

Génération intelligente de modèles procéduraux par prospection  
des paramètres du L-système

par

Andrianjafy SALOMON

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
COMME EXIGENCE PARTIELLE À L'OBTENTION  
DE LA MAÎTRISE EN GÉNIE DES TECHNOLOGIES  
DE L'INFORMATION

M. Sc. A.

MONTRÉAL, LE 20 FÉVRIER 2017

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC



Andrianjafy Salomon, 2017



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

**PRÉSENTATION DU JURY**

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Eric Paquette, Directeur de Mémoire  
Département de génie logiciel et des TI à l'École de technologie supérieure

M. Luc Duong, Président du Jury  
Département de génie logiciel et des TI à l'École de technologie supérieure

M. Hervé Lombaert, membre du jury  
Département de génie logiciel et des TI à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 25 JANVIER 2017

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE



## AVANT-PROPOS

### Notes sur les droits d'auteur

Afin de refléter l'état de l'art, certaines figures ont été tirées dans des publications antérieures. Ainsi pour chacune des figures concernées, les sources sont indiquées dans la légende, avec la référence. Ci-après les notes de droit d'auteur pour les extraits tirés d'ACM :

Copyright © 2015 by the Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of portions of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page in print or the first screen in digital media. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Send written requests for republication to ACM Publications, Copyright & Permissions at the address above or fax +1 (212) 869-0481 or email [permissions@acm.org](mailto:permissions@acm.org).

Et les notes de droit d'auteur pour Eurographics Association :

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than Eurographics must be honored.



## REMERCIEMENTS

Ce travail ne se veut de loin être l'œuvre de son seul auteur, car il ne put se réaliser sans les diverses institutions dans son contexte. En premier lieu, l'École de technologie supérieure, qui, citons-le, a maintenu son appui après les prolongations du projet de recherche. Je citerai dans un second lieu le réseau de recherche multidisciplinaire GRAND-NCE, pour leur soutien financier, mais aussi pour l'édification de balises pour ce projet durant les conférences annuelles.

Mes profondes reconnaissances vont également à mon directeur de recherche, le Professeur Eric Paquette, qui, tout au long du projet, n'a jamais arrêté de porter assistance. Ses multiples recommandations ont été riches du point de vue théorique, mais ses consignes ne se sont pas limitées sur le plan scientifique, car il m'a été d'un support inestimable pendant les moments clés dans mon parcours.

Mes remerciements vont ensuite à toutes les personnes qui ont contribué au projet de recherche en participant aux expériences. Les mérites vont aussi à mes collègues du laboratoire multimédia pour leurs maintes assistances, entre autres pour les analyses de données d'expérimentation. Je manifeste ma gratitude finalement à ma famille et à celle qui a répondu favorablement à mes sollicitations, car sans elles, le chemin à parcourir aurait paru beaucoup plus long.

Maman, tu nous as quittés il y a peu, avant même que je n'aie pu clore le présent rapport. Cette ligne est pour toi, merci, pour tout.



# **GÉNÉRATION INTELLIGENTE DE MODÈLES PROCÉDURAUX PAR PROSPECTION DES PARAMÈTRES DU L-SYSTÈME**

Andrianjafy SALOMON

## **RÉSUMÉ**

La modélisation procédurale est la solution alternative d'excellence à l'utilisation excessive des ressources à la fois en terme humain d'infographiste 3D mais aussi en ressources matérielles. Les nouvelles techniques des deux dernières décades répondent assez bien aux besoins requis dans la modélisation de scènes complexes. Le contrôle sur les objets modélisés constitue un constant défi afin d'éviter les motifs récurrents engendrés par les L-systèmes appliqués à la modélisation d'arbres. Dans cet esprit, ce mémoire propose un ensemble d'outils procéduraux permettant d'éditer des arbres suivant des critères perceptuels. Notre démarche consiste à s'appuyer sur une étude avec des participants afin d'automatiser l'ajustement des paramètres de L-systèmes, de sorte que les arbres résultants soient pertinents pour les utilisateurs. Sept grammaires L-système ont été utilisées pour l'étude. Les propriétés géométriques et topologiques de chaque grammaire vont constituer un système de métriques, que nous inspectons pendant les variations des paramètres des grammaires. Afin d'identifier les arbres les plus intéressants, les différences entre l'arbre initial et les arbres générés sont évaluées dans une analyse en composantes principales. Les tests ont établi une corrélation entre la pertinence des arbres générés et leurs emplacements dans l'espace des vecteurs caractéristiques des analyses de différence. Les expérimentations ont été effectuées pour acquérir des informations sur les relations entre les paramètres et les métriques, afin de proposer des modèles d'arbres pertinents pour les utilisateurs. Outre l'établissement d'un système de métriques dans le but d'évaluer et de comparer mathématiquement les arbres, ce projet contribue à la littérature en proposant une sélection intelligente basée sur les historiques de dérivation. Les utilisateurs modifient l'arbre en effectuant des sélections à partir desquelles la méthode de sélection intelligente peut inférer le nombre de régénération à faire sur chaque sous ensemble sélectionné de l'arbre. Les aperçus des modifications relatives à chaque qualificatif sont conjointement proposés aux utilisateurs pour faciliter les retouches sur l'arbre. Les résultats démontrent que notre approche permet d'identifier une riche variété d'arbres 3D en se basant sur la mesure des différences des arbres.

**Mots clés :** L-système, modélisation procédurale, modélisation procédurale interactive, sélection, métriques



# SMART GENERATION OF PROCEDURAL MODELS BY EXPLORING VARIATIONS OF L-SYSTEM PARAMETERS

Andrianjafy SALOMON

## ABSTRACT

Procedural modeling is the alternative by excellence to the excessive use of resources both in terms of graphics designer and in hardware costs. The discoveries in the last two decades meet some of the requirements of complex scene modeling. Nevertheless, the lack of flexibility and the appearance of recurring patterns in a scene are still a concern. In this context, this project suggests a set of new procedural tools which will allow users to edit 3D trees by means of interactive adjustment. Our approach relies on a study with participants to perform automatic adjustment of L-system parameters in a way that the resulting trees are meaningful to the users. Seven L-system grammars were used for the study. We derived geometric and topological metrics, which we inspected during the variations of the grammar parameters. In order to identify the most interesting trees, the differences between the initial tree and the generated trees are evaluated through principal component analysis. The tests have established a correlation between the relevance of the generated trees and their location in the characteristic features space of the difference analysis. Moreover, qualifiers assist the users on the type of available modifications on the initial tree. In addition to the establishment of a metrics system to mathematically evaluate and compare trees, this project contributes to the procedural editing by providing smart selection based on the derivation sequence. Users can edit tree parts by making a selection which is internally refined based on the derivation tree. The necessary number of regeneration on each subset of the tree is inferred from this derivation-based selection method. Also, an overview of the available edits, related to each qualifier, is available in the proposed interface. The results show that our approach can identify rich varieties of trees based on the measurement of the tree differences.

**Keywords:** L-system, procedural modeling, interactive procedural modeling, rule-based tree modeling, selection, metrics



## TABLE DES MATIÈRES

	Page
INTRODUCTION .....	1
CHAPITRE 1 ÉTAT DE L'ART .....	5
1.1 Techniques de modélisation procédurale .....	5
1.1.1 Domaines d'application .....	5
1.1.1.1 Environnement extérieur .....	6
1.1.1.2 Modélisation d'édifices .....	7
1.1.1.3 Production générique des arbres .....	8
1.1.2 Notion d'apprentissage et modélisation procédurale inverse .....	9
1.1.3 Techniques à base d'images .....	10
1.1.4 Approches sémantiques .....	11
1.1.4.1 Sémantique environnementale .....	11
1.1.4.2 Sémantique des objets .....	12
1.2 Contrôle des modèles finaux par interaction .....	12
1.2.1 Édition par graphe pondéré .....	13
1.2.2 Guidage par croquis .....	13
1.2.2.1 Notion de guide .....	13
1.2.2.2 Approche déclarative .....	14
1.2.2.3 Lobes de texture .....	15
1.2.2.4 Extrusions procédurales .....	15
1.3 Travaux sur les algèbres de la modélisation procédurale .....	17
1.3.1 Optimisation des espaces de production .....	17
1.3.2 Primitives de haut niveau .....	17
1.3.3 Édition visuelle des grammaires .....	18
1.4 Recherches sur les plantes procédurales .....	19
1.4.1 Colonisation de l'espace .....	19
1.4.2 Compétition pour la lumière .....	20
1.5 Entrée en matière sur les L-systèmes .....	20
1.5.1 Système de réécriture et interprétation de la tortue .....	21
1.5.2 Transformation géométrique .....	22
1.5.3 Formalisme des L-systèmes .....	23
1.5.4 L-systèmes paramétriques .....	24
1.5.5 Règles avec conditions .....	25
1.5.6 Traitement du prédécesseur et du successeur .....	25
1.5.7 Inspection du flot de grammaire .....	26
1.6 Problématiques et démarche de la recherche .....	26
1.6.1 Objectifs de la recherche .....	28
1.6.2 Méthodologie .....	28
CHAPITRE 2 REGÉNÉRATION DE CHAÎNES .....	31

2.1	Système de sauvegarde d'état de dérivation .....	31
2.2	Structuration en arborescence des dérivations .....	33
2.2.1	Sélection d'un élément d'historique .....	33
2.2.2	Correspondance d'une sélection avec un arbre de dérivation .....	33
2.2.3	Multisélection .....	36
2.3	Regénération .....	37
2.3.1	Redéfinition de constante .....	38
2.3.2	Regénération selective .....	39
2.4	Coupe minimum de graphe .....	39
2.4.1	Transcription de l'arbre de dérivation en graphe .....	40
2.4.2	Détermination des capacités des arêtes du graphe .....	43
2.4.2.1	Établissement des conditions initiales .....	43
2.4.2.2	Calcul des capacités des sélections pour chaque niveau .....	44
2.4.2.3	Contraintes sur la capacité initiale de sélection .....	46
2.4.3	Administration d'une contrainte de sélection .....	48
<b>CHAPITRE 3 MESURE DE DIFFÉRENCE ENTRE LES ARBRES .....</b>		55
3.1	Enveloppement des branches .....	55
3.1.1	Boîte englobante orientée .....	55
3.1.2	Calcul de la boîte orientée .....	56
3.2	Définition des paramètres de contrôle .....	58
3.2.1	Dimensions géométriques .....	59
3.2.1.1	Longueur des branches .....	59
3.2.1.2	Volume des branches .....	59
3.2.1.3	Surface des branches .....	59
3.2.1.4	Volume de la boîte .....	59
3.2.1.5	Surface de la boîte .....	60
3.2.2	Quantité d'éléments .....	60
3.2.3	Proportions .....	60
3.2.3.1	Densité de branches .....	60
3.2.3.2	Moyenne des longueurs des branches .....	60
3.2.3.3	Moyenne des rayons des branches .....	61
3.3	Variation de constantes des paramètres dans la grammaire .....	61
3.3.1	Proportion de modifications des constantes de la grammaire .....	61
3.3.2	Analyse en composantes principales sur les métriques .....	63
3.3.2.1	Format et dimensions de la population .....	63
3.3.2.2	Calcul de la moyenne et de la covariance .....	64
3.3.2.3	Vecteurs et valeurs propres .....	65
3.3.2.4	Choix des vecteurs caractéristiques .....	65
3.3.2.5	Visualisation des arbres générés pour l'analyse .....	67
3.4	Rendus des arbres nouvellement générés .....	67
<b>CHAPITRE 4 ÉTUDE EMPIRIQUE AVEC SUJET HUMAIN .....</b>		69
4.1	Désignation des grammaires à soumettre aux participants .....	69

4.2	Déroulement des expérimentations .....	71
4.3	Catégorisation des informations à recueillir .....	73
4.3.1	Groupement des images et images représentatives .....	73
4.3.2	Différences intergroupes et informations d'attribut .....	74
4.4	Observation des données sur les groupements et les images représentatives .....	75
4.4.1	Matrice de cooccurrence de groupement .....	75
4.4.2	Graphiques sur les regroupements .....	77
4.4.3	Réduction des ambiguïtés dans les matrices de cooccurrence .....	77
4.5	Pertinence des images .....	83
4.5.1	Choix de l'outil pour les tests statistiques .....	83
4.5.2	Analyse statistique d'interdépendance des grammaires .....	85
4.5.3	Analyse statistique d'interdépendance des points de grammaire .....	85
4.5.4	Analyse statistique des différences entre les points de grammaire .....	86
4.5.5	Interprétation des tests .....	88
4.6	Traitements sur les informations d'attribut .....	89
4.6.1	Collecte des qualificatifs .....	89
4.6.2	Visualisation graphique des informations .....	90
4.6.3	Collisions des qualificatifs dans l'espace PCA .....	91
4.7	Rendus des arbres regénérés à l'aide des qualificatifs .....	92
<b>CHAPITRE 5 SYNTHÈSE DES RÉSULTATS .....</b>		95
5.1	Implémentation du système .....	95
5.1.1	Technologies utilisées .....	95
5.1.2	Temps d'exécution .....	96
5.2	Conception du logiciel .....	99
5.3	Résultats et discussions .....	107
	Points forts .....	107
	Limitations .....	108
	Perspectives et recommandations .....	109
<b>CONCLUSION .....</b>		111
<b>ANNEXE I</b>	<b>SYMBOLES ET INTERPRÉTATION DE LA TORTUE .....</b>	115
<b>ANNEXE II</b>	<b>ALGÈBRE SUR LES CONDITIONS DES RÈGLES PROCÉDURALES .....</b>	117
<b>ANNEXE III</b>	<b>L-SYSTÈMES STOCHASTIQUES ET AVEC CONTEXTES .....</b>	119
<b>ANNEXE IV</b>	<b>ARCHIVAGE DES INFORMATIONS DE DÉRIVATIONS .....</b>	121
<b>ANNEXE V</b>	<b>QUESTIONNAIRE AUX PARTICIPANTS DES EXPÉRIMENTATIONS .....</b>	123

ANNEXE VI	CALCUL DES MÉTRIQUES POUR LES ANALYSES EN COMPOSANTES PRINCIPALES .....	125
ANNEXE VII	ANALYSE COMPLÉMENTAIRE SUR LA COUPE DE GRAPHE .....	127
RÉFÉRENCES BIBLIOGRAPHIQUES .....		129

## LISTE DES TABLEAUX

	Page	
Tableau 1.1	Comparatif des méthodes existantes avec notre approche .....	29
Tableau 2.1	Évolution de la chaîne à chaque dérivation .....	32
Tableau 2.2	Valeur de capacité ajoutée par niveau à partir de la feuille .....	47
Tableau 3.1	Exemple de valeur de constantes et ses variations .....	62
Tableau 3.2	Matrice de covariance calculée sur l'extrait 2.1 .....	64
Tableau 3.3	Résultat de la décomposition en vecteurs propres .....	65
Tableau 4.1	Données d'un participant sur l'utilité de chaque groupe d'images .....	74
Tableau 4.2	Différence entre les groupes selon le participant <i>105</i> .....	75
Tableau 4.3	Pertinence de chaque image selon les participants .....	84
Tableau 4.4	Test ANOVA sur la corrélation entre grammaires et pertinence .....	85
Tableau 4.5	Test ANOVA sur la corrélation entre points images et pertinence .....	86
Tableau 4.6	Les différences entre les groupes projeté à chaque image .....	87
Tableau 4.7	Test ANOVA sur la corrélation entre grammaires et différence .....	87
Tableau 4.8	Test ANOVA sur la corrélation entre points et différence .....	87
Tableau 5.1	Décompte des fichiers et lignes de code source du projet .....	95
Tableau 5.2	Résumé des coûts des sept grammaires .....	98
Tableau 5.3	Informations de coût d'exécution des PCAs sur les grammaires .....	99



## LISTE DES FIGURES

	Page	
Figure 1.1	Étapes de la modélisation inverse de routes .....	7
Figure 1.2	Génération d'environnement virtuel par <i>CityEngine</i> .....	9
Figure 1.3	Maniement de guide pour influencer un arbre procédural .....	14
Figure 1.4	Exemple de superposition de blocs architecturaux .....	16
Figure 1.5	Évolution géométrique du quadratique de Koch .....	22
Figure 1.6	État initial de la tortue dans les L-systèmes .....	23
Figure 1.7	Structure d'une règle de production.....	25
Figure 1.8	Interface utilisateur montrant les grammaires .....	27
Figure 2.1	Exemple d'arbre de dérivation .....	32
Figure 2.2	Cas d'une ascension dans un arbre d'historique.....	34
Figure 2.3	Aperçu du système de couleur TSV .....	34
Figure 2.4	Attribution des gradients de couleur lors d'une sélection d'un nœud .....	35
Figure 2.5	Relation entre la sélection et l'arbre de dérivation .....	37
Figure 2.6	Interface utilisateur montrant la sélection multiple.....	38
Figure 2.7	Sélection de feuilles sur l'arbre de dérivation .....	40
Figure 2.8	Choix des nœuds assignés aux régénérations.....	41
Figure 2.9	Aperçu d'une coupe dans un graphe orienté.....	42
Figure 2.10	Transcription d'un arbre de dérivation en un graphe orienté .....	49
Figure 2.11	Coupes $\Gamma_k$ aux niveaux $k$ .....	50
Figure 2.12	Transfert en amont de capacité des arêtes sélectionnées .....	51
Figure 2.13	Cas de sélection dans un graphe aléatoire .....	52
Figure 2.14	Illustration de toutes les coupes possibles .....	53

Figure 2.15	Personnalisation du paramètre $\rho$ de pourcentage de sélection.....	53
Figure 3.1	Illustration d'un enveloppement d'une sélection par un OBB .....	56
Figure 3.2	Descripteur de forme d'une branche .....	57
Figure 3.3	Application d'une OBB sur une branche sélectionnée .....	57
Figure 3.4	Distribution du nuage de points dans le repère des vecteurs propres .....	66
Figure 3.5	Variétés d'arbres obtenus à partir de la grammaire <i>Tree1</i> .....	68
Figure 4.1	Emplacement des 27 points d'intérêts retenus.....	70
Figure 4.2	Interface utilisée dans les phases d'expérimentation .....	72
Figure 4.3	Cooccurrence des images pour toutes les grammaires .....	76
Figure 4.4	Information de groupement des images selon chaque participant .....	78
Figure 4.5	Information de groupement des images selon les grammaires .....	79
Figure 4.6	Aperçu du groupement des images .....	80
Figure 4.7	Correspondance des points d'intérêts avec les points de grammaire .....	81
Figure 4.8	Cooccurrence des images en utilisant une nouvelle correspondance .....	82
Figure 4.9	Ajout de qualificatifs sur l'interface utilisateur.....	91
Figure 4.10	Visualisation 3D des boîtes englobantes pour chaque qualificatif .....	92
Figure 4.11	Collisions entre les boîtes englobantes des qualificatifs .....	93
Figure 4.12	Regénération de deux grammaires en utilisant les qualificatifs.....	94
Figure 5.1	Déroulement de la coupe minimum dans <i>ShowGraph</i> .....	97
Figure 5.2	Pipeline de lecture de la grammaire.....	100
Figure 5.3	Diagramme de communication entre la vue et le L-système .....	100
Figure 5.4	Aperçu et brève description de la fenêtre principale du logiciel.....	101
Figure 5.5	Diagramme de classe du système .....	102
Figure 5.6	Diagramme de séquence portant sur la regénération par qualificatif .....	103

Figure 5.7	Regénération des deux grammaires <i>Tree2</i> et <i>Ternary Branching</i> .....	105
Figure 5.8	Regénération intégrale de l'arbre de la grammaire <i>Tree3</i> .....	106



## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ANOVA	Analyse de variance ( <i>ANalysis Of VAriance</i> )
CGLA	Algèbre linéaire pour l'infographie ( <i>Computer Graphics Linear Algebra</i> )
ETS	École de technologie supérieure
L-système	Système de Lindenmayer
OBB	Boîte englobante orientée ( <i>Oriented Bounding Box</i> )
PCA	Analyse en composantes principales ( <i>Principal Component Analysis</i> )
TSV	Teinte saturation valeur ( <i>Hue, Saturation, Value, HSV</i> )
UML	Langage de modélisation unifié ( <i>Unified Modeling Language</i> )
XML	Langage de balisage extensible ( <i>Extensible Markup Language</i> )



## INTRODUCTION

### Enjeux de la modélisation procédurale

Les méthodes procédurales sont indéniablement un des meilleurs choix actuellement disponibles pour la production d'objets 3D complexes. Ces méthodes utilisent une procédure ou une fonction pour produire des contenus riches. Les possibilités ne cessent de s'accroître pour ne citer que les travaux de Longay *et al.* (2012) qui proposent un outil aussi ergonomique que les tablettes pour réaliser la modélisation procédurale.

### Fondement des grammaires formelles

Prusinkiewicz (1986) a amplement contribué aux connaissances sur la génération procédurale d'arbres grâce à ses travaux de formalisation des grammaires L-système. Son concept repose sur la génération de chaînes en utilisant le L-système, et d'interpréter ces mêmes chaînes en tant qu'une suite de commandes qui ont des significations géométriques traduites par une *tortue*. Cette méthode permet de dessiner notamment des fractales, mais à l'époque elle parvenait déjà à modéliser des arbres avec un bon niveau de réalisme.

### Perspectives d'interactivité et niveaux de détails accentués

Avoir un objet 3D, aussi réaliste soit-il, n'est pas une finalité absolue, mais plutôt une étape vers un rendu pertinent qui répond à des attentes de l'utilisateur. L'interactivité apporte une solution convenable selon l'approche proposée. Lintermann & Deussen (1999) ont misé sur le caractère intuitif de l'interface, en offrant un éventail de paramétrages, allant du facteur géométrique jusqu'à la déformation libre de formes. Dans la même lignée, Aliaga *et al.* (2007) se penchent sur les textures et proposent un algorithme d'analyse sur les couleurs diffuses et les transparences, en permettant aux utilisateurs de calquer de manière interactive les caractéristiques des autres édifices.

## Origine des applications aux arbres

Les travaux de Oppenheimer (1986) furent parmi les premiers à s'intéresser aux arbres. Les recherches en ce sens ont été nombreuses, sans doute à cause du fait qu'il n'est tout simplement pas envisageable de faire appel à chaque fois à un graphiste pour un objet aussi complexe et générique que les arbres. Plus récemment, Runions *et al.* (2007) exploitent le domaine de la colonisation de l'espace pour créer des modèles d'arbres variés. La colonisation de l'espace est une branche de la modélisation procédurale d'un arbre qui consiste à remplir méthodiquement les zones délimitées par le volume englobant l'arbre. Elle gère entre autres la collision des branches et autres éléments situés sur l'arbre. La colonisation de l'espace prend en compte la lumière qui se propage dans les parties intérieures de l'arbre. Des travaux comme celui de Palubicki *et al.* (2009) simulent des phénomènes biologiques de développement autonome tels les fléchissements et les coupures des branches.

## Cadre et problématiques de la recherche

L'interactivité peut être perçue comme un outil qui aide l'utilisateur à parvenir à l'objet 3D recherché. Cependant, cet objet recherché peut être parfois abstrait et imprécis. Il devient alors avantageux d'offrir des options de modification aux utilisateurs afin de pouvoir éditer un arbre existant. L'interactivité peut alors aboutir à un résultat encore plus intéressant si des modifications peuvent s'appliquer à des sous-ensembles éventuellement disparates ou encore dispersés dans l'arbre. Enfin, pour sélectionner des sous-ensembles dispersés, nous allons nous appuyer sur l'information du moment du développement auquel ces sous-ensembles se sont développés dans l'arbre. Les sélections pourront alors s'effectuer par prospection des stades de développement.

## **Contributions et organisation du document**

Ce projet propose un mécanisme de sauvegarde des historiques de dérivation, qui retrace l'apparition de nouveaux éléments dans l'arbre. Dans ce système, aucun élément n'est ignoré, même les transformations géométriques sont transcris dans l'arborescence des historiques. Ce mécanisme est exploité pendant la sélection d'un élément ou d'un sous-ensemble. Les éléments sélectionnés feront l'objet d'une analyse, où notre mécanisme d'interactivité apportera, en plus des modifications courantes de forme, des suggestions d'affinement du modèle 3D final. Cet ouvrage a été divisé comme suit : le premier chapitre visite l'état de l'art, les grammaires L-système sont approfondies dans le second chapitre, le troisième chapitre s'attaque à la régénération d'un élément ou d'un sous-ensemble, le quatrième chapitre incorpore les métriques et leur mode d'opération, le cinquième chapitre décrit l'étude empirique avec les sujets humains, les résultats sont montrés dans le sixième chapitre, et nous terminerons enfin par une conclusion sur les travaux.



