

*UNIVERSITES PAUL CEZANNE, de PROVENCE, de la MEDITERRANEE
UNIVERSITES du SUD TOULON-VAR
ECOLE NATIONALE SUPERIEURE d'ARTS et METIERS*

MASTER SIS
Sciences de l'Information et des Systèmes

Rapport de Stage

présenté le lundi 20 juin 2005 par
Guillaume THIBAUT

Sujet

Outils de déformation pour la Sculpture Virtuelle

Stage de recherche effectué au laboratoire : LSIS
Dans l'équipe : LXAO
Sous la direction de : Romain RAFFIN

Remerciements

Je souhaite particulièrement remercier Monsieur Romain RAFFIN pour son encadrement. Monsieur RAFFIN a su être présent, me conseiller et m'orienter chaque fois que je l'ai sollicité.

Merci à Monsieur Jean-Louis MALTRET pour son conseil.

Table des matières

1	Introduction	5
1.1	Objectif	5
1.2	Les problématiques	5
I	Etat de l'art	6
2	Modèles de représentation des objets : pièce et outil	6
2.1	Les facettes	6
2.2	Surfaces de subdivision	7
2.3	Les surfaces implicites	8
2.4	Le modèle CSG	10
2.5	La voxelisation	11
2.5.1	Compression des données : l'octree	12
2.5.2	Amélioration de rendu : surfaces de subdivision	13
2.6	Conclusion et choix des méthodes	13
3	Les méthodes de déformation	15
3.1	Modèles physiques	15
3.2	Déformation de surfaces implicites : sculpture du bois	16
3.3	Les Free Form Deformation (FFD)	17
3.4	SCODEF	18
3.5	Déformations libres sous contraintes	19
3.6	Esquisser le résultat	20
3.7	Méthode de Jörg Ayasse	20
3.8	Conclusion	21
II	Le Stage	22
4	Définitions	22
5	Amélioration algorithmique de la méthode de Jörg Ayasse	23
6	Notre outil de déformation	25
6.1	Extension de la voxelisation	25
6.2	Utilisations des normales pour retrouver les propriétés de l'outil	25
6.3	Déformation de la pièce par déplacement de matière	27
6.4	Simuler la viscosité	29
7	Conclusion	33
8	Améliorations possibles	33

Table des figures

1	Un exemple de représentation par facettes [34]	6
2	Stockage des informations lors de la représentation par facettes [17]	7
3	Méthode DCEL [17]	7
4	Un exemple récent de surfaces de subdivision : Némó, image PIXAR [4]	7
5	Subdivision d'un tétraèdre [18, 19]	8
6	Représentation d'une iso surface [26]	8
7	Exemples de quadriques [26]	9
8	Influences mutuelles de deux surfaces implicites [32]	9
9	Construction d'un squelette [27]	9
10	Surface implicite à squelette [20]	10
11	Quelques opérations ensemblistes utilisées par CSG. Réalisé avec POV-RAY [5]	11
12	Représentation d'un objet complexe avec CSG [19]	11
13	Exemple de discrétisation du "Stanford Bunny" [27]	12
14	Compression d'une surface par quadtree [2]	12
15	Compression d'un volume par octree [37]	13
16	Ajout d'une action de subdivision à un objet discrétisé [27]	13
17	Représentation des liaisons masse-ressort [22]	15
18	Résultats de déformation avec un modèle physique [22]	15
19	Exemples de résultats obtenus à l'aide de [29]	16
20	Déformation à l'aide de FFD [32]	17
21	AxDF, déformation par rapport à un axe [32]	17
22	EFFD, maillage non parallélépipédique [32]	18
23	Méthode SCODEF [32]	18
24	Déformations libres sous contraintes [32]	19
25	Déformation suivant une courbe de contrainte [32]	19
26	Extrusion de forme par simple dessin	20
27	Ajout et suppression de matière [6]	20
28	Sélection des voxels utiles [6]	21
29	Représentation des différentes phases de la déformation	22
30	Sphère de rayon 1 et sa partie utile pour un déplacement sur l'axe Z (en vert) .	23
31	Les angles de rotation du point de vue	24
32	Rotations identiques de la pièce et du point de vue en 2D avec vérification dans le buffer	24
33	Rotations identiques de la pièce et du point de vue	24
34	Utilité des normales dans la déformation	26
35	Calcul des normales à partir des voxels	26
36	Les normales (à droite) issues de la partie utile (à gauche)	27
37	Notre méthode de déformation	27
38	Résultat de la déformation à l'aide d'une sphère (figure 47) : à gauche, zone de pénétration de l'outil, à droite partie arrière	28
39	Problème du résultat de la déformation à l'aide d'un cube	28
40	Effets de la viscosité	29
41	Courbe de Bézier simulant la viscosité : la quantité de matière (axe Y) décroît en fonction de la distance (axe X)	29
42	Courbe de Bézier représentant une viscosité de 0.5 sur une zone d'influence de 5	30

43	Résultat de la déformation à l'aide d'un cube avec une notion de viscosité définie par la courbe de la figure 42	30
44	Résultat de la déformation à l'aide d'une sphère avec la notion de viscosité . . .	31
45	Résultat de la déformation avec un cube et une notion de viscosité faible	31
46	Résultat de la déformation avec une sphère (figure 47) et une notion de viscosité faible	32
47	Sphère de rayon 15, sa partie utile et les normales	34
48	Une pointe, sa partie utile et les normales	34
49	Un outil plus complexe en forme de croix, sa partie utile et les normales	35
50	Résultat de la déformation à l'aide d'une pointe (figure 44) et une viscosité nulle	35
51	Résultat de la déformation à l'aide d'une croix (figure 45) et une viscosité nulle	36
52	Résultat de la déformation avec une pointe (figure 44) et une viscosité importante	36
53	Résultat de la déformation avec une croix (figure 45) et une viscosité importante	36

1 Introduction

1.1 Objectif

Ce stage tutoré par Romain RAFFIN, s'inscrit dans le cadre de la réalité virtuelle et tout particulièrement en sculpture virtuelle. La réalité virtuelle consiste à faire entrer une personne dans un environnement de synthèse dans lequel il aura des sensations réelles et avec lequel il pourra interagir. La sculpture virtuelle doit donc fournir à l'utilisateur un environnement où il pourra agir sur des objets de manière intuitive et réelle (déformations de l'objet) à l'aide d'une palette d'outils pré-défini¹. Le sculpteur pourra également ressentir les effets physiques des différentes manipulations (résistance de la matière) sous forme de retours d'efforts. Nous souhaitons ainsi mettre en œuvre une méthode de déformation suffisamment générique pour qu'elle s'adapte à toutes les formes manipulées, capable d'accueillir un maximum d'outils de déformation et par la suite d'en fournir de nouveaux. Nous souhaitons également que l'utilisateur puisse lui-même créer ces outils afin de travailler selon son inspiration. Afin de se rapprocher au maximum de la réalité, une contrainte importante vient s'ajouter : notre système doit être interactif.

1.2 Les problématiques

Avant de commencer à sculpter, il faut définir comment représenter et stocker les informations relatives aux objets (pièces et outils) de topologie quelconque. En effet, il existe déjà plusieurs méthodes qui offrent chacune avantages et inconvénients dans des cas bien précis. La contrainte de l'interactivité du système impose un accès et une manipulation extrêmement rapide de nos données. Ensuite il nous faudra choisir parmi plusieurs méthodes de déformation déjà existantes afin de travailler sur l'outil ou la pièce elle-même.

¹Ceux de l'artisan sculpteur et les nouveaux que permettra le monde virtuel

Première partie

Etat de l'art

2 Modèles de représentation des objets : pièce et outil

2.1 Les facettes

La modélisation par facette est une représentation surfacique des objets. Elle est réalisée de la façon suivante :

1. Discretiser la surface de l'objet en n sommets (plus il y a de sommets, plus on se rapproche de la surface initiale).
2. Relier les sommets deux à deux afin de définir les arêtes (les arêtes devant suivre la surface sinon nous perdons la topologie de l'objet).
3. Regrouper les arêtes par face (une arête pouvant appartenir à plusieurs faces).

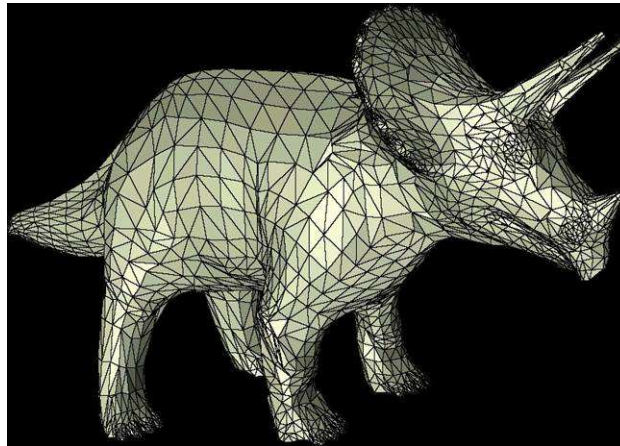


FIG. 1 – Un exemple de représentation par facettes [34]

L'objet d'origine est ainsi transformé en un parallélépipède quelconque (figure 1). Les faces triangulaires sont le plus souvent utilisées pour leur simplicité et une manipulation facile. Le principal problème de cette méthode va être le stockage de toutes les informations : sommets, arêtes et faces. Il y a ainsi un grand nombre de redondances, qui imposent une place importante en espace mémoire et de nombreux calculs de mise à jour si on manipule la surface. Ceci est mis en évidence dans la figure 2 où l'on s'aperçoit que chaque sommet apparaît de nombreuses fois pour un simple tétraèdre.

Une solution à toutefois été proposée pour manipuler plus rapidement cette méthode : la méthode DCEL (Doubly Connecting Edge List). Il s'agit d'une table des arêtes qui contient toutes les informations (cf figure 3).

Malgré l'utilisation de la méthode DCEL, l'accès aux données et leurs manipulations restent trop coûteux et rendent difficile cette méthode dans un système à temps réel. De plus, le rendu demeure assez pauvre mais peut être amélioré par l'utilisation de surfaces de subdivision (définies ci-après section 2.2).

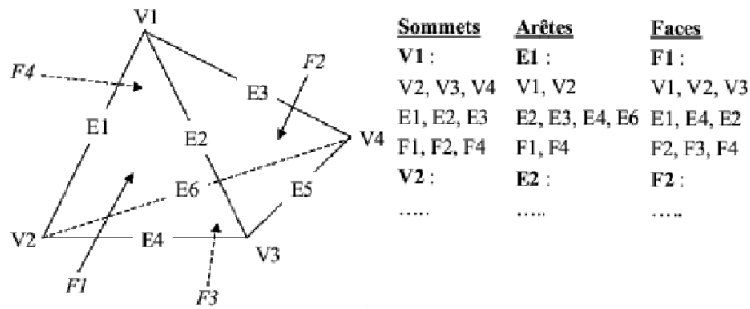


FIG. 2 – Stockage des informations lors de la représentation par facettes [17]

Arêtes	Sommets		Faces		Traverse gauche		Traverse droite	
	départ	arrivée	gauche	droite	préc.	succ.	préc.	succ.
a	A	D	3	1	e	f	b	c
b	A	B	1	4	c	a	f	d
c	B	D	1	2	a	b	d	e
d	B	C	2	4	e	c	b	f
e	C	D	2	3	c	d	f	a
f	A	C	4	3	d	b	a	e

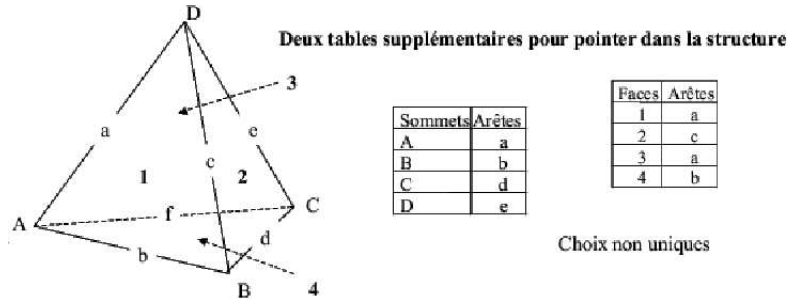


FIG. 3 – Méthode DCEL [17]

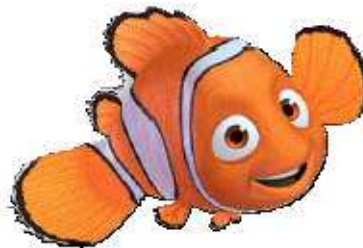


FIG. 4 – Un exemple récent de surfaces de subdivision : Némó, image PIXAR [4]

2.2 Surfaces de subdivision

Les surfaces de subdivision sont apparues en 1978 suite aux travaux de Catmull et Clark [13]. On utilise le maillage initial de l'objet (maillage quelconque de la surface) et on applique des règles de subdivision compatibles avec le maillage : $S_{i+1} = F(S_i)$ avec S_i la surface après i itérations et F la fonction de subdivision. On obtient ainsi un nouveau maillage avec plus de points, qui doit converger vers une surface lisse S (figure 5) :

$$\lim_{i \rightarrow \infty} S_i = S$$

Les surfaces de subdivision permettent de manipuler des objets très simples, tout en obtenant un excellent rendu. Il s'agit donc plus d'une technique d'amélioration du rendu que de modélisation. Elles souffrent toutefois de quelques défauts :

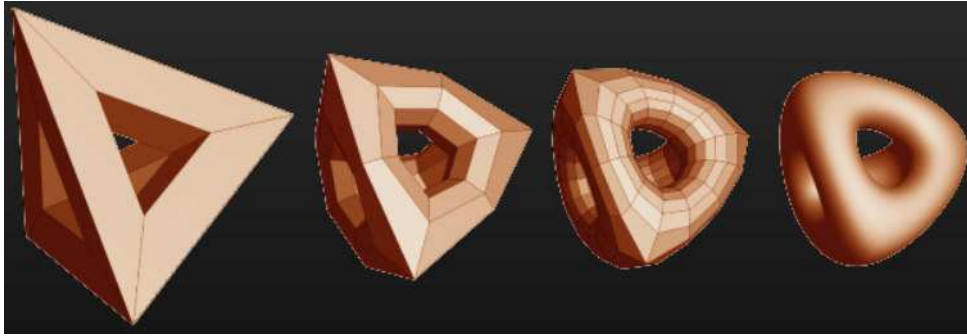


FIG. 5 – Subdivision d’un tétraèdre [18, 19]

- Risque de perte de la topologie du maillage après quelques itérations.
- Risque de saut de la résolution entre deux itérations.
- Elles n’ont aucune propriété différentielle.
- On manipule presque jamais directement la surface : certains algorithmes le permettent mais ils sont difficiles à implémenter.

