

# Déformation d'objet complexe par graphe d'interaction

---

## Résumé

Les objets virtuels complexes peuvent généralement être défini géométriquement par une structure hiérarchique représentant plusieurs niveaux de détails, et tels que chaque niveau respecte certaines propriétés spécifiques. Déformer de tels objets tout en préservant l'ensemble de ces propriétés est un défi. Nous introduisons dans cet article la notion d'interacteur : un élément encodant un comportement de déformation à un niveau de la hiérarchie de l'objet. Les interacteurs peuvent être interconnectés pour définir des comportements de déformation multi-échelle préservant les propriétés de l'objet auquel ils sont attaché. Nous identifions différents comportement de déformations de base et apportons un nouveau cadre de travail unifié permettant d'exprimer différentes transformations complexes d'une manière simple et générique. Nous montrons qu'il est possible d'exprimer par le biais de notre approche unifiée des méthodes de déformations complexes de l'état de l'art, ou de les étendre aisément par de nouveaux comportements.

## Abstract

*Complex geometrical objects can usually be defined by a hierarchical structure representing several layers of details whose distribution are following specific properties. Deforming such complex objects while preserving all the interconnected properties is a difficult task. We introduce the Interactor: an element representing a deformation behavior at a given scale of the corresponding object. Interactors can then be connected as a network representing a multi-scale deformation behavior allowing for preserving properties of the associated object. We identify useful basic deformation behaviors and provide a new unified framework enabling to express several complex transformations in a simple and generic way. We express state of the art shape-aware deformation methods in our framework and extends some of them to various new behaviors easily thanks to our general framework.*

---

**Mots-clés :** Modélisation géométrique, déformation, objet complexe.

## 1. Introduction

La modélisation de contenu 3D complexe est un élément majeur à prendre en compte pour la production de scène virtuels. Les artistes numériques doivent aujourd'hui produire toujours plus d'objets virtuels dont les différents niveaux de détails attendus par les utilisateurs ne cessent également de croître. Nous pouvons citer comme exemples caractéristiques de tels objets complexes les plantes et arbres avec leurs hiérarchies propres de troncs, branches et sous-branches ; ou encore les villages faits de maisons, murs, fenêtres, portes, briques, etc. De plus, peupler un monde virtuel tel qu'un paysage à l'aide d'arbres et maisons nécessite de construire de nombreuses occurrences d'objets

proches, mais possédants différentes variations dans leurs formes.

Générer l'ensemble de ces objets à partir de zéro est une tâche particulièrement coûteuse. Pour faciliter ce travail, la génération procédurale permet de remplir des mondes entiers d'objets virtuels. Cependant, la génération procédurale ne permet pas de contrôler directement la forme et la position des éléments de la scène ; le contrôle est indirect et dépend d'un ensemble de paramètres qui n'offrent pas toujours les degrés de libertés souhaités.

D'un autre côté, les approches par sculpture virtuelle permettent de modéliser précisément des formes spécifiques. En partant d'une forme initiale, des déformations lui sont appliquées et permettent ainsi de venir modéliser efficacement différentes occurrences de formes légèrement différentes. La déformation étant guidée par les gestes de l'artiste, celui-

ci peut alors contrôler finement la forme obtenue en accord avec ses attentes. Bien que ces approches de sculpture ont prouvé leur efficacité vis à vis de la déformation de formes lisses [ACWK04, vFTS06], elles ne s'appliquent aujourd'hui pas directement sur des formes complexes telle que celles décrites précédemment. En effet, l'application directe d'une déformation de l'espace non uniforme sur un objet comportant de nombreux éléments vient détruire la cohérence globale liant ceux-ci. Par exemple, l'application d'une homothétie vient affecter les espacements entre éléments, ce qui peut venir impacter la cohérence d'un système devant les préserver.

Les déformations dites *intelligentes* ont récemment permis d'appliquer avec succès des déformations sur des objets complexes tout en préservant les propriétés de l'objet dans des cas spécifiques [MWCS13]. Ce type de déformation possède un très fort potentiel, mais est principalement limité à l'utilisation sur des cas particuliers d'objets. Dans cet article, nous proposons *un nouveau cadre de travail unifié permettant l'application de déformations intelligentes sur des objets formés par une hiérarchie complexes générique* tout en préservant la cohérence de ces objets lors de la déformation. Nous proposons ainsi de combiner à la fois la puissance des approches de modélisation procédurale vis à vis de la préservation des propriétés de cohérence, avec l'expressivité des approches de sculpture pour modéliser des objets complexes.

Les méthodes standards de déformation placent généralement la géométrie au centre de leurs approches, puis viennent déduire les contraintes à appliquer à partir de celle-ci. À l'inverse, nous proposons dans ce travail de nous préoccuper déformations en premier lieu. Nous exprimons ainsi au premier plan la hiérarchie de contraintes s'appliquant sur les différentes structures de l'objet en utilisant des combinaisons de comportements de déformations basiques que l'on appelle *interacteurs*. La géométrie de l'objet peut alors n'être prise en compte que lors de la dernière étape, pour l'application effective de ces comportements de déformation sur l'objet. Cette vision apporte une plus grande liberté à l'utilisateur qui peut alors choisir plusieurs modes de représentation pour sa géométrie. Nous montrons ainsi qu'en autorisant l'utilisateur à construire une hiérarchie de comportement de déformation à haut niveau d'abstraction, cela permet de faciliter la sculpture d'objets complexes tout en permettant l'utilisation de nombreux types de déformations *intelligentes* permettant d'assurer la cohérence globale de l'objet déformé.

Les contributions décrites dans cet article sont les suivantes :

- La formalisation de l'interprétation de la déformation subie par un complexe. Cette formalisation est basée sur la notion de cohérence hiérarchique de l'objet, et est encapsulée dans ce qu'on appelle un interacteur. Chaque interacteur vient manipuler ainsi un élément de la hiérarchie

de l'objet tout en préservant certaines de ses propriétés initiales (partie 3).

- La modélisation du comportement de déformation d'un objet complexe sous la forme d'un graphe formant un réseau d'interacteurs. Ce graphe définit une *grammaire de déformation* et exprime la propagation des différentes déformations locales au travers de la hiérarchie de l'objet (partie 4).
- La définition d'une base d'interacteurs servant pour la définition d'une large famille de comportements de déformation (partie 5).

Ces contributions offrent un cadre de travail unifié permettant d'inclure certaines méthodes de l'état de l'art ainsi que d'en créer de nouvelles (partie 6).

## 2. État de l'art

Plusieurs approches ont été proposées afin de permettre de déformer des objets spécifiques tout en préservant la cohérence de ceux-ci. Nous pouvons classer ces approches suivant deux grandes catégories. Premièrement, les méthodes dont l'objectif est de préserver certaines propriétés géométriques pendant la déformation. Deuxièmement, les méthodes dupliquant ou réalisant des ré-arrangements de sous parties de l'objet original afin de déformer celui-ci ou d'en créer d'autres.