## CHAPITRE 1

### ÉTAT DE L'ART

Les champs d'application de la modélisation procédurale sont très vastes et ultimement variés. Reeves (1983) avait axé ses recherches sur la modélisation des objets flous tels la fumée et le feu en utilisant les systèmes de particule qui sont des représentations procédurales stochastiques contrôlés par des paramètres. Gagnon & Paquette (2011) ont étudié les formations des glaçons en utilisant les L-systèmes pour en définir les trajectoires. Les modélisations urbaines sont également un des thèmes les plus étudiés comme dans la publication de Watson *et al.* (2008) qui offre un aperçu sur l'exploitation des grammaires formelles dans la littérature de la modélisation procédurale. L'application aux plantes a commencé très tôt avec les travaux de Prusinkiewicz (1986), et continue d'attirer beaucoup de chercheurs dont parmi les plus récents Št'ava *et al.* (2014). Chacune de ces spécialisations favorise une technique de modélisation procédurale que nous allons résumer dans les prochaines sections.

#### 1.1 Techniques de modélisation procédurale

Les grammaires de forme, les grammaires de chaîne, ou encore les L-systèmes sont les fonctions les plus communes reliant les méthodes procédurales. Elles constituent un dénominateur commun pour les méthodes évoluées comme l'intégration d'interface visuelle d'édition de grammaire, l'apprentissage automatisé de grammaire, les librairies de règle, etc. Un bon aperçu de ces techniques peut s'obtenir en passant en revue les domaines d'application.

##### 1.1.1 Domaines d'application

Les domaines d'application sont nombreux, au point qu'il serait très difficile d'en fournir une liste exhaustive. Dans cette section, nous allons nous attarder sur des thèmes communs qui contiennent globalement les approches les plus connues en modélisation procédurale.

### 1.1.1.1 Environnement extérieur

La démarche la plus courante pour la modélisation d'un environnement extérieur à grande échelle comporte une subdivision du contenu virtuel en plusieurs structures environnementales traitées distinctement : la structure aquatique qui est la répartition des eaux, les réseaux de route et enfin les édifices ou constructions. Deux exemples de ces approches sont développés dans les prochains paragraphes.

Chen *et al.* (2008) se sont attaqués à la modélisation d'un réseau de routes en utilisant des champs de tenseur pour guider la génération du réseau de route. Leur système prend comme paramètres quatre cartes sous un format d'images. La première carte contient des plans sur les cheminements des eaux (figure 1.1a). La deuxième carte situe les parcs et les forêts. La troisième carte est une carte d'élévation montrant les dispositions des reliefs. La quatrième et dernière carte montre une densité des populations. Ensuite, l'utilisateur crée un champ de tenseur via des opérations d'édition (figure 1.1b). Les champs peuvent être créés en combinant des bases individuelles de champ, en calculant des tenseurs à partir des limites, en effectuant des rotations et en y ajoutant des bruits. Les réseaux de route sont générés grâce à des calculs de ligne aérodynamique sur les champs de tenseur (figure 1.1c). L'étape finale consiste à créer le tout en trois dimensions, en y ajoutant des bâtiments comme indiqué à la figure 1.1f. Une autre approche explorée par Galin *et al.* (2010) exploite des fonctions de pondération de coût pour relier deux points donnés en entrée. L'objectif est de calculer le chemin le plus court pour connecter un point initial et un point final. Les paramètres pour cette fonction sont les pentes constituant le terrain, ainsi que les obstacles naturels tels les forêts et les lacs. La fragmentation de la trajectoire dépend des élévations et des eaux, dans le but de déterminer le placement des tunnels et des ponts. L'algorithme cherche à convertir le problème en une optimisation de graphe implicite fini. Leur démarche consiste à effectuer une discréétisation du terrain en une grille. Un chemin est une concaténation entre des points de la grille. Ainsi pour un nombre fini de points dans la grille, la recherche du chemin continu le plus court se traduit en un calcul du plus court chemin sur un graphe fini. Finalement, la génération de la route s'effectue grâce à des modèles génériques procéduraux de route, de tunnel et de pont. Les trajectoires sont

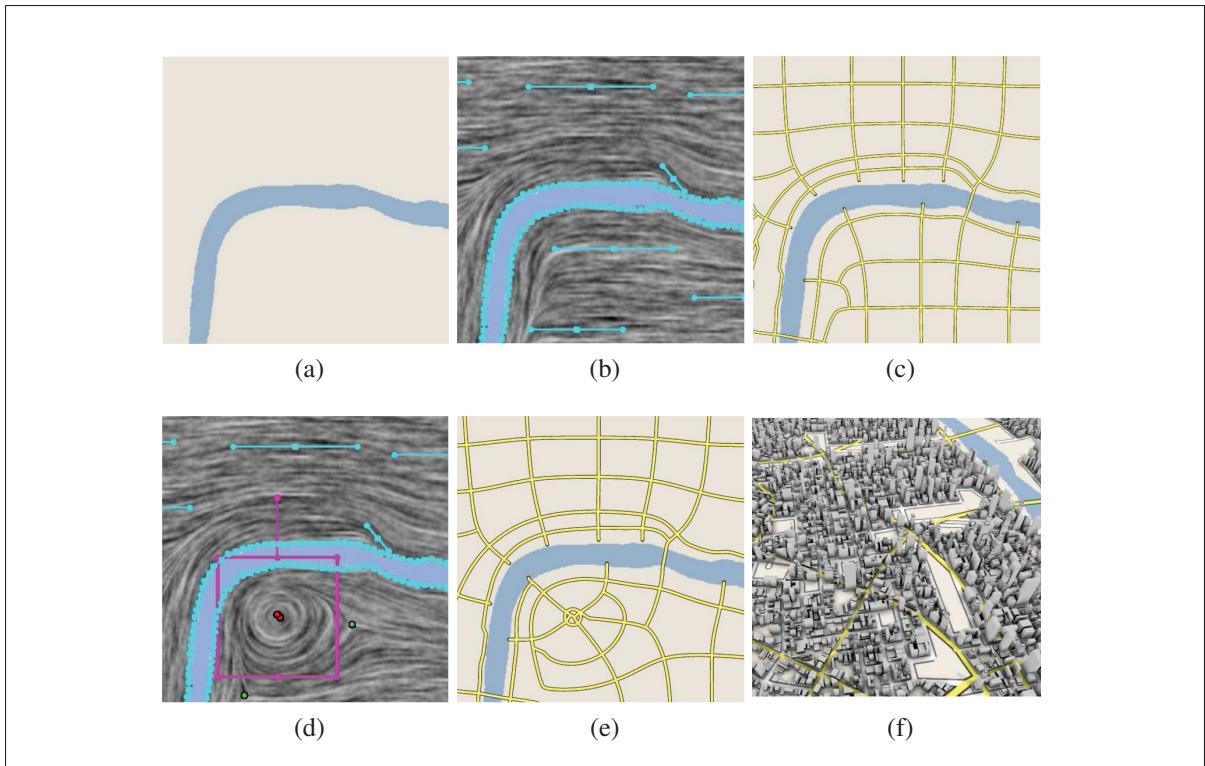


Figure 1.1 Illustration des étapes de la modélisation inverse d'un réseau de route.  
Tirée de Chen *et al.* (2008).

segmentées par échantillonnage linéaire, dans le but de déterminer quel type d'allée va être utilisé pour relier les points.

### 1.1.1.2 Modélisation d'édifices

Le L-système est la grammaire de chaîne utilisée par Parish & Müller (2001) pour générer des géométries avec des systèmes de texture à des fins de modélisation de bâtiments. Les variables d'entrée sont un ensemble de cartes sous format image. La technique cherche à produire à la sortie des réseaux routiers, divisant un environnement virtuel en plusieurs parcelles, pour créer à la fin des géométries compatibles à ces dernières. Pour parvenir à leur fin, les chercheurs ont créé des extensions aux L-systèmes qui permettent ainsi de prendre en compte des contraintes globales et locales.

Deux classes de cartes sont proposées dans la méthode intégrée dans *CityEngine* : d'abord les cartes géographiques qui constituent les élévations et les végétations, et ensuite les cartes sociostatistiques qui indiquent les paramètres de concentration telle la densité de population. La figure 1.2a est un exemple de sortie 3D obtenue à partir d'une carte sociostatistique. La génération des édifices et celle des rues sont gérées par deux L-systèmes. Les extensions du L-système implémentent un mécanisme de modules pour chaque portion de chaîne de caractères. Ces modules comprennent quant à eux des commandes avec des paramètres propres à chaque type de primitives à dessiner. Après que de nouvelles contraintes aient été ajoutées, des règles sont réécrites. Des notions de successeur idéal y sont introduites, consistant à créer des échantillons de chaîne pour chaque itération, au lieu de réécrire intégralement la chaîne de caractère courante.

Les grammaires de forme ont été utilisées par Müller *et al.* (2006), exploitant ce qu'on appelle dans la littérature la *sensibilité au contexte*. Il s'agit d'une condition booléenne de modélisation d'une forme : selon que la forme qui précède satisfasse un profil exigé, on décide de créer ou non les prochaines géométries. Le résultat est un contenu riche (figure 1.2b). Dans l'application de la grammaire de forme, des règles de production sont d'abord utilisées pour créer ce qu'on appelle le modèle de masse. Il s'agit d'un modèle volumétrique qui peut être perçu comme une esquisse pour le futur modèle 3D. Ensuite une application supplémentaire de règles de production ajoute une structure et une hiérarchie aux façades, formant ainsi des détails tels que les fenêtres et les bordures.

### 1.1.1.3 Production générique des arbres

Les approches procédurales pour la modélisation des arbres ont commencé il y a une trentaine d'années, et continuent aujourd'hui de faire l'objet d'investigations. Prusinkiewicz *et al.* (1990) furent parmi les pionniers du domaine en démontrant la pertinence de la modélisation procédurale à base de grammaires. Dans la réalité, les structures biologiques réelles comportent un nombre fini de cellules, rendant éventuellement inadéquate une analogie avec les fractales. Ils ont évoqué dans leurs études la différence entre une assimilation de plantes réelles en une

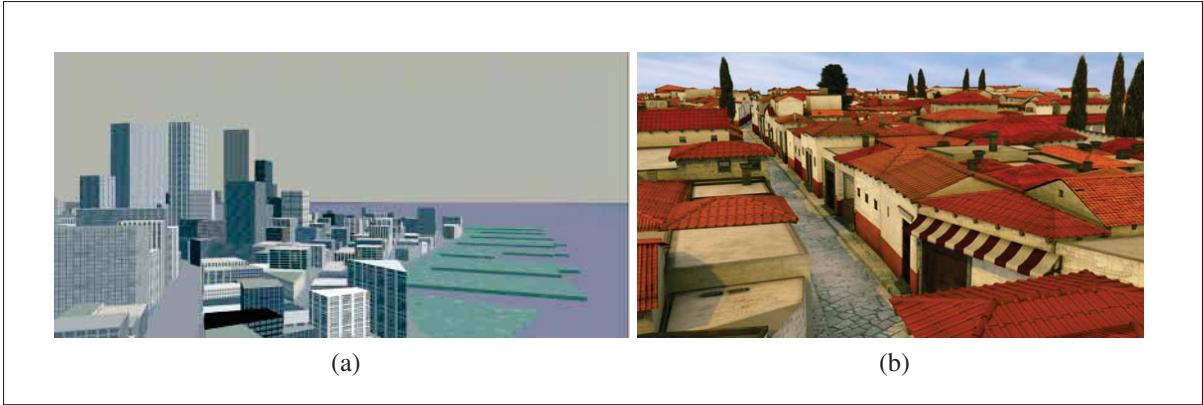


Figure 1.2 Génération d'environnement virtuel par *CityEngine*.

(a) Tirée de Parish & Müller (2001).

(b) Tirée de Müller *et al.* (2006).

approximation d'une structure fractale parfaite afin de valider les méthodes liées aux fractales. Plus récemment, Palubicki *et al.* (2009) ont travaillé sur les générations des plantes et arbustes en exploitant les compétitions pour la lumière et la colonisation des espaces.

### 1.1.2 Notion d'apprentissage et modélisation procédurale inverse

L'apprentissage dans le cadre de la modélisation procédurale inverse consiste à générer des règles procédurales grâce à une abstraction simple dans l'acquisition des entrées utilisateur, entre autres en traduisant ces entrées par des formes géométriques. C'est le fondement même de la modélisation procédurale inverse. L'apprentissage est motivé par le manque de caractère intuitif et de contrôle pendant l'écriture des règles procédurales. Les règles procédurales générées sont supposées produire d'une certaine manière des modèles qui se rapprochent des spécifications établies par les utilisateurs.

Št'ava *et al.* (2010) ont travaillé sur la génération d'un L-système paramétrique sans contexte respectant une image vectorielle fournie en entrée. Leur idée consiste à attribuer aux éléments graphiques identiques des symboles terminaux du L-système, notamment les alphabets. Cette détection de similitude s'effectue en utilisant une technique qui découpe une image en une collection de structures exiguës. Ces petites structures ont été morcelées de manière à abstraire

l'élément de sa position, de sa taille et de son orientation. Un coefficient de similarité est établi pour mesurer la ressemblance entre les éléments. Leur approche est efficace pour assister les artistes dans l'édition individuelle des règles, mais peut être moins avantageuse pour influencer l'aspect final du modèle 3D.

Une méthode proposée par Biggers & Keyser (2011) consiste à utiliser les croquis pour en inférer les contours d'un objet et modéliser des formes. Aussi, pour un croquis en forme de rectangle à l'entrée, leurs méthodes peuvent insérer des types d'objet dont l'allure peut correspondre, telle une planche de bois rectangulaire, ou encore une chaise qui rentrerait correctement dans le croquis. Ils ont démontré leur approche par la reproduction d'une scène réelle d'accumulation de débris de meubles où l'entrée fut un croquis formant une courbe fermée. Ils ont obtenu à la sortie des compositions d'objet 3D, qui sont empilés de façon à remplir la surface occupée par le croquis. Les objets 3D en sortie sont une combinaison de forme solide, arrangée de façon à rentrer correctement à l'intérieur du croquis dans une vue de dessus et de perspective.

Dans la même thématique de la modélisation procédurale inverse, Bokeloh *et al.* (2010) ont travaillé sur une méthode de conception de règles pouvant représenter un échantillon 3D fourni en entrée. Une exploration d'un voisinage local à l'intérieur de l'objet ainsi formé doit aboutir à un résultat identique à l'exploration d'un voisinage local de l'échantillon. C'est la notion de similarité locale. Pour satisfaire cette exigence, les chercheurs ont choisi d'inspecter les symétries partielles en mettant en correspondance les formes rigides. Des grammaires de forme sans contexte sont extraites de cette inspection.

### 1.1.3 Techniques à base d'images

Les techniques à base d'images trouvent leur application essentiellement dans la régénération d'édifice en 3D à partir d'un ensemble ou d'une unique façade. En ajoutant l'analyse des images aux routines de modélisation procédurale des grammaires de forme, Müller *et al.* (2007) ont pu reproduire en 3D assez fidèlement des façades d'un immeuble. Les façades sont

fournies sous forme d'une image en entrée. L'algorithme affiche ensuite comme sortie des subdivisions en étages et en carreaux de l'étage. Des informations de symétrie sont récupérées pour appuyer la mise en correspondance de points dans les carreaux. Chaque carreau identifié passe encore sous analyse récursive dans le but de découper une région, offrant ainsi les détails pour la structure des fenêtres et autres éléments présents dans la région. La prochaine étape consiste à fouiller une librairie 3D d'éléments architecturaux comme les fenêtres, afin de sélectionner l'élément le plus pertinent qui donnera un rendu similaire à l'entrée. Les subdivisions sont encodées en règles procédurales de grammaire de forme. Ces règles vont contenir des informations de hiérarchie et de dimension. Leur approche demeure toutefois sensible à la résolution de l'image d'entrée, ainsi que de la complexité de la structure.

#### 1.1.4 Approches sémantiques

Ce domaine de la modélisation procédurale cherche à dissocier l'utilisateur final des détails graphiques de la modélisation, pour diriger son attention aux liens et aux réciprocités connectant les contenus.

##### 1.1.4.1 Sémantique environnementale

Liu *et al.* (2006) ont appliqué la composition sémantique à la modélisation d'un décor urbain. Leur démarche consiste à traduire des éléments géométriques tels les triangles et les lignes en des éléments sémantiques telles que les maisons et les rues. Ce type de reconstruction peut se heurter à un haut degré de complexité engendré par des formes atypiques d'habitation. Les chercheurs ont établi deux catégories, dont les maisons résidentielles et les édifices commerciaux, pour supporter ce type d'environnement.

Le même thème a été repris plus tard par Yong *et al.* (2012) qui ont utilisé des techniques d'intelligence artificielle en plus des méthodes sémantiques. La conception des architectures complexes se fait par ontologie ayant comme but la reconnaissance des motifs récurrents, et la réaffectation des règles pour des formes différentes. L'ontologie vise aussi à définir des

procédés de notation systématique. Cette approche se heurte toutefois à la difficulté présentée par la qualité de la librairie de connaissance utilisée.

#### **1.1.4.2 Sémantique des objets**

Birch *et al.* (2001) ont traité la création d'environnement virtuel en accordant une grande priorité à la vitesse de rendu. Le style d'architecture visé est varié et s'étale de l'époque romaine jusqu'au Hong Kong du temps moderne, moyennant une librairie hétéroclite d'objet. Sont compris dans la sémantique des objets : la forme cubique des maisons, les toitures et les fenestrations. Les autres formes peuvent s'obtenir en utilisant des techniques standard de modélisation qui sont comprises dans l'outil.

Les techniques de modélisation procédurale présentées ci-haut proposent des approches intéressantes pour sculpter un objet 3D. Certaines de ces techniques, comme celles à base d'images, ne peuvent cependant pas être exploitées pour la modélisation des arbres en raison de la complexité des reconnaissances de motifs dans les arbres. Les modélisations procédurales inverses présentent l'avantage de pouvoir imposer, à travers un coefficient de similarité, des contraintes sur l'évolution de l'objet. Cette façon de procéder nécessite d'avoir une esquisse précise sur la forme finale de l'objet, une tâche de plus qui peut être laborieuse pour les utilisateurs. Une solution de modélisation plus accessible, reposant entre autres sur la sélection intelligente et la modification d'élément, est nécessaire pour outrepasser ces limites.

## **1.2 Contrôle des modèles finaux par interaction**

L'interactivité offre aux utilisateurs un moyen de maîtriser méthodiquement le modèle procédural final obtenu en sortie. Cette fonctionnalité peut être proposée sous différentes formes et options au sein du système.

### 1.2.1 Édition par graphe pondéré

Xu & Mould (2012) ont proposé une méthode d'édition et de personnalisation d'un arbre en 3D, consistant à créer un graphe pondéré pour dessiner des structures de branches sur le chemin de moindre coût. Dans leur démarche, l'utilisateur intervient très tôt dans le mécanisme en créant un croquis où les nœuds seront contenus. Il est également possible pour l'utilisateur de sélectionner par lui-même les points terminaux du graphe, le but étant de maximiser la diversité dans le modèle final. Pendant la construction du graphe pondéré, les poids des arêtes sont assignés aléatoirement. La dernière étape consistera à créer une géométrie le long du segment de chemin. Les formes géométriques, essentiellement des cylindres, substituent les chemins pour la modélisation. Les réductions des rayons des cylindres le long d'un chemin sont paramétrables moyennant une variable d'épaisseur. Une éventualité est d'avoir un même nœud traversé par de multiples chemins. Dans ce cas, le choix portera sur le chemin qui proposera au nœud une valeur d'épaisseur supérieure.

### 1.2.2 Guidage par croquis

L'objet 3D, dans cette technique, est manipulé suivant un croquis dessiné même par l'utilisateur. Il requiert donc que l'objet soit préalablement prêt pour les types de manipulation relatifs au guidage par croquis.

#### 1.2.2.1 Notion de guide

Les travaux de Beneš *et al.* (2011) sont une autre percée dans le domaine de la modélisation procédurale interactive des plantes. Leur démarche consiste à décomposer le système en plusieurs blocs maniables que l'on commande avec des guides (figure 1.3). La représentation haut niveau du modèle s'effectue par ces guides, tandis qu'au bas niveau l'utilisateur doit changer directement les règles procédurales de production. Toutefois, les blocs commandés par des guides différents ne peuvent pas entrer en interaction directe.

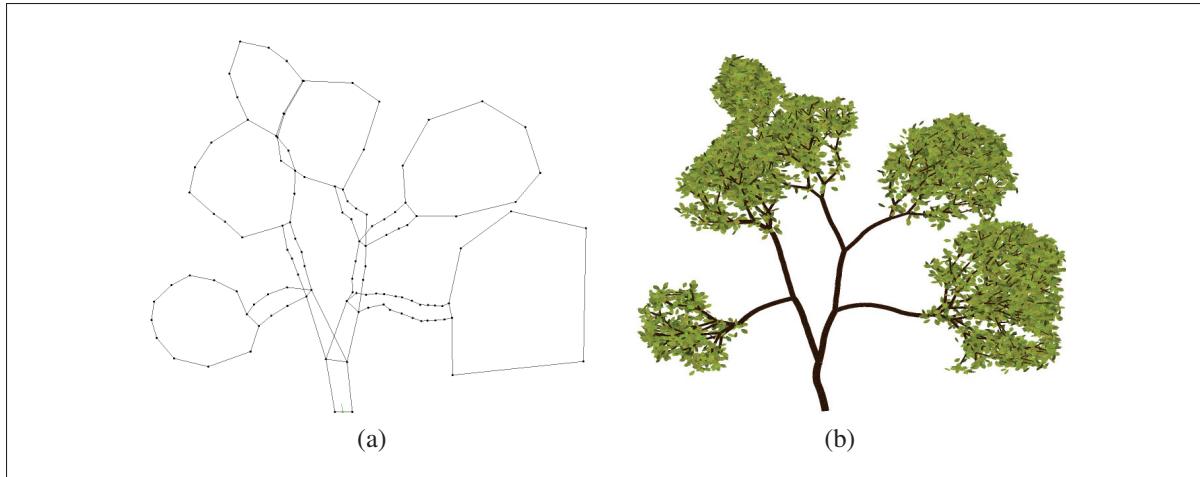


Figure 1.3 (a) Maniement de guide pour (b) influencer un arbre procédural.  
Tirée de Beneš *et al.* (2011).

Leur système de blocs est formé par les guides qui sont des courbes polygonales fermées, les sous-systèmes procéduraux internes aux guides, et enfin les liens connectant les guides. Les positions initiales ainsi que les orientations de la tortue du L-système sont stockées dans chaque guide, et il en va de même pour les axiomes des L-systèmes. Les itérations des L-systèmes se font de manière concurrente, les règles de production étant appliquées en parallèle. Les conditions initiales peuvent être reçues par le système de communication entre guides voisins, mais également de par l'intervention interactive de l'utilisateur.

### 1.2.2.2 Approche déclarative

Smelik *et al.* (2011) se sont penchés sur une approche dite déclarative, où l'utilisateur est censé mettre l'accent sur son attente, plutôt que sur le moyen d'y parvenir. Leur méthode repose sur l'exploitation des modélisations sémantiques en combinant avec les approches procédurales. Ils proposent une technique pour le croquis interactif procédural et le maintien de la pertinence du monde virtuel. Le croquis à fournir est un paysage vu du haut dessiné sur une grille à écotopes. Dans ces écotopes se trouvent des informations d'élévation et de type de sol. La vue est mise à jour en temps réel pour chaque action d'édition. La pertinence du monde virtuel est quant à elle une façon automatisée de garder la cohérence des éléments sur la carte, à savoir les profils

d’élévation, ainsi que la connexion entre éléments terrestres compatibles. Leurs applications s’orientent intrinsèquement à la génération procédurale de terrains.