Les surfaces de subdivision peuvent être utilisées afin d’améliorer le rendu lors d’une voxélisation. Cela permettrait de travailler sur des objets simples, allégés de leur aspect visuel, pour se concentrer sur les déformations.

2.3 Les surfaces implicites

Les fonctions implicites sont étudiées depuis longtemps en mathématiques, mais en dimensions quelconques. Plus récemment il y a eu une restriction à \mathbb{R}^3 pour la modélisation de formes. Elles connaissent un véritable essor depuis une vingtaine d’années, surtout depuis les travaux de [10, 33]. Contrairement aux modèles paramétriques qui explicitent les coordonnées des points, le formalisme implicite définit une surface comme un ensemble de points de l’espace vérifiant une propriété. Cette propriété est généralement liée à la valeur prise en ses points par une fonction $F : \mathbb{R} \rightarrow \mathbb{R}^3$, nommée fonction implicite, champ scalaire ou encore fonction de potentiel (analogie avec la physique). Une surface est donc définie comme une iso surface de niveau iso fixé par la fonction F (cf figure 6) :

$$S = \{P \in \mathbb{R}^3 / F(P) = iso\} = F^{-1}(iso)$$

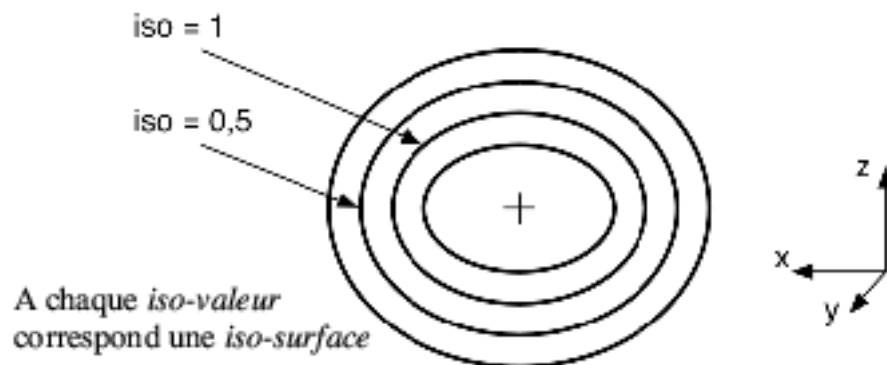


FIG. 6 – Représentation d’une iso surface [26]

Il est donc très facile de connaître la position d’un point p par rapport à la surface :

- $F(p) < iso$: le point est dans la surface
- $F(p) = iso$: le point est sur la surface
- $F(p) > iso$: le point est hors de la surface

Voilà quelques exemples d'iso surfaces (figure 7) définies à partir d'une équation de super quadriques [7] :

$$F(p) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}$$

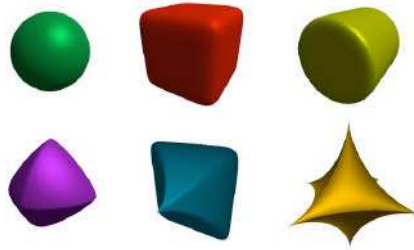


FIG. 7 – Exemples de quadriques [26]

Le passage du cube à la sphère se fait donc de manière continue par simple variations de ϵ_1 et ϵ_2 . Dans les surfaces implicites, il est ajouté à la notion de surface, un centre et une zone d'influence, afin de pouvoir combiner plusieurs surfaces (cf figure 8).

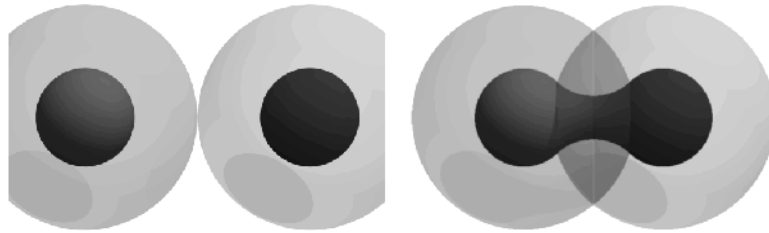


FIG. 8 – Influences mutuelles de deux surfaces implicites [32]

En sculpture virtuelle, la formalisation de surfaces implicites à base de squelette [11] est la plus souvent utilisée ([19, 20, 29]). Le squelette d'un objet est l'ensemble des centres des boules maximales contenues dans l'objet ; dans la figure 9, calculé pour un objet de dimension deux.

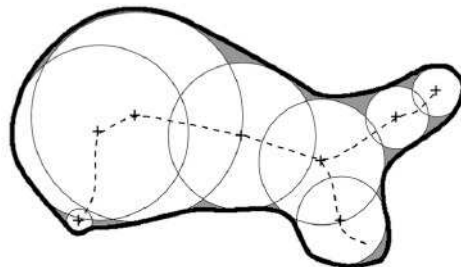


FIG. 9 – Construction d'un squelette [27]

Les surfaces sont définies par la sommation de potentiels locaux (les fonctions potentielles) et générées par les sources ponctuelles (les points du squelette). Nous trouvons différentes appellations pour les surfaces implicites à base de squelette :

- Modèles à blobs [10]
- Soft objects [39]
- Techniques par méta boules [30]

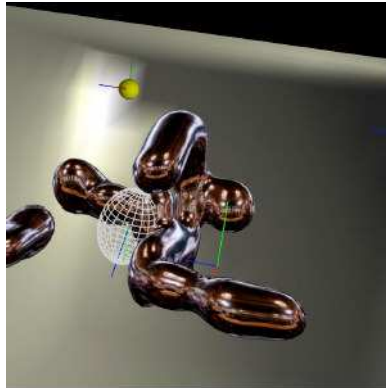


FIG. 10 – Surface implicite à squelette [20]

Les surfaces implicites ont un formalisme très puissant, notamment grâce au squelette qui permet de créer des objets de topologie quelconque en gardant un bon contrôle de la forme globale. La structure de données est compacte et permet un calcul immédiat de la notion intérieur / extérieur de l'objet et de la zone d'influence. Toutefois, les surfaces implicites présentent aussi des défauts importants :

- Les formes définies sont toutes arrondies et donc ne conviennent pas à tous les objets.
- La visualisation est coûteuse : il faut calculer tous les points de l'espace (par rapport à l'équation implicite) pour savoir s'ils sont dans l'objet, visible ou masqué.
- On ne peut se déplacer sur la surface de manière aisée : tester si les points proches sont sur la surface.
- Le contrôle local de la surface est difficile : variations de l'équation de la surface et de la zone d'influence.

Ces difficultés de contrôles et d'affichages gênent fortement la manipulation et ralentissent le système. Les surfaces implicites sont ainsi délicates à utiliser dans un système interactif et risquent de limiter les possibilités [29].

2.4 Le modèle CSG

Le modèle CSG (Constructive Solid Modeling) est un système de manipulations et de combinaisons de volumes prédéfinis (à l'aide de différents systèmes : facettes, surfaces implicites...). Il est défini à partir des trois ensembles suivants :

- Des primitives géométriques : sphère, cylindre, parallélépipède rectangle...
- Des transformations géométriques : rotation, translation, homothétie.
- Des opérateurs ensemblistes : union, intersection, différence...

Un objet défini par un modèle CSG est une composition des trois ensembles, dont la suite des manipulations est stockée sous forme d'arbre (cf figure 12).

Il est possible de créer des objets complexes mais qui dépendront des primitives de base. Toutefois, en sculpture virtuelle, chaque modification demandera une mise à jour de l'arbre (coûteuse) et le temps d'accès aux données est important. Elle est donc difficile à mettre en place, lourde à maintenir, mais quand même utilisée dans [29].

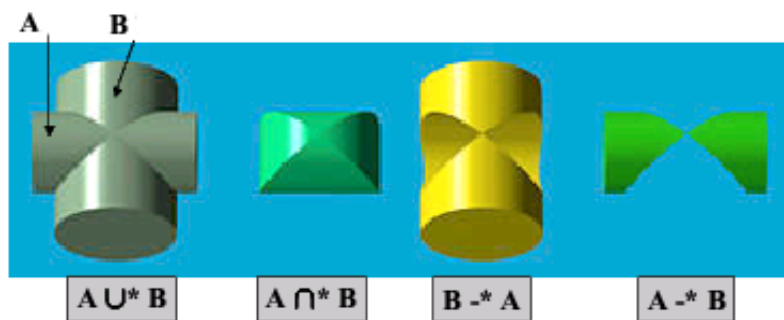


FIG. 11 – Quelques opérations ensemblistes utilisées par CSG. Réalisé avec POV-RAY [5]

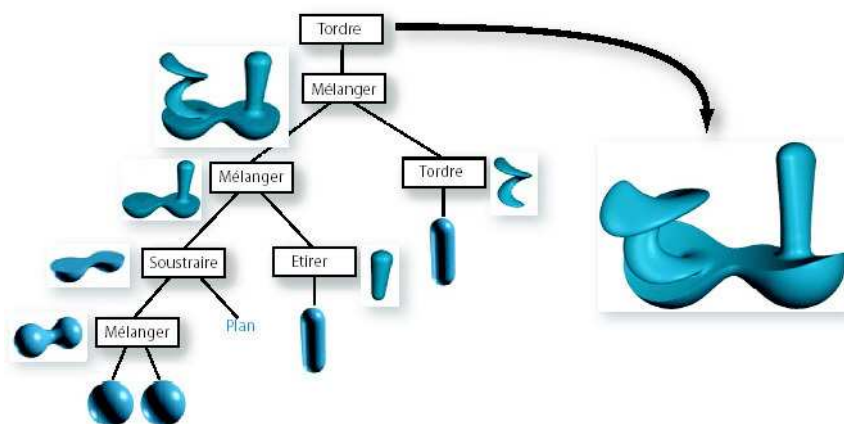


FIG. 12 – Représentation d'un objet complexe avec CSG [19]

2.5 La voxelisation

On vient de voir principalement deux méthodes de représentations par surfaces qui ont montré leurs limites. La voxelisation est la méthode la plus simple et la plus intuitive à base de volumes. L'espace de travail est discrétisé à l'aide de cube de taille identiques appelés "voxel"². Chaque voxel est repéré dans l'espace grâce au système de coordonnées paramétriques (x, y, z) ³ et peut prendre une valeur binaire :

- 1 le voxel appartient à l'objet.
- 0 il n'appartient pas.

Les voxels sont généralement stockés sous forme de matrice à trois dimensions. D'autres informations peuvent être ajoutées au voxel : couleur, données physiques... On peut donc représenter tout type d'objet de topologie quelconque et se déplacer sur la surface grâce à la notion de voisinage très naturelle dans ce système. Mais le système souffre de deux problèmes majeurs :

- L'espace requis pour stocker les voxels est important : si on prend un "petit" espace 100^3 , il faudra une matrice comportant 10^6 de cellules.
- Lors d'une voxelisation, les objets sont discrétisés ; on introduit donc une erreur d'approximation dès l'étape de modélisation (cf figure 13).
- L'affichage d'un objet discrétisé est particulièrement désastreux (cf figure 13).

La simplicité de cette méthode engendre quelques problèmes auxquels nous apportons une solution dans les sous-sections suivantes (2.5.1 et 2.5.2).

²le volume élément (un pixel en dimension 3)

³peuvent aussi être entières (i, j, k)

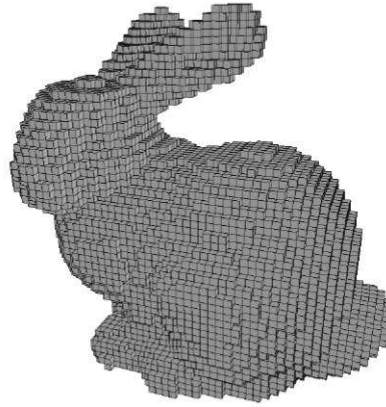


FIG. 13 – Exemple de discrétisation du "Stanford Bunny" [27]

2.5.1 Compression des données : l'octree

Un octree est un système de stockage qui permet de compresser les données lors d'une voxelisation. Il se présente sous la forme d'un arbre dont chaque nœud possède soit huit fils, soit aucun.

Le principe :

- Diviser l'espace de travail en huit (car on est en 3D).
- Pour chacun des huit sous-espaces, il faut regarder son contenu. S'il est totalement vide (marquer le nœud vide) ou totalement rempli par l'objet (marquer le nœud plein) alors s'arrêter, sinon subdiviser le sous-espace en huit et recommencer les tests.

Voici une représentation en dimension deux (donc à l'aide de "quadtree" et par conséquent une division de l'espace en 4) :

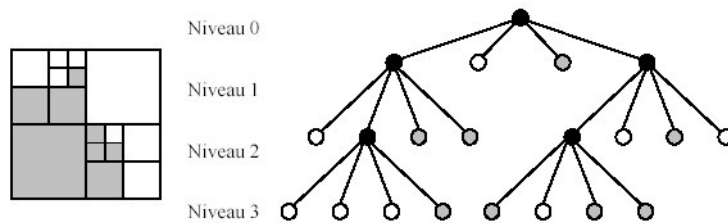


FIG. 14 – Compression d'une surface par quadtree [2]

Dans l'exemple de la figure 8, la taille de l'espace est de 64 pixels (8^2), dont 32 pour la surface grisée que l'on souhaite représenter sous forme de quadtree. Pour cela 21 sont nécessaires pour créer l'arbre mais seulement 8 pour stocker la surface grisée. Cette exemple met en évidence le gain en espace mémoire apporté par la méthode. La figure 15 est un exemple d'application d'un octree.

Ce système permet de diminuer considérablement la taille de l'espace mémoire requis pour stocker les objets lors d'une voxelisation. Toutefois il est absolument inutilisable dans le cas d'un nuage de points disjoints car chaque point devra être représenté par une feuille de l'arbre, donc il y aura encore plus de nœud dans l'arbre que de voxels dans l'objet. De plus, nous perdons toute les notions de voisinage ; dans le pire des cas, il faut parcourir toute l'arborescence pour trouver les voisins d'un voxel. Ce dernier inconvénient handicape fortement cette méthode nous insiste à ne pas la choisir dans le cadre de notre travail.

