UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE

INSTITUT D'ÉLECTRONIQUE ET D'INFORMATIQUE GASPARD-MONGE

M1 Informatique 2012/2013

Projet Infographie

Rendu par point d'arbre CSG

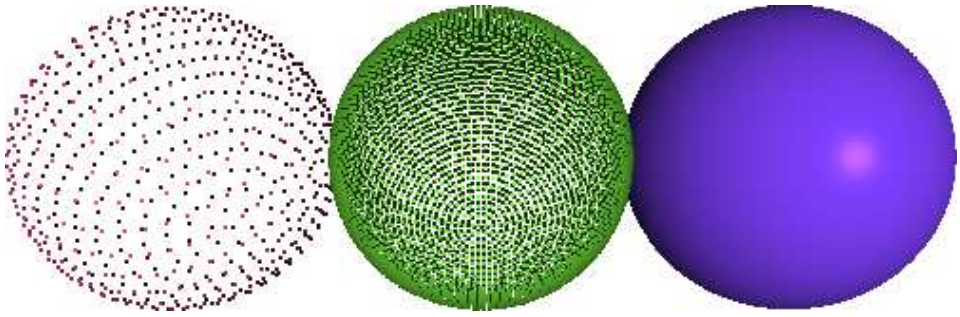
Une première étape de ce projet concerne la modélisation d'objets géométriques simples (sphere, cube, cylindre, cône, tore) et leur visualisation en utilisant la technique du **rendu par points** *(Point-Based Rendering).*
Une seconde étape est l'application de cette technique au rendu d'objets complexes obtenus par la technique des CSG *(Constructive Solid Geometry)*

Modélisation et rendu par points

Parmis les techniques classiques de rendu permettant de visualiser des objets 3D, la plus simple consiste à modéliser l'objet par un ensemble de sommets (ou vertex) et normales à partir desquels on définit facilement une topologie de facettes (triangles ou quadrilatères). L'avantage principal est que peu de sommets permettent, grâce aux performances du pipeline graphique, d'obtenir un rendu de très bonne qualité (cf. cours et TD). L'inconvénient majeur est qu'il est très difficile, avec cette technique, de modéliser des intersections entre objets (nécessité de redéfinir le maillage de sommets pour reconstruire les facettes au niveau des intersections).

Une autre technique classique, le lancer de rayon (*Ray Tracing*), permet de s'affranchir de toute topologie sur les objets en ne considérant que les interactions de la surface analytique avec la lumière. L'inconvénient est que cette technique est beaucoup plus coûteuse en calcul et n'est généralement pas gérée par le pipeline graphique (en particulier avec OpenGl).

Une troisième technique, beaucoup plus économique que le lancer de rayon et beaucoup plus simple que la facétisation, consiste à représenter l'objet par un nuage de points distribués sur sa surface. Si ce nuage est suffisemment dense on pourra avoir l'illusion d'une surface continue (cf. images suivantes). Comme pour la technique des facettes, le point de départ consiste à définir un ensemble de vertex et normales et de considérer chacun de ces éléments comme une *mini-facette* isolée représentant la surface localement : on s'abstrait ainsi de toute considération topologique.



une sphère en rendu par points à basse, moyenne et haute résolution

Objets Canoniques

Les objets de base seront représentés par leur objet canonique de référence sur lesquels devront par la suite être appliquées des transformations (cf. seconde partie : arbres CSG).