### 1.2.2.3 Lobes de texture

Une méthode liée à l’utilisation d’un croquis consiste à représenter un modèle d’arbres sous forme d’un ensemble de lobes géométriques telle que proposée par Livny *et al.* (2011). Les chercheurs sont partis de l’observation sur les feuillages des arbres qui peuvent être représentés en un ensemble canonique de structure géométrique. En entrée, le système reçoit les données des arbres sous la forme d’un nuage de points. Une librairie de diverses espèces d’arbre contenant plusieurs types de paramètres a également été construite. Pour le tronc ou la structure squelettique de l’arbre, les chercheurs se sont inspirés d’une technique existante consistant à créer un graphe avec les points voisins connectés entre eux. Les poids assignés aux arêtes du graphe servent à établir si le couple de points pourrait appartenir à une même branche. La structure de branche principale est formée en partant de la racine du graphe jusqu’à une arête à faible niveau de confiance. Les points des arêtes restants sont récupérés dans un groupe qui formera un lobe.

### 1.2.2.4 Extrusions procédurales

Un défi majeur dans la modélisation procédurale d’édifices consiste à gérer les surfaces irrégulières tels les lucarnes et les toits incurvés. Kelly & Wonka (2011) ont travaillé sur une solution interactive de modélisation architecturale traitant les architectures complexes en recourant à ce qu’ils ont défini par extrusion procédurale. La première option offerte à l’utilisateur pour reproduire les surplombs est de pouvoir dessiner des lignes polygonales dans les directions verticales. L’autre option de dessin de surplomb proposée est de contraindre l’utilisateur à produire des profils du modèle voulu toujours à travers des lignes polygonales. L’algorithme se déroule comme suit : le plan en entrée est extrudé ; à chaque hauteur du balayage, une coupe transversale est définie comme un nouveau plan 2D qui sera nommé plan actif. L’extrusion est

définie comme un déplacement des arêtes de façon colinéaire avec la direction du balayage. Une illustration de ce procédé est donnée à la figure 1.4.

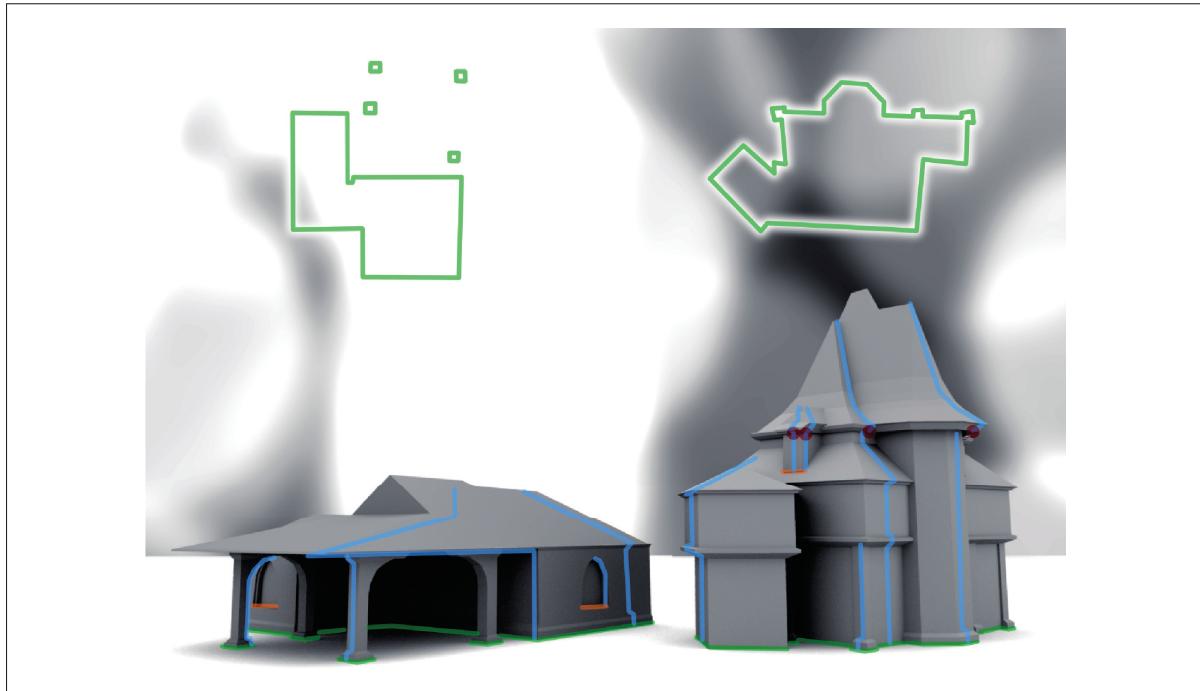


Figure 1.4 Exemple de superposition de blocs architecturaux avec l'entrée utilisateur.

La sortie 3D (grise) est obtenue en entrant les plans (vert) et les profils (bleu).

Tirée de Kelly & Wonka (2011)

Les méthodes de manipulation de graphe, comme celle proposée par Xu & Mould (2012), passent par l'attribution de poids sur des nœuds. Des poids sont assignés même si aucun élément de l'objet 3D ne passera forcément sur le nœud concerné. D'autre part, les différentes méthodes de guidage par croquis, comme celles de Beneš *et al.* (2011) et de Livny *et al.* (2011), permettent aux utilisateurs d'imposer une contrainte dès la première phase de modélisation de l'objet 3D. La conception du croquis peut cependant être une activité ardue pour les utilisateurs. D'autant plus, les éléments à l'intérieur du croquis sont difficilement gérés. Il devient alors nécessaire de travailler sur des solutions de modélisation, qui soient flexibles et moins coûteuses en termes de réalisation.

### 1.3 Travaux sur les algèbres de la modélisation procédurale

Les grammaires formelles sont des systèmes qui se sont bien établis dans la littérature de la modélisation procédurale. Les L-systèmes (Prusinkiewicz *et al.*, 1990) et les grammaires de forme (Stiny & Gips, 1971) en sont les plus fréquents et ont ainsi été explorées. Nous allons aborder dans les prochains paragraphes plusieurs avenues de modélisation procédurale.

#### 1.3.1 Optimisation des espaces de production

Un résultat significatif qui consiste à générer des règles de production selon des spécifications de haut niveau est issu de Talton *et al.* (2011). Les spécifications peuvent prendre plusieurs formes, tels des croquis, des volumes ou des objectifs analytiques. Cette méthode calcule la meilleure dérivation en formulant le problème en termes d’inférence de probabilité. Un produit est effectué entre un terme de vraisemblance de l’entrée et un modèle antérieur. La meilleure dérivation consiste alors à trouver la grammaire qui maximise ce produit. L’inférence d’une production est effectuée par les méthodes de Monte-Carlo par chaînes de Markov. La méthode calcule donc la meilleure règle de production possible à partir de la grammaire. L’application de la méthode Monte-Carlo par chaînes de Markov sur les grammaires formelles passe par la définition de deux opérations de contrôle. La première opération s’appelle la *diffusion*, elle consiste à laisser la topologie de la dérivation constante, et modifier les paramètres descriptifs pris aléatoirement sur un terminal de la forme 3D. La deuxième opération de contrôle est le *saut*, elle revient à influencer les paramètres stochastiques de la grammaire. Leur formulation parvient à une optimisation aboutie. Toutefois, leur méthode peut générer un objet 3D incohérent selon les spécifications à l’entrée.

#### 1.3.2 Primitives de haut niveau

La méthode de Krecklau & Kobbelt (2012) propose une visualisation en temps réel d’une grammaire appelée *grammaire de scission*. Il s’agit d’un type de grammaire où les objets et les règles de construction sont organisés en hiérarchie dans la scène. La description d’une gram-

maire de scission est également textuelle. L'idée pour leur méthode est d'offrir interactivement un aperçu visuel de ce qui s'effectue au cours d'une édition ou production de règles procédurales. Le manipulateur 3D est l'outil qui sert à faire apparaître dans l'espace de modélisation l'aspect visuel des paramètres ajustés. Les outils de contrôle, que Krecklau & Kobbelt (2012) proposent, s'exécutent à travers des primitives, c'est-à-dire des formes géométriques élémentaires tels des cubes ou des sphères, ou une composition de ces formes. Comme dans la plupart des recherches offrant cette fonctionnalité, des bases de données sont utilisées pour représenter chaque classe d'éléments.

### 1.3.3 Édition visuelle des grammaires

Similairement à la section précédente, les chercheurs Lipp *et al.* (2008) ont proposé des paradigmes d'édition visuelle des grammaires de forme. Leurs méthodes enlèvent la possibilité pour l'utilisateur d'écrire les règles. Ils substituent ce processus par un système de contrôle interactif sur l'instance d'objet 3D. Des modifications locales sont faites à la demande par l'utilisateur, en introduisant de nouvelles règles conçues de façon visuelle. Les modifications peuvent être circonscrites, dans le sens où le changement d'une règle a un effet local seulement, et non dispersé sur tout l'objet. La sélection pour les fins de modification est effectuée sémantiquement à travers des attributs de l'objet, en spécifiant par exemple le numéro de la face de l'élément ciblé, ou encore son étage et sa colonne.

L'ensemble des travaux sur l'algèbre de modélisation présentent l'avantage d'être moins exigeant sur le plan de la connaissance des grammaires formelles. Divers outils de contrôle sont proposés de façon à ce que les utilisateurs parviennent aux formes souhaitées sans passer par une manipulation ardue des grammaires. Ces travaux sur l'algèbre permettaient déjà de manipuler les paramètres de la grammaire. Cependant, les contrôles, notamment ceux proposés par Lipp *et al.* (2008), sont très rigides, car ils proposent uniquement d'appliquer un changement soit localement, soit sur tous les arbres. Cette limitation justifie la nécessité de créer un système de sélection multiple bien plus souple, afin que les modifications puissent se propager simultanément dans des sous-ensembles choisis de l'objet.

## 1.4 Recherches sur les plantes procédurales

L’application aux plantes a toujours été un sujet d’intérêt dans la littérature de la modélisation procédurale. Les orientations des recherches ainsi que les approches sur les problèmes traités y sont très variées. Parmi les résultats les plus intéressants, on peut citer les travaux de Št’ava *et al.* (2012) sur des modèles d’arbres auto adaptatifs. Leur but était de concevoir des arbres complexes capables d’interagir avec leur environnement. Pour y parvenir, ils exploitent les variations dans la distribution de la lumière et la proximité avec les obstacles. En entrée, l’algorithme reçoit des grammaires de production d’un squelette d’arbre ou aussi des modèles reconstruits par laser. L’analyse de l’arbre s’effectue en incorporant divers paramètres comme la réponse des branches individuelles. Leurs calculs se font à partir des dispositions géométriques et des informations topologiques de l’arbre. L’âge de l’arbre est estimé par son taux de croissance, assisté de la longueur des segments sans bourgeons. L’âge de l’arbre sera ensuite utilisé pour évaluer les conditions de lumière à toutes les étapes de croissance. Ce sera seulement après que les conditions environnementales de développement soient déterminées que l’on parviendra à calculer des valeurs de tropisme. Le tropisme est la tendance de l’arbre et de ces branches à s’incliner dans une direction.

### 1.4.1 Colonisation de l’espace

Prusinkiewicz *et al.* (1994) avaient approfondi les L-systèmes sensibles à l’environnement, puis incorporé dans les règles de réécriture même des extensions s’y rapportant. L’expansion à une nouvelle dimension, le partitionnement des espaces et la compétition pour l’espace sont les démarches de Prusinkiewicz & de Reuille (2010) pour traiter le problème de colonisation de l’espace, mais cette fois en étroite dualité avec des propriétés botaniques telles que les régulations génétiques et les densités des organes.

### 1.4.2 Compétition pour la lumière

Nous avons déjà vu dans les sections précédentes des cas d'exploitation de la lumière pour décider du développement de l'arbre. Palubicki *et al.* (2009) ont également approfondi la compétition pour la lumière et l'espace. Des mécanismes de signalisation interne sont proposés par les chercheurs pour véhiculer des informations dans les éléments composant l'arbre. En procédant comme tel, leur objectif consiste à produire une balance conforme et variable de distribution de branches. L'étape préliminaire consiste à effectuer des calculs sur les conditions environnementales. Elle retourne l'espace disponible pour chaque bourgeon, et la direction optimale pour la croissance des pousses. Dans l'étape suivante, il est alors possible de déterminer les bourgeons qui pourront produire de nouvelles pousses, avec les contraintes de volume.

Les précédentes méthodes appliquées aux plantes procédurales répondent surtout aux exigences de réalisme des modèles de plantes et arbres. Entre autres, le tropisme induit une courbure des plantes, phénomène dont le seul contrôle revient aux quantités d'éléments naturels reçus dans les constituants. Ces techniques sur les plantes procédurales sont très efficaces pour simuler les développements naturels des plantes, mais n'offrent toutefois aucun support pour modifier la structure de ces plantes. Les méthodes de Št'ava *et al.* (2012) présentent l'avantage d'être sensible aux stimuli externes, cependant leurs résultats se limitent à des flexions de certains éléments de l'arbre, en plus d'exiger des entrées aussi complexes que la numérisation d'un objet. De ce fait, des méthodes plus accessibles sont utiles aux besoins des utilisateurs. Dans l'approche que nous proposons dans ce mémoire, les sous-ensembles à éditer de l'arbre peuvent être directement supprimés ou regénérés. Des qualificatifs sont d'autre part disponibles pour assister les utilisateurs dans ses étapes de modification de l'arbre.

## 1.5 Entrée en matière sur les L-systèmes

Les L-systèmes étant à la base du projet présenté dans ce mémoire, ils seront présentés plus en détail dans cette section. Proposée par Lindenmayer (1968), il s'agit d'une théorie mathématique du développement des plantes, compatible à des applications graphiques. Ces derniers

ont été introduits comme un formalisme permettant la simulation du développement des organismes multicellulaires, reposant sur les langages formels. Le fondement du développement des plantes, du point de vue du L-système, est un mécanisme de réécriture parallèle. La réécriture consiste en un remplacement successif d'un objet initial en un ensemble d'objets suivant des *règles de production*. Le système de réécriture le plus répandu repose sur les grammaires formelles.

### 1.5.1 Système de réécriture et interprétation de la tortue

Le développement d'une algue, selon le modèle de Lindenmayer, peut s'exprimer à l'aide de deux mots *A* et *B*, partant de l'axiome :

$\omega : A$

et avec le couple de règles de production suivant :

$p : A \rightarrow AB$

$B \rightarrow A$

Ainsi, la chaîne courante pour chaque itération  $n$  serait :

$n = 0 : A$

$n = 1 : AB$

$n = 2 : ABA$

$n = 3 : ABAAB$

$n = 4 : ABAABABA$

$n = 5 : ABAABABAABAAB$

...

Du point de vue de l'interprétation graphique, plusieurs résultats ont été publiés. Toutefois, une idée générale est adoptée dans le domaine des L-systèmes, consistant à définir un triplet  $(x, y, \alpha)$  composé des coordonnées cartésiennes  $(x, y)$  de la position d'une tortue, et d'un angle  $\alpha$  qui est l'orientation de la tête de la tortue. À ces informations s'ajoutent deux autres éléments utiles pour l'interprétation, dont la taille d'un pas noté  $d$  et un angle d'incrémentation noté  $\delta$ .

Les grammaires ainsi que ses interprétations n'ont pas toujours été rigoureusement similaires de publication en publication dans la littérature du L-système. Cependant dans le système d'application que nous avons conçu pour le projet, nous attestons une fidélité avec l'algèbre et les symboles proposés par Prusinkiewicz (1986). Pour illustrer le fonctionnement de notre système, considérons l'exemple du quadratique de Koch défini par la grammaire suivante :

$$\omega : F - F - F - F$$

$$p : F \rightarrow F - F + F + FF - F - F + F$$

Le mot F (en majuscule) est une commande de tracé de ligne. Les symboles + et – sont des commandes de rotation de la tortue respectivement à gauche et à droite par un angle  $\delta$ . Les images de la figure 1.5 montrent, à partir de la gauche, l'interprétation de la tortue du quadratique de Koch en commençant par la dérivation  $n = 0$  qui est l'axiome ; l'image immédiatement sur la droite est l'interprétation de la tortue pour la dérivation  $n = 1$ , et ainsi de suite.

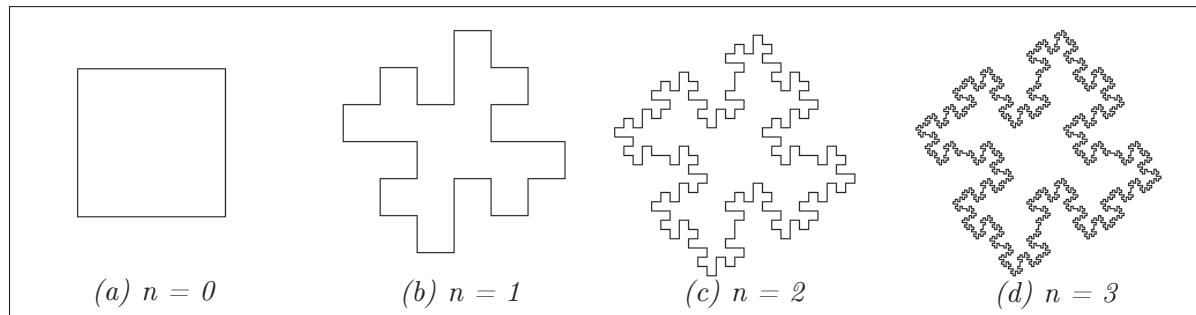


Figure 1.5 Évolution géométrique du quadratique de Koch. De gauche à droite, l'interprétation de la tortue après chaque dérivation montre une progression régulière de la forme initiale.

### 1.5.2 Transformation géométrique

Les transformations s'effectuent à partir de l'état courant de la tortue. Les trois vecteurs  $\vec{H}$ ,  $\vec{L}$ ,  $\vec{U}$  représentent respectivement la direction de la tête ou *head*, la direction de la gauche ou *left*, et la direction du haut ou *up* (figure 1.6). Les vecteurs sont unitaires et orthogonaux entre eux, et répondent à  $\vec{H} \times \vec{L} = \vec{U}$ . Comme nous pouvons le voir à la figure 1.5, nous pouvons assigner

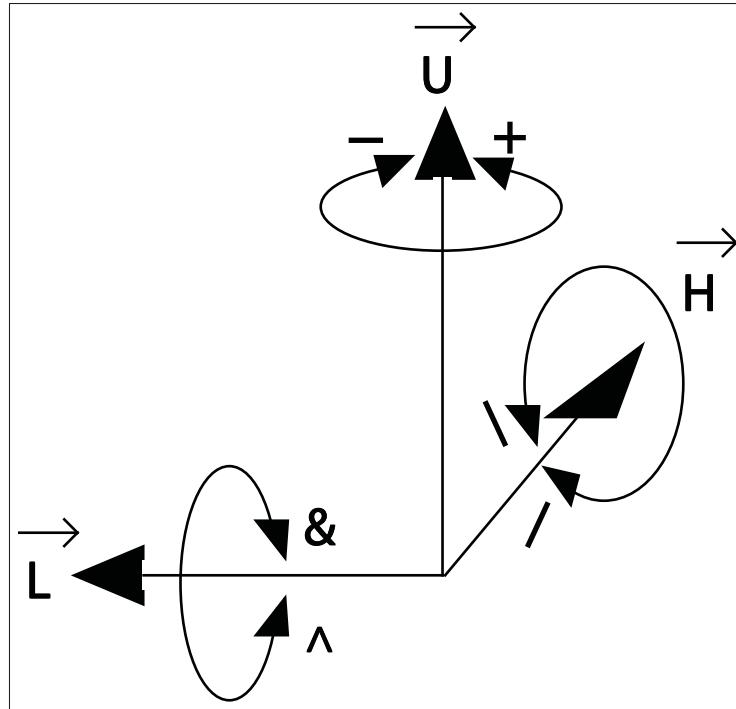


Figure 1.6 État initial de la tortue dans les L-systèmes.  
Tirée de Prusinkiewicz *et al.* (1990).

à chaque mot de l’alphabet du L-système une forme géométrique propre. De même, plusieurs autres symboles sont dédiés à des commandes de rotation sur un axe donné. Ces symboles font partie des grammaires du L-système présentées dans l’annexe I.

Une autre fonctionnalité importante de l’interprétation de la tortue est de pouvoir garder dans une pile l’état de la tortue, à savoir sa position et son orientation, et éventuellement des attributs de couleur et d’épaisseur de tracé. Ce mécanisme de sauvegarde d’état s’effectue avec la commande « [ », tandis que sa réciproque, « ] » est la commande qui restaure la pile ou l’état de la tortue.

### 1.5.3 Formalisme des L-systèmes

Soit  $V$  un alphabet.  $V^*$  est l’ensemble de tous les mots dans  $V$ , et  $V^+$  l’ensemble des mots non vides dans  $V$ . Le triplet  $G = \langle V, \omega, p \rangle$  définit un L-système déterministe et sans contexte, où  $\omega \in V^+$  est un mot appelé *axiome* et  $p \subset V \times V^*$  un ensemble de *productions*. Une production

s'écrit sous la forme  $a \rightarrow \chi$  où le mot  $a$  est le *prédecesseur* et le mot  $\chi$  le *successeur*. La production identité  $a \rightarrow a$  s'applique par défaut lorsque pour un mot  $a$ , aucun successeur  $\chi$  n'est spécifié. Le processus d'application des règles de production sur la chaîne courante est appelé une dérivation.

Les L-systèmes répondent à un critère de déterminisme et n'impliquent aucun paramètre aléatoire lors des dérivations. Une grammaire reproduira ainsi  $N$  objets 3D rigoureusement similaires si exécutés  $N$  fois.

#### 1.5.4 L-systèmes paramétriques

Les L-systèmes paramétriques viennent ajouter plus de flexibilité aux L-systèmes standards dont une des limitations les plus contraignantes est de ne pas pouvoir faire varier la longueur des lignes. Des paramètres numériques ont alors été ajoutés aux L-systèmes pour résoudre ce type de problème.

Les L-systèmes paramétriques consistent en des *modules* composés par une lettre ou un mot appartenant à l'alphabet  $V$  et des paramètres associés appartenant à l'ensemble des réels  $\mathbb{R}$ . D'une manière plus formelle, un module est composé d'un mot avec comme paramètres  $m_1, m_2, \dots, m_n \in \mathbb{R}$ , et est noté  $A(m_1, m_2, \dots, m_n)$ . Si  $\Sigma$  est l'ensemble des paramètres formels, alors on peut écrire  $C(\Sigma)$  l'ensemble des expressions logiques à paramètres dans  $\Sigma$ , et  $E(\Sigma)$  dénote les expressions arithmétiques, toujours à paramètres dans  $\Sigma$ . Ces expressions renferment toutes des paramètres formels avec des constantes numériques, et liées par :

- des opérateurs arithmétiques  $+, -, *, /,$
- un opérateur exponentiel  $\wedge,$
- des opérateurs de relation  $<, >, =,$
- des opérateurs logiques  $\&\&, ||, !.$

Dans une grammaire paramétrique, il est commun d'avoir des mots qui ne portent aucun paramètre. Les paramètres globaux qui s'appliquent aux L-systèmes standards sont assumés pour

ces mots. Ensuite, pour les mots qui admettent plus d'un paramètre, l'usage veut que le premier paramètre soit toujours relié à la tortue du L-système.

### 1.5.5 Règles avec conditions

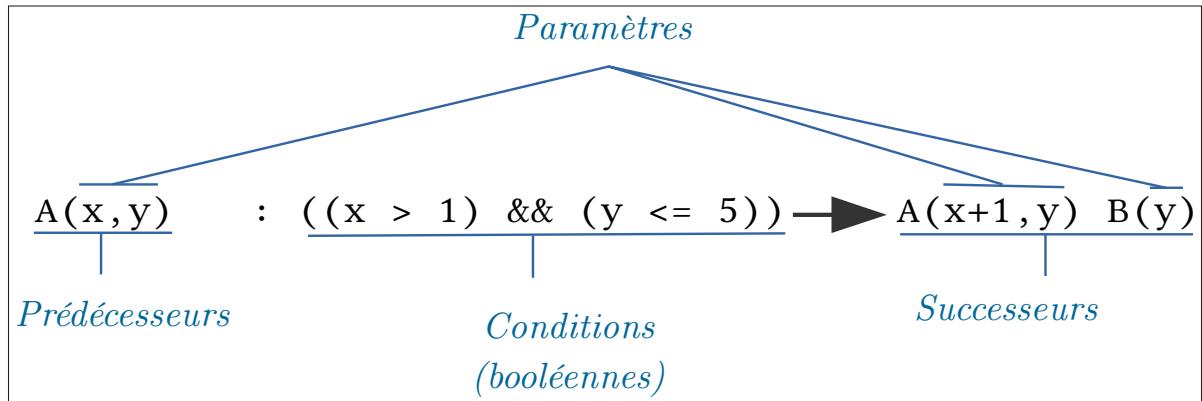


Figure 1.7 Structure d'une règle de production.