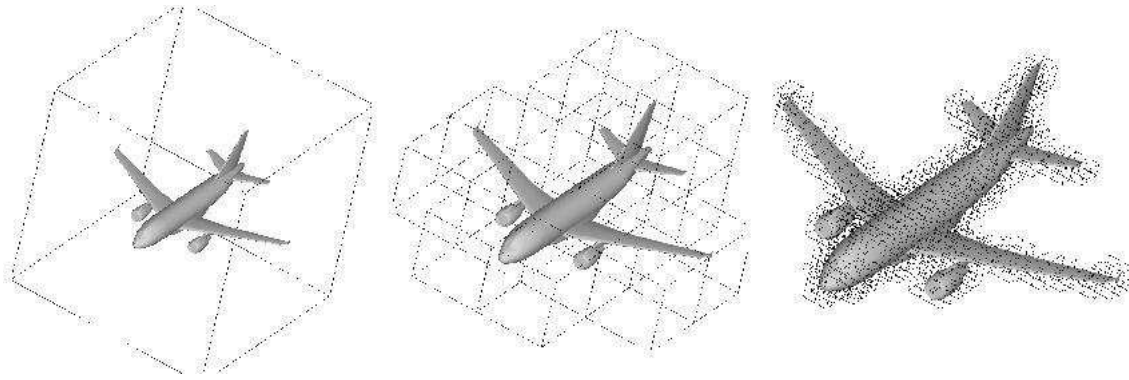


FIG. 15 – Compression d'un volume par octree [37]

2.5.2 Amélioration de rendu : surfaces de subdivision

Comme il a été dit en conclusion sur les surfaces de subdivision (section 2.2), elles peuvent être utilisées pour améliorer le rendu. Voici le "lapin" de la figure 13 après les opérations de subdivision (figure 16) :

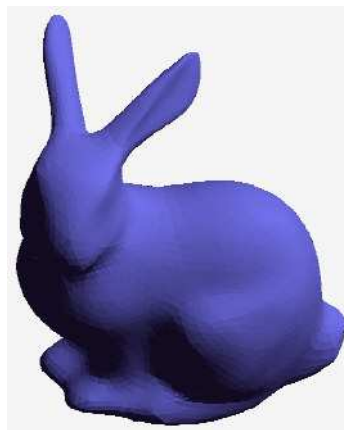


FIG. 16 – Ajout d'une action de subdivision à un objet discrétisé [27]

La méthode des "marching cube" [16] est aussi utilisée dans [6] mais nous ne l'aborderons pas ici.

2.6 Conclusion et choix des méthodes

Nous venons de présenter les principales méthodes de stockage des données, de la géométrie, en évoquant systématiquement leurs défauts et leurs avantages. Dans le cadre du stage, nous avons décidé de travailler par voxelisation, pour la simplicité de la méthode et les possibilités de maîtriser interactivement les objets. C'est avec une voxelisation que l'on obtient le mieux la représentation d'un volume (donc de l'objet et de l'outil) et de plus, la quasi inexistence de contraintes pendant la déformation correspond au souhait de développer une méthode générique. Deux pistes seront par ailleurs étudiées durant ce stage. La première consistera à ajouter un rendu par surfaces de subdivision pour améliorer l'aspect et tout particulièrement un algorithme de subdivision spécifique à chaque outil de sculpture afin d'afficher au mieux le résultat. De plus les surfaces de subdivision seront bientôt implémentées en "Hardware" dans les cartes vidéo, ce qui augmentera leur efficacité et diminuera leur coût d'utilisation. Dans la deuxième piste

l'objet serait représenté par un ensemble de cubes de taille non uniforme afin de diminuer le nombre de cubes à manipuler sans s'encombrer du formalisme des octrees. Mais il y a une perte immédiate de la donnée de voisinage.

3 Les méthodes de déformation

3.1 Modèles physiques

Ajouter un modèle physique à un système de sculpture virtuelle augmente grandement le réalisme. L'objet réagit comme un matériau réel lors de la déformation et résiste à la main du sculpteur (réponse haptique) lors du retour de force, en fonction de ses caractéristiques physiques. Mais dans les deux cas, le formalisme physique se traduit par la résolution d'une équation différentielle du second degré. Il est extrêmement difficile de résoudre des équations différentielles aussi bien formellement (moins de 5% des équations) que numériquement. Une méthode de résolution fréquemment utilisée est celle des "éléments finis" ([28, 8, 36]). Le résultat en terme de réalisme est excellent (déformation de la peau sur le corps [15, 25, 31] ou de vêtements sur une personne) mais le temps de calcul est désastreux. Toutefois, une équipe est parvenue à contourner partiellement le problème [22]. Pour cela, elle a utilisé un système de masse-ressort : chaque sommet est affecté d'une masse et relié à chacun de ses voisins par un ressort [22].

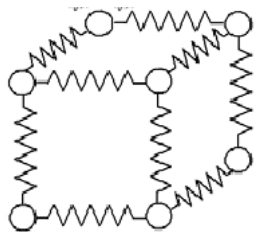


FIG. 17 – Représentation des liaisons masse-ressort [22]

Il faut donc résoudre une équation différentielle du second degré de la forme : $Md^{(2)} + Dd^{(1)} + Kd = F_d$. Avec M matrice des masses, D "damping" matrice, K matrice de raideur, F_d la fonction du second membre. Cette équation sera résolue par une méthode d'Euler explicite [1] :

$$d_{i+1}^{(1)} = d_i^{(1)} + hF(d_i^{(2)})$$

$$d_{i+1} = d_i + hF(d_i^{(1)})$$

Les méthodes d'Euler sont très rapides pour un pas h bien choisi et faciles à mettre en place. Mais elles connaissent de gros problèmes d'instabilité et de convergence vers la solution. Elles sont déconseillées dans la plupart des cas au profit des méthodes Runge Kutta [3] d'ordre deux et quatre, qui sont cependant trop lourdes en temps de calcul pour obtenir du temps réel.

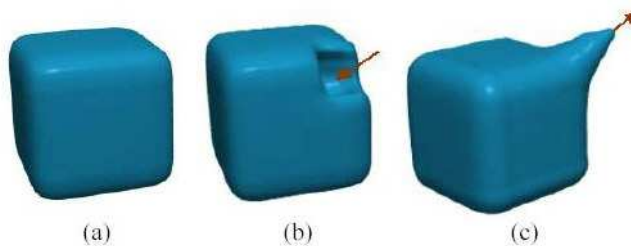


FIG. 18 – Résultats de déformation avec un modèle physique [22]

Une fois la déformation accomplie, il faut réinitialiser les ressorts pour remettre l'objet dans un état stable, sans quoi il reviendrait à sa forme stable précédente. Cette méthode offre l'avantage de posséder un retour haptique mais diminue le nombre d'outils de déformation disponible à cause de son modèle physique. Le point faible essentiel de ce modèle restera l'utilisation de la méthode d'Euler qui introduit tous ces défauts mathématiques.

3.2 Déformation de surfaces implicites : sculpture du bois

Mizuno et al [9], présentent en 1999 une méthode dont les résultats simulent parfaitement la sculpture du bois. Ils utilisent des surfaces implicites dont les modifications sont stockées par la méthode CSG. Les ciseaux à bois ont une forme courbe, qui peut être apparentée à une portion d'ellipse. Par conséquent, l'outil de sculpture est un ellipsoïde que l'on déplacera sur l'objet à sculpter (quelques ciseaux sont aussi cubiques, donc simulés par des parallélépipèdes cubiques). Avec la formulation implicite, il est aisé de calculer les intersections entre l'objet et l'outil (égalité entre deux fonctions implicites). Pour afficher le résultat il faut connaître toutes les arêtes visibles créées par l'outil. Elles sont donc stockées au fur et à mesure dans chaque nœud de l'arbre CSG. Or, il est fréquent de repasser deux fois au même endroit en sculptant ; il faut donc remettre à jour l'arbre CSG après chaque action de l'outil, car certaines arêtes disparaissent totalement ou partiellement et d'autres sont alors créées.

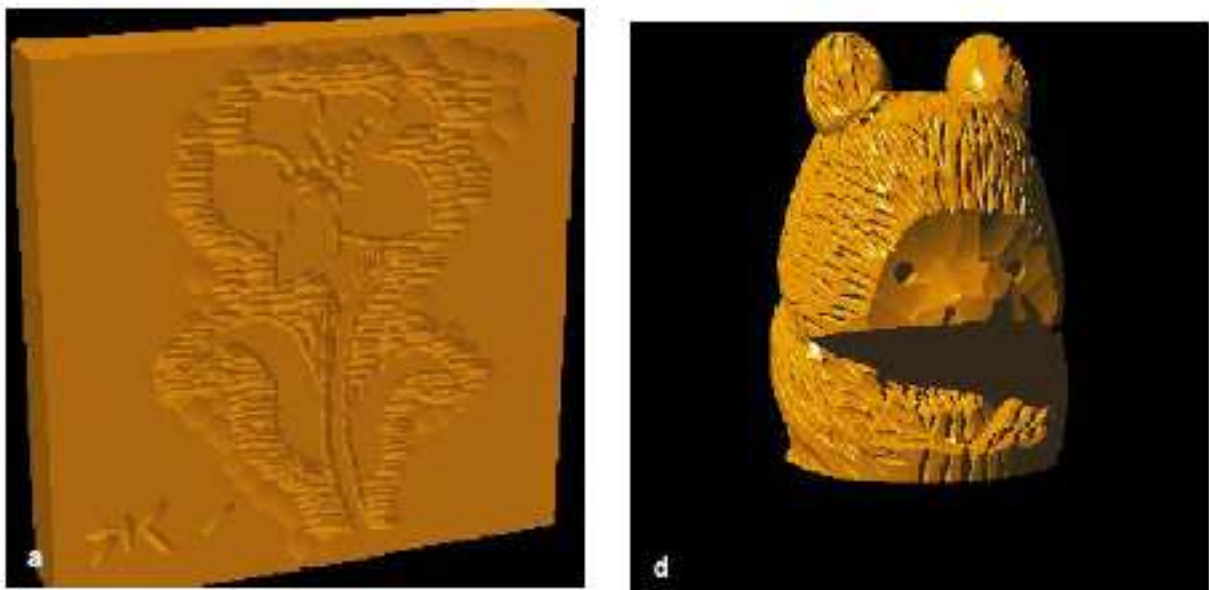


FIG. 19 – Exemples de résultats obtenus à l'aide de [29]

Comme on peut le constater sur la figure 19 le résultat esthétique est excellent. En revanche, le risque majeur de cette méthode est l'augmentation du temps de calcul pour un grand nombre d'opérations de déformation. Dans ce cas, la vitesse d'affichage n'est que de quelques images par seconde. Il semble ainsi impossible d'enrichir cette méthode (modèle physique,...) afin d'augmenter le réalisme lors du travail.

3.3 Les Free Form Deformation (FFD)

La méthode des FFD fut créée par Soderberg et Parry en 1986 [35]. Elle consiste à englober un objet quelconque à l'aide d'un maillage parallélépipédique, puis à calculer les coordonnées de chaque sommet de l'objet par rapport au sommet du maillage le plus proche. Donc lorsque l'on modifiera la position d'un sommet du maillage, les sommets de l'objet suivront de manière conséquent (figure 20).

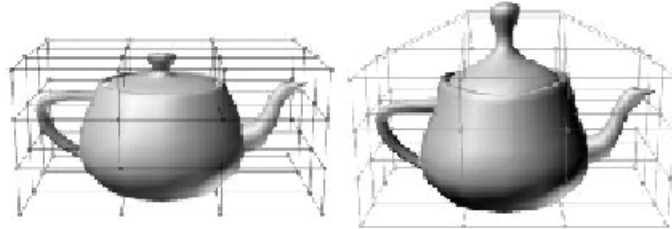


FIG. 20 – Déformation à l'aide de FFD [32]

Le maillage englobant est défini de préférence par un produit tri varié de polynômes de Bernstein :

$$X_{ffd} = \sum_{i=0}^l C_l^i (1-s)^{l-i} s^i \left[\sum_{j=0}^m C_m^j (1-t)^{m-j} t^j \left[\sum_{k=0}^n C_n^k (1-u)^{n-k} u^k P_{ijk} \right] \right]$$

Grâce à cette formulation, la manipulation d'un point du maillage influe sur les voisins et permet un mouvement uniforme. Les FFD fournissent une déformation rapide et intuitive d'un objet quelconque, mais ne gère pas la topologie. Elle est limitée à cause de sa notion de maillage englobant (pour l'instant de forme unique) et ne permet pas de trous l'objet.

Il existe quelques extensions des FFD :

- Hsu et al. [21] proposent de masquer le maillage englobant et de manipuler directement l'objet. Le maillage devient alors virtuel et inexistant pour l'utilisateur.
- AxDF : ce modèle fut proposé par Lazarus et al. en 1993 [24] ; il consiste à placer un axe virtuel à travers l'objet et à exprimer les coordonnées de chaque sommet de l'objet par rapport à l'axe. On se sépare ainsi de la notion de maillage englobant. La déformation est induite par la manipulation de l'axe de référence (figure 21).
- EFFF : Extended FFD. Cette méthode, exposée par Sabine Coquillart en 1990 [14] propose de nouveaux maillages englobants non parallélépipédiques qui permettent de nouvelles déformations (voir figure 22).
- NFFF : Nurbs FFD, proposé dans [23] par Lamousin et Waggenspack en 1994. Le maillage est constitué de Nurbs et n'est plus régulier. Il s'adapte en fonction de la densité de l'objet.