### 2.1. Préservation de caractéristiques géométriques

La première catégorie de méthode vise à préserver des propriétés pré-déterminées de la surface initiale au cours de la déformation. Pour cela, une approche de déformation standard est considérée, puis, celle-ci est modifiée pour satisfaire au mieux un ensemble de contraintes au sens des moindres carrés. Sorkine et al. [SA07] proposent de contraindre la déformation d'un maillage à être aussi rigide que possible. Cette contrainte est exprimée par la minimisation des déformations non rigides de chaque triangle tout en tentant de satisfaire des contraintes de positions pour un sous ensemble de sommets du maillage. Kraevoy et al. [KSSCO08] développent une méthode permettant de préserver la courbure des objets. Pour cela, la forme globale est plongée dans une grille volumique qui est ensuite déformée par l'utilisateur par une déformation volumique. Cette grille guide la déformation de la surface mais est elle-même contrainte par la forme locale de celle-ci. Ainsi, chaque cellule est contrainte à garder sa taille initiale si elle contient un morceau de surface courbée à l'intérieur, et est libre de s'étirer lorsque cela n'impacte pas la courbure de la surface sous jacente. L'approche permet ainsi d'allonger une surface localement planaire, mais préserve ses régions courbes. Gal et al. [GSMCO09] interprètent les objets manufacturés comme un ensemble de courbes décrivant les bords marqués de ceux-ci. Ces courbes peuvent alors être éditées dynamiquement, et sont déformés de manière à préserver cer-

taines propriétés tels que leurs longueurs, angles de connexions ou encore le parallélisme. Récemment, Dekkers et Kobbelt [DK14] étendent l'approche de *seam carving* sur un maillage 3D et permettent ainsi à certaines parties d'un objet de déformer tout en minimisant l'élongation des détails.

Toutes ces approches permettent de déformer un contenu existant, mais elles ne permettent pas d'en synthétiser. De ce fait, appliquer ces approches dans le cas de déformation très importantes aboutit inexorablement soit à ne plus pouvoir préserver les contraintes initiales, soit à perdre la cohérence de l'objet globale qui fini par posséder de larges zones planaires ne représentant plus fidèlement l'objet original.

Notons également qu'une manipulation directe d'objet généré par des approches procédurale à été proposée par Lipp et al. [LWW08]. Cependant, la déformation ne préserve que les propriétés qui sont spécifiées directement par la grammaire de génération de l'objet.

## 2.2. Ré-arrangement de sous-éléments

La seconde catégorie d'approches permet à des sous parties de l'objet original d'être dupliqué ou de remplacer d'autres parties trop déformées. Ainsi, un objet peut s'étendre sans nécessairement impliquer l'allongement de ses sous parties, à la place, de nouveaux éléments apparaissent.

Stanculescu et al. [SCC11] développent une méthode de préservation d'un maillage quasi-uniforme pour des déformations quelconques. L'approche vient insérer ou enlever des triangles du maillage lorsqu'une région est trop fortement déformée, et garantit de garder une triangulation de bonne qualité. Cette approche est également étendue de manière à préserver les bords francs de la surface déformée [SCCS13]. Notre approche s'inspire de ce travail, mais nous ne nous restreignons pas uniquement à la préservation de la qualité de la triangulation. À la place, nous généralisons l'approche afin de pouvoir gérer un ensemble d'éléments organisés hiérarchiquement. Dans le cas 2D, Lipp et al. [LSWW11] introduisent une méthode permettant de déformer la structure modélisant une carte de ville en utilisant une approche basée sur les *coupes de graphes*. La trace des bâtiments est générée procéduralement pour chaque parcelle, et la carte de la ville peut être déformée soit de manière continue par de petites déformations, soit changer de topologie lors de déformations plus importantes.

Lorsque la forme de départ correspond à un objet manufacturé possédant un arrangement répétitif de sous éléments, Bokeloh et al. [BWSK12] proposent une méthode permettant à la fois de détecter la répétition de ces sous éléments, et permet une déformation préservant l'espacement entre ceux-ci lors d'un étirement unidirectionnel. Milliez et al. [MWCS13] étendent cette approche à la duplication d'éléments le long d'un axe 1D courbé, et permet également

l'application d'une légère déformation *aussi rigide que possible* sur les détails sous la forme. Notre cadre de travail permet la gestion du cas de déformation unidirectionnelle d'objets dupliqués, mais s'étend également à d'autres modèles.

Finalement, Alhasim et al. [AZL12] encodent les détails présents sur une surface lisse sous jacente en tant que carte de déplacements. Lorsque la surface est étirée, une méthode de synthèse de texture est utilisée pour générer d'autres détails au niveau de cette carte de déplacement. Cette approche permet de gérer des déformation de distributions de détails complexes, mais ne convient que pour des détails organisés comme un champs de hauteur que l'on peut paramétrer au dessus d'une surface lisse.

Nous pouvons noter que les méthodes récentes prennent de plus en plus en compte la structure de la surface à déformer, et s'adaptent à celle-ci par le biais de spécificités géométriques, ou de préservation de relations entre éléments. Ces approches permettent de préserver la cohérence de l'agencement des sous-éléments, mais sont par contre limités à un contexte précis d'éléments ou de déformations. Ainsi, appliquer des déformations sur d'autres types d'éléments peut nécessiter de redéfinir entièrement l'approche.

Nous proposons un cadre général permettant de définir de manière unifié les éléments, sous-éléments, relations, et principes de préservation de cohérence lié à une déformation. Ainsi, une fois qu'une approche de *déformation intelligente* est définie dans notre cadre de travail, le type de forme ou de contraintes à préserver peuvent être aisément modifiés afin d'explorer une plus large gamme de déformation.

## 3. Interprétation de déformation

Nous nous plaçons dans le contexte où un objet reçoit une déformation au sens général, c'est à dire que l'un des paramètres définissant cet objet (position d'un élément, orientation, couleur) est modifié. Cette déformation peut par exemple être de nature géométrique, générée par un outil de déformation dans un logiciel manipulation 3D. Classiquement, un objet quel qu'il soit applique cette déformation à sa représentation surfacique. Dans le cas d'un maillage par exemple, la déformation sera appliquée aux sommets par le déplacement ou le changement de couleur.

L'idée au coeur de notre méthode et de permettre à un objet de ne pas appliquer directement à sa représentation surfacique la déformation. Au lieu de cela, nous allons lui donner la capacité d'interpréter la déformation, pour en appliquer effectivement une version modifiée à sa représentation. Le but de cette modification est de permettre à l'objet d'appliquer au mieux la déformation demandée tout en préservant une certaine cohérence.

### 3.1. Exemple d'interprétation

Nous nous plaçons dans un cas simple pour décrire un comportement de déformation.

Soit  $R$  l'espace représentant l'ensemble des rectangles, paramétrés par leur largeur  $l$  et leur hauteur  $h$  (voir figure 1). On définit  $C$  un sous-espace de  $R$  qui représente l'ensemble des carrés à partir de la contrainte  $p$  qui définit  $l = h$ . Pour un élément de  $C$ , on dit que  $p$  est une propriété de cohérence.

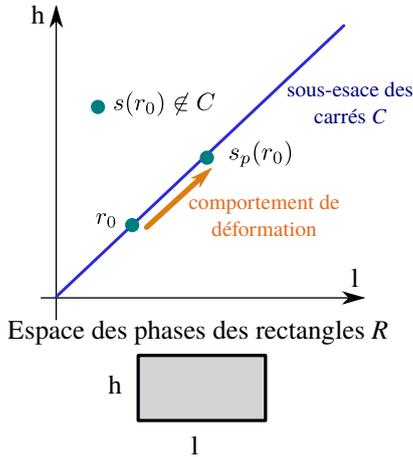


Figure 1: Un rectangle ayant la propriété  $p$  d'être un carré perd cette propriété lorsqu'il est déformé. En revanche, un comportement de déformation peut lui permettre de subir la déformation tout en préservant  $p$ .