Un aperçu de la structure d'une règle de production est donné à la figure 1.7. Les conditions sont placées après le symbole « : » comme indiqué dans la figure 1.7. La validation des conditions peut comporter des évaluations arithmétiques et booléennes. Les opérations booléennes ont lieu quand il existe plusieurs contraintes arithmétiques à remplir.

### 1.5.6 Traitement du prédécesseur et du successeur

Le prédécesseur contient le mot qui sera remplacé dans la chaîne courante. Le successeur est une chaîne de caractère d'une règle de production qui est placée après le séparateur « → ». Lors d'une dérivation, le mot prédécesseur est remplacé par une séquence de caractères dans le successeur de la règle qui correspond. Les paramètres sont remplacés puis l'expression calculée en conséquence. Le résultat du calcul constituera le paramètre du mot dans la prochaine dérivation.

### 1.5.7 Inspection du flot de grammaire

Une grammaire commence usuellement par une section portant sur des définitions d'expressions constantes. Cette section figure au début de l'ensemble des données constituant la grammaire. Tout d'abord, le concept de préprocesseur a été mis en place pour traiter ces définitions. Dans cette section de définitions, il est offert à l'utilisateur de créer des mots, c'est-à-dire en addition aux 26 caractères de l'alphabet latin classique. Ces mots sont une suite de lettres et peuvent être plus expressifs comme « *branche* » ou « *feuille* », etc. Une fois les mots collectés, on définit les constantes de grammaire. Après que ces étapes soient effectuées, la lecture de l'axiome et des règles procédurales est entamée comme d'habitude. À chaque élément est attribuée une signature particulière :

- Les préprocesseurs commencent par le signe « # » en début de ligne ;
- Un axiome est introduit par le caractère *w* suivi de « : » ;
- Le nombre de dérivation à effectuer peut se spécifier en entrant en début de la ligne le caractère *n* suivi de « : » ;
- Les règles de production débutent par le caractère *p* suivi de « : ».

La figure 1.8 montre un extrait de l'interface utilisateur où l'on peut voir une grammaire décomposée en plusieurs parties.

Les approches que nous suggérons dans le cadre du projet exploitent intensivement plusieurs caractéristiques des L-systèmes. Les formalismes, qui viennent d'être présentées, sont au centre de la méthodologie exposée dans les sections qui suivent.

## 1.6 Problématiques et démarche de la recherche

Les recherches antérieures ont abordé le problème sous divers angles, tels la compétition pour la lumière et l'espace ou l'ajout de guides et de contraintes. Toutefois, dans les compétitions pour la lumière, telles que proposées par Prusinkiewicz & de Reuille (2010), l'utilisateur ne peut qu'influencer indirectement l'évolution de l'arbre, et aucun mécanisme de contrôle direct

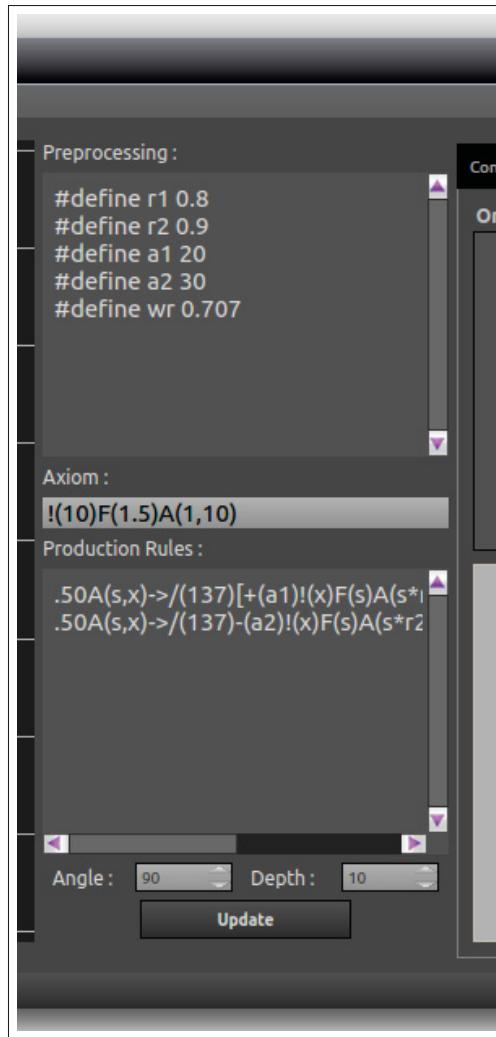


Figure 1.8 Extrait de l’interface utilisateur montrant les grammaires.

n’y est proposé. Palubicki *et al.* (2009) avaient proposé des opérations d’élagage et de fléchissement des branches dans leurs méthodes de compétition pour l’espace et pour la lumière, mais leur démarche requiert une connaissance en biologie végétale avant de pouvoir manipuler l’environnement de croissance de l’arbre. L’approche adaptative de Št’ava *et al.* (2012), qui interagit avec des obstacles solides, présente l’avantage d’être intuitive, mais ne modifie pas substantiellement la structure de l’arbre en ce sens où le squelette ne change pas.

En parcourant la littérature de la modélisation procédurale, nous avons constaté que les créateurs de contenu ne sont pas assistés comme il se doit pendant la conception d’un objet 3D.

Le tableau 1.1 établit un comparatif des méthodes existantes, avec en dernière ligne l'approche que nous proposons. Les procédés actuels permettent une interaction avec l'objet mais n'autorisent pas, comme indiqué dans le tableau, une modification ponctuelle sur une sélection, et ne proposent pas de regénérer un sous-ensemble de l'objet. Les paramètres des règles procédurales peuvent être incompris par l'utilisateur, et l'édition visuelle, comme celle proposée par Lipp *et al.* (2008), ne résout que partiellement le problème. En effet, l'utilisateur peut voir en temps réel l'effet d'un changement de paramètre, mais ne parviendra pas nécessairement à en déduire lesquels des multiples paramètres aboutiront à l'effet voulu. Parallèlement, les créateurs de contenu ignorent les combinaisons de paramètres qui maximisent les modifications dans la scène 3D. Finalement, aucune des techniques découvertes jusqu'ici n'a tenté de relier systématiquement des ensembles de paramètres à des qualificatifs caractéristiques.

### 1.6.1 Objectifs de la recherche

Constatant les lacunes dans l'assistance des créateurs de contenu, ce projet s'est marqué comme objectif de fournir des outils avancés d'édition intuitive d'un arbre procédural. Initialement, l'utilisateur peut interagir avec l'objet en sélectionnant directement une branche ou un sous-ensemble qui l'intéresse. Ensuite, des opérations d'élagage et de régénération de branches sous un autre ensemble de paramètres sont proposées. Le projet cherche également à proposer plusieurs combinaisons de paramètres qui ont le plus d'effet visuel sur l'objet. Finalement, l'utilisateur pourra choisir d'être assisté par des qualificatifs, comme *large*, ou *dense*, pour l'assister dans le processus de régénération.

### 1.6.2 Méthodologie

L'utilisateur peut fournir sa propre grammaire, ou travailler à partir d'une grammaire existante. Nous intégrons un système de métrique dans le but d'évaluer les paramètres qui affectent le plus une grammaire en entrée. Des calculs seront exécutés en interne afin de retenir les valeurs où les combinaisons de paramètres qui maximisent les variations des métriques. Une expérience auprès de participants a permis d'évaluer la pertinence des arbres retenus par notre

Méthodes	Pré-contrôle	Interactivité/Post-contrôle	Temps réel	Application 3D	Observations
Grammaire CGA	X Axiome	Ajout de restriction de règles au cours d'un cycle de dérivation	X	Ville	Très haut niveau de détail, mais présence d'irrégularité (irréaliste)
Procédurale inverse	Croquis	X		Ville, édifice, arbre	Résultat intéressant pour les spécifications multiples données en entrée
Guidage croquis	Croquis	Mise à jour manuelle des guides	X	Arbre	Résultat intéressant à condition d'avoir des guides précis et fournis
Chaîne de Markov	Croquis	X		Ville, édifice, arbre	Temps d'exécution élevé
Règle visuelle	X Axiome	Sélection sémantique, modification locale		Forme géométrique, édifice	Résultat combinant la modélisation procédurale et la modélisation 3D standard
Notre approche	X Axiome	Absolue : sélection libre automatiquement ajustée, modification de sous-ensemble d'arbre spécifique d'une façon différente		Arbre	Assistant utilisateur via des qualificatifs et des vignettes

Tableau 1.1 Comparatif des méthodes existantes avec notre approche.

approche à base de métrique, et de recueillir des qualificatifs qui caractérisent le modèle 3D. La multisélection est prise en charge : l'utilisateur peut donc effectuer simultanément plusieurs changements de même nature. La sélection est ajustée en fonction de l'arbre de dérivation de la grammaire, et sera transcrit en un graphe orienté. Un calcul de la coupe minimale sur ce graphe s'ensuit pour isoler des parties de l'arbre qui seront destinées à la régénération. Les qualificatifs fournis par les participants seront retenus dans notre démarche pour guider l'utilisateur final dans les types de régénération disponibles.

## CHAPITRE 2

### REGÉNÉRATION DE CHAÎNES

Les dérivations forment la chaîne sur laquelle on voudra appliquer certains types de modification. Dans ce chapitre, nous allons présenter une méthode pour manipuler cette chaîne. Cette méthode est basée sur un historique de l'évolution de cette chaîne, que nous avons structuré sous une forme de hiérarchie, et que l'on nomme arbre de dérivation.

#### 2.1 Système de sauvegarde d'état de dérivation

Au cours d'une itération de dérivation, l'objectif consiste à sauvegarder d'une manière ordonnée l'historique de l'évolution de la chaîne. Le principe qui régit notre méthode est une exploitation d'un arbre de dérivation, où l'origine de chaque mot peut être retracée en parcourant les parents dans la hiérarchie de cet arbre de dérivation.

Une grammaire peut contenir plusieurs arbres de dérivations, le nombre d'arbres de dérivations pour une grammaire est de ce fait lié directement au nombre de symboles dans l'axiome. Chacun de ces arbres va se développer en fonction du nombre de dérivations demandées.

L'organisation des historiques de dérivation s'effectue de telle manière qu'aucune instance d'objet n'est nouvellement créée quand le mot existait déjà dans la chaîne précédente. Ce choix de mode opératoire est positif pour l'organisation en hiérarchie de l'arbre de dérivation, car à un mot correspondra un seul et unique objet.

Illustrons le fonctionnement de l'historique à travers la grammaire ci-après :

$w : A(1)$

$A(x) \rightarrow A(x+1)B(x+1)C(1)$

$B(y) \rightarrow B(y+3)$

Effectuons  $n = 2$  dérivations sur cette grammaire. Le tableau 2.1 affiche l'évolution de la chaîne au cours des dérivations successives.

Tableau 2.1 Évolution de la chaîne à chaque dérivation.

Indices	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Axiome	A	(	1	)																
Dériv. n=1	A	(	2	)	B	(	2	)	C	(	1	)								
Dériv. n=2	A	(	3	)	B	(	3	)	C	(	1	)	B	(	5	)	C	(	1	)

La figure 2.1 montre l’arbre de dérivation correspondant. Depuis cet arbre, il est facile de constater d’ores et déjà que la chaîne courante est en effet située sur les feuilles de l’arbre de dérivation.

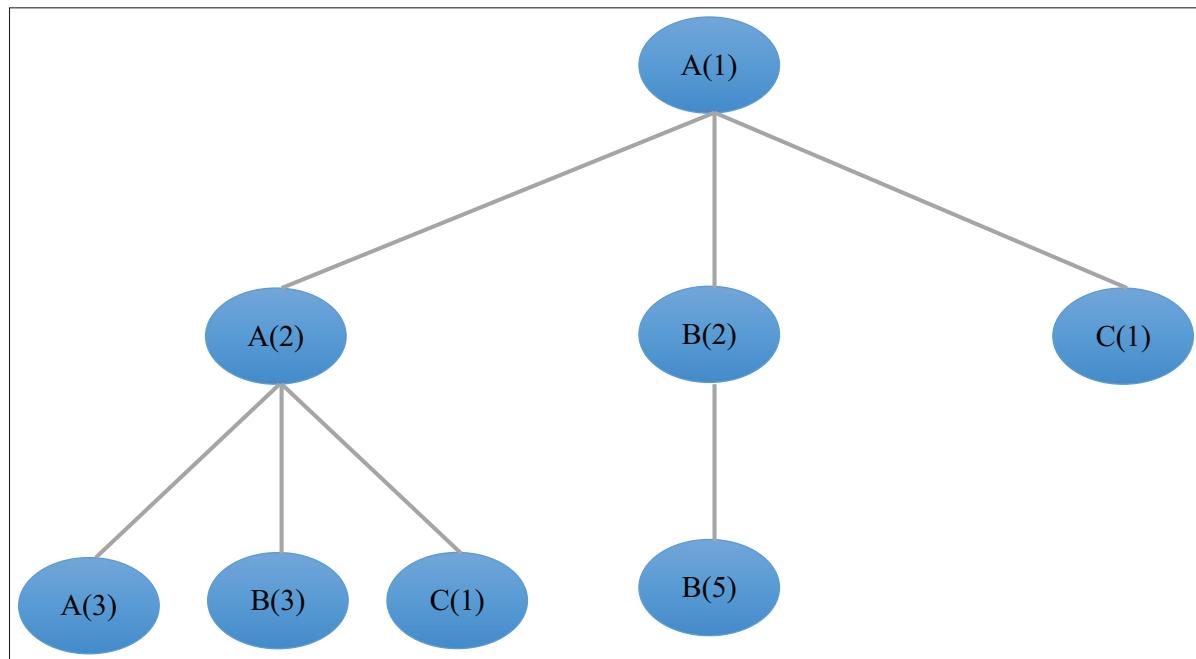


Figure 2.1 Exemple d’arbre de dérivation.

Chaque élément d’historique dans l’arbre de dérivation constitue la description d’un symbole dans la chaîne de dérivation. Les symboles comme ceux de sauvegarde et de chargement d’état de la tortue, respectivement « [ » et « ] », ainsi que les divers symboles de transformation de la tortue, font état d’une instance d’historique de dérivation. Le stockage de ces symboles implique aussi de manière interne le stockage des paramètres respectifs.

## 2.2 Structuration en arborescence des dérivations

À chaque fois qu'une application de règle sur un symbole a lieu, l'historique est mis à jour en ajoutant des nœuds enfants au nœud (donc au symbole) de départ. Les nœuds enfants seront les symboles nouvellement formés après l'application de la règle. La classe d'historique a été conçue en une structure hiérarchique, où l'instance connaît l'emplacement mémoire de son parent, mais aussi de chacun de ses enfants dans l'arbre de dérivation.

### 2.2.1 Sélection d'un élément d'historique

La sélection d'un élément d'historique est appelée à partir de l'action utilisateur sur l'objet 3D, qui indiquera à son tour le mot correspondant à l'élément pointé. Ceci implique que la sélection sur l'arbre de dérivation s'effectuera exclusivement sur les nœuds feuilles de l'arbre. Ainsi, la sélection d'un élément dans la scène correspond exactement à la sélection d'une feuille dans l'arbre de dérivation. Dans le cadre du projet, effectuer une ascension dans l'arborescence d'historique consiste à monter en hiérarchie dans les nœuds parents et éliminer récursivement tous les nœuds enfants (figure 2.2). Cette opération consiste donc tout simplement à une suppression de nœuds dans l'arbre de dérivation.

### 2.2.2 Correspondance d'une sélection avec un arbre de dérivation

Pendant la conception du système, une méthode de visualisation à base de gradient de couleur est introduite afin de donner un aperçu à l'utilisateur de l'impact dans la hiérarchie d'une manipulation de l'arbre de dérivation. En utilisant le système de couleur *TSV* (*Teinte, Saturation, Valeur*), nous offrons à l'utilisateur le moyen d'évaluer l'impact de la sélection courante. En se référant à la figure 2.3, la teinte correspond à la valeur de l'angle autour de l'axe vertical central du cylindre.

À chaque nœud feuille d'un arbre de dérivation, nous aimerais attribuer une couleur. La figure 2.4 donne un aperçu du système de gradient de couleur que nous aimerais assigner aux nœuds feuilles de l'arbre de dérivation. Les nœuds qui ne sont pas des feuilles ne nous intéressent

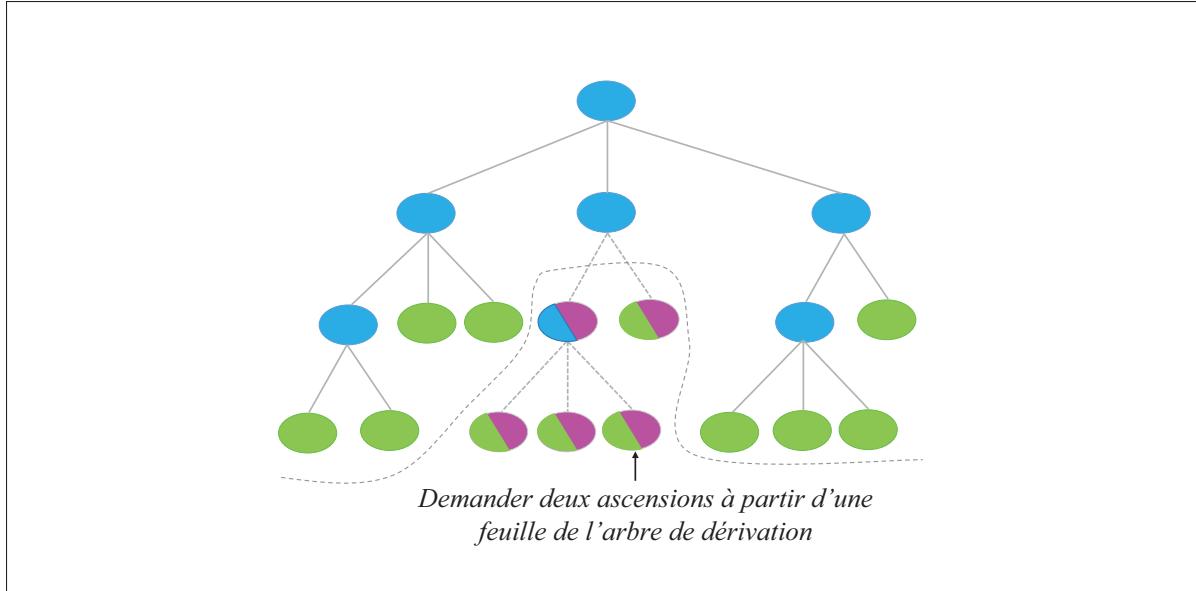


Figure 2.2 Cas d'une ascension dans un arbre d'historique. Les couleurs bleue, verte et magenta sont associées aux nœuds parent, feuille et ceux supprimés par la régénération.

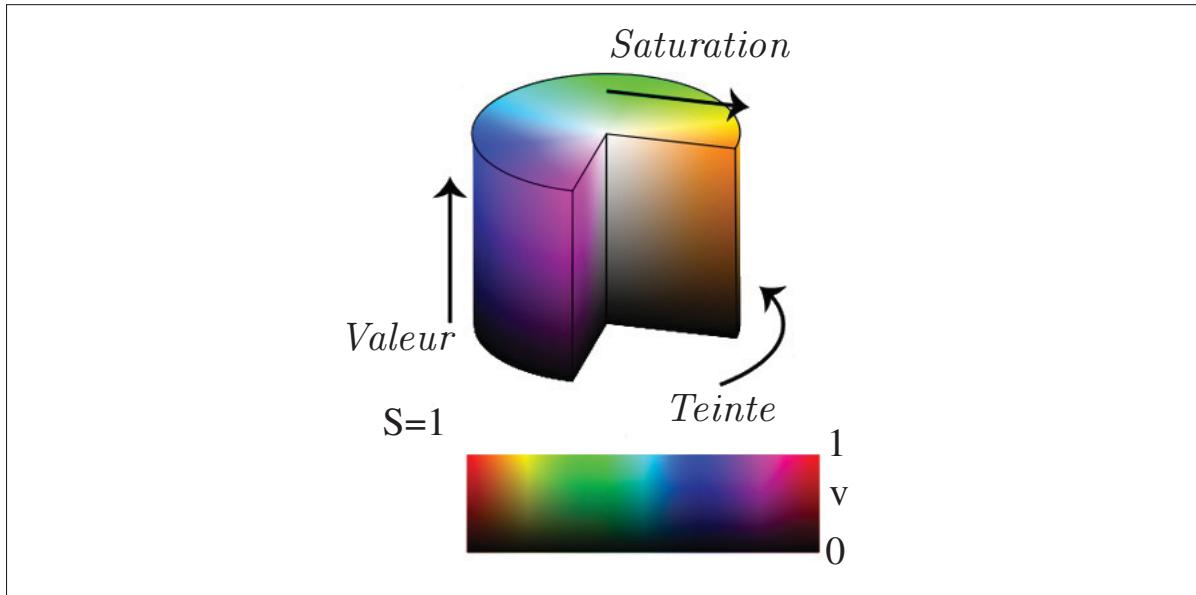


Figure 2.3 Aperçu du système de couleur TSV.

pas, car seuls les nœuds feuilles apparaissent dans la scène 3D. Dénotons par  $F$  un nœud feuille sélectionnée par l'utilisateur, identifiée en rouge dans la figure 2.4. Nous définissons

$\varphi(F) \in \mathbb{N}$ , pour tous les nœuds feuilles, avec  $F$  qui désigne le nœud sélectionné. Les feuilles  $\varphi(F) = 1$  représentent les nœuds frères ayant le même parent immédiat. Elles sont identifiées en jaune dans la figure 2.4. Notre objectif consiste à assigner de façon progressive en fonction de  $\varphi(F)$  un attribut de couleur. Comme dans la figure 2.4 la sélection courante n'a qu'un unique frère  $\varphi(F) = 1$ , il n'y a alors qu'un seul nœud qui sera représenté par cette couleur jaune. Les nœuds  $\varphi(F) = 2$  sont les nœuds additionnels accessibles en remontant au deuxième parent de la sélection. Leurs couleurs sont alors identiques. Le même principe est appliqué pour les nœuds  $\varphi(F) = 3$ , dont le troisième parent est commun à la sélection. On appelle ainsi les nœuds frères de  $F$  d'ordre  $j$  ceux pour lesquels  $\varphi(F) = j$ . La valeur maximale de  $j$  est égale au nombre de dérivations  $n$  de la grammaire.

