduit les cubes voisins qui intersectent la surface. Ainsi, le résultat produit sera identique à la technique précédement mise en œuvre mais sera beaucoup plus efficace en terme de temps.