On définit maintenant  $S$  un espace représentant l'ensemble des homothéties orthogonales, paramétrées par leur coefficients d'étirement selon les axes horizontaux  $a$  et verticaux  $b$  (voir figure 2). Les éléments de  $S$  sont des automorphismes de  $R$  mais pas de  $C$ .

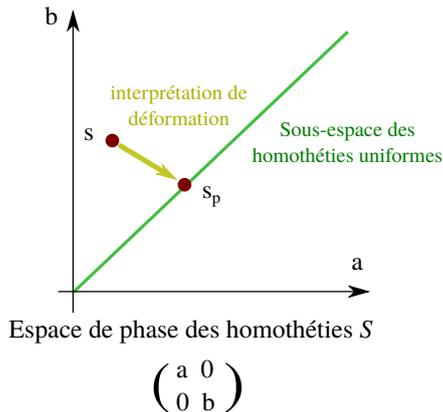


Figure 2: Une déformation est modifiée pour préserver une propriété de l'objet qu'elle déforme : celui-ci interprète la déformation.

Soit  $c_0 \in C$  un carré ; il vérifie donc  $p$ . Soit une homothétie quelconque  $s$ . Le résultats de l'application de  $s$  à  $c_0$  ne vérifie plus  $p$ . En revanche, il existe un sous-ensemble  $S_p$  de  $S$  tel que si l'on projette  $s$  sur  $S_p$ , on définit alors  $s_p$  de manière à ce que  $s_p(c_0)$  préserve  $p$ .  $S_p$  est l'ensemble des homothéties uniformes.

Cette projection d'une déformation  $s$  sur un espace de déformations admissibles pour l'objet est ce que nous appelons l'interprétation de la déformation de l'objet.

### 3.2. Comportement de déformation

Supposons qu'un objet  $o$  disposant d'une propriété de cohérence donné  $p$  subisse une déformation  $d$  ne préservant pas  $p$ . Nous souhaitons que  $d$  soit interprété avant d'être appliqué à  $o$  afin que celui-ci conserve  $p$ .

Nous créons un nouveau type d'éléments chargés d'interpréter les déformations subies par un objet afin d'en préserver les propriétés de cohérences, que nous nommons *interacteur*. Les interacteurs sont des éléments associés à un objet et chargés d'interpréter les déformations subies par celui-ci (voir figure 3).

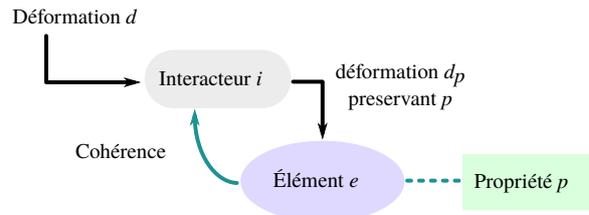


Figure 3: Un interacteur est l'unité fondamentale d'interprétation de la déformation.

Dans notre cas, l'interacteur  $i_o$  associé à  $o$  va interpréter  $d$  en  $d_p$  avant de lui appliquer. Ce faisant,  $o$  va montrer un certain *comportement de déformation*.

Un interacteur doit faire un compromis entre la fidélité avec laquelle il va interpréter la déformation et le respect de la cohérence de l'objet auquel il est associé.

## 4. Déformation d'objets complexes

### 4.1. Description hiérarchique

Dans le cadre de cet article, nous définissons récursivement un *objet complexe* comme un objet formé de sous-objets simples ou complexes. Un *objet simple* est un objet au sens commun du terme, c'est à dire disposant d'une représentation non compositée. Autrement dit, les objets complexes sont des groupements hiérarchiques d'objets.

Comme on peut le voir en figure 4, les objets complexes sont naturellement hiérarchisés.

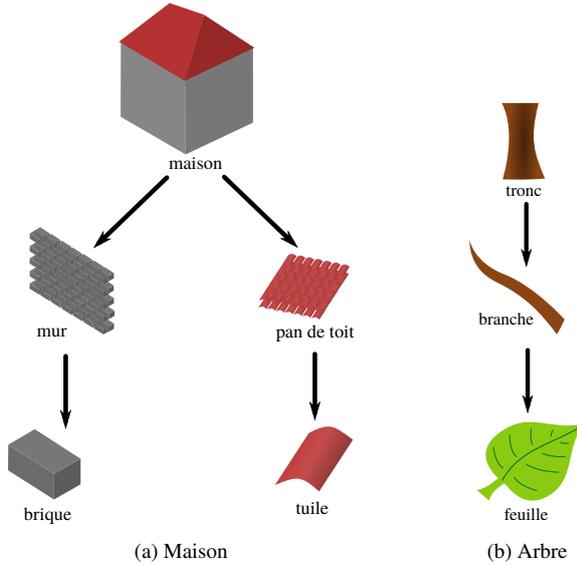


Figure 4: Exemples de descriptions hiérarchiques.

**Partition d'objet complexe** Comme nous avons pu le voir dans l'état de l'art (partie 2), les méthodes de déformation intelligentes existantes visent à partitionner un objet en deux catégories. Une de ces parties est *déformable*, et l'autre est *préservable*. Cette partition se fait généralement sur des critères géométriques, et la préservation selon un critère spécifique dépendant de la méthode.

Dans notre approche, nous considérons une partition de l'objet plus complexe : un objet est décomposé en un nombre arbitraire de niveaux. Notons que nous ne proposons pas de manière de créer cette partition, mais nous considérons qu'elle fait partie des données d'entrée, au même titre que l'objet lui-même. Ceci nécessite un pré-traitement, qui au demeurant peut être effectué par l'une des méthodes citées en partie 2 ou par toute autre méthode de segmentation.

#### 4.2. Arbre d'interaction

Lorsqu'on associe un interacteur à un objet complexe pour interpréter sa déformation, nous sommes confrontés au problème suivant : l'objet n'a pas de modèle de représentation. En effet, dans le cas de la figure 4a par exemple, la maison est uniquement composée de ses sous-parties, elle n'a pas de représentation propre.

Ceci amène à plusieurs remarques. Premièrement, la cohérence d'un objet complexe est-elle aussi multi-niveaux : elle est définie à partir de l'agencement des sous-parties ainsi que de la cohérence propre de chaque sous-partie. Deuxièmement, ceci implique que chaque sous-partie doit également être associée à un interacteur pour maintenir sa cohérence. On assiste donc à la construction d'un empilement arborescent d'interacteurs (voir figure 6), structuré comme la

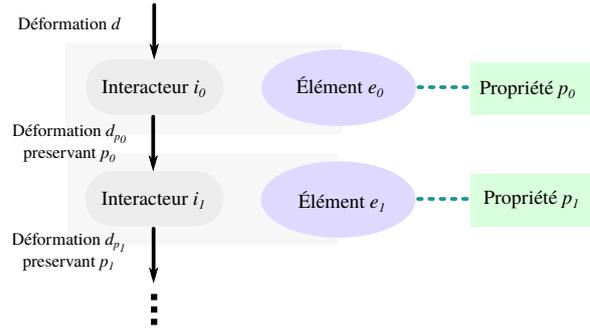


Figure 5: L'interacteur d'un objet non terminal adresse ses déformations aux interacteurs des éléments inférieurs. Les interacteurs forment donc un graphe caractérisant la transmission de la déformation dans l'objet.

description hiérarchique de l'objet auquel il est associé (voir figure 6).