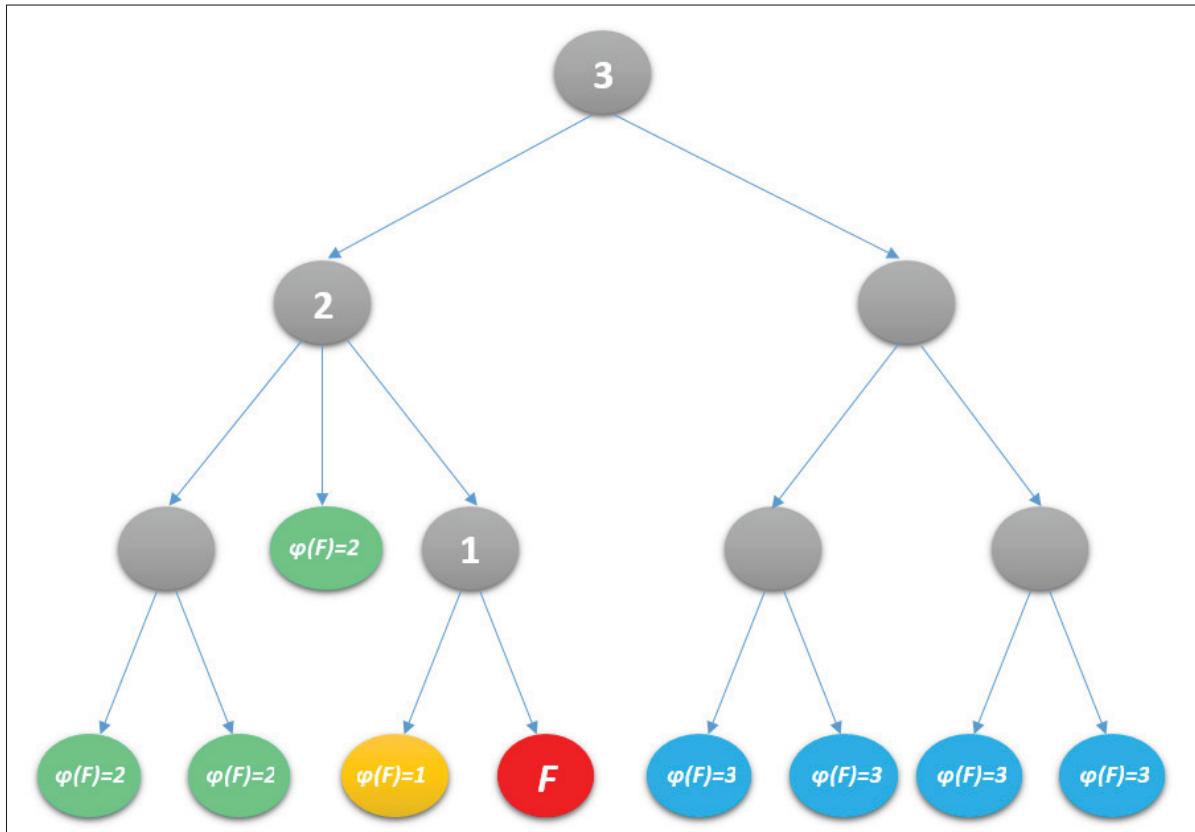


Figure 2.4 Attribution des gradients de couleur lors d'une sélection d'un nœud.

L'idée est d'assigner à la sélection la couleur de teinte correspondant à l'angle  $0^\circ$  (couleur rouge), et ensuite d'attribuer à l'angle  $240^\circ$  (couleur bleue) les nœuds pour lesquels  $\varphi(F) = n$ . Les attributs de saturation quant à eux  $S$  sont gardés constants à  $S = 1$ , de même la valeur  $V$  est aussi constante et est de  $V = 1$ . La couleur bleue est par conséquent censée indiquer les éléments du modèle 3D qui correspondent aux nœuds frères les plus éloignés. Pour remplir ce critère d'évolution progressive de couleur, nous calculons un paramètre dit valeur d'impact dont la formule est :

$$impact = \frac{\varphi(F)}{n}$$

Par définition,  $\varphi(F) = 0$  pour le nœud  $F$ . La valeur d'impact est donc une valeur positive dans  $[0, 1]$ . Nous en déduisons facilement une valeur de la teinte  $T$  qui répondrait au critère d'évolution :

$$T = 240 \times impact$$

La figure 2.5 donne un aperçu du résultat de l'application de la formule : l'objet 3D non sélectionné est à gauche (figure 2.5a), tandis qu'à droite une sélection est effectuée (figure 2.5b) et est mise en surbrillance de rouge. On remarque qu'à mesure que l'on s'éloigne de la sélection, la couleur s'éloigne de la couleur de sélection pour s'approcher progressivement de la couleur bleue.

### 2.2.3 Multisélection

La sélection multiple utilise la méthode de Nykolas Clément-Corriveau, étudiant à la maîtrise au laboratoire multimédia. La sélection multiple s'effectue en englobant dans une zone de forme rectangulaire toutes les branches qu'on veut inclure dans la sélection. Une liste de sélection est constituée pour contenir l'indice de tous les éléments de l'arbre compris dans la sélection. Cette fonctionnalité a été enrichie par l'auteur du présent rapport en offrant plus de personnalisation à cette liste de sélection. En effet, il est possible d'enlever ou d'ajouter un

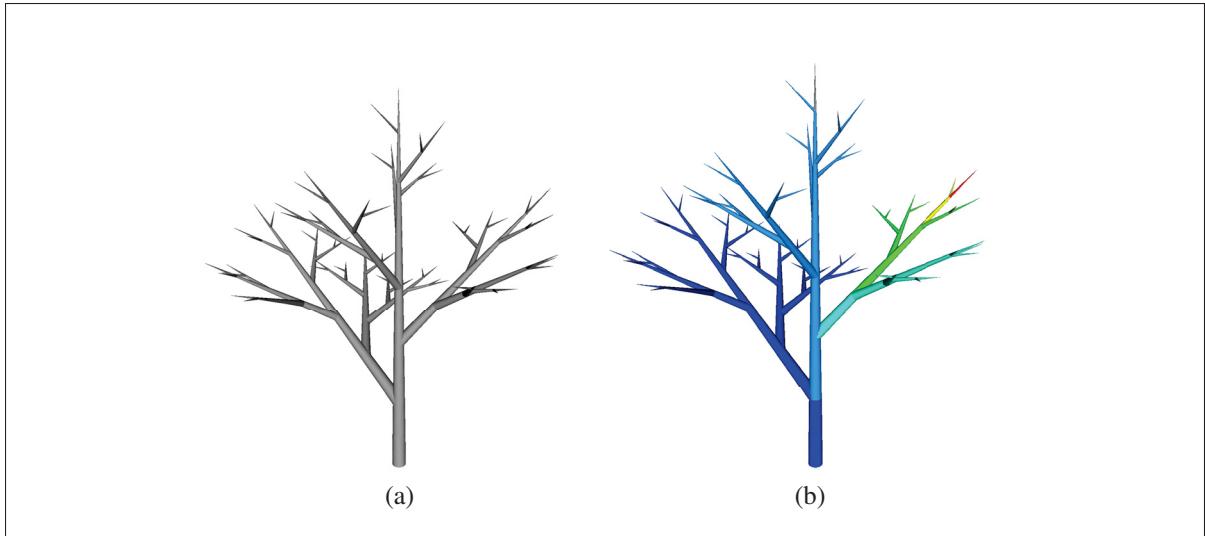


Figure 2.5 Illustration par un gradient de couleur de la relation entre la sélection et l'arbre de dérivation.

ou plusieurs éléments dans la liste de sélection. La multisélection ne se limite donc pas à une zone rectangulaire, mais peut se faire par l'ajout de plusieurs zones. La figure 2.6 montre par exemple la sélection de trois groupes de branches répandus dans de multiples endroits dans l'arbre 3D.

### 2.3 Regénération

Étant donné un nœud parent désigné pour une génération, ses nœuds enfant sont retirés de façon à ce qu'il devienne lui-même un nœud feuille. Ainsi, quand un nœud est sélectionné pour y effectuer une regénération, la première étape consiste à monter dans la hiérarchie c'est-à-dire vers les nœuds parents. La génération, c'est-à-dire la formation d'un nouvel ensemble de hiérarchie de nœuds, commence alors à partir du nœud jusqu'auquel nous sommes remontés dans l'arbre de dérivation.

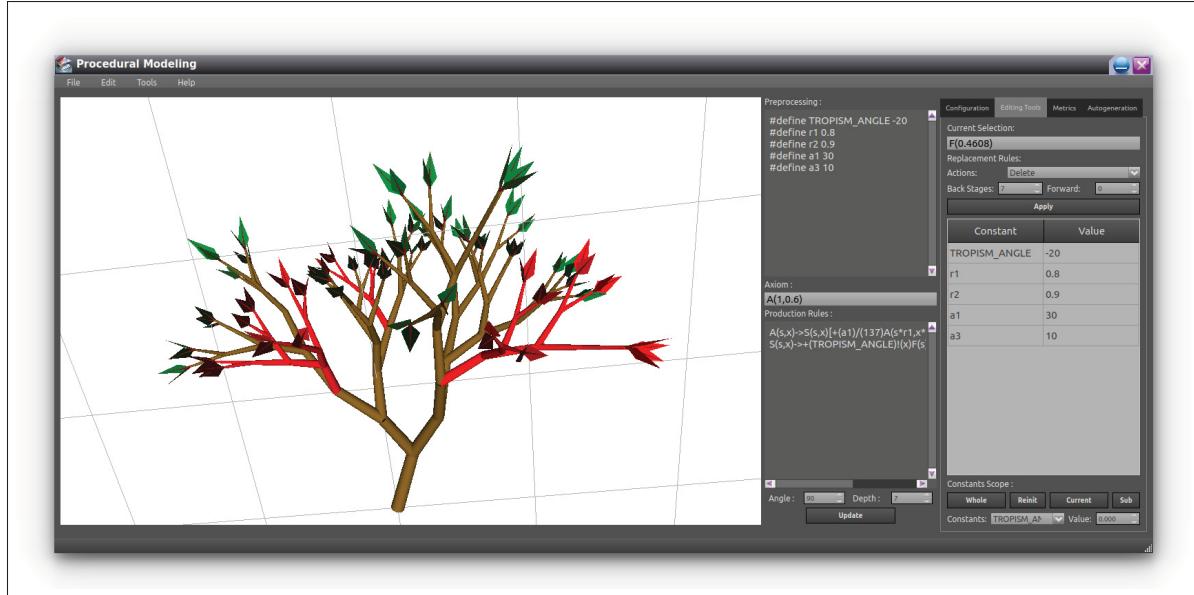


Figure 2.6 Interface utilisateur montrant la sélection multiple.

### 2.3.1 Redéfinition de constante

Pour changer l'aspect du modèle, nous proposons de modifier les paramètres en agissant sur les constantes de la grammaire. L'extrait 2.1 montre un exemple de grammaire où cinq constantes ont été définies en utilisant la directive `#define`. Avant la régénération, l'utilisateur peut agir via l'interface en modifiant ces constantes avant d'appliquer de nouvelles dérivations.

**Extrait 2.1** Exemple type de grammaire où une régénération peut s'appliquer

```
#define r1 0.9           /* contraction ratio 1 */
#define r2 0.7           /* contraction ratio 2 */
#define a1 10            /* branching angle 1 */
#define a2 60            /* branching angle 2 */
#define wr 0.707         /* width decrease rate */

n:      10
w:      A(1,10)
p:

A(1,w):* -> !(w)F(1)[&(a1)B((1*r1),(w*wr))]
```

```

/(180)[&(a2)B((1*r2),(w*wr))]
B(1,w):* -> !(w)F(1)[+(a1)$B((1*r1),(w*wr))]
[-(a2)$B((1*r2),(w*wr))]

```

### 2.3.2 Regénération sélective

Ce cas de regénération se présente lorsque l'utilisateur effectue une multisélection dans le but de regénérer plusieurs éléments dans l'arbre 3D. La sélection multiple introduit de nouvelles contraintes lors d'un processus de regénération. Des conflits peuvent en effet survenir dans le cas où l'ascension à partir d'une feuille de l'arbre de dérivation d'un élément rend obsolète l'ascension d'un autre élément. Ce cas se présente lorsqu'en demandant l'ascension à partir d'un élément, un autre élément faisant l'objet d'une requête séparée d'ascension pourrait disparaître lors de l'exécution de la première ascension. Dans ces circonstances, nous comparons les multiples requêtes d'ascension et retenons celle dont le nombre d'ascension est maximal, annulant ainsi toutes les autres requêtes.

Selon la quantité d'éléments sélectionnés et l'étendue de l'impact qu'une ascension peut avoir, on voudrait balancer le nombre d'ascensions qui seraient effectuées à partir des feuilles. Entre autres, si un utilisateur a par exemple effectué deux groupes de sélection (figure 2.7), il serait contre-indiqué de faire pour chaque groupe un même nombre d'ascensions dans l'arbre de dérivation. Effectuer deux montées dans l'arbre de dérivation dans les deux groupes n'aura pas le même effet. Dans la figure 2.7, deux montées dans une sélection ne contenant qu'une feuille supprimeraient trois feuilles non sélectionnées. Une telle modification irait en bonne partie à l'encontre de la sélection de l'utilisateur. À l'inverse, deux montées dans l'arbre de dérivation pour le groupe de sélection contenant quatre feuilles aurait l'effet escompté.

### 2.4 Coupe minimum de graphe

Notre méthode doit contenir un mécanisme qui peut décider du nombre de regénérations à faire selon le contenu de la sélection. Ce mécanisme est expliqué dans un exemple de la figure 2.8

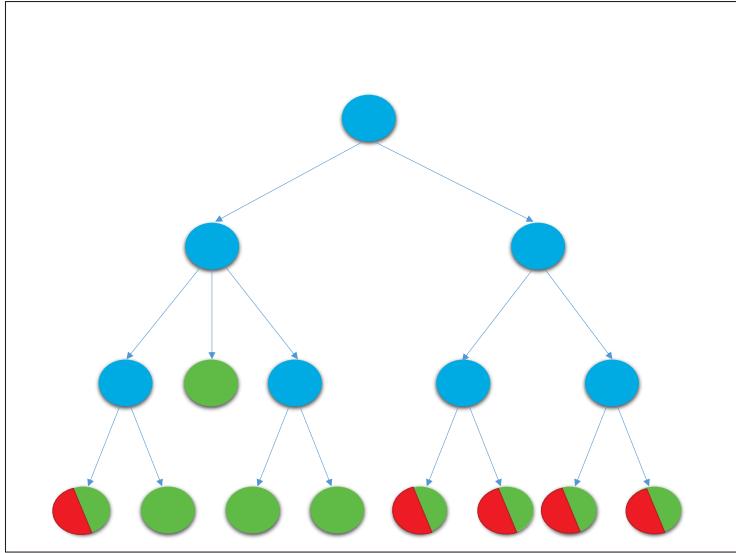


Figure 2.7 Illustration d'une sélection (nœuds comportant du rouge). Les nœuds comportant du vert sont des feuilles. Les nœuds comportant du bleu sont des nœuds parents.

où les nœuds feuille correspondent à la sélection de l'utilisateur. Les nœuds en jaune sont les nœuds qui seront choisis par l'algorithme pour des fins de regénération, donc les nœuds enfants de ces nœuds en jaune seront effacés pour donner lieu à une autre configuration. Les nœuds qui seront enlevés sont les nœuds en dessous de la coupe en ligne pointillée.

#### 2.4.1 Transcription de l'arbre de dérivation en graphe

La recherche des nœuds à choisir pour la regénération peut être transcrise en un problème consistant à départager les nœuds dans l'arbre de dérivation en deux groupes : ceux qui seront gardés et ceux qui seront effacés pour des fins de regénération. Ce problème peut être ramené à un problème de minimisation d'énergie par des coupes dans un graphe, où tous les nœuds dans le graphe appartiennent soit à la source, soit au drain (Kolmogorov & Zabih, 2001).

Le graphe orienté  $G = (V, E)$  est constitué des sommets du graphe  $V$ , et  $E \subset V^2$  l'ensemble des arêtes. À l'intérieur du graphe, on distingue deux sommets  $s$  et  $t$  respectivement la source et le drain. Aucune arête n'entre dans  $s$ , et aucune arête ne sort de  $t$ . Considérons également une fonction  $w : E \rightarrow \mathbb{R}^+ \cup +\infty$  qui associe un poids (ou capacité) aux arêtes. Tel qu'indiqué

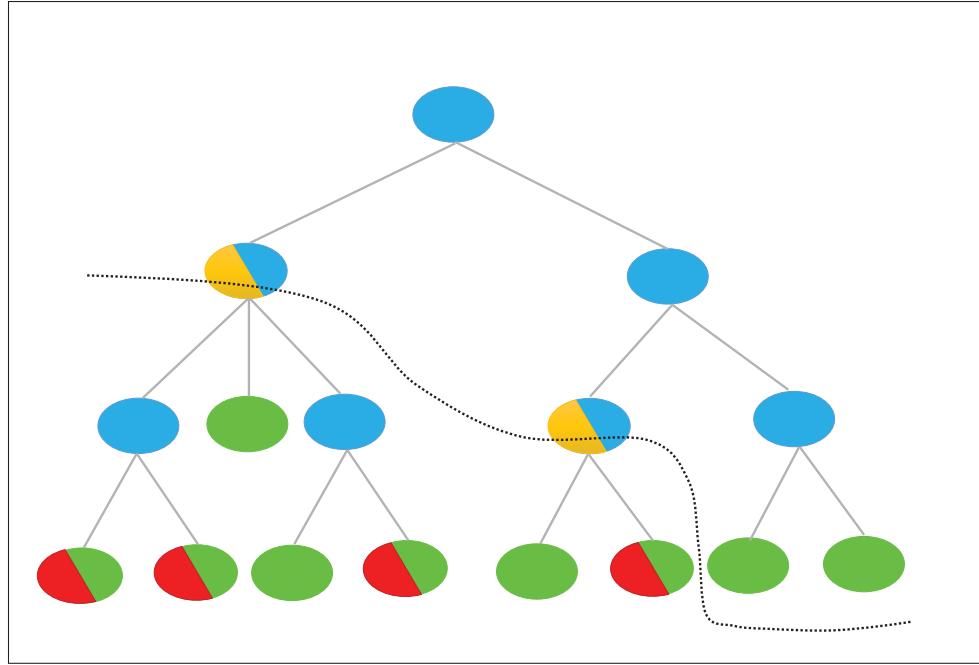


Figure 2.8 Choix des nœuds assignés aux régénérations (nœuds comportant du jaune) en fonction des sélections de l'utilisateur. Les couleurs bleue, verte et rouge sont associées respectivement aux nœuds parents, aux feuilles et aux sélections.

plus haut, dans les graphes orientés, la coupe se fait sur les arêtes  $E$  de manière à enlever toute liaison entre  $s$  et  $t$ . Le poids d'une coupe  $(S, T)$  est le terme  $\sum_{\substack{(x,y) \in E \\ x \in S, y \in T}} w(x,y)$ . Une coupe minimale pour le graphe  $G$  est par définition une coupe de poids minimal. La figure 2.9 est une illustration de la découpe d'un graphe orienté. La sortie de cet outil de découpe correspond sous certains aspects au type de sortie que nous voulons avoir pour notre arbre de dérivation. Le problème consiste alors à transcrire l'arbre de dérivation en une forme de graphe orienté.

Par contre dans les arbres de dérivation, la coupe sera un peu plus complexe. En effet, dans les arbres de dérivation, il faut prendre en compte qu'il doit être impossible de couper et séparer un nœud enfant de ses nœuds frères. Plus précisément, dans les arbres de dérivation, on ne peut pas couper une arête en tant que tel. La découpe se fait plutôt au niveau des nœuds comme dans la figure 2.8, car toute coupe au niveau d'une arête séparerait un ou plusieurs nœuds de son ou ses frères. Finalement, on peut partir de la convention que les nœuds traversés par la coupe (en jaune dans la figure 2.8) resteront dans l'arbre de dérivation. Une fois ces contraintes établies,

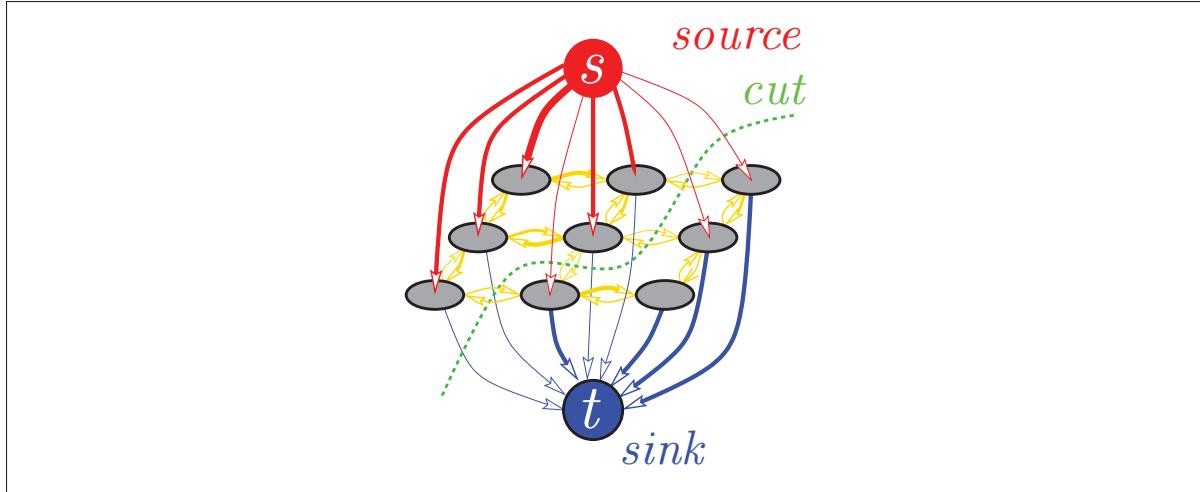


Figure 2.9 Aperçu d'une coupe dans un graphe orienté.  
Image tirée de Boykov & Kolmogorov (2004).

nous pouvons convertir l'arbre de dérivation en un graphe orienté. Quelques principes régissent cette conversion :

- Les arêtes de l'arbre de dérivation, ne pouvant pas être traversées par une coupe, seront converties dans le graphe orienté en des arêtes avec une capacité infinie ;
- Chaque nœud de l'arbre de dérivation est remplacé dans le graphe orienté par une succession de trois éléments, dont respectivement un nœud, une arête avec une capacité à déterminer, et un dernier nœud. Cette conversion est justifiée, comme vue précédemment, par le fait que l'on avait cherché à couper l'arbre de dérivation dans les nœuds plutôt que dans les arêtes ;
- Une source est ajoutée au graphe orienté et connectée aux nœuds racines ;
- Le drain quant à lui se connecte aux nœuds feuilles.

La figure 2.10 illustre en image le processus de conversion d'un arbre de dérivation (figure 2.10a) en un graphe orienté (figure 2.10b). Les nœuds  $N_i$  de l'arbre deviennent dans le graphe les deux nœuds  $v_i$  et  $v'_i$ , connectés ensuite par l'arête  $(v_i, v'_i)$ . Les arêtes dans l'arbre de dérivation de la figure 2.10a gardent la même couleur dans sa transcription en graphe orienté de la figure 2.10b.

### 2.4.2 Détermination des capacités des arêtes du graphe

L’assignation des capacités des arêtes dans le graphe orienté part du principe que l’ascension vers le nœud parent a lieu lorsque la moitié ou plus de ses nœuds enfants sont sélectionnés. Lorsqu’exactement la moitié des nœuds enfants est sélectionnée, on aurait donc une équiprobabilité entre une coupe sur le nœud parent et une coupe sur les nœuds enfants. De plus, nous souhaitons que lorsqu’aucune sélection n’est faite par l’utilisateur, la coupe va s’effectuer vers la partie en aval du graphe. Chaque nœud sélectionné dans l’arbre va ajouter aux capacités des arêtes concernées, mais aussi devrait transférer une certaine quantité de capacité vers les arêtes en amont.

Dans les prochaines sections, nous suggérons un système de transfert de capacité pour chaque feuille sélectionnée. Sa mise en place s’est faite à partir d’un graphe balancé, c’est-à-dire que tous les nœuds parents possèdent un nombre pair d’enfant, mais aussi que tous les nœuds parents possèdent le même nombre d’enfants. La figure 2.11 montre une telle configuration, où tous les nœuds parents possèdent exactement deux enfants. À partir de ce graphe, les capacités attribuées à chaque sélection, ainsi qu’aux parents qui sont liés, seront calculés. Une fois que les capacités des sélections sont obtenues, nous allons appliquer les résultats pour tous les autres types de graphe. Le système a été éprouvé expérimentalement et donne des résultats satisfaisants dans toutes les configurations que nous avons testées.

#### 2.4.2.1 Établissement des conditions initiales

Les capacités initiales correspondant aux feuilles sont initialisées à  $w = 1$ . La capacité initiale de l’arête correspondant à un nœud parent de l’arbre est égale au double de la somme des capacités des arêtes correspondant à ses nœuds enfants de façon à ce que la coupe se fait initialement aux feuilles. La figure 2.11 montre en image ce système d’assignation des capacités initiales. Dans cette figure, nous pouvons voir les coupes  $\Gamma_k$  traversant horizontalement le graphe à un niveau  $k$  ( $k \in [0, n]$ ) à partir des feuilles du graphe.

Quand l'utilisateur effectue une sélection, il y a des ajouts de capacité sur les nœuds sélectionnés, et ensuite un ajout récursif de capacité vers toutes les arêtes en amont. Le problème revient ensuite à déterminer l'équation qui régit la quantité de capacité à ajouter par rapport à la quantité ajoutée lors d'une sélection utilisateur. L'objectif est de calculer la quantité  $\alpha_k$  qui est la capacité que l'on ajoute à la capacité initiale à chaque fois qu'une feuille est sélectionnée.

