
TP Surfaces Implicites

Tom CLABAULT - p2205453

14 octobre 2023

1 Surfaces implicites

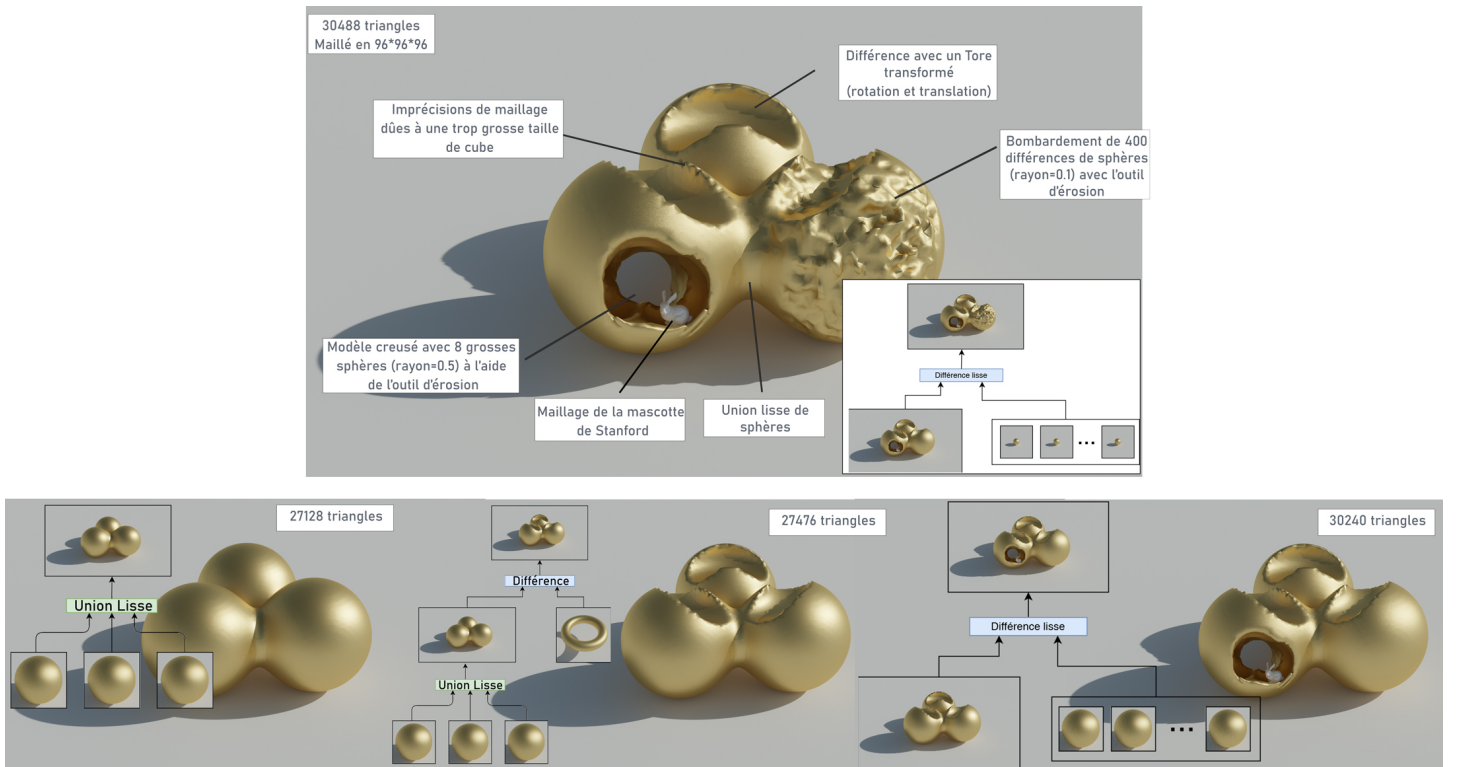


FIGURE 1 – Rendu Blender d'une surface implicite modélisée et maillée avec TinyMesh

Les opérateurs binaires de modélisation ont été implémentés au moyen de 7 classes, d'héritage de fonctions virtuelles en C++.