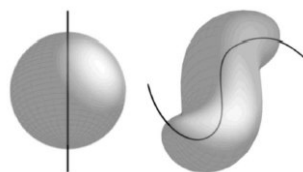


FIG. 21 – AxDF, déformation par rapport à un axe [32]

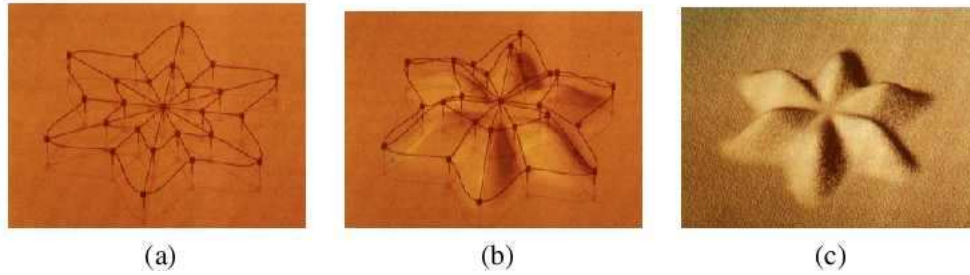


FIG. 22 – EFFT, maillage non parallélépipédique [32]

3.4 SCODEF

La SCODEF (Simple CONstrained DEFormation) a été introduite par Borel et Rappoport en 1994 [12]. Il s'agit en fait d'une reformulation de la méthode *B-Spline Space Times* [9]. L'utilisateur va choisir un ou plusieurs points de l'espace, puis définir pour chacun d'eux un déplacement et un rayon d'action. Une SCODEF est donc définie de la façon suivante :

- De \mathbb{R}^n dans \mathbb{R}^n , avec n dimension de l'espace
- r le nombre de contraintes qui définissent la SCODEF
- C_i la position d'un point de contrainte dans l'espace original
- dC_i son déplacement dans l'espace déformé
- R_i le rayon d'action de la contrainte C_i
- f_i la fonction de déformation de C_i

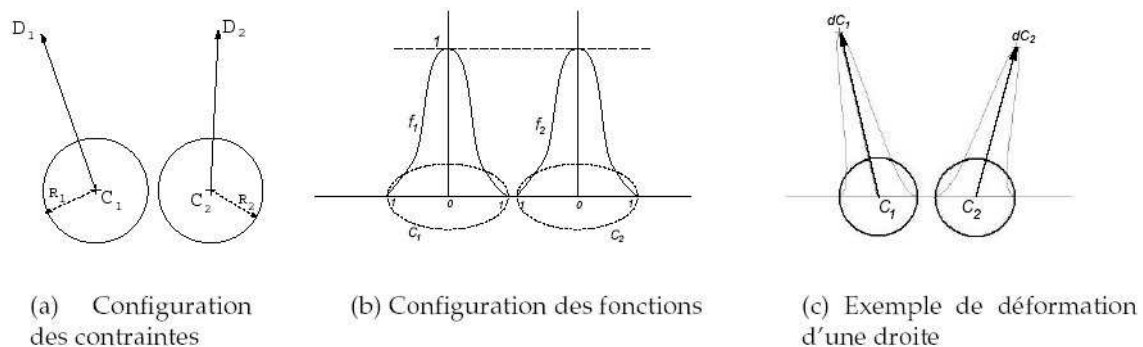


FIG. 23 – Méthode SCODEF [32]

Ici (figure 23) les contraintes sont disjointes et déforment ainsi la droite à deux points différents. Mais elles peuvent aussi avoir une zone d'influence mutuelle (voir même être confondues), auquel cas elles s'influencent mutuellement.

La SCODEF est une des méthodes les plus intuitives. On choisit un point, le rayon d'action, la fonction d'influence et le déplacement ; on peut ainsi très facilement "deviner" le résultat. Ceci permet des déformations intuitives qui se rapprochent de l'imagination de l'utilisateur. Mais ces déformations sont surfaciques, alors que nous travaillerons sur des volumes. Cela impliquera de "remplir" la déformation. De plus, si nous souhaitons appliquer ces déformations sur un outil, la méthode étant définie de manière continue, il faudra donc discrétiser la surface "résultat" pour pouvoir travailler. En revanche, le résultat surfacique évitera de calculer quels sont les pixels "utiles ou actifs" lors de l'utilisation de l'outil (cf section 3.7 [6]).

3.5 Déformations libres sous contraintes

Les déformations libres sous contraintes sont la contribution de Romain RAFFIN lors de son travail de thèse [32]. L'aboutissement a été un enrichissement de la méthode SCODEF, dont voici une explication simplifiée des trois principales extensions :

1. Le cercle d'influence de la fonction devient un volume étoilé (figure 24c et 24d).
2. Le vecteur de déplacement d'un point devient une courbe de variation. Nous ne sommes plus limités à des déplacements rectilignes (figure 24a).

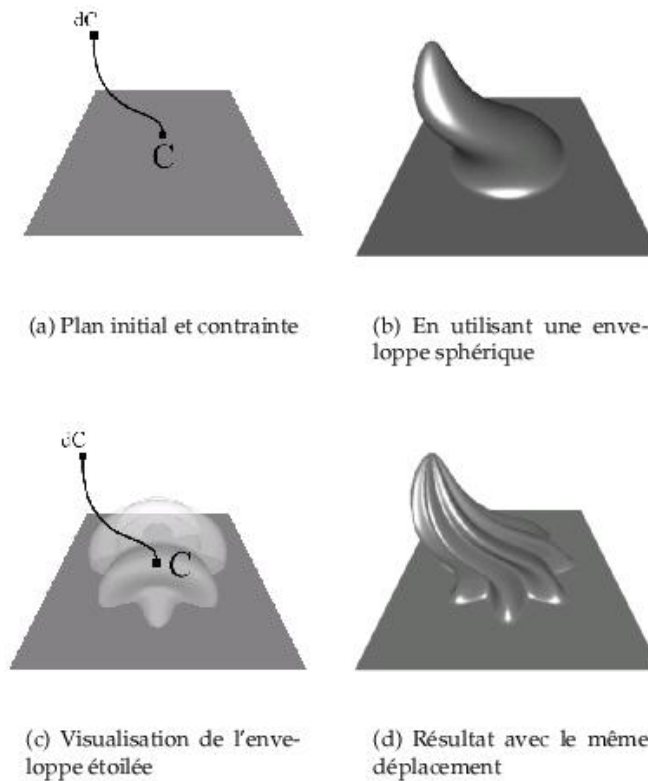


FIG. 24 – Déformations libres sous contraintes [32]

3. Le point d'application de la contrainte est étendu à une courbe (figure 25).

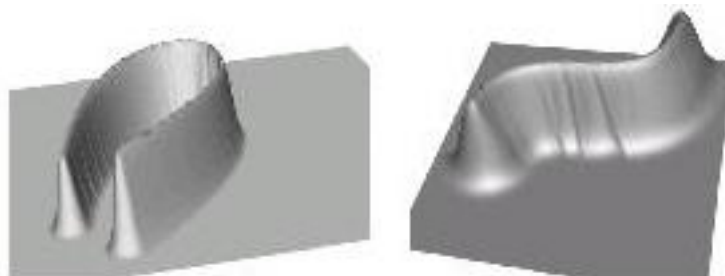


FIG. 25 – Déformation suivant une courbe de contrainte [32]

Si on choisit une zone d'influence sphérique et un déplacement rectiligne appliqués à un point, on retrouve la méthode SCODEF.

3.6 Esquisser le résultat

Dans l'article de Wang et Yuen paru en 2003 [38], les auteurs souhaitent "dessiner" ou "esquisser" le résultat, comme le ferait un dessinateur sur une feuille de papier. Pour cela, ils partent d'un objet quelconque qui servira de support, sur lequel ils dessinent à l'aide de la souris (ou d'un autre système) une forme ouverte ou fermée. La forme doit donc être composée de deux parties : une base fermée sur l'objet et une partie extraite. Leur méthode va alors déplacer la base le long de la partie extraite afin de créer le résultat tout en respectant les contraintes scalaires induites par le dessin de cette même partie. De plus, au fur et à mesure du déplacement, un maillage est créé afin de représenter et de stocker le résultat.

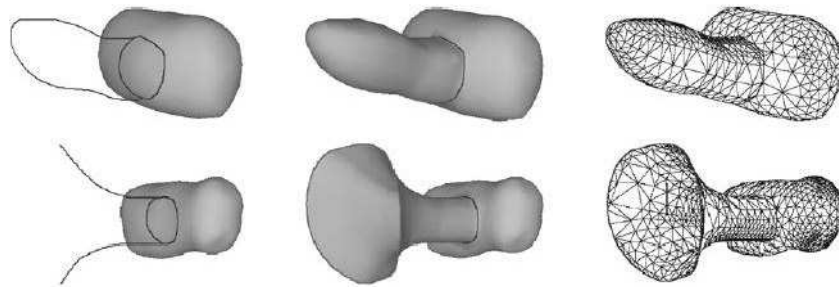


FIG. 26 – Extrusion de forme par simple dessin

Dessiner à la main une forme automatiquement implémentée par l'ordinateur est le rêve de toute personne faisant de la modélisation géométrique. Grâce à cette méthode on peut rajouter une forme quelconque à un objet existant en la dessinant. Malheureusement, on ne peut supprimer des parties (ce qui est primordial en sculpture virtuelle). De plus, les surfaces résultantes sont difficilement déformables par d'autres outils extérieurs à la méthode.

3.7 Méthode de Jörg Ayasse

En mars 2000, Jörg Ayasse présente dans [6] une méthode très simple mais efficace et interactive de sculpture virtuelle. Il utilise une voxélisation pour stocker les données et un algorithme des "Marching Cubes" pour améliorer le rendu. Un outil de sculpture est créé à partir d'une primitive quelconque (sphère, cube, ...); puis il est ensuite déplacé dans l'espace de travail suivant un axe prédéfini. Chaque fois qu'un voxel de l'outil rencontre un voxel de l'objet, le voxel de l'objet est supprimé. Il n'y a donc pas de déformation, mais une suppression systématique par rencontre de voxels. Inversement, on rajoute de la matière de la même façon (figure 27).

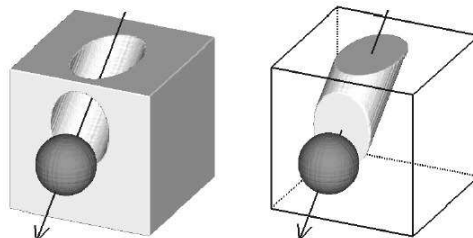


FIG. 27 – Ajout et suppression de matière [6]

Avec ce modèle, nous avons la façon la plus simple de supprimer ou de rajouter de la matière. Lors de la publication de cet article, il était possible de manipuler des objets de taille

400^3 et des outils 80^3 .

Pendant le déplacement de l'outil de sculpture, les voxels qui suppriment de la matière sont toujours les mêmes ! En effet, seuls les voxels se situant à la surface de l'outil peuvent rencontrer un voxel de l'objet, car ils sont toujours les premiers au contact. Les voxels à l'intérieur de l'outil deviennent alors "inutiles" ; rien ne sert alors de les stocker et de les manipuler. On effectue ainsi une économie en espace mémoire et surtout en temps de calcul. De plus, si le mouvement est rectiligne, seuls les voxels situés en surface et "à l'avant" (par rapport au sens de déplacement) de l'outil seront utiles : on divise donc encore par deux le nombre de voxels à manipuler. On va donc pré-calculer les voxels de l'outil utiles lors de la sculpture (figures 28).

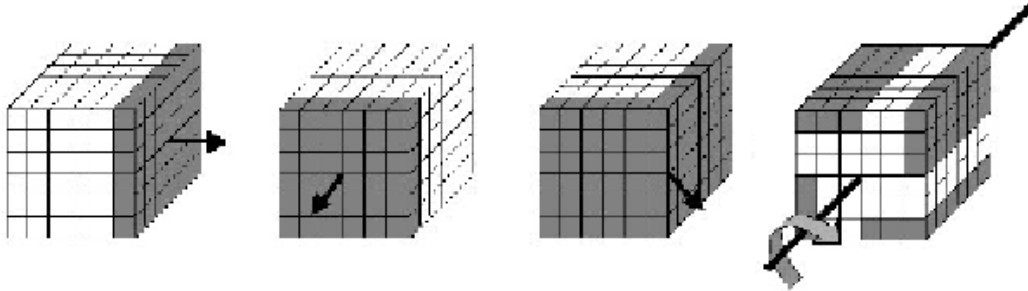


FIG. 28 – Sélection des voxels utiles [6]

L'auteur utilise une méthode très simple mais soulève un point intéressant dans son article (calcul des voxels utiles) qui sera réutilisé dans nos travaux.

3.8 Conclusion

Nous venons de présenter un certain nombre d'outils de déformation. Nous pouvons constater que créer un outil de forme quelconque est plus simple que d'en calculer les effets (problèmes de manipulation des surfaces, de calcul des intersections, de temps de calcul, de rendu, ...). Certaines pourront toutefois être implémentées afin de modifier l'outil de sculpture. Mais il nous faudra surtout définir une nouvelle méthode générique qui puisse combiner différents outils et agir sur des formes de topologie quelconque. Pour éviter les calculs trop lourds, nous nous intéresserons au comportement de matériaux géométriques, ce qui devrait simuler des caractéristiques de modèles physiques sans l'inconvénient de résolution d'équations différentielles.

Deuxième partie

Le Stage

4 Définitions

Pour commencer à travailler, il est important de définir notre cadre de travail. Nous allons ainsi donner quelques définitions relatives aux matériaux afin de préciser le contexte :

- Homogène : ses propriétés sont les mêmes en tous points.
- Isotrope : ses propriétés ne dépendent pas du repère dans lequel elles sont observées ou mesurées.
- Plastique et élastique : la plasticité désigne la capacité d'un matériau à se déformer sous l'effet d'une contrainte et à conserver sa déformation une fois la contrainte supprimée. L'élasticité désigne la capacité d'un matériau déformé à retrouver sa forme initiale et s'oppose en cela à la plasticité.
- Souple et rigide : la souplesse fait référence à la facilité à plier une structure et s'oppose à la rigidité qui désigne la résistance à la pliure.

Ceci est résumé par la courbe suivante :

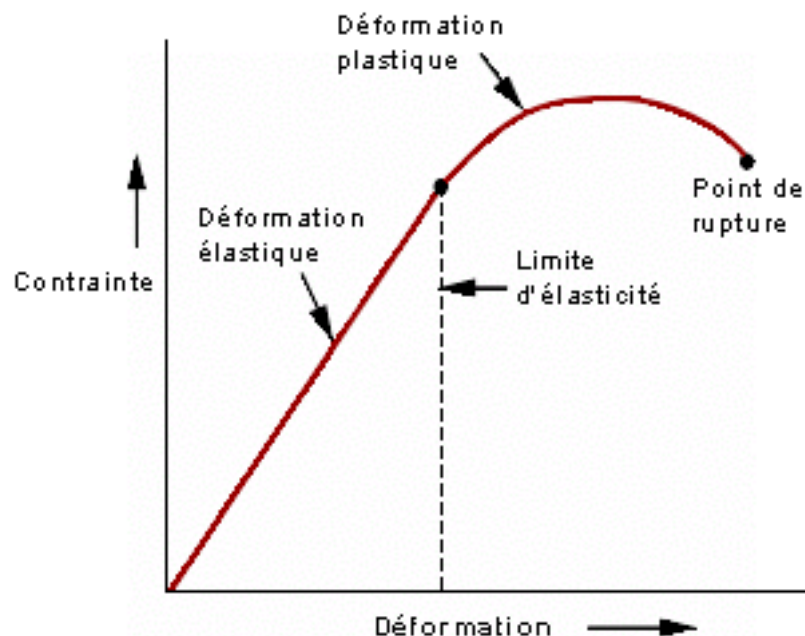


FIG. 29 – Représentation des différentes phases de la déformation

Dans ce stage, nous avons décidé de travailler avec des matériaux homogènes, plastiques et isotropes ; les matériaux ayant ce type de caractéristique sont la glaise, l'argile, ... C'est le cadre le plus simple imposant le moins de contraintes physiques. En effet, il serait difficile de travailler avec des matériaux non homogènes et élastiques car il faudrait prendre en compte les variations d'homogénéité, le retour de la matière dans une position stable, etc ... La complexité des paramètres qu'il faudrait alors gérer décuplerait les calculs et ne permettrait pas de faire du temps réel (vu dans section 3.1).