#### 2.4.2.2 Calcul des capacités des sélections pour chaque niveau

Soit  $\alpha_0$  la capacité ajoutée à une arête sélectionnée (donc sur une feuille),  $\alpha_k$  sera la capacité ajoutée à chaque nœud parent de niveau  $k$ . Si on dénote par  $\Gamma'_k$  les nouvelles valeurs des coupes  $\Gamma_k$  après que les sélections soient faites,  $\Gamma'_k$  contiendra alors deux termes dont la capacité initiale, et la capacité qui a été ajoutée. La capacité initiale des feuilles étant initialisée à  $w = 1$ , alors, pour une coupe au niveau des feuilles, la valeur initiale de la coupe est égale à la somme des capacités des feuilles :  $\Gamma_0 = \eta$ ,  $\eta$  le nombre de nœuds feuilles. La nouvelle coupe sera donnée par  $\Gamma'_0 = \Gamma_0 + \alpha_0 \times \frac{\eta}{2}$  où  $\frac{\eta}{2}$  est la quantité de feuilles sélectionnés et  $\alpha_0$  la capacité qu'on ajoute à une feuille si elle est sélectionnée. En suivant le même fil de raisonnement, on construit alors la suite arithmétique :

$$\begin{aligned}\Gamma'_0 &= \alpha_0 \times \frac{\eta}{2} + \eta \\ \Gamma'_1 &= \alpha_1 \times \frac{\eta}{2} + 2\eta \\ \Gamma'_2 &= \alpha_2 \times \frac{\eta}{2} + 4\eta \\ &\dots \\ \Gamma'_k &= \alpha_k \times \frac{\eta}{2} + 2^k\eta\end{aligned}$$

Comme on veut que la coupe sur l'arête parent soit égale à celle sur les enfants quand exactement la moitié des enfants est sélectionnée, alors on a :

$$\Gamma'_0 = \Gamma'_1 = \dots = \Gamma'_k$$

Pour  $\Gamma'_0 = \Gamma'_1$  :

$$\alpha_0 \times \frac{\eta}{2} + \eta = \alpha_1 \times \frac{\eta}{2} + 2\eta$$

$$\alpha_1 = \alpha_0 - 2$$

Pour  $\Gamma'_1 = \Gamma'_2$  :

$$\alpha_1 \times \frac{\eta}{2} + 2\eta = \alpha_2 \times \frac{\eta}{2} + 4\eta$$

$$\alpha_2 = \alpha_1 - 4$$

On en déduit alors par généralisation :

$$\alpha_{k+1} = \alpha_k - 2^{k+1} \quad (2.1)$$

Afin d'obtenir l'expression générale de l'expression du transfert de capacité, on développe à partir de l'équation 2.1 :

$$\begin{aligned} \alpha_k &= \alpha_{k-1} - 2^k \\ \alpha_{k-1} &= \alpha_{k-2} - 2^{k-1} \\ \implies \alpha_k &= \alpha_{k-2} - 2^{k-1} - 2^k \\ &\quad \dots \\ \implies \alpha_k &= \alpha_{k-2} - 2^{k-k+1} - \dots - 2^{k-1} - 2^k, \end{aligned}$$

d'où :

$$\alpha_k = \alpha_0 - (2^1 + 2^2 + \dots + 2^k) \quad (2.2)$$

L'expression 2.2 peut être simplifiée encore en analysant la suite géométrique :  $U_{p+1} = 2 \times U_p$  dont la raison est  $q = 2$ .

La somme de ses  $k$  premiers termes sera :

$$\sum_{j=0}^k U_j = U_0 \times \frac{1-q^{k+1}}{1-q}.$$

En remplaçant les termes :

$$\begin{aligned} \sum_{j=0}^k U_j &= U_0 + U_1 + \dots + U_{k-1} + U_k = 2^0 + 2^1 + \dots + 2^k = \frac{1-2^{k+1}}{1-2} \\ \implies 2^0 + 2^1 + \dots + 2^k &= 2^{k+1} - 1. \end{aligned}$$

On en déduit :

$$2^1 + \dots + 2^k = 2^{k+1} - 2. \quad (2.3)$$

En portant 2.3 dans 2.2, on obtient :

$$\alpha_k = \alpha_0 - (2^{k+1} - 2). \quad (2.4)$$

#### 2.4.2.3 Contraintes sur la capacité initiale de sélection

Le terme  $\alpha_0$  étant la capacité ajoutée sur les arêtes sélectionnées dans le graphe ;  $\alpha_k$  est la capacité ajoutée au  $k$ -ième parent de l'arête sélectionnée. Rappelons que l'arête sélectionnée dans le graphe n'est autre que la conversion du nœud sélectionné dans l'arbre de dérivation. Le choix de  $\alpha_0$  peut être arbitraire. Toutefois, l'équation 2.1 permet d'évaluer que  $\alpha_{k+1} < \alpha_k$ , la suite  $\alpha_k$  est de ce fait décroissante. L'élément  $\alpha_k$  peut être une valeur négative à un moment donné. Ceci nous expose alors à une coupe  $\Gamma'_k$  de valeur négative, ce qui serait inadapté au graphe orienté que nous voulons construire. Par conséquent, la contrainte sur  $\alpha_0$  part de la

condition sur une sélection totale (de toutes les feuilles de l'arbre) :

$$\begin{aligned}\Gamma'_k &= \eta \times \alpha_k + 2^k \eta \geq 0 \\ \alpha_k &\geq -2^k \\ (2.4) \implies \alpha_0 - (2^{k+1} - 2) &\geq -2^k\end{aligned}$$

D'où la condition sur la capacité initiale :

$$\alpha_0 \geq 2^k - 2 \quad (2.5)$$

Il faut alors prendre une valeur de niveau maximal  $k_{max}$  égale au nombre de dérivation  $n$ . Le tableau 2.2 donne les valeurs à ajouter à la capacité initiale, en partant de la feuille si  $n = 10$ .

Tableau 2.2 Valeur de capacité ajoutée par niveau à partir de la feuille.

k	0	1	2	3	4	5
$\alpha_k$	1024	1022	1018	1010	994	962

Les équations dérivées jusqu'à maintenant assument qu'exactement la moitié des feuilles sont sélectionnées pour chaque nœud parent. La figure 2.12 donne un exemple où l'utilisation de ces équations donne le résultat attendu pour une autres configuration. Dans toutes nos expérimentations, ces équations ont donné les résultats attendus en termes de remontée dans l'arbre de dérivation. La figure 2.13 illustre un graphe qui n'est pas balancé et où la moitié des feuilles est sélectionnée. Dans cette figure, on peut vérifier que les coupes aux niveaux  $k = 0, 1, 2$  ont la même valeur. Cette configuration donne un résultat similaire à notre spécification initiale pour le calcul des  $\alpha_k$ , en ce sens que les coupes sont de valeur égale si la moitié des feuilles est sélectionnée. L'approche sur l'attribution des capacités démeure donc stable par rapport au résultat obtenu dans cette configuration, le principe étant que la coupe peut se faire à n'importe quel niveau lors d'une valeur de coupe égale. La figure figure 2.14 présente la même configuration que dans la figure 2.13, mais avec toutes les propositions de coupes possibles. La coupe minimale retrouvée va être  $\Gamma'' = 2051$  : la coupe proposée dans notre approche est

meilleure. D'autres cas de sélection (plus que la moitié ou moins que la moitié des feuilles) sont aussi gérés correctement lors de nos études expérimentales comme en témoignent les exemples des figures VII-1 et VII-2 dans l'annexe VII sur les analyses complémentaires de la coupe de graphe.

### 2.4.3 Administration d'une contrainte de sélection

Jusque-là, la suppression des nœuds enfants a lieu si la moitié ou plus de ces nœuds enfants sont sélectionnés. Cette façon de procéder peut être handicapante, car elle enlève un certain contrôle à l'utilisateur. Pour pallier ce problème, nous ajoutons un paramètre  $\rho$  à l'équation. Ce paramètre devient le pourcentage de nœuds enfants sélectionnés, pour qu'une suppression de nœuds enfants survienne. Les démarches précédentes sont équivalentes à l'utilisation de  $\rho = 0.5$ . Nous reprenons ensuite les raisonnements précédents, cette fois en partant de la valeur de la coupe :

$$\Gamma'_k = \alpha_k \times \rho \eta + 2^k \eta \quad (2.6)$$

Le terme général du transfert de capacité pour la sélection avec contrainte devient :

$$\alpha_k = \alpha_0 - \frac{1}{\rho} \times (2^k - 1) \quad (2.7)$$

Le paramètre  $\rho$  est finalement ajouté pour contrôler la coupe (figure 2.15). Il permet ainsi d'imposer une contrainte sur le comportement de la régénération dans le cas des multisélections initiées par l'utilisateur. Dans l'interface utilisateur,  $\rho$  est une valeur comprise dans l'intervalle  $[0, 100]$ , qui sera ramenée entre zéro et un, conformément aux expressions mathématiques précédemment exposées.

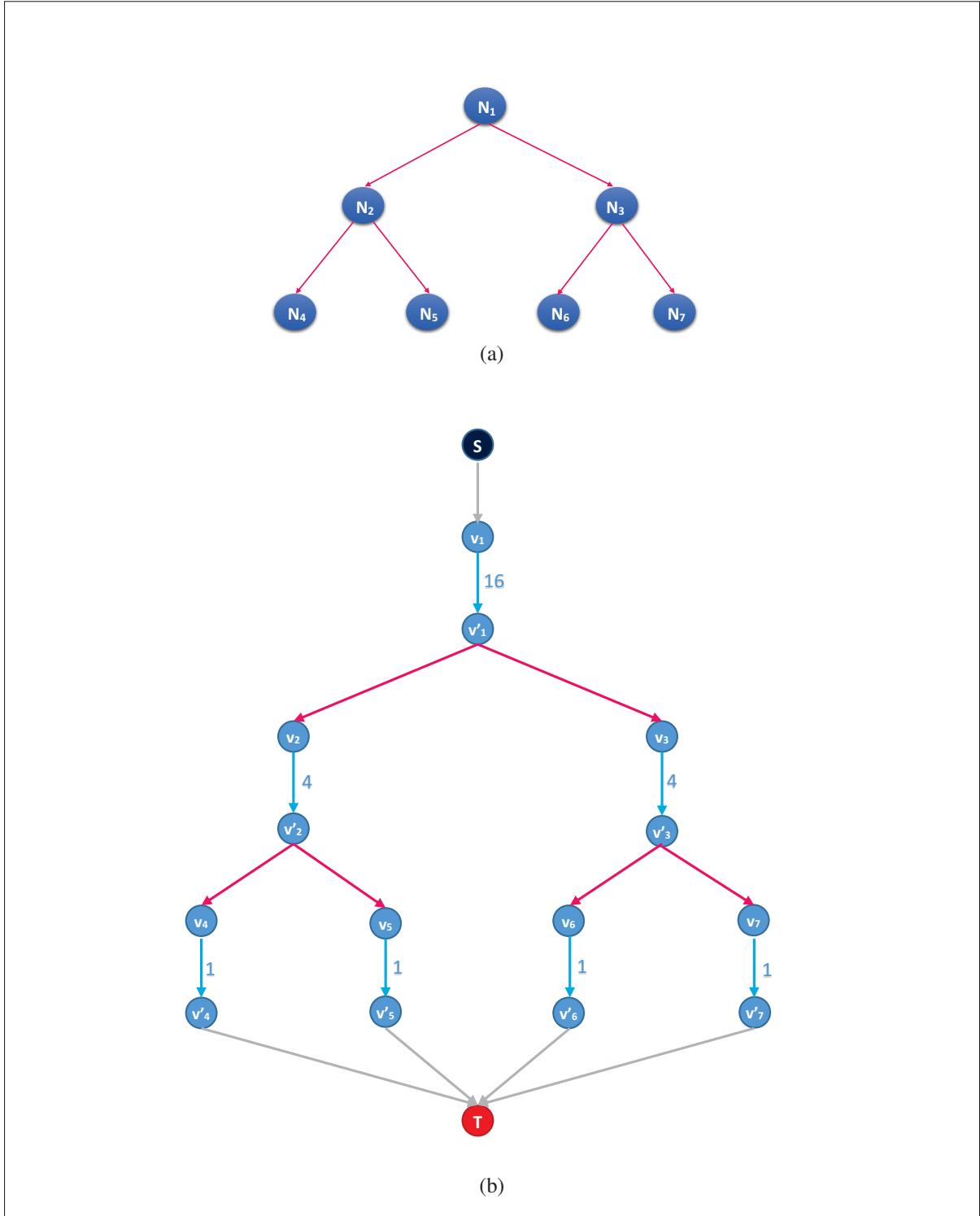
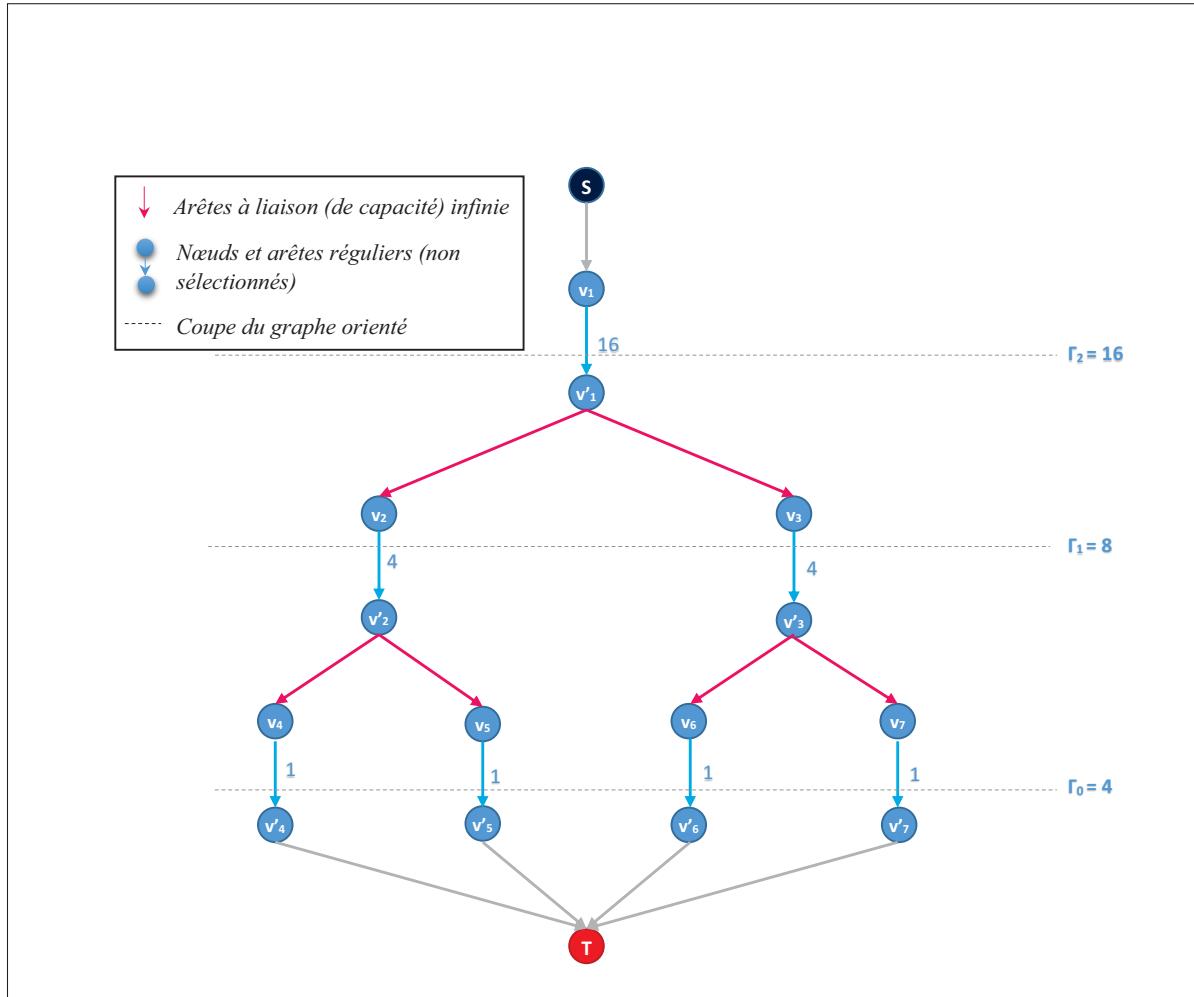


Figure 2.10 Exemple de processus de transcription d'un arbre de dérivation (a) en un graphe orienté (b).

Figure 2.11 Coupes  $\Gamma_k$  aux niveaux  $k = 0, 1, 2$ .

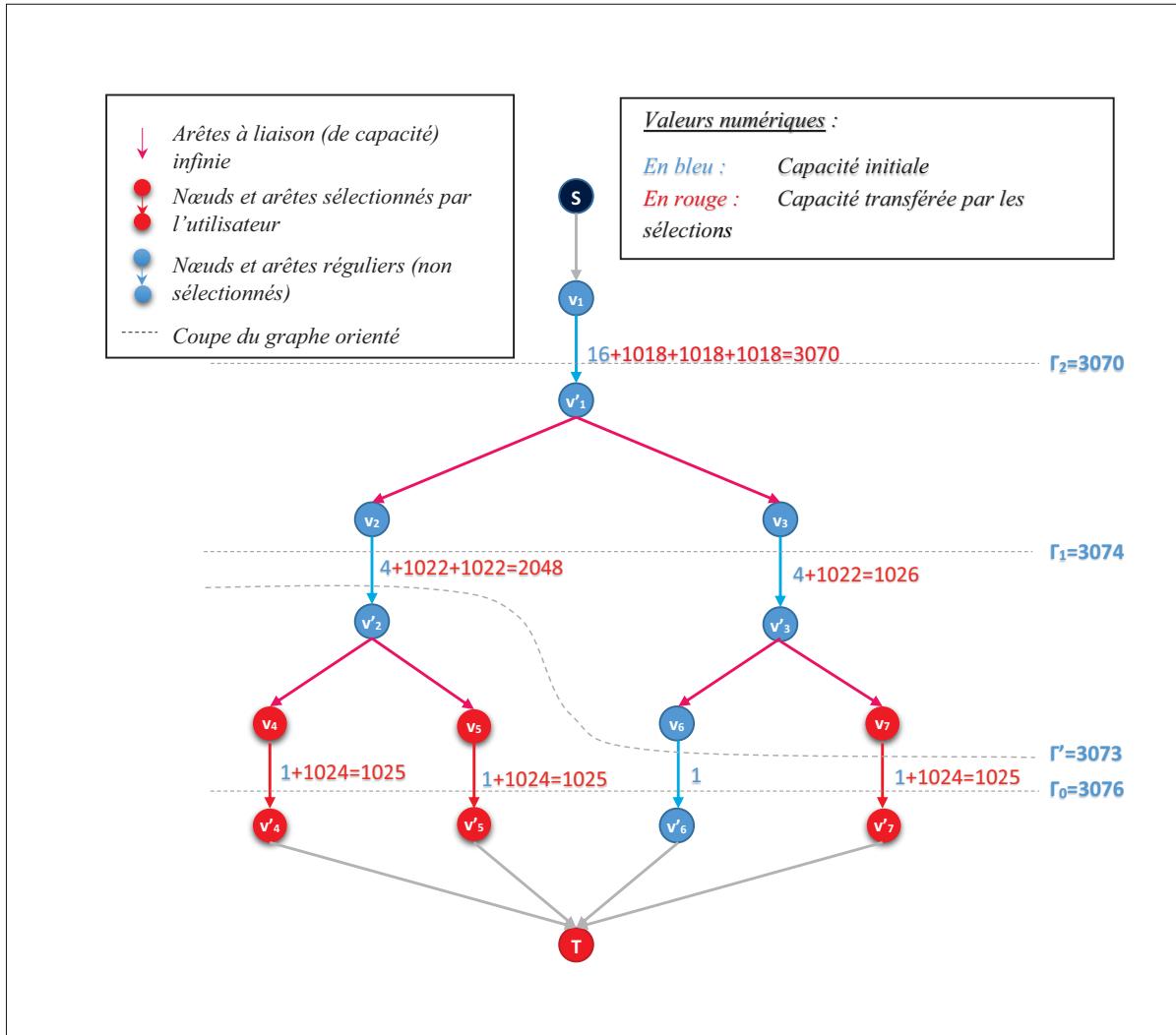


Figure 2.12 Illustration du transfert des capacités des arêtes sélectionnées vers les arêtes en amont.

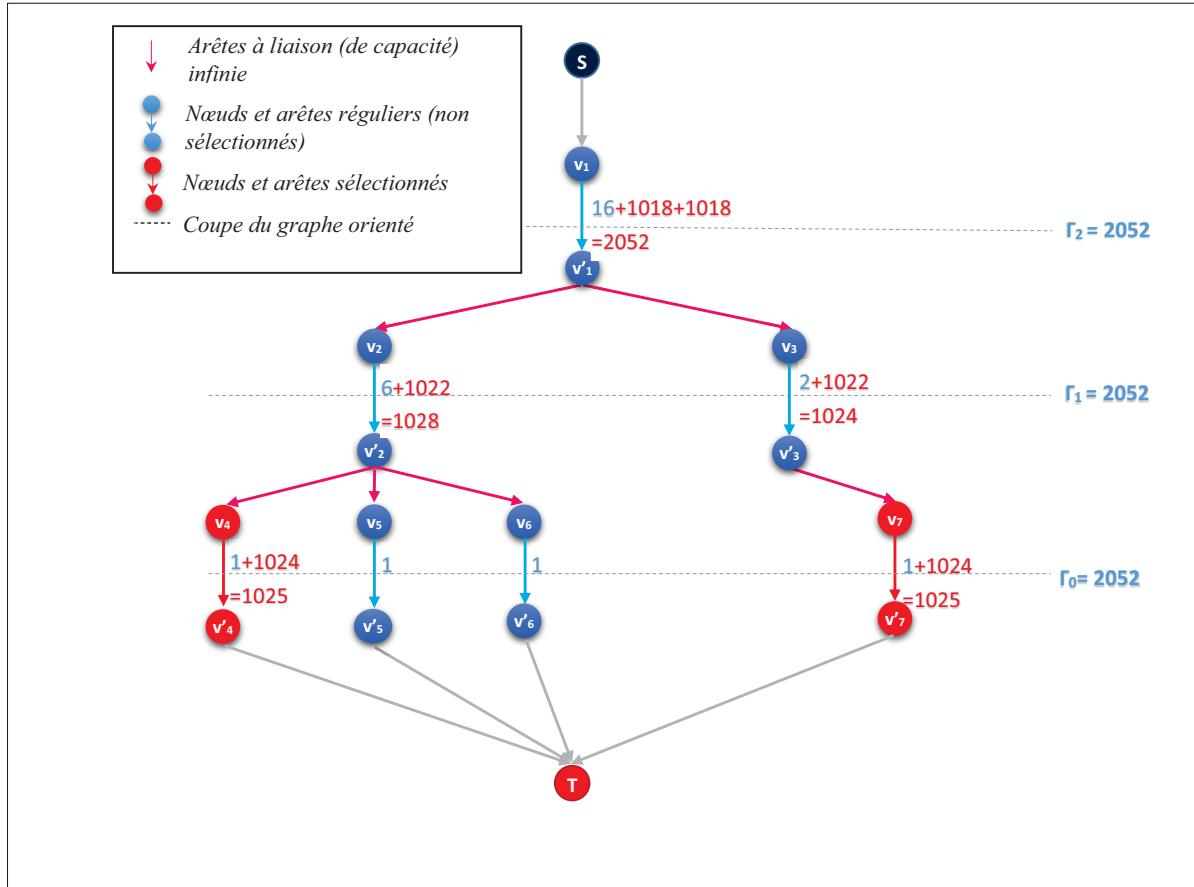


Figure 2.13 Cas de sélection de la moitié des feuilles dans un graphe aléatoire.