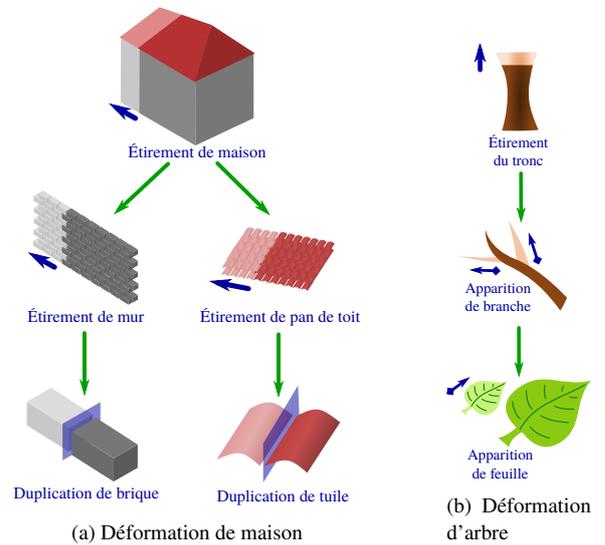


Figure 6: Les comportements de déformation de chaque élément d'un objet forment un comportement global.

Cet arbre d'interaction traduit un comportement de déformation respectant la cohérence multi-niveaux issue de la hiérarchie descriptive de l'objet auquel il est associé. Chaque déformation appliquée à l'objet est subdivisée en un ensemble de déformations appliquées sur ses sous-parties. Ce processus est itéré jusqu'à atteindre les éléments ultimes qui appliquent la déformation résultante au modèle visuel.

#### 4.3. Graphe d'interaction

Si, comme nous venons de le voir, chaque élément constituant l'objet dispose de son propre interacteur, il peut être intéressant de manipuler directement un élément. En effet, un

utilisateur peut souhaiter avoir accès à une déformation plus précise que celle offerte par la déformation de l'objet dans sa totalité. Dans le cas d'une maison par exemple, l'utilisateur pourrait vouloir manipuler un mur ou une brique indépendamment du reste de la maison.

Ceci pose un problème de cohérence : en effet, l'élément "mur" n'a connaissance que de ses sous-éléments, à savoir les briques. Donc si l'utilisateur déforme un mur, cette déformation sera interprétée pour chaque brique afin de préserver la cohérence du mur. Or la déformation d'un mur se fait sans passer par l'interacteur de la maison, la cohérence de celle-ci ne peut donc être préservée par l'arbre d'interacteurs.

La solution à ce problème est de casser la structure arborescente du réseau d'interacteurs en créant des connexions entre la sortie de l'interacteur d'un élément et l'entrée de l'interacteur associé à son parent. Ceci a pour conséquence de permettre au parent de superviser la déformation de l'élément tout en préservant sa propre cohérence.

Le réseau d'interacteur permet maintenant la descente et la remontée de déformations. Nous parlons dans ce cas de *graphe d'interaction*.

**Provenance de la déformation** Le point précédent a pour conséquence que les interacteurs doivent maintenant prendre en compte l'émetteur d'une transformation. En effet, lorsqu'une déformation remonte d'un enfant vers un parent dans la hiérarchie, l'interacteur du parent doit à son tour déformer l'enfant qui vient de l'activer (afin de rendre de compte de la déformation) ainsi que ses autres enfants (pour maintenir sa cohérence). Ceci implique de faire une distinction entre l'élément ayant émis la déformation et les autres éléments.

De manière similaire, le réseau d'interacteurs étant maintenant un graphe comportant des cycles, il est nécessaire de prendre en compte l'émetteur d'une transformation pour éviter une interprétation en boucle qui n'aboutirait jamais.

De manière générale, nos expériences nous ont amené à établir les règles suivantes :

- un interacteur recevant une déformation en provenance de l'utilisateur doit toujours en référer au parent de l'élément auquel il est associé ;
- un interacteur recevant une déformation de son parent ne doit jamais faire remonter de déformation.

#### 4.4. Grammaire de déformation

Les graphes d'interacteurs peuvent être vus comme comme des *grammaires de déformation*. Dans ce cadre, les interacteurs appliquent des règles de subdivisions qu'ils encodent. Les symboles sont les déformations des éléments. Les symboles terminaux sont les déformations s'appliquant au modèle visuel, ils sont produits par les interacteurs terminaux.

Contrairement aux grammaires de forme, les grammaires de déformation n'induisent pas de hiérarchie de l'objet.

## 5. Généricité de notre approche

### 5.1. Classification de propriétés de cohérence

Comme nous l'avons précisé en section 3, un objet peut être associé à une métrique de cohérence. Cette cohérence peut avoir différentes significations. Nous avons classé les différents types de cohérence que nous avons rencontré.

- propriétés géométriques : types de face (quadrangle, triangle), variété, quasi-uniformité [SCC11], présence d'un bord, courbure [KSSCO08], saillance [SCCS13], non intersection des sous-éléments [LSWW11], angle de connexion des sous-parties [LK11], alignement [MWH\*06], volume [ACWK04]
- propriété de distribution de détails : densité, homogénéité, amas, anisotropie [ÖG12]
- propriétés sémantiques : absence de cycles, stabilité d'un bâtiment [WOD09], adjacence de sous-éléments [LVW\*15], relation entre éléments d'un paysage [GGG\*13, EPCV14]

Certaines de ces propriétés sont équivalentes. Par exemple, on peut reformuler certaines propriétés sémantiques en termes géométriques.

Dans tout le cas, maintenir ces propriétés revient à manipuler les sous-éléments pour les conformer à un critère donné. Cette manipulation se fait au travers d'opérations de base que nous avons identifiées et qui constituent une base d'interacteurs terminaux. Nous reviendrons sur ce point à la partie 5.3

### 5.2. Taxonomie de comportements de déformation

À la connaissance des auteurs, il existe dans l'état de l'art seulement une classification de comportements de déformation. Elle se trouve dans [SCCS13]. Cette classification identifie trois comportements de déformation : *mutable*, *immutable*, and *effaçable*. Cette taxonomie est limitée au cas où :

- chaque élément est déterminé par sa dimension
- chaque élément ne contient qu'un seul type de sous-élément de dimension immédiatement inférieur
- chaque élément est contenu dans son parent
- chaque élément interagit avec les autres éléments de même dimension

Ces limitations proviennent du fait que cette classification a été conçue pour la déformation des éléments saillants lors de la sculpture d'un maillage uniforme. Nous retenons l'idée de classer les comportements de déformation, mais essayons d'en créer une plus générale.

### 5.3. Base d'interacteurs terminaux

Un interacteur est le récepteur d'une déformation. Ainsi un interacteur est complètement déterminé par l'élément auquel

il est associé, la déformation qu'il est capable d'interpréter, et la propriété qu'il va préserver lors de l'interprétation. On ne peut donc pas établir d'interacteurs génériques.