5 Amélioration algorithmique de la méthode de Jörg Ayasse

La méthode de Jörg Ayasse est le point de départ de notre outil de déformation. Dans l'article [6], il est décrit une méthode qui permet d'améliorer la complexité pour un modèle de sculpture virtuelle.

En effet, soit P le nombre de voxel de la pièce sculptée et N le nombre de voxel de l'outil. Chaque fois que l'outil se déplaçait, pour chaque voxel de la pièce, il fallait tester chaque voxel de l'outil. La complexité était ainsi en $O(NP)$, ce qui devenait vite important pour des outils ou pièces un tant soit peu volumineux. Avec sa méthode, Jörg Ayasse réduit l'outil à sa partie utile de taille n (figure 28). Donc la nouvelle complexité est en $O(nP)$. Dans le cadre d'un outil comme le cube (figure 28), seulement un huitième de l'outil est nécessaire pour sculpter, donc un temps de calcul huit fois inférieur.

Dans notre travail, nous nous sommes intéressés uniquement à des mouvements rectilignes de l'outil, donc seuls les voxels "en avant" de l'outil sont utiles. Ainsi, dans le sens du déplacement, il ne pourra jamais y avoir un voxel utile derrière un autre. Nous pouvons ainsi les stocker dans un buffer à deux dimensions X et Y qui contiendrait les coordonnées en Z de l'outil : un cube de taille $10 * 10 * 10$ aura un buffer de taille $10 * 10$. De même un sphère de rayon quinze aura un buffer de taille $30 * 30$, mais seules certaines cases contiendront les coordonnées en Z de l'outil, les autres étant initialisé à $-\infty$. Illustrons ceci par un exemple : une fois discrétisée une sphère de rayon 1 est une croix en trois dimension, n'ayant que cinq voxels utiles (figure 30) :

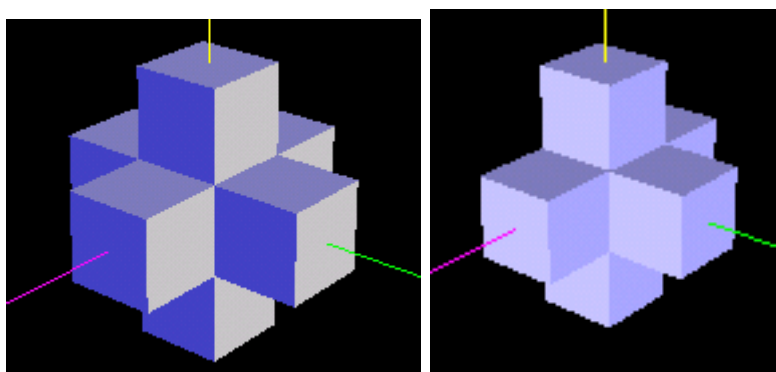


FIG. 30 – Sphère de rayon 1 et sa partie utile pour un déplacement sur l'axe Z (en vert)

Les figures 43, 44 et 45 montrent d'autres exemples de parties utiles. Cette sphère de rayon 1 aura ainsi comme buffer :

$-\infty$	0	$-\infty$
0	1	0
$-\infty$	0	$-\infty$

Ce buffer se traduit ainsi :

- 0 : la coordonnée en Z du voxel est égale à la position du buffer sur l'axe des Z .
- $n > 0$: le voxel est n cran en avant par rapport à la position du buffer sur l'axe des Z .
- $-\infty$: il n'y avait pas de voxel utile pour cette case, cette valeur assure ainsi que ce voxel de l'outil (en fait inexistant) ne rencontrera jamais un voxel de la pièce.

Nous allons maintenant décrire le processus de suppression. L'outil se déplace en suivant le sens et l'orientation du point de vue, mais en infographie lors de la visualisation de la scène, le

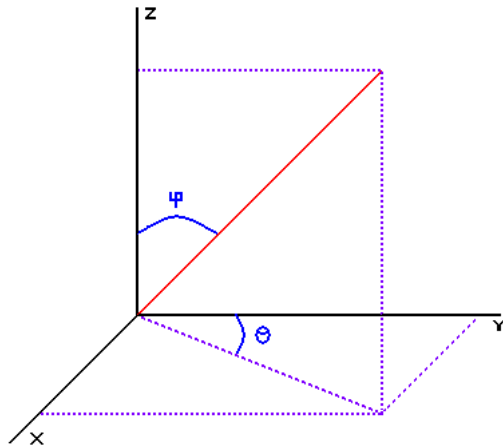


FIG. 31 – Les angles de rotation du point de vue

point de vue subit systématiquement des rotations d'angle $-\theta$, $-\phi$ afin d'être situé sur l'axe des Z (figure 31).

Pour garder une scène identique et cohérente, la pièce doit subir les mêmes rotations que le point de vue (figure 32 en 2D et figure 33 en 3D).

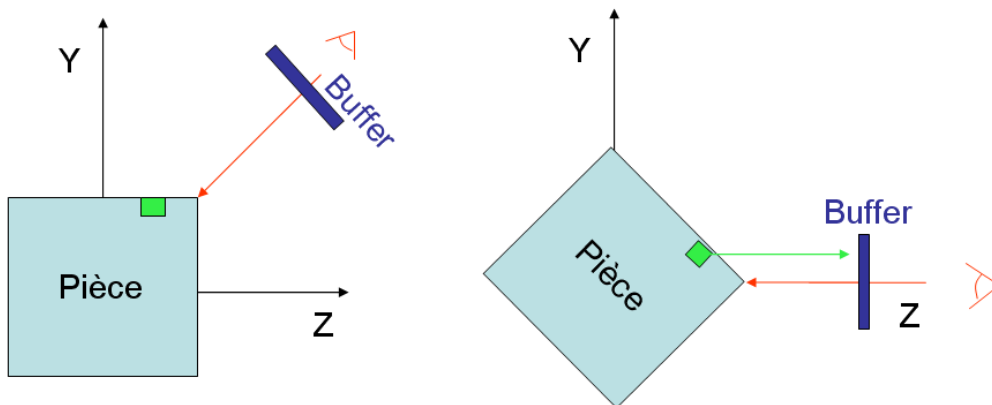


FIG. 32 – Rotations identiques de la pièce et du point de vue en 2D avec vérification dans le buffer

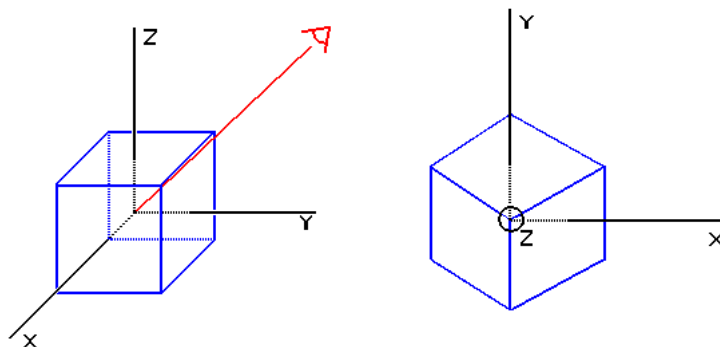


FIG. 33 – Rotations identiques de la pièce et du point de vue

Lors de la suppression des voxels, l'outil (maintenant le buffer) se déplace sur l'axe du point de vue (après rotations l'axe des Z) et nous calculons ces intersections avec les voxels de la pièce. Pour cela, il suffit de tester si la coordonnée en Z du voxel de la pièce après rotation est identique à celle du buffer. Si c'est le cas le voxel est supprimé. Algorithmiquement, cela se traduit comme suit :

Algorithme :

```

Pour chaque voxel  $V$  de la pièce
    | Si  $V.Z = \text{Buffer}[V.X][V.Y]$  Alors supprimer  $V$ 
    | Sinon Passer au voxel suivant
  
```

Donc un simple parcours de la pièce est désormais nécessaire, soit une complexité en $O(P)$. Nous obtenons ainsi une complexité linéaire dépendante de la taille de la pièce. Les calculs sont alors beaucoup plus rapides, ce qui nous permet de travailler avec des pièces plus volumineuses et comme nous le verrons plus loin d'ajouter des caractéristiques physiques. Le tableau ci-après résume l'évolution de la complexité en fonction des méthodes pour une pièce de taille P et un outil de taille N avec une partie utile de taille n :

Méthode	Complexité
Normale	$O(NP)$
Jörg Ayasse	$O(nP)$
Notre méthode	$O(P)$

6 Notre outil de déformation

6.1 Extension de la voxelisation

La forme binaire de la voxelisation (0 vide, 1 plein) qui fait sa simplicité est insuffisante dans notre travail. Sous cette définition, il nous est impossible de gérer un voxel qui ne serait pas "totalement plein" (ou réciproquement pas "totalement vide"). Nous parlerons plus loin de quantité de matière contenue dans un voxel qu'il nous faudra diviser en plusieurs sous-quantités. Nous devons donc étendre la notion binaire vide/plein à une quantité de matière variant entre 0 et 1. Mais il est difficile de représenter un voxel qui ne serait que partiellement "rempli". Nous tenterons de jouer sur la couleur (dégradé de bleu) et la transparence des voxels afin de visualiser la quantité de matière contenue à l'intérieur.

6.2 Utilisations des normales pour retrouver les propriétés de l'outil

Lorsque nous avons commencé à travailler, nous nous sommes immédiatement heurtés à un problème de taille : la voxelisation fait perdre toutes les connaissances mathématiques de la surface. En effet, si nous travaillons sur des iso-surfaces ou des objets définis de manière continue, nous connaissons la surface de l'objet et ses caractéristiques en tous points : tangente, normale, courbure, ... Toutes ces notions sont perdues lors de la discrétisation de la pièce. Or la normale en un point d'une surface nous donne le sens et la direction dans lesquels ce point de l'outil doit pousser la matière de la pièce (figure 34).

Il est donc essentiel dans la première étape de la construction de notre outil de déformation, de retrouver au moins partiellement la notion de normale de la surface pour chaque voxel afin

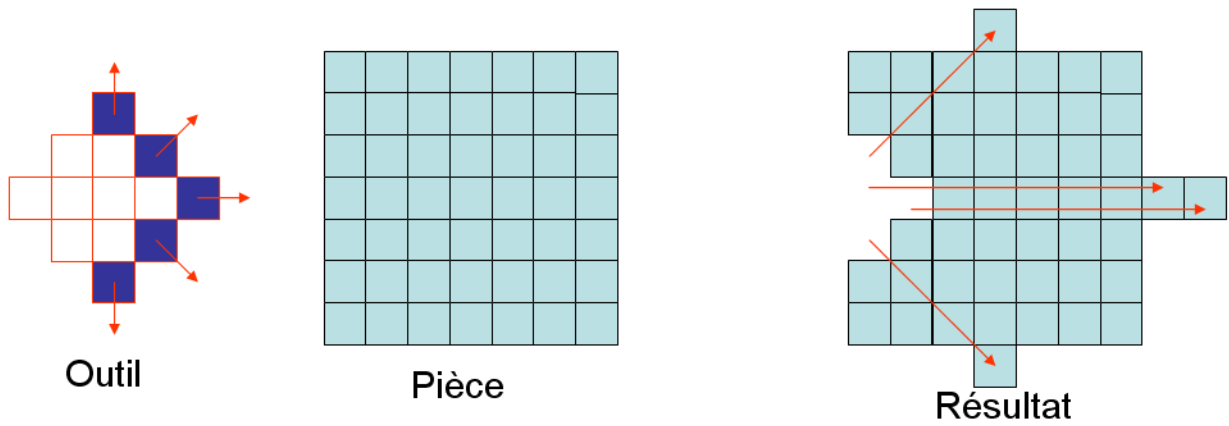


FIG. 34 – Utilité des normales dans la déformation

de pouvoir travailler avec un outil.

Mathématiquement une normale existe que sur une courbe ou une surface, donc nous devons recréer ce contexte (des surfaces) à partir des voxels de l’outil (ou plus précisément du buffer). Dans une huit connexité, chaque voxel V possède huit voisins V_0, \dots, V_7 (numéroté dans le sens inverse du sens trigonométrique). Nous pouvons alors utiliser deux voisins consécutifs (V_i, V_{i+1}) pour créer deux vecteurs $\vec{v}_i = \overrightarrow{VV_{i \bmod 8}}, \vec{v}_{i+1} = \overrightarrow{VV_{(i+1) \bmod 8}}$ (figure 35a). Ces deux vecteurs forment la limite d’une surface S_i , dont il est aisé de calculer la normale avec le produit vectoriel $\vec{N}_i = \vec{v}_i \wedge \vec{v}_{i+1}$ (figure 35b). Nous répétons cette opération pour les huit surfaces S_0, \dots, S_7 ainsi créées à l’aide des voisins. Il suffit maintenant de faire la moyenne des huit normales et nous obtenons la direction de travail v du voxel V :

$$v = \frac{1}{8} \sum_{i=0}^7 v_i$$

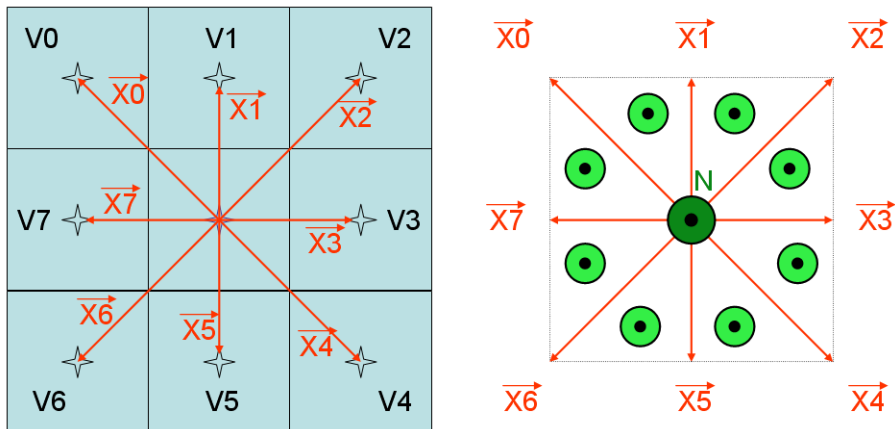


FIG. 35 – Calcul des normales à partir des voxels

Nous avons ainsi obtenu la normale (et ses composantes) à la surface de l’outil pour chaque voxel. Nous savons maintenant comment l’outil doit pousser la matière de la pièce lors d’un contact.