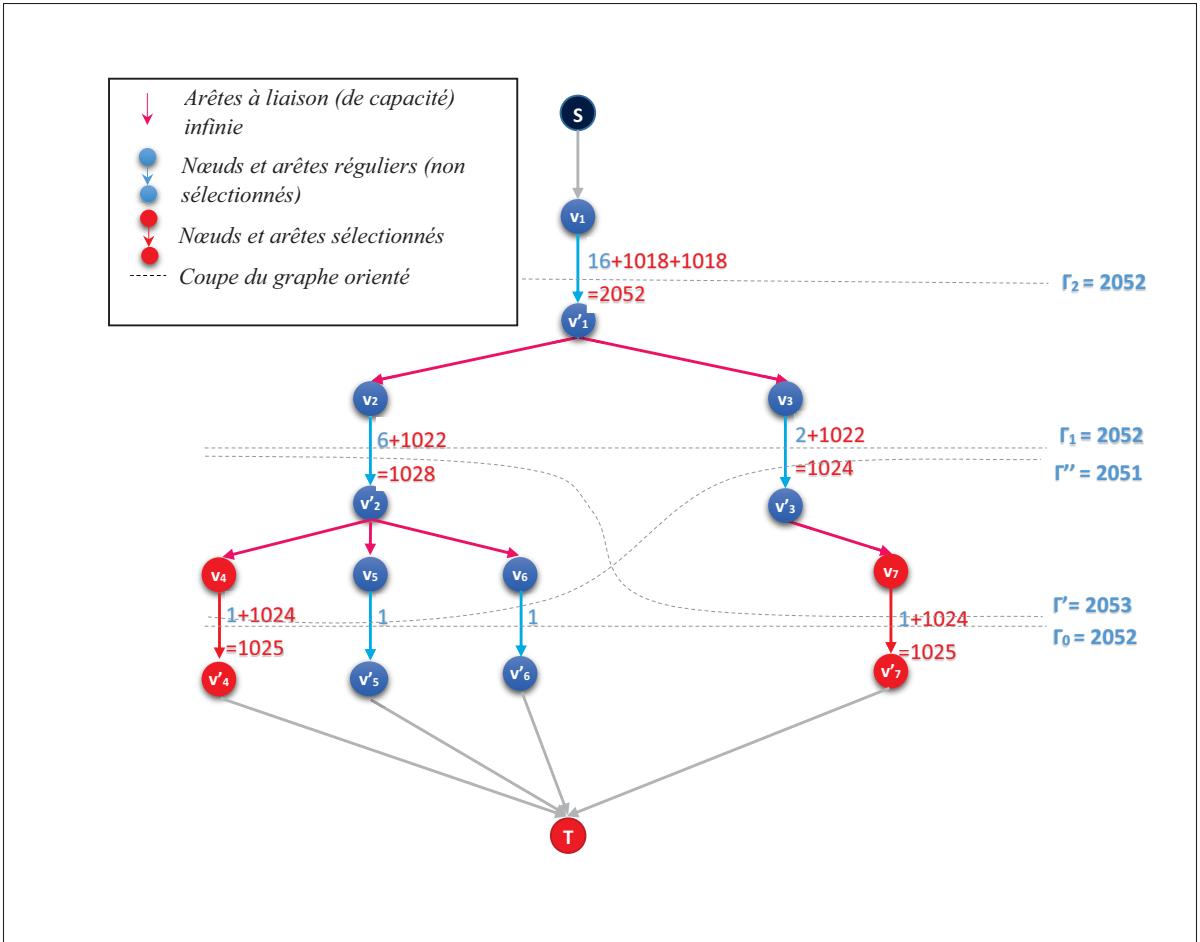


Figure 2.14 Illustration de toutes les coupes possibles dans un arbre non balancé.

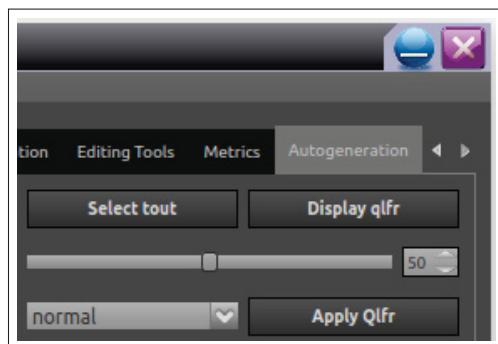


Figure 2.15 Interface utilisateur de la personnalisation du paramètre  $\rho$  de pourcentage de sélection à considérer.



## CHAPITRE 3

### MESURE DE DIFFÉRENCE ENTRE LES ARBRES

L'objectif d'une régénération dans le cadre de ce projet est de reconstituer une forme qui serait nuancée par rapport à l'arbre 3D original. Les changements dans l'arbre s'obtiennent en faisant varier des paramètres qui ont été définis dans les directives de la grammaire du L-système. Dans ce chapitre, nous allons introduire une méthode pour obtenir des diversités à partir de l'arbre de départ en analysant les relations avec les variations des paramètres du L-système.

#### 3.1 Enveloppement des branches

Afin d'avoir des mesures communes sur toutes les variétés d'arbres, les méthodes mises en place vont consister d'abord en un système de mesure adapté aux divers changements de forme. Tel est le cas d'une boîte englobante orientée sur les axes, qui est applicable pour tout type de forme 3D. Cette boîte formera une enveloppe autour de toutes les branches de l'arbre 3D. À partir de cette boîte englobante, nous pouvons dériver d'autres métriques qui seront détaillées dans les sections qui vont suivre.

##### 3.1.1 Boîte englobante orientée

Les deux types de boîte englobante les plus utilisés sont : les boîtes englobantes alignées sur les axes, et les boîtes englobantes orientées. Celles orientées présentent la caractéristique d'être plus petite que toutes les boîtes qui contiennent l'ensemble des points considérés. Les boîtes alignées sur les axes quant à elles ont une restriction d'orientation.

La boîte englobante orientée (ou *OBB*, pour *Oriented Bounding Box*) est une boîte qui renferme un ensemble de points définis ( $P$ ). Dans un espace à trois dimensions, cette boîte est le minimum, en volume, de toutes les boîtes pouvant contenir l'ensemble de points ( $P$ ).

Dans la figure 3.1, la sélection de l'utilisateur est en surbrillance et enveloppée par une OBB en fil de fer rouge. Le choix des points à inclure est conçu de manière à ce que chaque type de

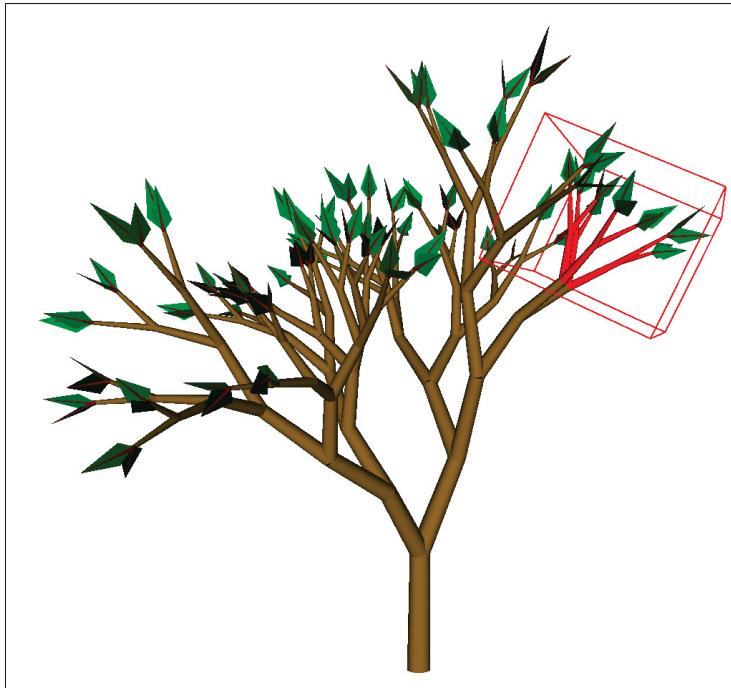


Figure 3.1 Illustration d'un enveloppement d'une sélection par une OBB.

forme géométrique possède un ensemble de points qui sera pris en compte pour le calcul de la boîte englobante. Par exemple, une branche d'un arbre en 3D possède la forme d'un cylindre généralisé, où les points inclus dans le calcul se trouvent sur la bordure du cylindre (sur la périphérie des faces supérieures et inférieures), et y sont ajoutés deux points, respectivement le centre de la face supérieure et celui de la face inférieure. La figure 3.2 montre les points, en rouge, qui sont les descripteurs dans cet exemple de cylindre généralisé.

### 3.1.2 Calcul de la boîte orientée

Notre objectif est d'aligner l'OBB avec l'objet 3D à envelopper. Dans le cas d'une branche, la forme attendue est intuitivement longue et fine. Dans la figure 3.3, les éléments en surbrillance sont des branches individuelles et l'OBB correspondante se trouve à être dans la même direction que la branche.

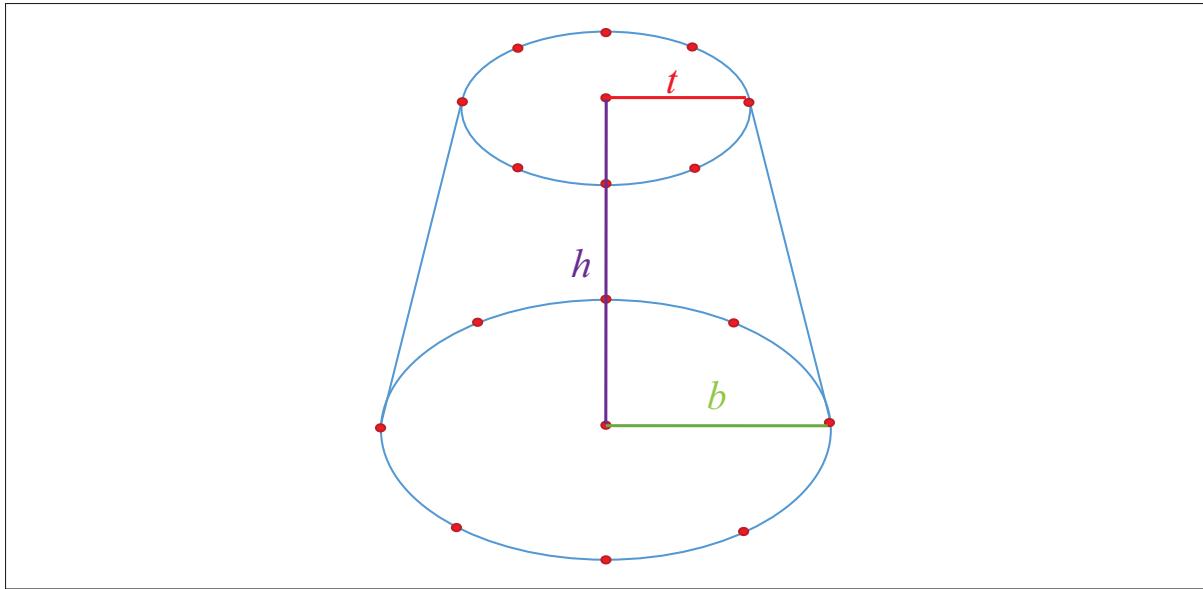


Figure 3.2 Descripteur de forme d'une branche, représenté par les points rouges.

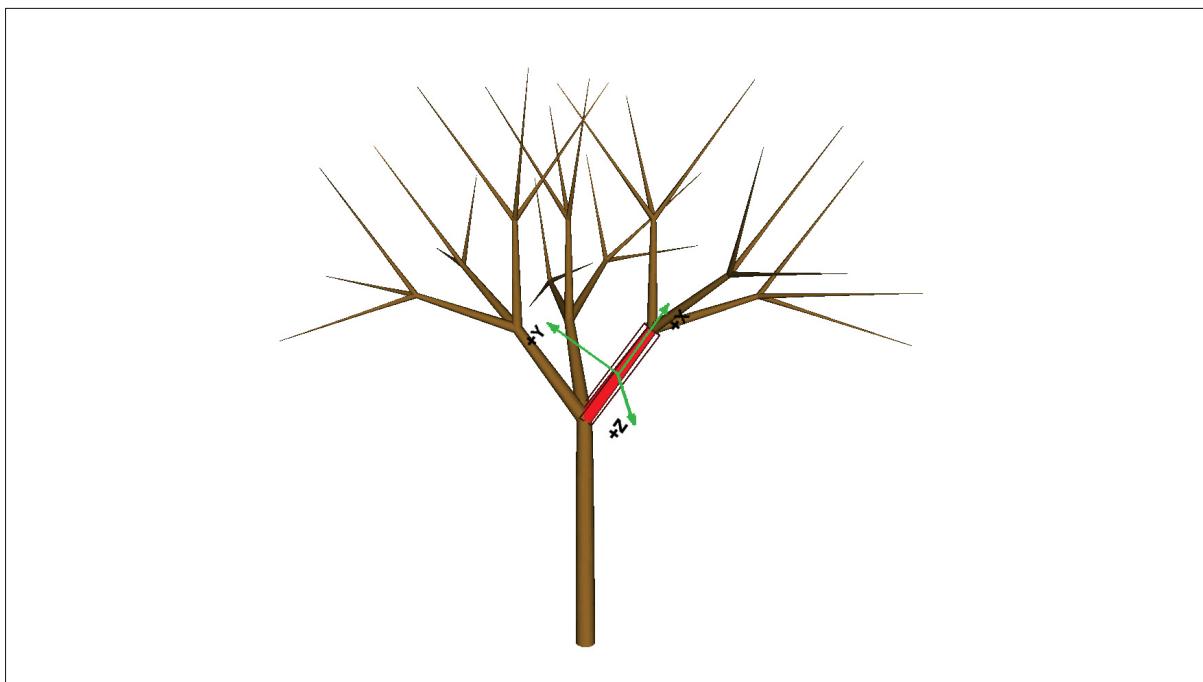


Figure 3.3 Application d'une OBB sur une branche sélectionnée. Les axes naturels de la boîte sont indiqués en vert.

L'approximation de l'orientation d'un nuage de points peut se déterminer à l'aide d'une matrice de covariance. Cette matrice est symétrique et réelle, et est de dimension  $3 \times 3$ . Les vecteurs propres de la matrice sont orthogonaux entre eux et dénotent les directions qui maximisent et minimisent la variance.

La matrice de covariance pour  $n$  variables aléatoires est une matrice carrée dont les éléments  $a_{i,j}$  sont :

$$a_{i,j} = \text{Cov}(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)] \quad (3.1)$$

où  $\mu_i$  dénote la moyenne de  $X_i$ , et  $E(X_i)$  l'espérance mathématique (dans notre cas la moyenne des  $X_i$ ). Ainsi pour les points  $P(x, y, z)$  à envelopper par l'OBB, les éléments de la matrice vont consister aux covariances entre chaque paire des coordonnées  $x$ ,  $y$  et  $z$ . Intuitivement, cette matrice est une généralisation de la covariance pour une dimension multiple, c'est-à-dire une mesure de corrélation entre deux variables aléatoires. La détermination de la boîte englobante consiste donc à calculer cette matrice de covariance, ensuite les vecteurs propres de cette matrice, et enfin analyser les sommets sur chacune des directions afin d'y trouver les extrêmes.

### 3.2 Définition des paramètres de contrôle

Les boîtes englobantes calculées dans la section précédente édifieront les calculs des mesures géométriques et des proportions de grandeurs diverses, qui, au final, procureront des valeurs quantitatives pour différencier deux arbres procéduraux. Plus les écarts observés entre les mesures effectuées sont grands, plus les arbres générés sont variés en apparence. 10 paramètres sont calculés, regroupés en trois classes de mesure :

- Les dimensions géométriques ;
- Les quantités d'éléments ;
- Les proportions.

### 3.2.1 Dimensions géométriques

Les dimensions géométriques sont des grandeurs calculées sur tout ou sur une partie de l’arbre.

#### 3.2.1.1 Longueur des branches

En observant la figure 3.2, la longueur de la branche s’identifie à la hauteur du cône tronqué, c’est-à-dire la distance cartésienne entre le centre de la base et le centre du sommet du cône.

#### 3.2.1.2 Volume des branches

Si on dénote par  $b$  le rayon de la base du cône, et  $t$  le rayon au niveau du sommet, le volume d’une branche se calcule grâce à la formule :

$$Vol_{branche} = \frac{1}{3}\pi(b^2 + bt + t^2)h$$

où  $h$  est hauteur du cône, qui se trouve à être la longueur de la branche.

#### 3.2.1.3 Surface des branches

Lors de l’évaluation des surfaces des branches, le calcul va se limiter aux surfaces latérales des cônes. Ces surfaces latérales sont les seules visibles pendant la visualisation de l’arbre, car d’autres branches sont d’une manière générale accolées sur la base et le sommet des branches. Ainsi, la surface de la branche est donnée par la formule :

$$Surf_{branche} = \pi(b + t)\sqrt{(b - t)^2 + h^2}$$

#### 3.2.1.4 Volume de la boîte

Le volume de la boîte englobante est donnée par :

$$Vol_{obb} = longueur \times largeur \times hauteur$$

### 3.2.1.5 Surface de la boîte

Pour chacune des six faces de la boîte englobante, la surface est calculée puis ajoutée à la surface totale. En tout, trois paires de surface sont à calculer.

$$Surf_{obb} = 2 \times \text{hauteur} \times \text{largeur} + 2 \times \text{hauteur} \times \text{longueur} + 2 \times \text{largeur} \times \text{longueur}$$

### 3.2.2 Quantité d'éléments

Une façon supplémentaire de différencier d'une manière concrète est de faire le décompte des éléments constituant l'arbre. Un parcours de la chaîne permet de dénombrer les branches en prenant en compte uniquement les mots de l'alphabet du L-système qui dessinent des branches.

### 3.2.3 Proportions

Une troisième catégorie de métrique met en relation les deux catégories précédentes, c'est-à-dire les dimensions géométriques et les quantités d'éléments.

#### 3.2.3.1 Densité de branches

Nous pouvons évaluer la densité en calculant le rapport entre le nombre de branches et le volume de la boîte englobante concernée :

$$D_{branches} = \frac{N_{branches}}{Vol_{obb}}$$

#### 3.2.3.2 Moyenne des longueurs des branches

Pour la deuxième métrique de cette catégorie, nous faisons le calcul de la moyenne des longueurs toutes les branches à l'intérieur de la boîte englobante.

$$M_{longueur} = \frac{\sum_{branches} \text{longueur}}{N_{branches}}$$

### 3.2.3.3 Moyenne des rayons des branches

Nous évaluons aussi la moyenne des longueurs de toutes les branches à l'intérieur de la boîte englobante :

$$M_{rayon} = \frac{\sum_{\text{branches}} t + \sum_{\text{branches}} b}{2 \times N_{\text{branches}}},$$

$t$  et  $b$  étant respectivement le rayon du sommet et de la base du cône.

## 3.3 Variation de constantes des paramètres dans la grammaire

Après avoir conçu nos différentes métriques pour évaluer les modèles d'arbres procéduraux, la prochaine étape est de modifier les grammaires et de mesurer les différences à l'aide des métriques. Pour rappel, les constantes de grammaire pour un L-système donné sont des valeurs définies en début de la grammaire. Elles s'identifient par la directive `#define` et sont enregistrées lors la première lecture de la grammaire.

### 3.3.1 Proportion de modifications des constantes de la grammaire

L'idée est de varier chaque constante par une valeur  $\beta$ . Pour chaque constante  $k_i$ , on évalue les métriques des arbres pour les nouvelles grammaires utilisant les constantes suivantes :

- $k_i - \beta k_i = k_i^-$
- $k_i$
- $k_i + \beta k_i = k_i^+$

Nous cherchons à générer toutes les combinaisons d'arbre possibles sur le principe de variation de constantes sur trois échelons. Le premier choix de la valeur du paramètre  $\beta$  a été assigné à  $\beta = 0.1$  après une séance d'expérimentation avec un étudiant en maîtrise dans le laboratoire multimédia. Cette valeur correspond à la valeur où le changement au niveau de la forme et de la structure de l'arbre est suffisamment perceptible.

Afin de mieux contrôler l'évaluation de l'impact d'une modification de constantes, chaque combinaison de variation de constantes s'appliquera à une règle à la fois. Le nombre d'arbres à générer est donc fonction du nombre de règles procédurales et du nombre de constantes dans la grammaire. La formule relative au nombre d'arbres à générer est alors :

$$N_{arbres} = 3^m \times N_{regles}, \quad (3.2)$$

où  $m$  est le nombre de constantes, et  $N_{regles}$  le nombre de règles procédurales dans la grammaire.

Une illustration du nombre de générations à exécuter peut s'effectuer en reprenant l'exemple de l'extrait 2.1. Toutes les valeurs de  $\{k_i^-, k_i, k_i^+\}$  sont présentées au tableau 3.1, et de nouvelles grammaires à base de ces valeurs sont produites. En tout, le nombre d'arbres qui seront générés, d'après la formule 3.2, est :

$$\begin{aligned} N_{arbres} &= 3^m \times N_{regles} \\ &= 3^5 \times 2 \\ &= 243 \times 2 \\ &= 486. \end{aligned} \quad (3.3)$$

Tableau 3.1 Exemple de valeur de constantes dans une grammaire et ses variations.

Constante	$k_i^-$	$k_i$	$k_i^+$
r1	0.81	0.9	0.99
r2	0.63	0.7	0.77
a1	9	10	11
a2	54	60	66
wr	0.6363	0.707	0.7777

### 3.3.2 Analyse en composantes principales sur les métriques

L’analyse des grammaires et arbres générés est réalisée à l’aide d’une analyse en composantes principales sur la différence des métriques de chaque nouvel arbre avec l’arbre initial. Les différences sont calculées en pourcentage suivant la formule :

$$\delta = 100 \times \left( \frac{\mu'}{\mu} - 1 \right) \quad (3.4)$$

avec :

- $\delta$  la différence en pourcentage ;
- $\mu'$  la nouvelle valeur de la métrique mesurée avec la grammaire générée ;
- $\mu$  la nouvelle valeur de la métrique de l’arbre initial.

Le choix de la méthode d’analyse est motivé par le fait que l’analyse en composantes principales permet de transformer des variables linéaires statistiques corrélées entre elles en variables indépendantes, c’est-à-dire, linéaires décorrélées. Comme dans le calcul des OBB, l’analyse en composante principale s’opère en décomposant en valeur propre la matrice de covariance obtenue à partir des données initiales. L’examen des données sur les axes indépendants met en avant la variance des données.

#### 3.3.2.1 Format et dimensions de la population

Les 10 différences calculées avec l’équation 3.4 formeront les données sur lesquelles nous appliquerons l’analyse en composantes principales. Notre population statistique sera donc constituée de  $N_{arbres}$ , avec 10 dimensions à évaluer pour chacun d’entre eux.

En partant des 10 métriques qui sont nos variables aléatoires, nous choisissons de retenir les trois premiers vecteurs propres de l’analyse en composantes principales, ces vecteurs propres étant les plus importants en termes de variance. La projection des données sur les trois premiers vecteurs propres offre des informations plus profitables, admettant que les pertes de dimensionnalité n’altèrent pas trop les caractéristiques des données.

### 3.3.2.2 Calcul de la moyenne et de la covariance

Comme dans le cas des OBB, la covariance est calculée avec l'équation 3.1. La covariance est nulle lorsque la corrélation entre les deux variables est nulle. Dans le tableau 3.2, le volume de la boîte englobante et le nombre de branches sont non corrélés parce que leur covariance est nulle. Ce tableau représente la matrice de covariance des métriques calculées sur la base de la grammaire de l'extrait 2.1.

Tableau 3.2 Matrice de covariance calculée sur l'extrait 2.1.

	Vobb	Sobb	Sb	Vb	Db	Nb	Remp	Lb	Ml	Mr
Vobb	2000	1200	590	400	-1600	0	-1200	820	820	0
Sobb	1200	780	370	260	-1100	0	-800	520	520	0
Sb	590	370	210	140	-540	0	-380	290	290	0
Vb	400	260	140	100	-370	0	-260	200	200	0
Db	-1600	-1100	-540	-370	1800	0	1300	-730	-730	0
Nb	0	0	0	0	0	0	0	0	0	0
Remp	-1200	-800	-380	-260	1300	0	970	-520	-520	0
Lb	820	520	290	200	-730	0	-520	410	410	0
Ml	820	520	290	200	-730	0	-520	410	410	0
Mr	0	0	0	0	0	0	0	0	0	0

Les lignes et colonnes du tableau 3.2 sont constituées des types de métriques dont :

- Vobb : Volume de la boîte englobante ;
- Sobb : Surface de la boîte englobante ;
- Sb : Surface de la branche ;
- Vb : Volume de la branche ;
- Nb : Nombre de branche à l'intérieur de la boîte englobante ;
- Db : Densité de branche ;
- Remp : Taux de remplissage de la boîte englobante ;
- Lb : Longueur de branche ;
- Ml : Moyenne de longueur des branches ;

- $Mr$  : Moyenne des rayons des branches.