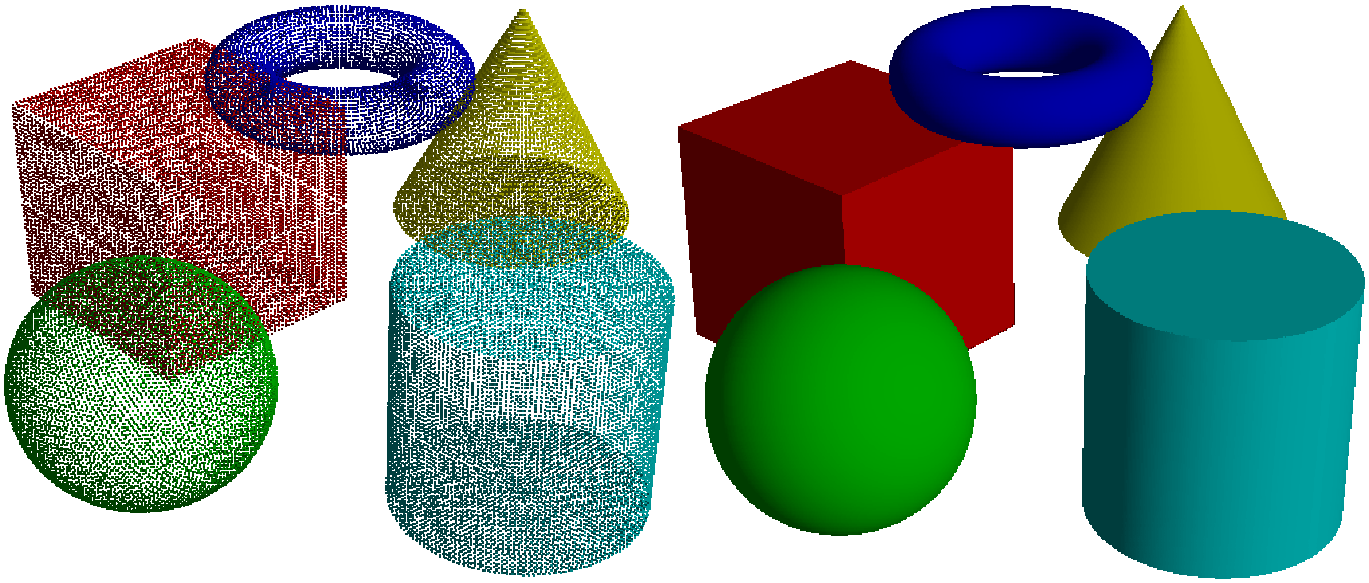
On se contentera, dans un premier temps, d'utiliser les transformations proposées par `OpenGl`, ce qui permettra de se concentrer sur la modélisation des objets canoniques.

Les formes demandées sont simples : la sphere de rayon 1, le cube de côté 2, le cylindre de rayon 1 et de hauteur 2 et, si possible, le cône et le tore.

Les seules difficultés à ce niveau concernent les formulations (basées sur les équations paramétriques) donnant les distributions de points sur les surfaces : il faut en effet essayer de répartir ces points de manière plus ou moins homogène pour obtenir un bon rendu. Des techniques très sophistiquées existent pour réaliser celà, mais on peut s'en tirer très simplement avec un peu d'intuition.

De même, il peut être bon de disposer d'un *facteur de zoom* pour tenir compte des tailles relatives des objets dans la scène : un cube et une sphère de même taille devront présenter des résolutions voisines et inversement une grosse sphère nécessitera plus de points qu'une petite... A vous d'être astucieux (économe et efficace).

Une piste est de construire chaque objet canonique sous la forme d'un gros tableau de points. La lecture de la totalité des points correspond à la résolution maximale, et/ou à l'exemplaire le plus gros de l'objet. Un exemplaire plus petit (ou plus loin) sera affiché en sous-échantillonnant le tableau (un point sur deux...). L'idéal est que la "densité" de points reste à peu prés constante quel que soit l'objet représenté (un cube, une sphère), sa déformation, sa taille, sa position... (cette amélioration est la plus difficile – à garder pour la fin).



les objets de base en basse et haute résolution

Constructive Solid Geometry

La seconde étape de ce projet consiste en la mise en œuvre de la technique des arbres CSG. Le principe est de créer des objets *composites* formés par l'association, via des opérateurs booléens, de plusieurs objets simples. Les opérateurs disponibles sont l'*union*, l'*intersection* et la *soustraction*. On formera ainsi des arbres binaires dont les nœuds internes sont des opérateurs et les feuilles des objets canoniques.

Les opérateurs ont pour rôle de définir quelles parties de chaque objet est toujours visible et quelle partie doit être éliminée. La valeur booléenne associée à l'opérateur doit donc nous indiquer si l'on se trouve à l'intérieur ou à l'extérieur d'un objet. Cela se fera grâce à une fonction de distance spécifique associée à l'objet canonique (une méthode contientpoint(M) retournant une valeur booléenne).