Voilà le résultat du calcul des normales d’un outil sur l’exemple du cube $10 * 10 * 10$ où nous affichons les composantes de chaque normale et non pas les normales elles mêmes :

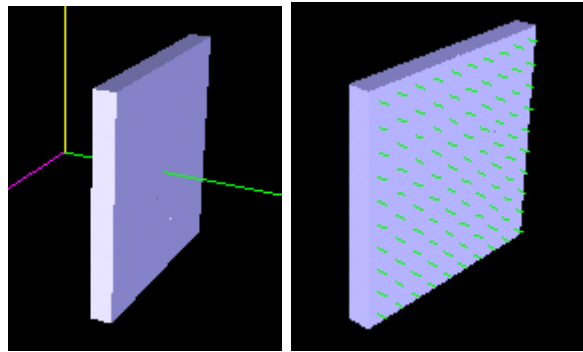


FIG. 36 – Les normales (à droite) issuent de la partie utile (à gauche)

Les figures 48, 49 montrent les résultats obtenus avec un outil en forme de pointe et un dernier outil plus complexe en forme de croix (plus exactement deux cylindres orthogonaux). Un simple parcours du buffer (car on ne fait ce calcul que sur les voxels utiles de l’outil) permet de calculer toutes les normales, donc ceci se fait avec une complexité linéaire. Les résultats obtenus sont tout à fait correctes et nous permettent de connaître une caractéristique essentielle de l’outil, mais ne sont pas aussi précis que dans le cas où l’outil serait défini de manière continue. Ces imperfections sont essentiellement dues aux erreurs d’arrondie et numériques introduites lors de la discrétisation de l’outil (section 2.5). Elles sont toutefois négligeable, car nous essaierons de rester au plus pres de la réalité sans vouloir absolument la simuler.

6.3 Déformation de la pièce par déplacement de matière

Maintenant que nous avons retrouver les caratéristiques de l’outil, nous allons pouvoir réaliser notre modèle de déformation. Pour cela nous déplaçons l’outil de manière rectiligne pas à pas (un pas représentant la taille de un voxel) jusqu’à ce que au moins un voxel de l’outil pénètre dans la pièce, c’est-à-dire jusqu’au moment où un voxel de l’outil prenne la place d’un voxel de la pièce. La quantité de matière contenu dans le voxel de la pièce doit donc maintenant être déplacé ; pour cela elle est divisée en sous quantité, proportionnellement à la taille de chaque composante de la normale du voxel de l’outil. Il y aura donc au plus trois sous quantités et au minimum une. Chaque sous quantité suivra la direction et le sens d’une composante de la normale, jusqu’à rencontrer un voxel partiellement rempli ou vide dans lequel elle ajoutera sa propre quantité de matière (figure 37).

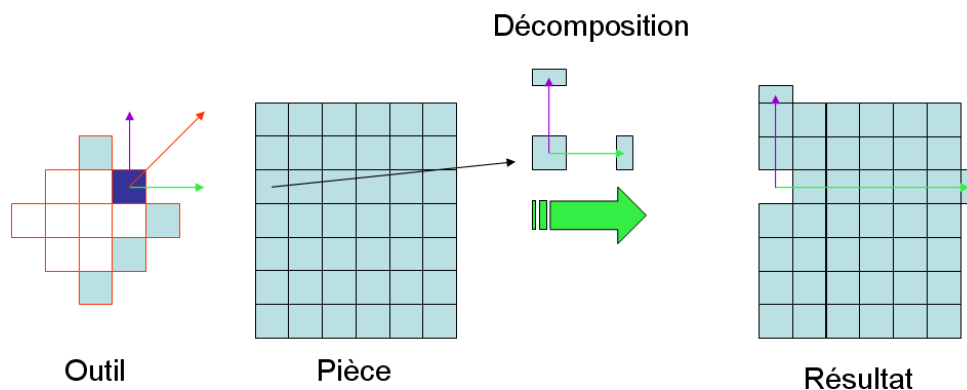


FIG. 37 – Notre méthode de déformation

Voilà un exemple avec une sphère comme outil (figure 38) :

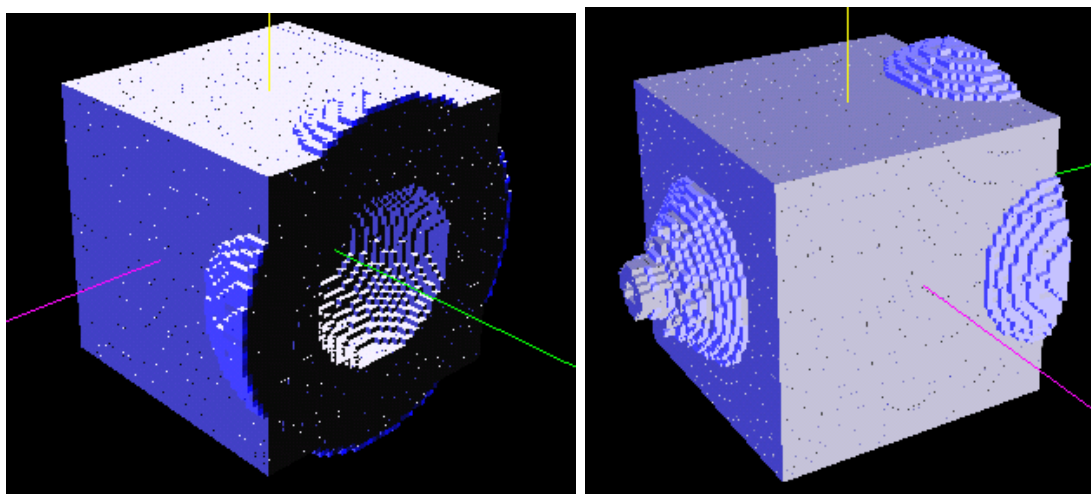


FIG. 38 – Résultat de la déformation à l'aide d'une sphère (figure 47) : à gauche, zone de pénétration de l'outil, à droite partie arrière

Les figures 50 et 51 montrent le résultat obtenu avec d'autres outils.

Pourquoi ne pas avoir déplacé la matière en fonction des normales et non pas des composantes des normales ? Cela se comprend avec l'exemple de la sphère comme outil : aucun des voxels de la pièce n'aurait été déplacé latéralement afin de simuler la déformation lors de la pénétration de l'outil. Dans la réalité, la pièce se serait immédiatement déformé latéralement sous l'effet de l'outil. Il est donc plus cohérent au niveau du réalisme de travailler avec les composantes de chaque normale.

Que se passe t'il si notre outil est un cube ? La partie utile sera une surface carrée et les normales seront toutes orthogonales à la surface utile. Donc toutes les normales seront identiques (figure 36) et elles déplaceront tous les voxels de la pièce rencontrée dans la même direction (figure 39).

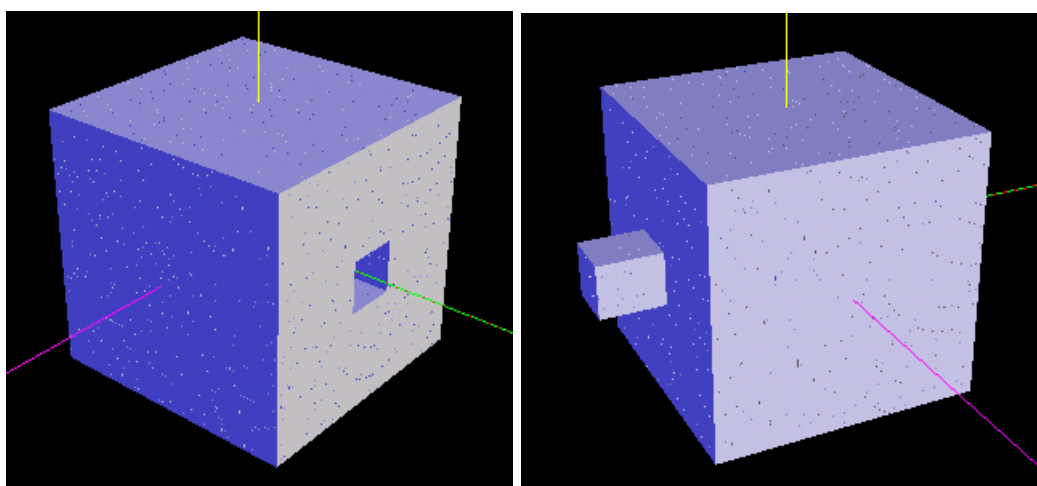


FIG. 39 – Problème du résultat de la déformation à l'aide d'un cube

Ce résultat est cohérent par rapport au modèle que nous venons de mettre en place, mais pas dans la réalité. Grâce à sa viscosité, un bloc de glaise se serait déformé de façon homogène même s'il était déformé avec un cube. Or pour des raisons de complexité, nous ne pouvons pas

incorporer un modèle physique afin d'avoir une déformation homogène, mais en revanche nous pouvons essayer de simuler un comportement physique.

6.4 Simuler la viscosité

L'argile se déformé de manière homogène grâce à sa viscosité. Nous pouvons décrire grossièrement son comportement par : "chaque particule entraîne ses voisins dans son déplacement et ses voisins la retiennent", ce que nous avons vu dans le modèle physique avec le système de masse ressort (section 3.1). En réalité, cette interaction a pour effet : à l'opposé du point de pression, la déformation est importante, puis diminue progressivement en fonction de la distance (figure 40).

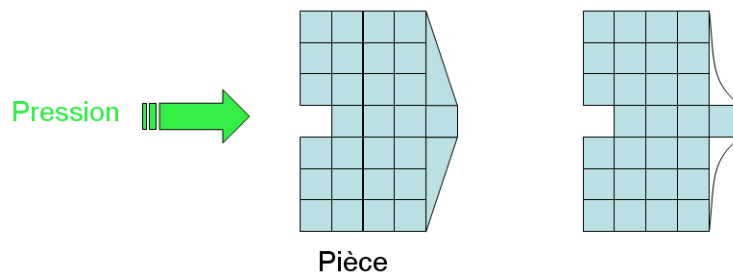


FIG. 40 – Effets de la viscosité

Il faut donc simuler ce phénomène : la solution est de rajouter de la matière aux voisins du voxel déplacé. La quantité de matière attribuée aux voisins devra être décroissante en fonction de la distance. Pour connaître la quantité de matière distribuée au voxel central et à chacun des voisins, nous allons utiliser une courbe de Bézier avec trois points de contrôle (figure 41).

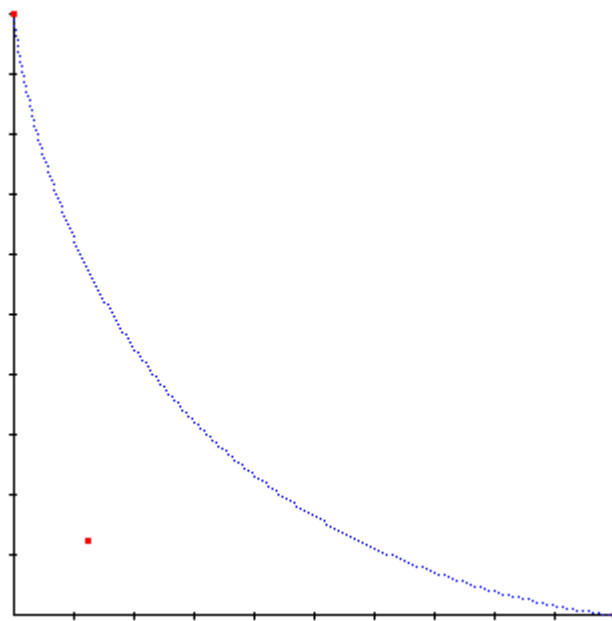


FIG. 41 – Courbe de Bézier simulant la viscosité : la quantité de matière (axe Y) décroît en fonction de la distance (axe X)

La viscosité de la glaise varie en fonction de la quantité d'eau et de la qualité de la matière. Nous pouvons donc jouer sur deux paramètres afin de simuler différentes viscosités : la taille de la zone sur laquelle nous ajoutons de la matière (zone d'influence) et la courbure de la courbe pour répartir cette matière. Plus la matière est visqueuse, plus la déformation est homogène et donc la courbe doit décroître de manière linéaire. Sur la figure 42, la viscosité est maximum, donc la répartition de matière décroît linéairement en fonction de la distance. En revanche si la matière est peu visqueuse (figure 41), la courbure est plus importante et la zone d'influence moins grande. De même, si la viscosité est nulle (ce qui est virtuellement possible, mais physiquement impossible) nous sommes alors dans le cas de la section précédente avec une zone d'influence nulle.

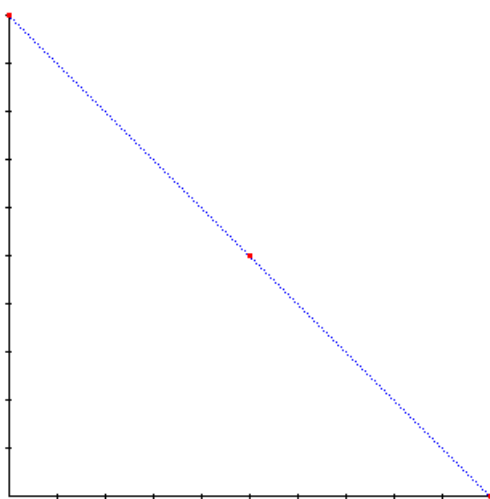


FIG. 42 – Courbe de Bézier représentant une viscosité de 0.5 sur une zone d'influence de 5

Nous pouvons maintenant ajouter une simulation de la viscosité dans notre outil de déformation. Voici le nouveau résultat avec comme outil le cube et une matière très visqueuse (une décroissance linéaire et une zone d'influence de 10 voxels, courbe de la figure 42) :

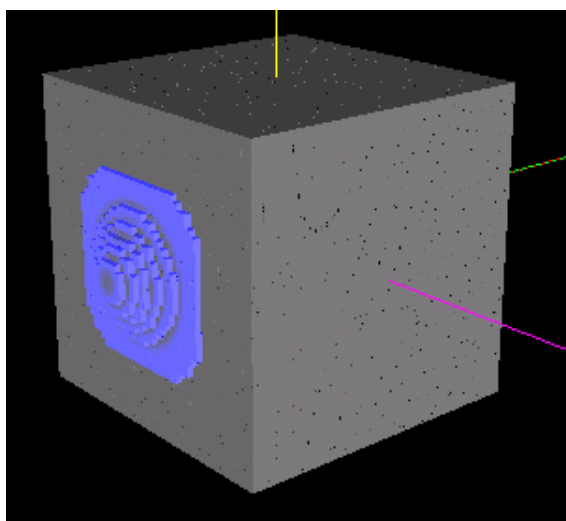


FIG. 43 – Résultat de la déformation à l'aide d'un cube avec une notion de viscosité définie par la courbe de la figure 42

La déformation produite à l'arrière de la pièce est plus homogène, donc beaucoup plus cohérente. Toutefois, dans la réalité, une légère déformation latérale se serait produite, que nous ne retrouvons pas ici car nous nous fions uniquement aux normales de la partie utile. En revanche, pour certains outils (une sphère , ...) la déformation se rapproche plus de la réalité (figure 43).