Nous avons identifié les interacteurs suivants comme étant fondamentaux :

- *mutation* : changement d'un paramètre de l'élément associé (position, orientation, ...)
- *création* : production d'un élément
- *destruction* : suppression d'un élément

Cette liste peut être étendue avec deux interacteurs supplémentaires issus des premiers :

- *fusion* : destruction d'un groupe d'éléments et création d'un nouveau (à partir des paramètres des éléments initiaux)
- *division* : création d'un élément à partir d'un groupe d'éléments (à partir des paramètres des éléments initiaux)

Ces interacteurs terminaux peuvent être combinés pour générer des comportements de déformations préservant les propriétés données en section 5.1.

Une base d'interacteurs non terminaux est difficile à établir : un interacteur est associé à un type de déformation et une propriété de cohérence. Ainsi, lister ces interacteurs revient à lister les déformations possible, les propriétés de cohérence (qui sont intimement liés au type d'élément), et de mettre les deux en correspondance, ce qui résulterait en une explosion combinatoire.

#### 5.4. Contrainte de forme

La cohérence d'un objet peut inclure que celui-ci ne tombe pas dans une configuration donnée. Par exemple, dans l'exemple de la section 3, le carré doit rester carré. Dans ce cas précis, il existe un moyen simple de transformer une homothétie en déformation conservant  $p$ .

Dans d'autres cas, il n'existe pas de modifications de la déformation permettant de préserver une propriété de cohérence donnée. Dans ce cas, il est possible pour un interacteur d'effectuer directement la transformation, rompant ainsi la cohérence de l'objet ; puis d'appeler un autre interacteur générant des transformations rétablissant cette cohérence. Par exemple, si un élément doit maintenir ses sous-éléments alignés selon un axe, son interacteur peut déplacer les sous-éléments puis les re-projeter sur l'axe initial. Un autre exemple est celui d'un élément devant maintenir la taille de ses sous-parties. L'interacteur de celui-ci peut par exemple déformer les sous-parties, puis diviser celles ayant dépassé un certain seuil.

Ultimement, si aucune restauration de la cohérence ne peut être trouvée, l'interacteur peut finalement revenir à son état antérieur à l'application de la déformation. Ainsi, dans le cas des objets très contraints par le maintien de leur cohérence, il peut s'avérer difficile de maintenir celle-ci, ou

alors ceci se fait au détriment de la fidélité à la déformation initiale. Ces objets ne sont donc *a priori* pas adaptés aux graphes d'interaction.

#### 5.5. Propriété de déformation

Un des avantages de l'utilisation d'un graphe d'interaction pour la définition d'un comportement de déformation est qu'il est relativement aisé de montrer que celui-ci respecte certaines propriétés. Le cas que nous avons jusqu'à présent longuement discuté est celui du maintien de la cohérence des éléments constitutifs de l'objet. Il est cependant envisageable de considérer que d'autres propriétés de la déformation soit souhaitée.

Par exemple, dans [LSWW11], il est avancé qu'un outil de déformation se doit d'être *local* et *réversible*. Dans certains contextes, il pourrait être également souhaitable d'avoir une déformation *continue* [ALX\*14], ou bien préservant le volume [ACWK04] par exemple. Il est aisé de constater que si tout les interacteurs constituant un graphe d'interaction respectent ces propriétés, le comportement décrit par le graphe tout entier les respectera également.

Une exception est à noter. Dans le cas particulier de la localité d'une déformation, celle-ci se définit relativement à l'échelle de l'élément considéré. Ainsi, lors d'une déformation interprétée de manière ascendante, on utilise une notion de localité de plus en plus large, ce qui va à l'encontre de la définition de déformation locale. La localité d'un comportement de déformation ne peut donc être assuré que pour un graphe d'interaction respectant la hiérarchie de l'objet.

### 6. Résultats

Afin de mettre à l'épreuve les idées de cet article, nous avons créé un logiciel qui propose de déformer des objets hiérarchiques auxquels nous avons associé des graphes d'interaction, grâce à une métaphore de type sculpture.

#### 6.1. Intégration de méthodes de l'état de l'art

Comme nous l'avons mentionné plus haut (partie 1), le cadre que nous avons défini permet d'inclure certaine méthode de déformation de l'état de l'art. Pour ce faire, nous reformulons le coeur de la méthode considérée sous la forme d'un graphe d'interaction.

L'intégration d'une méthode pré-existante dans le cadre des graphes d'interaction se fait en plusieurs étapes, que nous décrivons ici dans le cadre des maillages quasi-uniformes de [SCC11].

Premièrement, il faut décrire l'objet à déformer sous la forme d'une hiérarchie, comme nous l'avons fait dans la figure 7a. Remarquons que nous aurions pu définir un maillage sous la forme classique : maillage, face, arête, puis sommet ;

cependant la version simplifiée que nous avons adopté suffisait pour notre objectif. Cette description n'inclue pas tout les éléments du maillage (qui contient bien-sûr des faces), mais constitue une interface pour la définition de graphe d'interaction.

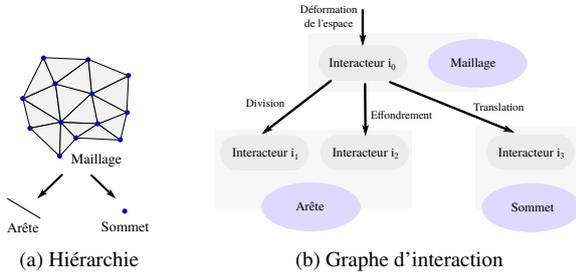


Figure 7: La méthode de [SCC11] a pu être décrite sous la forme d'un graphe d'interaction (b) associé à un objet décrit pas une hiérarchie (a).

Ensuite, il faut identifier le type de déformation que va subir l'objet : dans notre cas, il s'agit d'une déformation arbitraire de l'espace. L'interacteur associé au maillage devra donc pouvoir interpréter ce type de déformation.

Il s'agit maintenant d'identifier le type de déformation que devront pouvoir interpréter les sous-éléments, à savoir les arêtes. Pour ce faire, notons que la déformation de maillages quasi-uniforme se déroule en trois temps :

1. *déplacement* des points contenus dans les sommets
2. *effondrement* des arêtes plus courtes qu'un seuil  $d_{min}$
3. *division* des arêtes plus longues qu'un seuil  $d_{max}$

Nous en déduisons qu'il est nécessaire pour un sommet de pouvoir se déplacer (étape 1) ; les arêtes quant à elles doivent pouvoir se diviser et s'effondrer (étape 2 et 3).

Nous aboutissons ainsi au graphe d'interaction de la figure 7b. Le comportement de déformation obtenu est bien conforme à celui attendu d'un maillage quasi-uniforme, et nous a permis de créer les résultats de la figure 8.

Dans l'état actuel, nous avons uniquement partiellement intégré le travail de [SCC11] : en effet, la structure que nous décrivons ne gère pas la fusion d'éléments n'étant pas adjacents dans le maillage (et ne permet donc pas le changement de genre topologique de l'objet). Cette extension pourrait elle aussi s'exprimer à l'aide d'un graphe d'interaction. De même, plusieurs autres travaux que nous avons présenté dans cet article pourraient se retrouver ainsi unifiés.

## 6.2. Comportements de déformation hiérarchiques

Nous avons implémenté différents comportements de déformations afin d'apporter une preuve de concept de notre approche. Pour cela, nous considérons deux cas spécifique: un cas de distribution de sphères, et un cas d'arrangement de segments sous formes d'arbres.

**Distribution de sphères.** Le premier exemple propose une modélisation simple, mais permettant de tester aisément différents comportements. Pour cela, nous considérons une distribution de sphères dans l'espace. La hiérarchie de l'objet contient deux niveaux : un objet conteneur est le parent de toutes les sphères de la distribution.