La Figure1 montre les interactions possibles avec différents opérateurs binaires de modélisation. Le temps nécessaire au maillage (ou au rendu par ray-marching) du modèle dépend directement de la complexité de l'arbre de construction. Pour cet exemple, le plus gros consommateur de temps de calcul est l'outil d'érosion qui peut rajouter jusqu'à 100 différences de sphères à la SDF à chaque clic. Chaque appel à la fonction *Value()* de la SDF doit alors prendre en compte toutes ces nouvelles sphères.

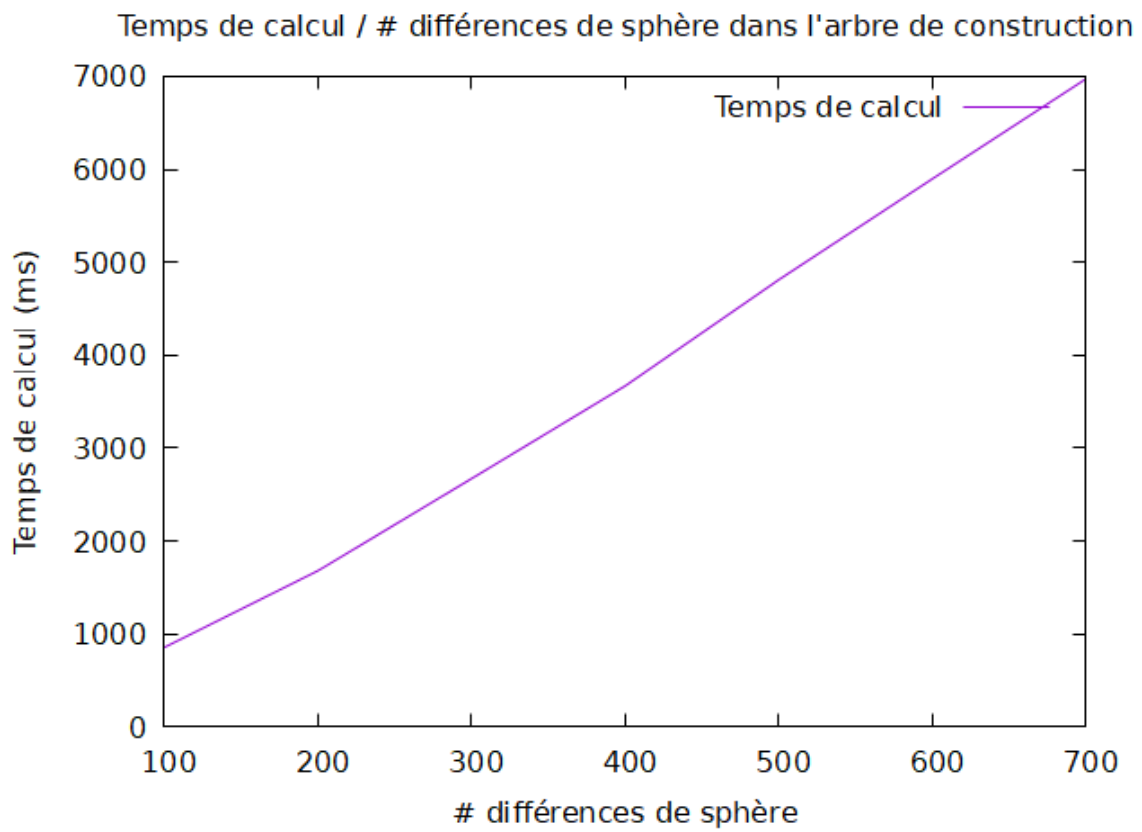


FIGURE 2 – Temps d’exécution de l’algorithme de maillage de la surface implicite en fonction de la complexité de la SDF (nombre de différences de sphère utilisé dans la SDF)

1.1 Outil d’érosion

L’outil d’érosion a été implémenté grâce à un algorithme de ray marching. On lance des rayons aléatoirement autour de la position où l’utilisateur a cliqué et on rajoute des différences de sphère à la SDF aux points d’intersection.

L’algorithme de ray marching faisant appel à la fonction *Value()* de la SDF, il est de moins en moins efficace (linéairement) au fur et à mesure que la SDF se complexifie. Un lancer de rayons classique + BVH sur le maillage de la surface implicite se révèle être 3 ordres de grandeur plus rapide mais cela nécessite cependant d’avoir maillé la surface implicite au préalable.