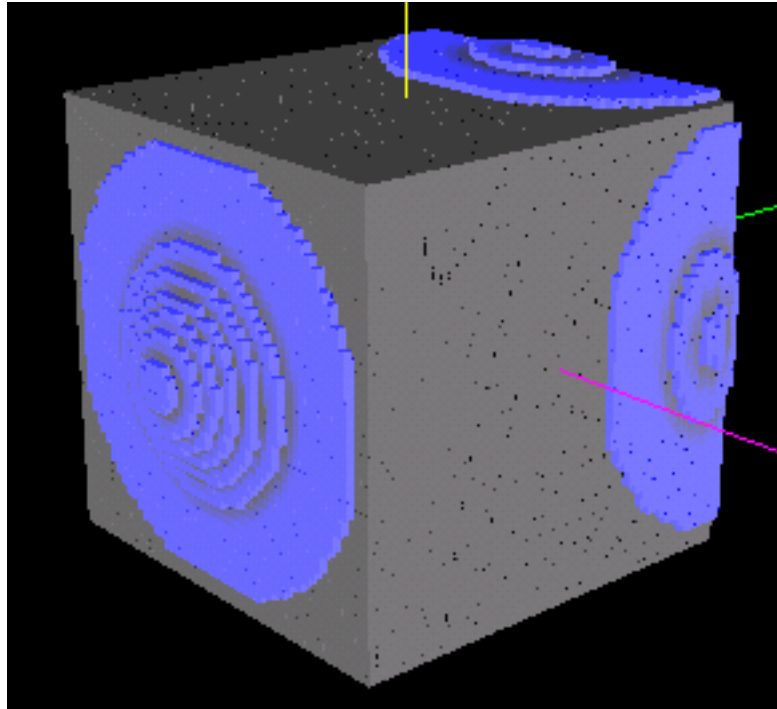


FIG. 44 – Résultat de la déformation à l'aide d'une sphère avec la notion de viscosité

Les figures 52 et 53 montrent d'autres résultats de déformation avec simulations de la viscosité pour d'autres types d'outils. Ci-dessous, deux autres exemples de déformations, réalisées avec une viscosité moins importante (courbe de la figure 41) à l'aide du cube et de la sphère.

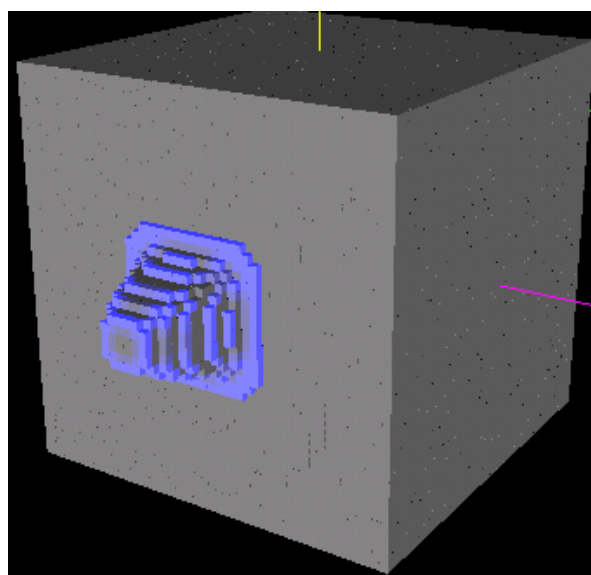


FIG. 45 – Résultat de la déformation avec un cube et une notion de viscosité faible

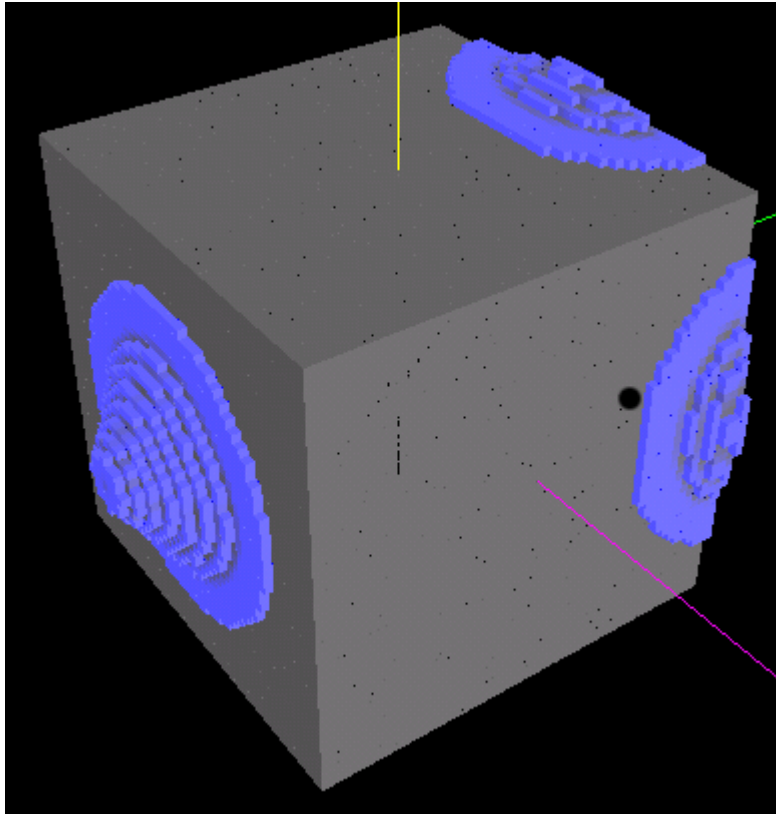


FIG. 46 – Résultat de la déformation avec une sphère (figure 47) et une notion de viscosité faible

Ces déformations sont conformes à nos attentes : elles sont moins homogènes et traduisent donc un matériau moins visqueux.

Mais la courbe de Bézier avec trois points de contrôle ne permet pas d'obtenir toutes les courbures possibles. En effet, d'après ces caractéristiques mathématiques, une courbe de Bézier est contenue dans son polygone de contrôle ; elle est aussi tangente au premier et dernier segment du polygone de contrôle. Donc le deuxième point de contrôle doit rester dans la partie supérieure droite du repère afin que la courbe y soit également contenue. Ainsi, si le polygone (de trois points) n'est pas contenu dans la partie supérieure droite du repère, la courbe ne serait pas non plus entièrement contenue dans cette même partie et cela se traduirait par une courbe qui aurait des valeurs négatives. Cela impliquerait que l'on enlève de la matière à certains voisins. De plus, l'amélioration algorithmique apportée dans la section 5 accélère d'avantage ces déformations qui sont déjà très rapides et le tout s'effectue donc en temps réel. Le tableau suivant montre quelques mesures de vitesse effectuées pendant le stage. Le temps affiché est le temps nécessaire entre le départ du calcul et l'affichage du résultat pour le déplacement de l'outil d'un voxel.

Pièces	Diamètre	
	50	100
Cube	1/12 s	1/6 s
Sphère	1/10 s	1/5 s

7 Conclusion

Nous venons de présenter un outil de déformation simple mais générique qui agit par déplacement de matière. Il peut agir sur des pièces de topologie quelconque et à l'aide d'outils de toutes les formes. Notre point de départ fût la méthode de Jörg Ayasse que nous avons nettement améliorée du point de vue algorithmique. Cette amélioration accélère grandement le processus de traitement et accélère encore plus l'exécution de notre outil qui est déjà rapide. Nous pouvons ainsi faire du temps réel avec des objets beaucoup plus volumineux. De plus, cet outil permet de simuler des comportements physiques en intégrant le comportement de la viscosité représenté à l'aide d'une courbe de Bézier. En modifiant les paramètres de cette courbe, nous pouvons simuler différentes viscosités et ainsi s'adapter à d'avantages de matériaux.

En revanche, cette méthode ne fait qu'approcher la réalité et ne simule pas parfaitement le comportement réel des matériaux lors de leur déformation.

Le point fort de cette méthode restera sa rapidité de calcul et son intégration de caractéristiques physique telle que la viscosité en temps réel.

8 Améliorations possibles

Au vu de notre travail, quelques pistes sont possibles afin de poursuivre ou d'améliorer :

- Ajouter des points au polygone de contrôle : cela permettrait d'obtenir des courbures plus importantes et de simuler d'avantage de comportement de viscosité.
- Trouver un autre modèle de courbe mieux adapté que celle de Bézier afin de simuler la viscosité.
- Ajouter un modèle physique lors de la frappe d'un objet : si nous frappons à l'aide d'un outil ayant une masse m avec une vitesse v , la force est donnée par l'équation $F = mv$. En fonction de la viscosité de la matière, de sa résistance et de sa densité, nous pourrions créer un modèle qui déterminerait l'enfoncement de l'outil dans la pièce.

Quelques figures supplémentaires :