L'utilisateur peut générer des déformations spatiales interactivement à la souris. Ces déformations sont interprétées par l'interacteur du conteneur qui traduit ce champs en une translation de chaque sphère (cf. figure 9a). Cet interacteur à pour objectif de préserver l'orientation et la forme des éléments du conteneur.

Ensuite, l'interacteur du conteneur fusionne les sphères qui sont en collision. Ceci permet de préserver une propriété de cohérence de l'objet qui exclue les intersection entre sous-éléments.

Il est intéressant de noter qu'on peut associer aux sphères différents interacteurs d'interprétation de la fusion. Nous avons créé un interacteur qui remplace les deux sphères initiales par une sphère dont les caractéristiques (rayon, couleur, position) sont interpolées. Cet interacteur essaye de maintenir l'information contenue dans les sphères

Ensuite, nous avons également créé un interacteur qui génère une sphère dont le volume est égal à la somme des volumes des sphères initiales. Tout les interacteurs des sphères préservent leur volume comme expliqué à la section 5.5, la distribution de sphère est donc déformée à volume constant dans son ensemble.

**Arbre.** Le second exemple est celui d'un modèle d'arbre déformable. Notre but ici n'est pas de présenter une méthode d'édition d'arbre telle que proposé par Longay et al. [LRBP12], dont le but est spécifiquement de déformer des arbres, mais d'explorer les possibilités offertes par les graphes d'interaction sur un cas concret.

Nous définissons un arbre sous la forme d'un ensemble de branches. Une branche est représentée par un cylindre à bout arrondi ; ses enfants dans la hiérarchie de l'objet sont ses sous-branches. On voit que la hiérarchie de l'objet n'a pas une taille fixe comme précédemment, mais qu'un contraire elle contient autant de niveaux qu'il y a de générations de branches.

L'arbre initial est généré par un L-system, que l'utilisateur peut déformer similairement au cas précédent des sphères. Le même interacteur est associé à toutes les branches : celui ci déplace les extrémités du cylindre associé à la branche, puis calcule une transformation rigide qu'il transmet à toute les sous-branches pour les maintenir connectées (cf. figure 9b).

La figure 11 montre différents résultats obtenus à partir du même arbre et de la même déformation initiale, pour différents comportements de déformation.

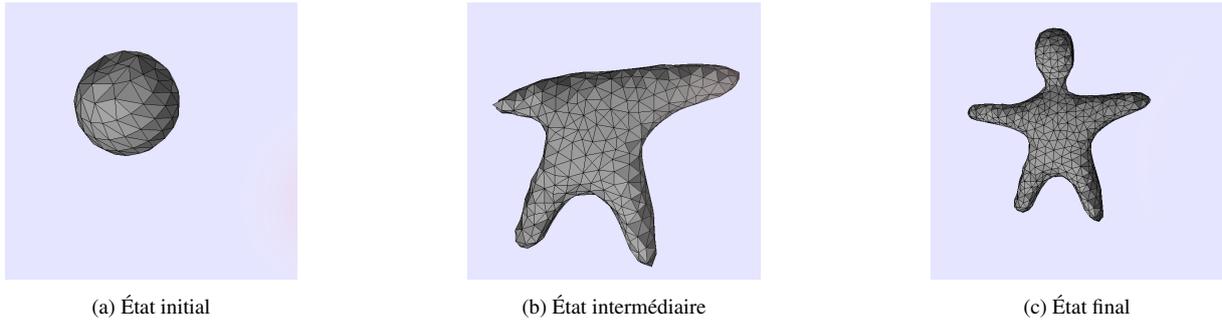


Figure 8: Un personnage sculpté grâce à un graphe d'interaction reproduisant le comportement de Freestyle [SCC11].

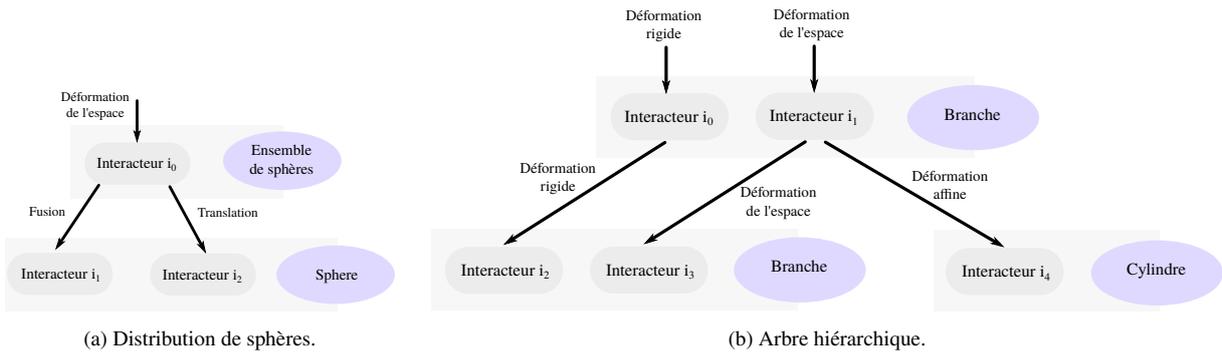


Figure 9: Les graphes d'interaction associés aux objets.

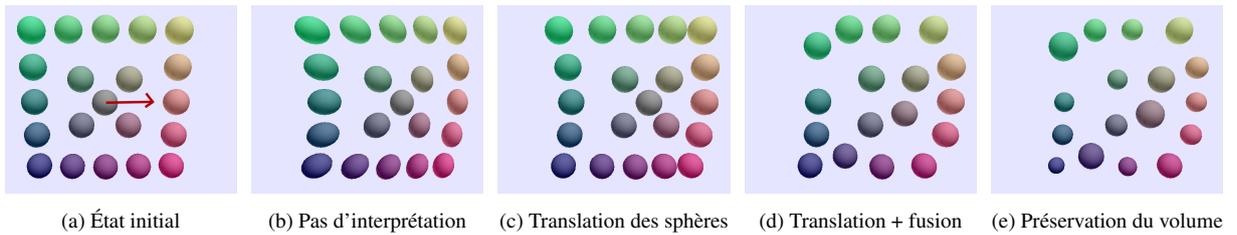


Figure 10: Différentes déformations appliquées sur un ensemble de sphères.

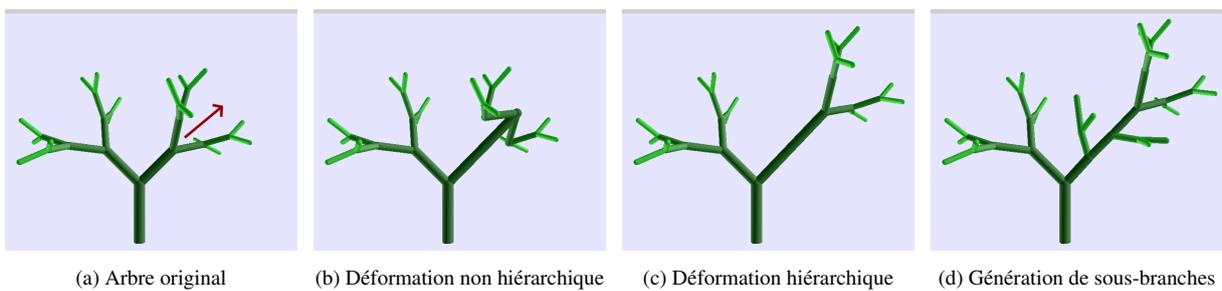


Figure 11: Différents comportements de déformations sont obtenus par l'utilisation de graphes d'interaction différents.