A chaque nœud ou feuille sont associées des matrices de transformation permettant de déformer, orienter et positionner les objets. Ces transformations sont propagées aux nœuds fils de sorte que lorsque l'on translate un nœud, cette translation s'applique à tous les objets situés dans ce sous-arbre.

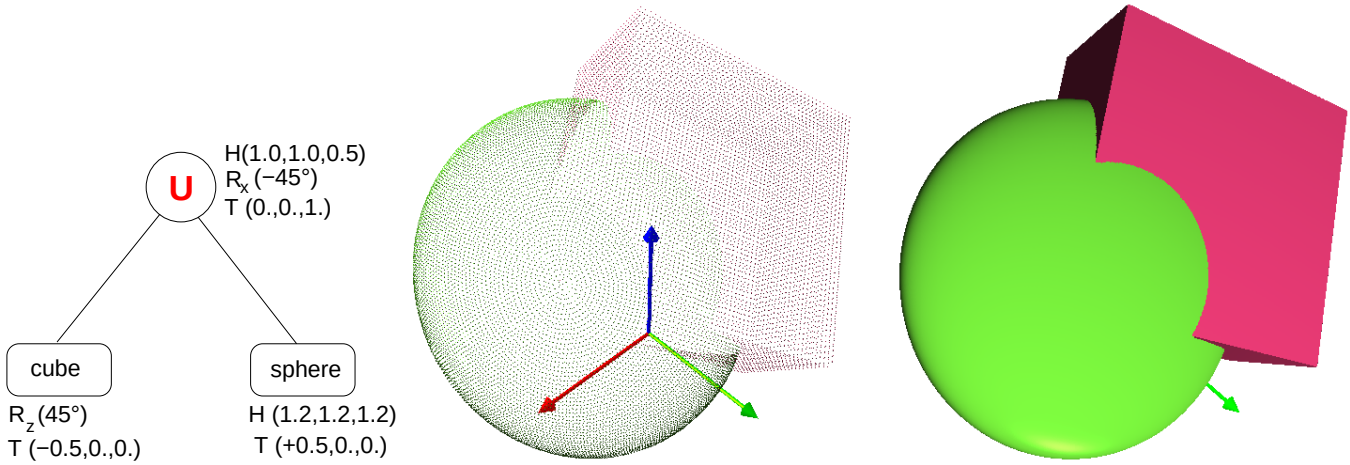
Les opérateurs

L'union

C'est l'opérateur le plus simple : l'union de deux objets $\mathcal{A}$ et $\mathcal{B}$ est formée par l'ensemble des points de $\mathcal{A}$ situés à l'extérieur de B et l'ensemble des points de $\mathcal{B}$ situés à l'extérieur de $\mathcal{A}$.

Pour créer concrètement l'union de deux objets $\mathcal{A}$ et $\mathcal{B}$ il faut :

① appliquer les transformations locales (celles compilées au niveau des nœuds feuilles) aux objets. Cette opération positionne les points de chaque objet relativement à l'autre objet.
② pour chaque point de l'objet $\mathcal{A}$ tester sa position (cf. plus loin) par rapport à $\mathcal{B}$:
⇒ si il est à l'**ext**érieur il est conservé en l'état, sinon il est éliminé.
③ faire de même pour chaque point de $\mathcal{B}$, par rapport à $\mathcal{A}$.



union d'un cube et d'une sphère : arbre, basse et haute résolution

L'intersection

L'intersection de deux objets $\mathcal{A}$ et $\mathcal{B}$ est formée par l'ensemble des points de $\mathcal{A}$ appartenant à B et l'ensemble des points de $\mathcal{B}$ appartenant à $\mathcal{A}$.