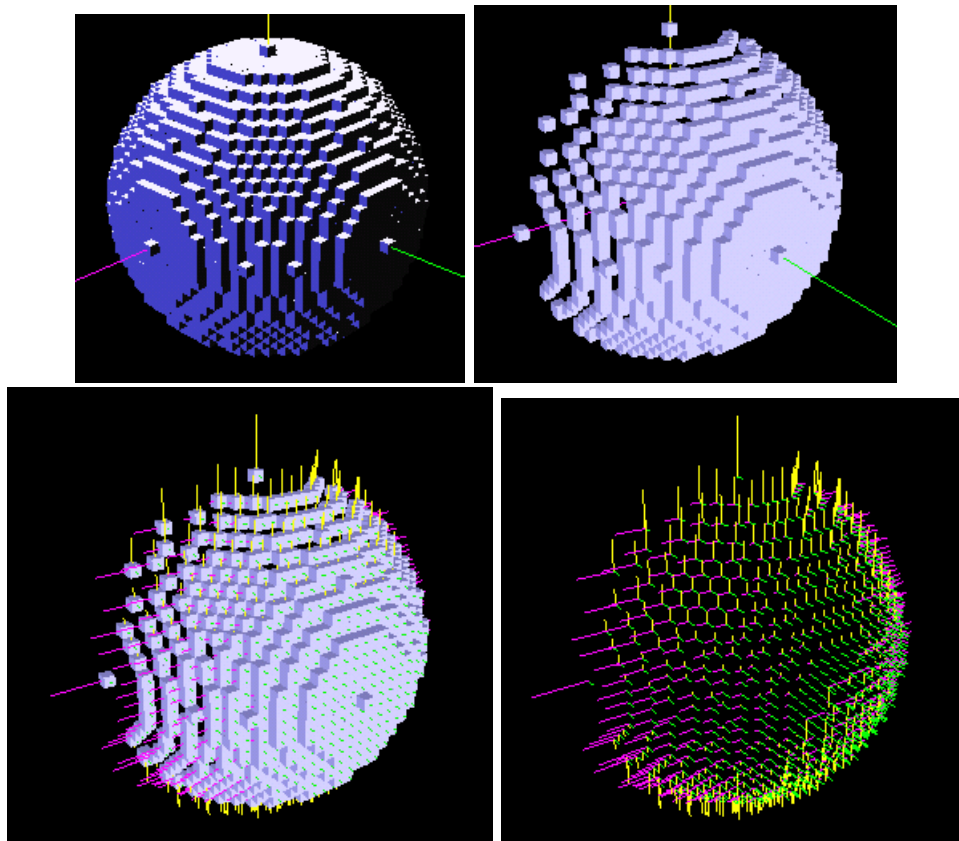


FIG. 47 – Sphère de rayon 15, sa partie utile et les normales

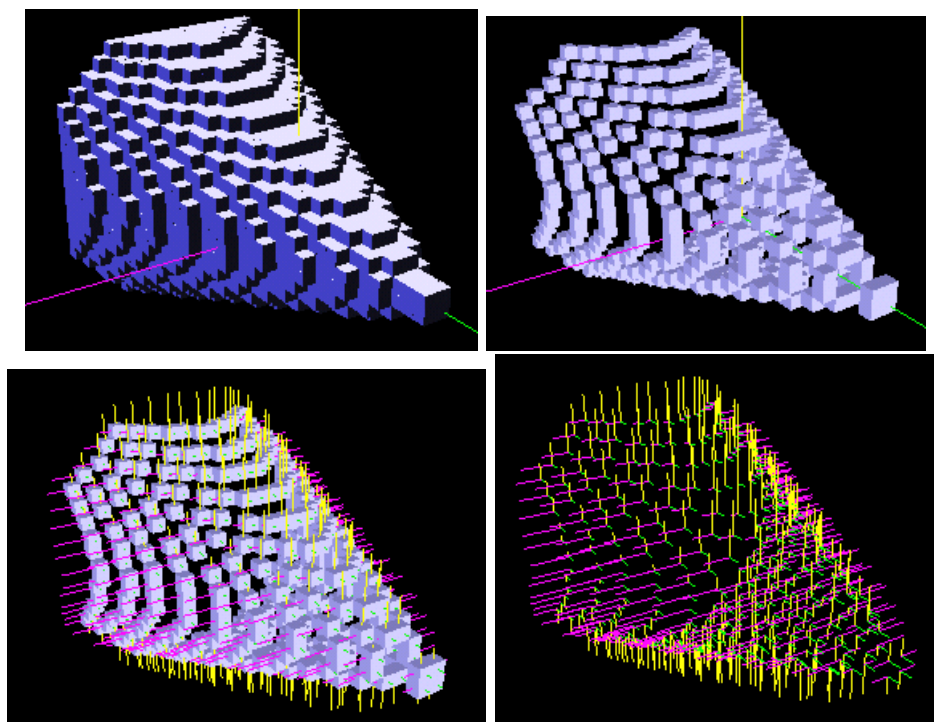


FIG. 48 – Une pointe, sa partie utile et les normales

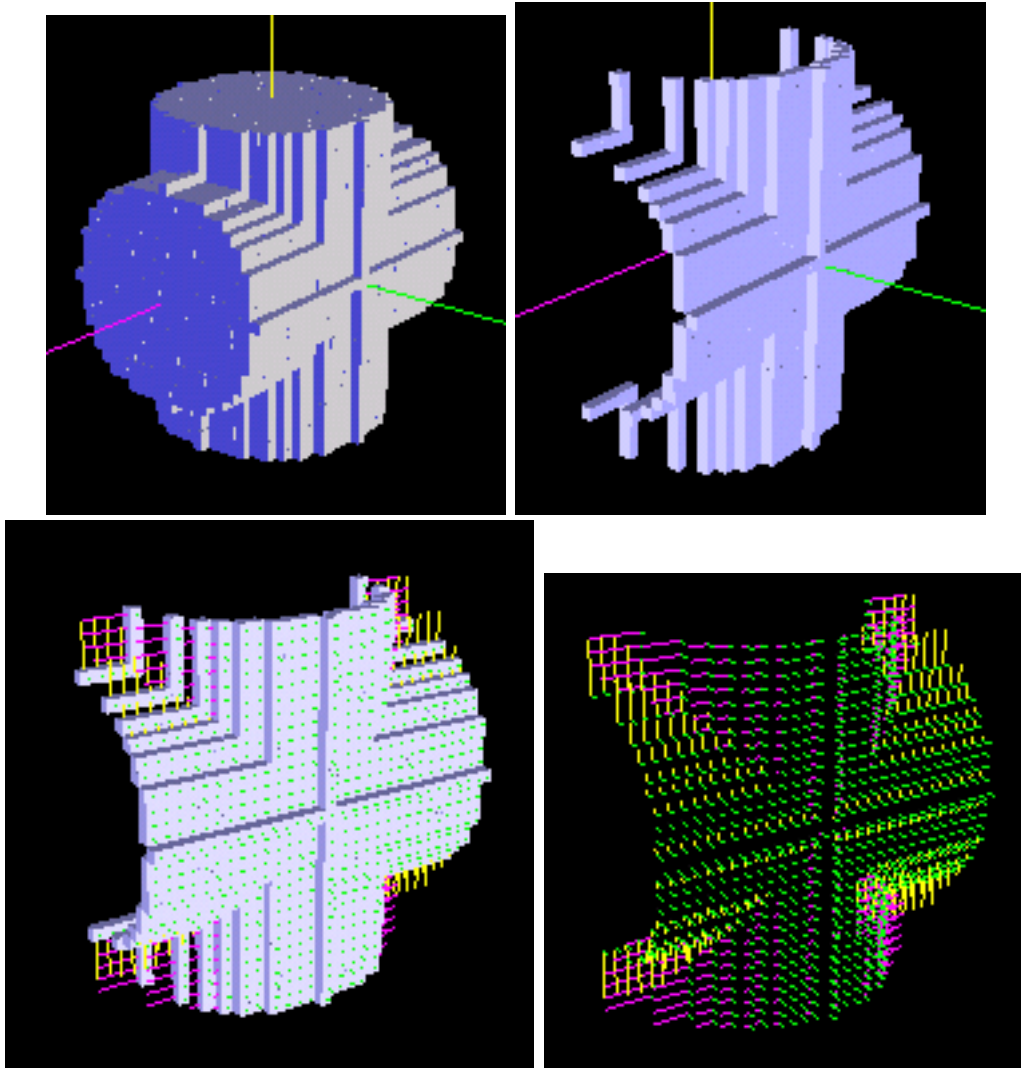


FIG. 49 – Un outil plus complexe en forme de croix, sa partie utile et les normales

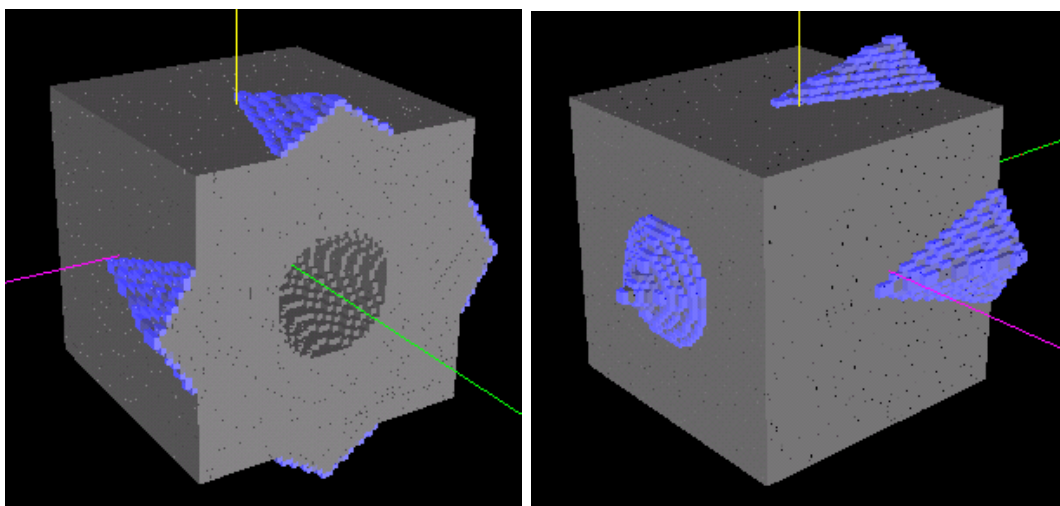


FIG. 50 – Résultat de la déformation à l'aide d'une pointe (figure 44) et une viscosité nulle

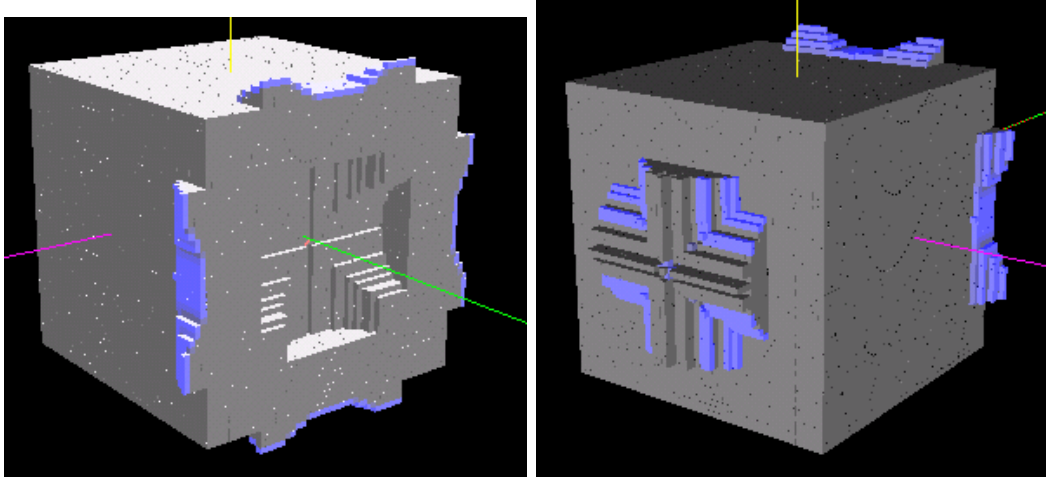


FIG. 51 – Résultat de la déformation à l'aide d'une croix (figure 45) et une viscosité nulle

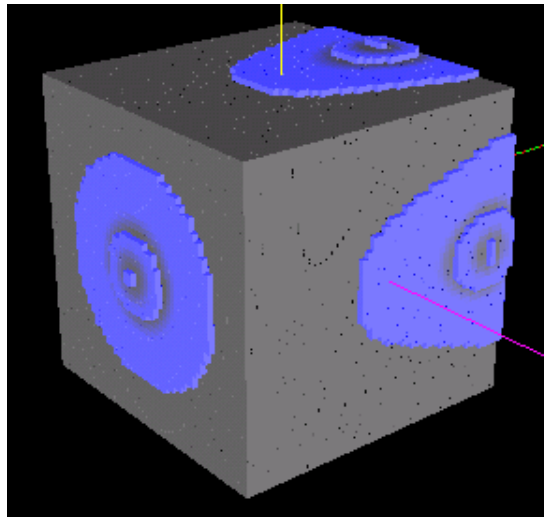


FIG. 52 – Résultat de la déformation avec une pointe (figure 44) et une viscosité importante

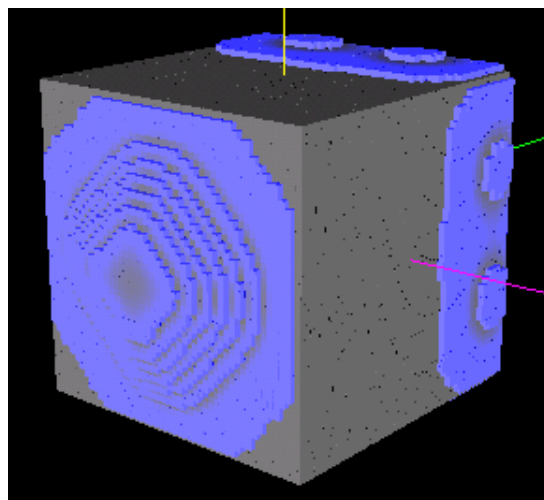


FIG. 53 – Résultat de la déformation avec une croix (figure 45) et une viscosité importante

Bibliographie

- [1] Page web : Euler explicite. <http://dmawww.epfl.ch/rappaz.mosaic/Support/support/node72.html>.
- [2] Page web : les quadtree. <http://www.greyc.ismra.fr/chris/DEA2002.pdf>.
- [3] Page web : méthode de runge kutta. www.ac-nancy-metz.fr/enseign/physique/divers/MethodNum/Schw-doc/RK402.pdf.
- [4] Page web : Pixar. www.pixar.com.
- [5] Page web : Pov ray. www.povray.org.
- [6] Jörg Ayasse and Heinrich Müller. Sculpturing on discrete displacement fields. *Comput. Graph. Forum*, 21(3), 2000.
- [7] A Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics Application*, 1981.
- [8] K-J. Bathe. Finite element procedures. *Prentice Hall, Englewood*, 1996.
- [9] D Bechmann and N Dubreuil. Animation through space and time based on a space deformation model. *The journal of visualisation and computer animation*, 1993.
- [10] James F. Blinn. A generalization of algebraic surface drawing. *ACM Trans. Graph.*, 1(3) :235–256, 1982.
- [11] H Blum. A transformation for extracting new descriptors of shape. *MIT*, 1967.
- [12] A Borel, P et Rappoport. Simple constrained deformation for geometric modeling and interactive design. *ACM Transaction on Graphics*.
- [13] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological surfaces. *Computer-Aided Design*, 1978.
- [14] S Coquillart. Extended free-form deformation : A sculpturing tool for 3d geometric modeling. *Computer Graphics*, 1993.
- [15] Piero Cosi, Emanuela Magno Caldognetto, Giulio Perin, and Claudio Zmarich. Labial coarticulation modeling for realistic facial animation. In *ICMI*, pages 505–510, 2002.
- [16] Olivier Coulon. Page web : les marching cube. www.esil.univ-mrs.fr/ocoulon/cours/ESIL3A_marchingcube.pdf.
- [17] Marc Daniel. Page web : cours de modélisation géométrique : facettes. www.esil.univ-mrs.fr/mdaniel.
- [18] Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In *SIGGRAPH*, pages 85–94, 1998.
- [19] Eric Ferley. *Sculpture Virtuelle*. PhD thesis, INPG, septembre 2002.
- [20] Eric Ferley, Marie-Paule Cani, and Jean-Dominique Gascuel. Practical volumetric sculpting. *the Visual Computer*, 16(8) :469–480, dec 2000. A preliminary version of this paper appeared in *Implicit Surfaces'99*, Bordeaux, France, sept 1999.
- [21] William M. Hsu, John F. Hughes, and Henry Kaufman. Direct manipulation of free-form deformations. In *SIGGRAPH*, pages 177–184, 1992.
- [22] Jing Hua and Hong Qin. Haptics-based volumetric modeling using dynamic spline-based implicit functions. In *VolVis*, pages 55–64, 2002.
- [23] W. Lamousin, H. et Waggenspack. Nurbs-based free-form deformations. *IEEE Computer Graphics and Applications*, 1994.

- [24] Jancene P Lazarus F, Coquillart S. Interactive axial deformations. *Institut National de Recherche en Informatique et Automatique*, 1993.
- [25] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Realistic modeling for facial animation. In *SIGGRAPH*, pages 55–62, 1995.
- [26] Jean Luc Mari. Page web : cours de surfaces implicites. [www.esil.univ – mrs.fr/ jlmari](http://www.esil.univ-mrs.fr/~jlmari).
- [27] JL Mari. *Modélisation de formes complexes intégrant leurs caractéristiques globales et leurs spécificités locales*. PhD thesis, Faculté des sciences de Luminy, 2002.
- [28] D Martin. Page web : les éléments finis. [http : //perso.univ – rennes1.fr/daniel.martin/melina/www/doc_html/guide/mefl.html](http://perso.univ-rennes1.fr/daniel.martin/melina/www/doc_html/guide/mefl.html).
- [29] Shinji Mizuno, Minoru Okada, and Jun ichiro Toriwaki. An interactive designing system with virtual sculpting and virtual woodcut printing. *Comput. Graph. Forum*, 18(3) :183–194, 1999.
- [30] S. Murakami and H Ichihara. On a 3d display method by meta-ball technique. 1987.
- [31] S. Platt and N. Badler. Animating facial expressions. *Computer Graphics*, 1981.
- [32] Raffin R. *Déformations de formes libres*. PhD thesis, Université de Bourgogne, 2000.
- [33] A. Ricci. A constructive geometry for computer graphics. *Comput. J.*, 16(2) :157–160, 1973.
- [34] Marie Samozino. Page web : photo des facettes. www.sop.inria.fr/geometrica/team/Marie.Samozino.
- [35] T Sederberg and S Parry. Free-form deformation of solid geometrics models. *ACM Transactions on Graphics*, 1986.
- [36] L. Segerlind. Applied finite element analysis. 1984.
- [37] Guillaume Sylvand. Page web : les octree. [www – sop.inria.fr/caiman/personnel/Guillaume.Sylvand/main/node79.html](http://www-sop.inria.fr/caiman/personnel/Guillaume.Sylvand/main/node79.html).
- [38] Charlie C. L. Wang and Matthew Ming-Fai Yuen. Freeform extrusion by sketched input. *Computers & Graphics*, 27(2) :255–263, 2003.
- [39] Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Data structure for *soft* objects. *The Visual Computer*, 2(4) :227–234, 1986.