Notons que, comme nous l'avons souligné à la section 5.5, les comportements des figures 11c et 11d ne sont pas locaux. Ceci est dû à la structure hiérarchique de l'arbre telle que

nous la définissons : les enfants d'une branche ne sont pas contenus dans la branche. Ainsi la transmission de la défor-

mation d'une branche à ses enfants ne permet pas d'assurer la localité de la déformation.

### 6.3. Comportements de déformation bi-directionnels

Comme mentionné en partie 4.3, la création dans le graphe d'interaction de connexion non contraintes par la hiérarchie de l'objet permet une remonté de l'information de déformation dans l'objet impliquant un aspect que nous désignons par *bi-directionnel*. L'avantage de cette approche est de permettre de manipuler directement les sous-éléments de l'objet tout en maintenant la cohérence de celui-ci. Nous avons mis en place de tels graphes d'interaction dans les deux exemples présentés précédemment.

Ainsi dans le cas le cas des distributions de sphères, il est possible à l'utilisateur de manipuler directement une sphère tout en déléguant la gestion de la fusion et de la division des sphères au conteneur de sphères pour maintenir le critère de distances inter-sphères.

Dans le cas des arbres, un graphe d'interaction bi-directionnel permet à l'utilisateur de manipuler directement des branches de l'arbre, ce qui présente l'avantage de pouvoir contrôler celles-ci séparément (ce qui est peu aisé à faire avec un champ de déformation). Ce faisant, les sous-branches s'adaptent à la déformation de leur parent, ce qui permet de préserver la cohérence globale de la branche.

Par ailleurs, la possibilité de créer de la remonté de déformation dans un objet amène à d'autres essais. Par exemple dans le cas des arbres, lorsqu'on déplace une branche il peut être intéressant de transmettre une partie de ce déplacement au parent. Ce dernier va appliquer le déplacement, puis en transmettre à son tour une fraction à son propre parent. On observe alors un déplacement de tous les sous-éléments de l'objet. Dans notre cas, nous avons choisi d'atténuer l'effet du déplacement par une fonction de type exponentielle décroissante suivant la distance dans la hiérarchie.

On observe que dans ce cas, la remonté de déformation ne vise plus simplement à maintenir la cohérence de l'objet, mais à créer une déformation hiérarchique non contrainte par la hiérarchie de l'objet. Notons que ce type de déformation pourrait servir à décrire des comportements de l'état de l'art tels que ARAP [SA07].

### 6.4. Modification dynamique du comportement de déformation

Nous avons vu que les graphes d'interaction offrent la possibilité de créer des comportements de déformation relativement simplement. Or pour un même objet, il peut y avoir différents comportements souhaitables. Selon l'utilisateur, la phase de l'édition, ou un besoin spécifique, il peut être intéressant d'offrir des comportements de déformation dédiés.

Nous avons mis en place cette idée dans un logiciel de

sculpture d'arbre à comportement variable. Dans cette application, un arbre est décrit selon la même hiérarchie que celle présentée précédemment.

L'utilisateur a la possibilité de paramétrer dynamiquement le comportement de déformation grâce à des interactions au clavier, tandis que la sculpture se fait à la souris. Les paramètres qui peuvent être gérés par l'utilisateur sont les suivants :

- comportement local / non local (héréditaire)
- interprétation de la déformation tel qu'un changement de rayon (héréditaire) de la branche
- activation / désactivation de la fusion de branches
- activation / désactivation de la division de branches
- activation / désactivation de la génération de sous-branches lors de la division
- remonté de la déformation au parent

Cette application a permis à un artiste 3D de réaliser des modèles variés tels que ceux présentés en figure 12. Cette utilisation par un artiste d'un outil basé sur les graphes d'interaction confirme l'applicabilité du travail présenté dans cet article. Plus spécifiquement, l'artiste a apprécié la possibilité de naviguer entre un comportement local et un comportement hiérarchique, ainsi que la possibilité d'interagir avec l'arbre tout entier, ou bien simplement avec les branches.

## 7. Limitations

Certains objets ne se prêtent pas à l'interprétation de déformation. Il y a plusieurs raisons à cela.

D'une part la cohérence de certains objets peut être trop compliquée à traduire en terme d'interacteurs. Le cas se présente pour des propriétés de cohérence abstraite, difficiles à traduire en termes géométriques. Par exemple, préserver la cohérence de vêtements développable en collision sur un personnage virtuel est très difficile à préserver lors de la déformation [BSBC12].

L'estimation de la cohérence d'un objet doit également être rapide à calculer, sinon le problème peut rapidement se retrouver intraitable en temps interactif.

Enfin, certains objets peuvent au contraire posséder un critère de cohérence simple à exprimer, mais dont les contraintes sont trop fortes pour permettre la déformation par sculpture. C'est typiquement le cas des modèles de bâtiments, pour lesquels nos essais de création d'un graphe d'interaction n'ont pas abouti. Le maintien de l'alignement des éléments, de leur non intersection, des angles droits, limite énormément la variété des déformations que l'on peut apporter à un tel modèle. Dans ce cas, la stratégie optimale semble d'identifier les degrés de liberté de la forme et de limiter toute les transformations à des opérations sur ces paramètres (cf. [BWSK12]).

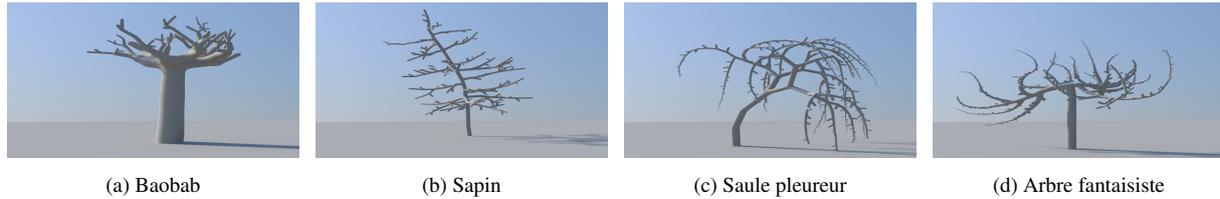


Figure 12: Une série d'arbres créés grâce à un comportement de déformation paramétrable dynamiquement. Dans ces images, la structure de l'arbre a servi comme squelette de convolution pour le rendu.

## 8. Perspectives

Dans cet article nous avons proposé un nouveau cadre pour la modélisation géométrique, qui peut être étendu dans de multiples directions.

### 8.1. Édition itérative

Lorsqu'on déforme un objet hiérarchique, nous sommes contraints de procéder depuis l'échelle la plus grossière vers la plus fine. En effet, si on apporte des modifications sur un élément, puis sur son parent, la première modification risque d'être écrasée par la seconde. Cette ordre forcé des opérations va à l'encontre du processus de création artistique, qui est de nature itérative.

Avec un graphe d'interaction, il est possible de mettre au point un comportement de déformation qui préserve les éléments qui ont été apportés par l'utilisateur. Pour ce faire, il faudrait marquer tout les éléments ayant été déformés par l'utilisateur, et créer un graphe d'interaction dont tout les interacteurs génèrent une déformation préservant au mieux les éléments marqués. Par exemple, lors de la fusion de deux éléments, l'élément résultant pourrait être une copie de celui des éléments originaux étant marqué.