Pour créer concrètement l'intersection de deux objets $\mathcal{A}$ et $\mathcal{B}$ il faut :

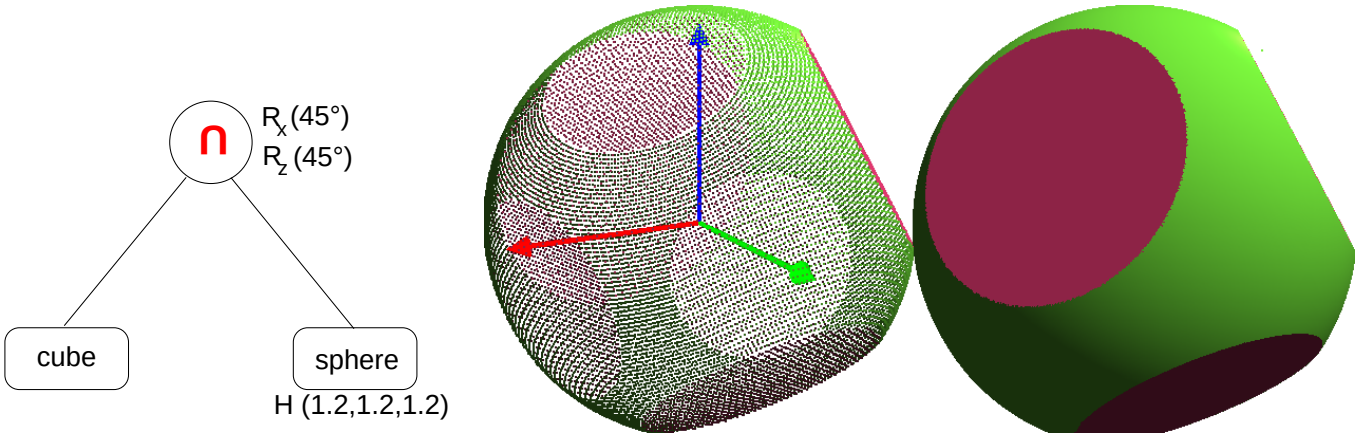
① appliquer les transformations locales aux objets.
② pour chaque point de l'objet $\mathcal{A}$ tester sa position par rapport à $\mathcal{B}$:
⇒ si il est à l'**int**érieur il est conservé sinon il est éliminé.
③ faire de même pour chaque point de $\mathcal{B}$, par rapport à $\mathcal{A}$.

La Soustraction

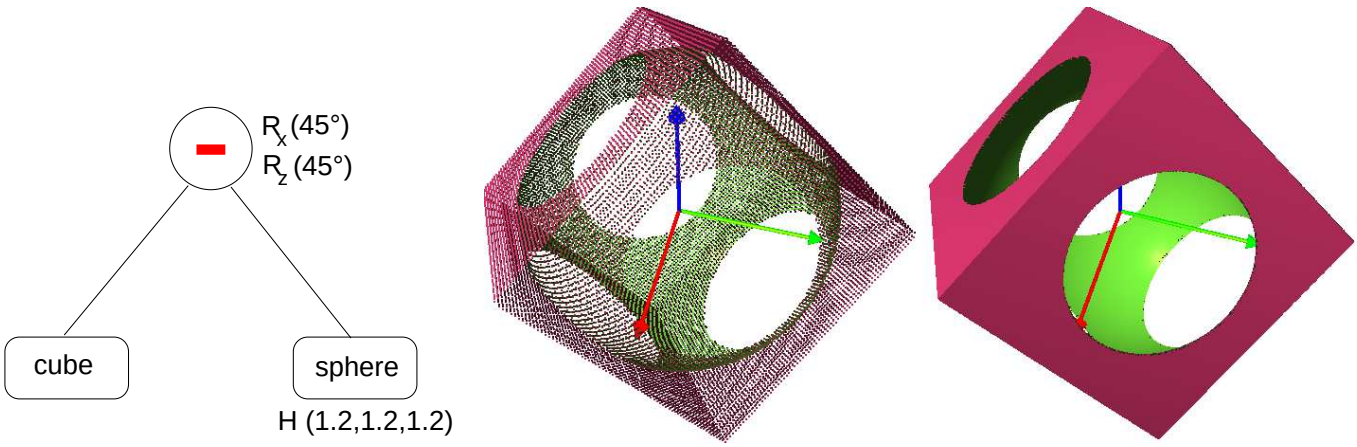
Enfin, le plus complexe des trois opérateurs : la soustraction de deux objets $\mathcal{A}$ et $\mathcal{B}$ est formée par l'ensemble des points de $\mathcal{A}$ n'appartenant à $\mathcal{B}$ et l'ensemble des points de $\mathcal{B}$ appartenant à $\mathcal{A}$.