La difficulté de cet axe est la gestion des cas limites : que faire lorsque la préservation de tout les éléments marqués est impossible ?

### 8.2. Sculpture de contenu animé

Joardao et al. [JPCC14] proposent de sculpter du contenu animé sous la forme de patch spatio-temporels, avec une méthode basée sur l'approche de Milliez et al. [MWCS13]. De manière analogue, du contenu animé pourrait être associé à un graphe d'interaction pour décrire un comportement de déformation adapté.

### 8.3. Représentation et compression d'objets

Dans le domaine de la compression, un principe bien connu consiste à encoder non pas un état donné mais le déplacement qu'il a fallu pour arriver à cet état depuis l'état antérieur. De manière similaire, un objet pourrait être représenté par un objet de base associé à un graphe

d'interaction, et l'ensemble des déformations qu'il a fallu pour déformer l'objet de base en l'objet final. Ainsi un comportement de déformation pourrait être utilisé pour représenter un objet donné sous la forme de l'ensemble des déformations qui l'ont engendré.

Notons qu'il s'agit d'un cas particulier de compression de Kolmogorov, où le modèle procédural est la grammaire de déformation, et le mot encodant l'objet est la séquence de symbole-déformations.

### 8.4. Problème inverse

Étant donné que nous avons ici décrit un modèle procédural hiérarchique de déformation, il pourrait s'avérer intéressant d'étudier le problème inverse : la modélisation de comportement de déformation à partir d'une base de donnée d'exemples. Cette base de donnée pourrait être utilisée soit pour recréer le graphe d'interaction.

## 9. Conclusion

Dans ce travail, nous avons introduit la notion d'interprétation de déformation. Associé à un objet, un comportement de déformation consiste à projeter les déformations subies par l'objet sur un sous-espace de déformations préservant certaines propriétés.

Nous avons créé le concept de graphe d'interaction, qui permet de décrire par morceaux le comportement de déformation d'un objet hiérarchisé. Nous avons démontré la faisabilité de notre approche sur différents exemples, et exploré différentes pistes : nous avons montré que certains travaux de l'état de l'art rentraient dans le cadre des graphes d'interaction ; nous avons créé le premier comportement de déformation hiérarchique, dans le cas d'un arbre ; enfin, nous avons montré la possibilité et l'intérêt de comportements bi-directionnels et de modification dynamique de comportement.

En conclusion, nous constatons que la modélisation de comportement de déformation est une piste de recherche prometteuse en modélisation 3D.

## Références

- [ACWK04] ANGELIDIS A., CANI M.-P., WYVILL G., KING S.: Swirling-sweepers: Constant-volume modeling. *Graphical Models, Proceedings of Pacific Graphics* (2004).
- [ALX\*14] ALHASHIM I., LI H., XU K., CAO J., MA R., ZHANG H.: Topology-varying 3d shape creation via structural blending. *ACM Transactions on Graphics (TOG)* (2014).
- [AZL12] ALHASHIM I., ZHANG H., LIU L.: Detail-replicating shape stretching. *The Visual Computer* (2012).
- [BSBC12] BROUET R., SHEFFER A., BOISSIEUX L., CANI M.-P.: Design preserving garment transfer. *ACM TOG, Proceedings of Siggraph* (2012).
- [BWSK12] BOKELOH M., WAND M., SEIDEL H.-P., KOLTUN V.: An algebraic model for parameterized shape editing. *ACM TOG, Proceedings of SIGGRAPH* (2012).
- [DK14] DEKKERS E., KOBBELT L.: Geometry seam carving. *Computer-Aided Design. Vol. 46* (2014), 120–128.
- [EPCV14] EMILIEN A., POULIN P., CANI M.-P., VIMONT U.: Interactive procedural modelling of coherent waterfall scenes. *Computer Graphics Forum* (2014).
- [GGG\*13] GÉNEVAUX J.-D., GALIN E., GUÉRIN E., PEYTAIE A., BENEŠ B.: Terrain generation using procedural models based on hydrology. *ACM TOG. Proceedings of SIGGRAPH* (2013).
- [GSMCO09] GAL R., SORKINE O., MITRA N. J., COHEN-OR D.: iwires: an analyze-and-edit approach to shape manipulation. *ACM TOG, Proceedings of SIGGRAPH* (2009).
- [JPCC14] JORDAO K., PETTRÉ J., CHRISTIE M., CANI M.-P.: Crowd sculpting: A space-time sculpting method for populating virtual environments. *Computer Graphics Forum, Proceedings of Eurographics* (2014).
- [KSSCO08] KRAEVOY V., SHEFFER A., SHAMIR A., COHEN-OR D.: Non-homogeneous resizing of complex models. *ACM TOG, Proceedings of SIGGRAPH* (2008).
- [LK11] LARS KRECKLAU L. K.: Procedural modeling of interconnected structures. *Computer Graphics Forum, Proceedings of Eurographics. Vol. 30, Num. 2* (2011), 335–344.
- [LRBP12] LONGAY S., RUNIONS A., BOUDON F., PRUSINKIEWICZ P.: Treesketch: interactive procedural modeling of trees on a tablet. *Proceedings of the international symposium on sketch-based interfaces and modeling* (2012).
- [LSWW11] LIPP M., SCHERZER D., WONKA P., WIMMER M.: Interactive modeling of city layouts using layers of procedural content. *CGF, Proceedings of Eurographics* (2011).
- [LVW\*15] LIU H., VIMONT U., WAND M., CANI M.-P., HAHMANN S., ROHMER D., MITRA N. J.: Replaceable substructures for efficient part-based modeling. *CGF, Proceedings of Eurographics* (2015).
- [LWW08] LIPP M., WONKA P., WIMMER M.: Interactive visual editing of grammars for procedural architecture. *ACM Transactions on Graphics (TOG). Proceeding of ACM SIGGRAPH. Vol. 27, Num. 3* (2008).
- [MWCS13] MILLIEZ A., WAND M., CANI M.-P., SEIDEL H.-P.: Mutable elastic models for sculpting structured shapes. *CGF, Proceedings of Eurographics* (2013).
- [MWH\*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM TOG, Proceedings of SIGGRAPH* (2006).
- [ÖG12] ÖZTIRELI A. C., GROSS M.: Analysis and synthesis of point distributions based on pair correlation. *ACM TOG. Proceedings of SIGGRAPH* (2012).
- [SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. *Symposium on Geometry Processing* (2007).
- [SCC11] STANCULESCU L., CHAINE R., CANI M.-P.: Freestyle: Sculpting meshes with self-adaptive topology. *Computers & Graphics* (2011).
- [SCCS13] STANCULESCU L., CHAINE R., CANI M.-P., SINGH K.: Sculpting multi-dimensional nested structures. *Computers & Graphics* (2013).
- [vFTS06] VON FUNCK W., THEISEL H., SEIDEL H.-P.: Vector field based shape deformations. *ACM TOG, Proceedings of SIGGRAPH, Proceedings of ACM SIGGRAPH* (2006).
- [WOD09] WHITING E., OCHSENDORF J., DURAND F.: Procedural modeling of structurally-sound masonry buildings. *ACM Transactions on Graphics (TOG)* (2009).