Pour créer concrètement la soustraction de deux objets $\mathcal{A}$ et $\mathcal{B}$ il faut :

① appliquer les transformations locales aux objets.
② pour chaque point de l'objet $\mathcal{A}$ tester sa position par rapport à $\mathcal{B}$:
⇒ si il est à l'**ext**érieur il est conservé sinon il est éliminé.
③ pour chaque point de l'objet $\mathcal{B}$ tester sa position par rapport à $\mathcal{A}$:
⇒ si il est à l'**int**érieur il est conservé sinon il est éliminé.
④ l'objet $\mathcal{B}$ étant vue *en négatif*, il faudra, avant de l'afficher, penser à inverser ses normales.



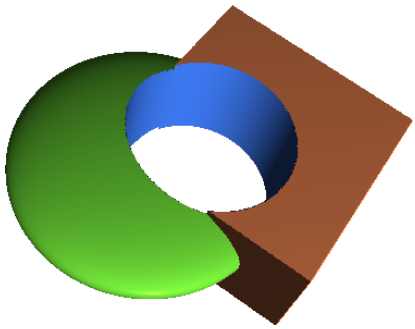
intersection d'un cube et d'une sphère : arbre, basse et haute résolution



soustraction d'un cube et d'une sphère : arbre, basse et haute résolution

Remarque :
Lorsqu'un même objet $\mathcal{C}$ est soustrait à plusieurs objets $\mathcal{A}$ et $\mathcal{B}$ (par exemple une soustraction sur une union, comme dans l'exemple précédent) un problème apparait :

- lors de la première soustraction $(\mathcal{A} - \mathcal{C})$ tous les points de $\mathcal{C}$ situés hors de $\mathcal{A}$ sont (virtuellement) éliminés alors que certains peuvent se trouver dans $\mathcal{B}$ et devront donc réapparaître lors de la seconde soustraction $(\mathcal{B} - \mathcal{C})$.
- de même lors de cette seconde soustraction, il ne faudra pas éliminer les points de $(\mathcal{C})$ déjà sélectionnés lors de la première soustraction.



soustraction d'un cylindre sur l'union d'un cube et d'une sphère

Mise en œuvre

L'application des ces opérateurs sur les différents objets est basée sur la définition, pour chaque objet canonique, d'une *fonction d'appartenance*, assimilable à une distance, permettant de tester si un point P est à l'intérieur ou à l'extérieur de l'objet.

- Dans le cas de la sphère $\mathcal{S}_O$, cette fonction est trivialement basée sur la distance euclidienne : $P(x, y, z) \in \mathcal{S}_O \Leftrightarrow (x^2 + y^2 + z^2) \leq 1$.
- Dans le cas du cube $\mathcal{C}_O$, on utlisera la *norme du max* : $d(O, P) = Max(|x|, |y|, |z|)$.
- Pour les autres objets (cylindre et tore), il faudra chercher un peu...

En pratique, pour déterminer si un point P est à l'intérieur ou à l'extérieur d'un objet $\mathcal{A}$ il faudra ramener P dans le répère de l'objet canonique $\mathcal{A}_0$, via la matrice de transformation inverse de $\mathcal{A}$ – on obtient un point Q – pour ensuite tester la position de Q par rapport à $\mathcal{A}_0$.

Rendu / évaluation

Ce projet sera à rendre à la fin du semestre.

Cette évaluation portera sur
- le nombre et la qualité des objets canoniques modélisés
- la structuration et la modularité des graphes de scènes
- la qualité et la pertinence des exemples proposés (complexité des scènes, animations éventuelles)

Un court rapport est demandé, dans lequel vous détaillerez la modélisation des objets, le principe de construction de vos graphes de scène, les méthodes associées aux objets canoniques, l'architecture globale du projet, ainsi que les procédures de compilation et exécution.

Exemples en vrac