### 3.3.2.3 Vecteurs et valeurs propres

La matrice de covariance est décomposée en vecteurs propres unitaires. À l'image de ce que nous avons vu dans les calculs des boîtes englobantes, les vecteurs propres mettent en avant les tendances générales des données.

### 3.3.2.4 Choix des vecteurs caractéristiques

Les vecteurs propres qui correspondent aux valeurs propres les plus élevées forment les composantes principales de l'ensemble des données. Dans le tableau 3.3, les valeurs propres sont déjà disposées dans l'ordre croissant, et les vecteurs propres qui correspondent pour chaque valeur sont dans la même colonne. Les composantes principales sont de ce fait classées en ordre croissant d'importance dans le même tableau. Il est maintenant possible d'éliminer des composantes qui ont peu ou pas de signification.

Tableau 3.3 Résultat de la décomposition en vecteurs propres de la matrice de covariance.

Vecteurs propres										
[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	
0	0	0	0	0.36	0.32	0.02	-0.43	0.53	0.55	
0	0	0	0.01	-0.78	-0.39	0.17	-0.22	0.22	0.35	
0	0	0.05	-0.84	0.15	-0.2	0.25	0.36	0.11	0.17	
0	0	0	0.48	0.36	-0.41	0.63	0.24	0.06	0.12	
1	0	0	0	0	0	0	0	0	0	
0	0	0	-0.01	0.14	-0.45	-0.29	-0.10	0.64	-0.52	
0	1	0	0	0	0	0	0	0	0	
0	0	0	0.02	-0.30	0.57	0.48	0.21	0.40	-0.38	
0	0	0.70	0.22	-0.08	0.06	-0.32	0.52	0.19	0.24	
0	0	-0.72	0.14	-0.08	0.06	-0.32	0.52	0.19	0.24	
Valeurs propres										
0	0	0	0	0.51	1.3	4.3	150	390	6200	

Nous avons décidé de retenir trois vecteurs propres pour représenter nos données. Les données seront alors représentées dans un espace 3D. À partir des trois vecteurs retenus, nous allons constituer une matrice :

$$MatCaract = (vprop_1 \ vprop_2 \ vprop_3)$$

Les données seront transformées dans cet espace en projetant sur les vecteurs propres :

$$NouvDonnees = MatCaract^T \times Donnees$$

Le système intègre un outil de visualisation des données compressées dans les trois dimensions retenues. La figure 3.4 est une capture de cet outil, lorsqu'appliqué à l'exemple précédent. Chaque point dans cette figure correspond à un arbre. Dans la figure 3.4, l'axe des abscisses (en rouge) correspond à la valeur propre la plus influente ; ensuite vient l'axe en vert, et finalement en bleu.

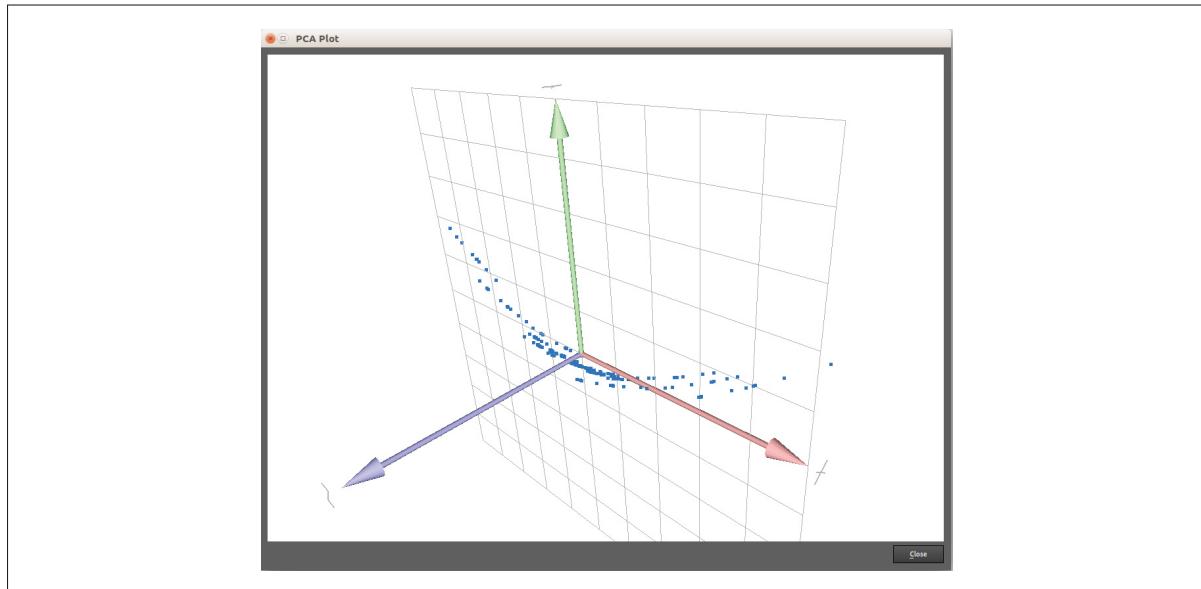


Figure 3.4 Aperçu de la distribution du nuage de points dans le repère formé par les vecteurs propres.

### 3.3.2.5 Visualisation des arbres générés pour l'analyse

Léo Villalba, stagiaire au laboratoire multimédia, a travaillé sur le positionnement des caméras de façon à bien visualiser les arbres. L'aboutissement de ce positionnement permettra aux participants des expérimentations de visualiser l'arbre à partir d'un emplacement déterminé en fonction de la taille de l'arbre. En effet, à chaque fois qu'une constante est modifiée, la taille de l'arbre 3D peut changer drastiquement, ainsi que sa structure. Il convient alors de recadrer la caméra en prenant en compte le volume de l'arbre ainsi que sa longueur.

## 3.4 Rendus des arbres nouvellement générés

La figure 3.5 est un aperçu des arbres proposés aux participants des expérimentations. L'arbre initial est dessiné à la figure 3.5a tandis que les images 3.5b à 3.5f sont les arbres obtenus en effectuant les variations sur les constantes de la grammaire *Tree1*.

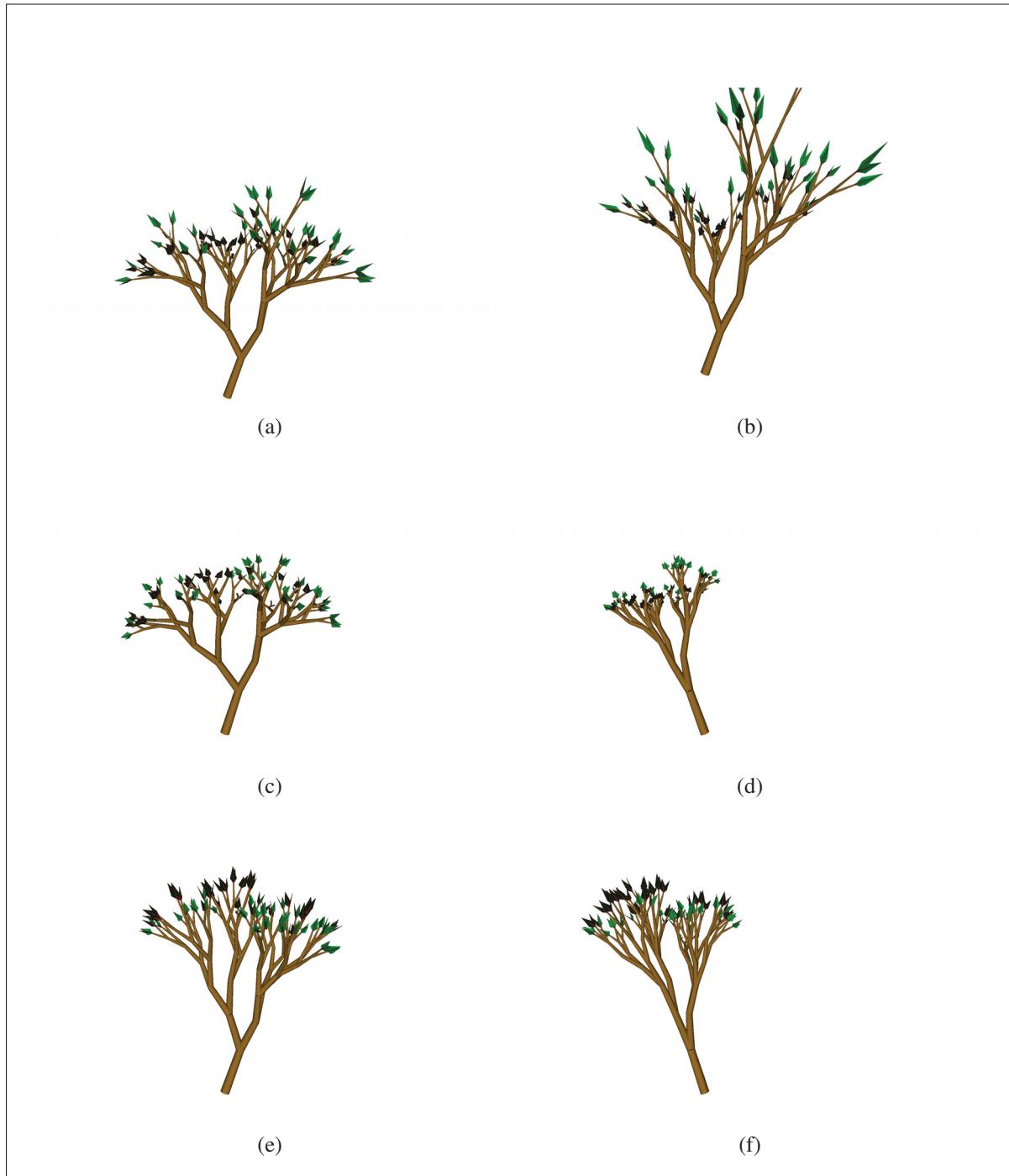


Figure 3.5 Variétés d'arbres obtenus à partir de la grammaire *Tree1*.

## CHAPITRE 4

### ÉTUDE EMPIRIQUE AVEC SUJET HUMAIN

Dans le chapitre précédent, nous avons vu qu'un nombre élevé d'arbres pouvait être généré en variant les constantes de grammaire. Étant donné  $N$  modèles d'arbres générés automatiquement, le problème consiste à retenir un nombre réduit d'arbres  $n$  qui engloberait du point de vue de l'apparence toutes les modifications possibles. Ce chapitre énumère les étapes effectuées pour sélectionner les arbres retenus pour les études ultérieures, le déroulement des études avec les participants et le traitement des informations recueillies auprès de ces derniers.

#### 4.1 Désignation des grammaires à soumettre aux participants

La détermination des vecteurs caractéristiques aboutit à la représentation des modifications dans un nuage de points 3D comme dans la figure 3.4. Chaque point du nuage représente un modèle d'arbre 3D avec ses propres combinaisons de constantes. Il sera assumé que plus la distance entre deux points est petite, plus les arbres vont se ressembler. En effet, des variations de métriques similaires résultent à des superpositions de points dans l'espace PCA. L'idée consiste alors à balayer l'espace PCA tout en éliminant les points qui sont jugés très proches.

Une autre contrainte qu'il faut prendre en compte est le nombre final de modèles d'arbre 3D à présenter aux participants. Le projet a sélectionné sept grammaires à soumettre dans les expérimentations. Ces grammaires possèdent des nombres variés de constantes mais aussi de règles. Il en résulte que le nombre de points varie d'une grammaire à une autre. Toutefois, dans le cadre du projet, un nombre constant d'arbres retenus  $n$ , indépendant du nombre de points, a été choisi suivant des concepts d'analyse exposés en détail dans les sections qui viennent.

La technique consiste à calculer pour chaque grammaire la boîte alignée sur les axes qui englobe les points. Par la suite, les points qui seront choisis seront les points les plus proches des points aux extrémités et aux milieux de la boîte.

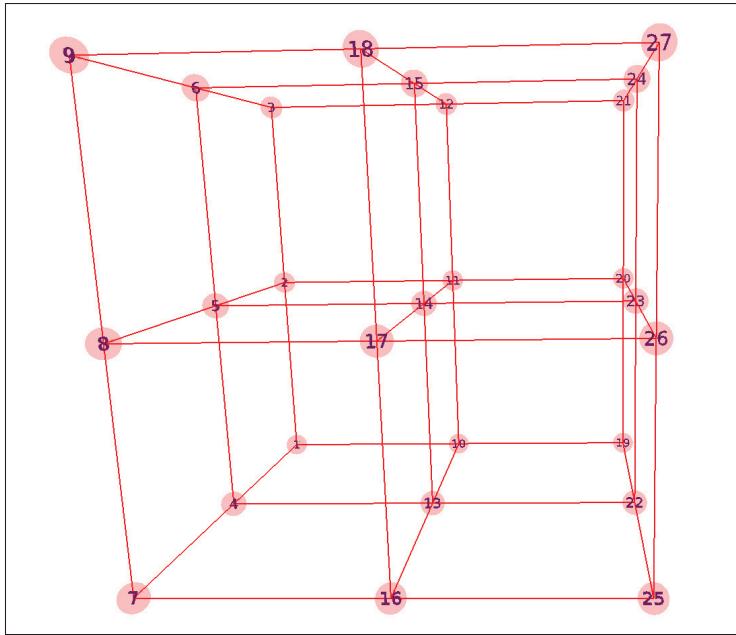


Figure 4.1 Emplacement des 27 points d'intérêts retenus.

La figure 4.1 est une illustration de la disposition de ces 27 points dans la boîte englobante, sans les nuages de points.

Les points les plus proches, en termes de distance, de ces points d'intérêts de la boîte seront donc retenus pour l'analyse, indépendamment du nombre de points du nuage. Ainsi, pour chacun des points dans le nuage, la distance euclidienne avec chacun des points d'intérêts de la boîte englobante sera calculée. Pour chaque point d'intérêt de la boîte, le point du nuage ayant la distance minimale par rapport à ce point sera retenu. Des  $N$  arbres pouvant être générés, 27 seront donc retenus.

À chacun des points est attribué un identificateur. Les identificateurs sont des entiers compris entre 1 à 27. Ainsi, dans la suite du document, on se réfèrera à l'identificateur  $i$ , comme l'arbre numéro  $i$ . Cette attribution est indépendante du type de grammaire à analyser.

## 4.2 Déroulement des expérimentations

Dans le cadre du projet, nous avons soumis à des utilisateurs des modèles procéduraux d’arbre en 3D créés en modifiant méthodiquement les constantes d’une grammaire initiale. À cet effet, des procédures ont été entamées pour s’assurer que les expérimentations se déroulent dans les normes et éthiques des recherches scientifiques. Le projet a été alors soumis pour évaluation au comité d’éthique de la recherche de l’ÉTS (CÉR), sous l’intitulé « *Analyse des variations d’apparence d’arbres virtuels* », avec le Professeur Eric Paquette comme *Responsable du projet*.

À cette fin, les membres de l’équipe du projet ont préparé plusieurs documents qui ont été envoyés au CÉR :

- Formulaire de présentation d’un projet de recherche : il s’agit de la présentation du projet, entre autres ses objectifs et ses motivations ;
- Formulaire d’information et de consentement : ce formulaire explique au participant son implication, et la confidentialité qui y est rattachée ;
- Fiche d’identification : remplie par le chercheur responsable des expérimentations, elle enregistre un code anonyme du participant et la date de son intervention ;
- Interface utilisateur : application, où les arbres seront présentés aux participants (figure 4.2) ;
- Questionnaire : ce document est l’exemplaire des questions auxquelles les participants auront à répondre ;
- Recrutement des participants : c’est un appel public aux participants, pouvant être affiché éventuellement sur les emplacements prévus à cet effet ;
- Demande et confirmation d’octroi : ce document relate le financement du projet.

Le recours à l’étude empirique avec sujets humains a pour but d’évaluer si l’approche pour identifier des variables intéressantes d’arbre est valable. À la fin des expérimentations, nous

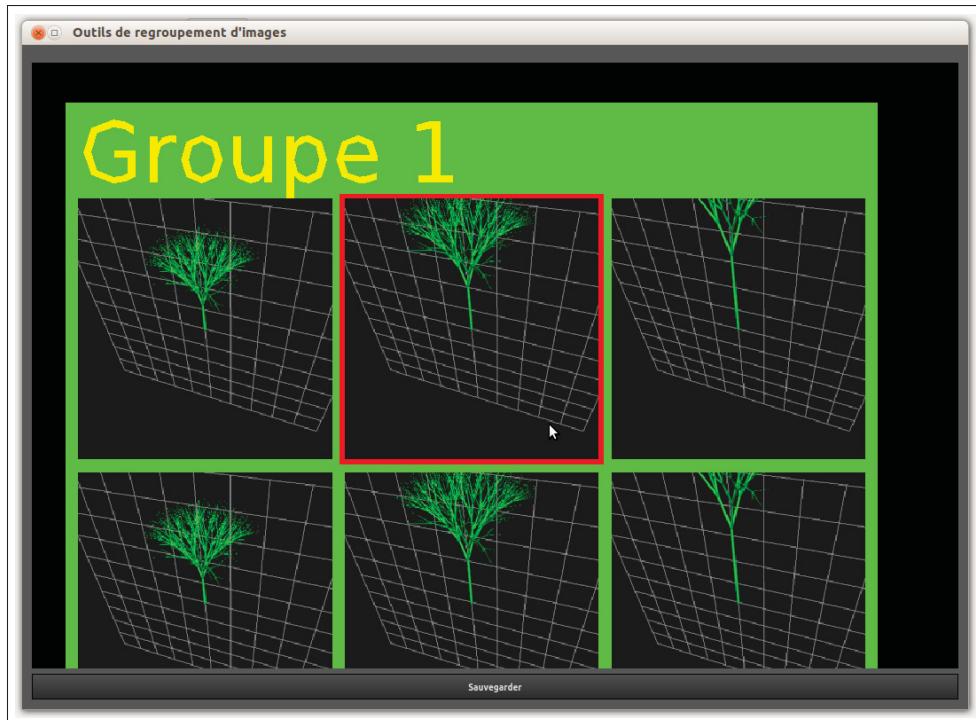


Figure 4.2 Interface utilisée dans les phases d’expérimentation

cherchons à obtenir un système capable de répondre à des directives de haut niveau telles que « *augmenter la densité* » ou encore « *réduire l’étalement* ».

Le recrutement se fait par voie publicitaire et invitation par courriel. Les candidats recrutés dans les réseaux de connaissances des chercheurs du projet sont privilégiés. Les candidats feront leurs interventions à l'aide d'un logiciel conçu spécialement pour cette phase d'expérimentation du projet. Le temps imparti pour une intervention est de 30 minutes. Afin d'avoir une marge d'erreur de 20 %, 9 fois sur 10, nous désirions recruter au moins 17 candidats. Le processus de recrutement a abouti à un total de 20 participants pour les expérimentations.

Dès le début de la séance avec un participant, un formulaire de consentement lui est présenté. Le participant va ainsi réaliser un à cinq tests selon le temps disponible. À la fin des 30 minutes, le candidat pourra quitter ou terminer le test qu'il réalise à ce moment. Un test équivaut à une grammaire, tandis qu'un total de sept grammaires sera présenté au travers de l'étude. Ce-

pendant, pour des fins d'analyse statistique, une grammaire en particulier a été exclusivement choisie afin de figurer parmi les grammaires analysées par la majorité des candidats.

### **4.3 Catégorisation des informations à recueillir**

Les participants auront à classifier des images qui leur seront présentées, et qualifier les images qu'ils ont identifiées. Le cadre conceptuel du projet s'appuie essentiellement sur l'analyse des classifications effectuées par les participants. Les données collectées détermineront si des couples d'images sont groupés systématiquement ensemble. Ces mêmes données mettront en relation les groupements et l'analyse en composante principale.

Chaque participant mettra à l'écrit ses appréciations sur la différence d'un groupe parmi les autres et si un groupe est utile. Une section dans les questionnaires collecte finalement des qualificatifs proposés par le participant, utilisant ses propres termes.

#### **4.3.1 Groupement des images et images représentatives**

Au cours du test, chaque participant regroupera les images de sorte que les arbres dans chaque groupe soient significativement différents des arbres dans les autres groupes, et que les arbres dans un même groupe soient semblables. Un groupe peut être composé d'une ou plusieurs images.

Conformément à l'exploitation de notre analyse en composantes principales, 27 images sont présentées pour chaque test. Pour chacun des groupes formés, exactement une image au sein du groupe est choisie pour représenter le groupe. Le remplissage des questionnaires par les participants commence seulement une fois les groupes identifiés. Le participant fournira ensuite une mesure subjective de l'apparence des arbres de chaque groupe. Une question demande au participant de choisir une image au sein du groupe pour représenter ce groupe. Pour chaque mesure demandée, une échelle de Likert (Rasmussen 1989) est proposée aux participants : « *Mauvaise* », « *Passable* », « *Bonne* » et « *Excellente* ».

La question sur la pertinence des images est formulée au participant comme suit : *Dans le contexte de la modélisation d'arbres 3D, indiquez si vous trouvez que les groupes que vous avez identifiés seraient utiles.* À titre d'exemple, pour la grammaire appelée *honda sympodial* listée dans l'extrait 2.1. Le participant portant l'identificateur 105 a formé 8 groupes à partir des 27 images qui lui ont été présentées. Le tableau 4.1 détaille ses appréciations des groupements qu'il a formés.

Tableau 4.1 Données reçues d'un participant sur l'échelle d'utilité de chaque groupe d'images.

Groupe	1	2	3	4
	Excellent	Excellent	Excellent	Excellent
	Bonne	Bonne	Bonne	Bonne
	Passable	Passable	Passable	Passable
	Mauvaise	Mauvaise	Mauvaise	Mauvaise
Groupe	5	6	7	8
	Excellent	Excellent	Excellent	Excellent
	Bonne	Bonne	Bonne	Bonne
	Passable	Passable	Passable	Passable
	Mauvaise	Mauvaise	Mauvaise	Mauvaise

#### 4.3.2 Différences intergroupes et informations d'attribut

Le participant devra donner une mesure subjective de la différence de chacun des groupes par rapport aux autres. Le même participant que dans l'exemple précédent a rapporté les données de différences intergroupes présentées dans le tableau 4.2 pour la même grammaire *honda sympodial*.

Dans cet exemple, on peut souligner entre autres les informations suivantes : les groupes d'arbre 3D n°1 et n°3 se distinguent à la fois entre eux et par rapport aux autres groupes. Aussi, d'après le même tableau, le groupe n°4 contient au moins une image qui a une forte ressemblance avec les modèles d'arbre 3D d'autres groupes.

Tableau 4.2 Information de différence entre les groupes selon le participant portant l'identificateur 105.

Groupes	1	2	3	4
	Excellent	Excellent	Excellent	Excellent
	Bonne	Bonne	Bonne	Bonne
	Passable	Passable	Passable	Passable
	Mauvaise	Mauvaise	Mauvaise	Mauvaise
Groupes	5	6	7	8
	Excellent	Excellent	Excellent	Excellent
	Bonne	Bonne	Bonne	Bonne
	Passable	Passable	Passable	Passable
	Mauvaise	Mauvaise	Mauvaise	Mauvaise

Finalement, il est demandé au participant de fournir un ou plusieurs qualificatifs pour chaque groupe. Les qualificatifs devraient être des termes qui seraient utiles pour la modélisation d'arbres virtuels.

#### 4.4 Observation des données sur les groupements et les images représentatives

Toutes les informations reçues depuis les questionnaires sont compilées dans une base de données. Ces données sont ensuite analysées sous différentes configurations, notamment par le profil des grammaires, ou encore la pertinence des images.

##### 4.4.1 Matrice de cooccurrence de groupement

Une information intéressante serait déjà de déterminer si un couple d'images tend à être regroupé plus souvent que d'autres couples, tant bien pour une grammaire que plusieurs. Rapelons que l'identification de nos images se fait par des indices allant de 1 à 27. Le but dans le calcul de la matrice de cooccurrence est tout d'abord de comparer pair à pair les images de manière à identifier lesquelles sont groupées ensemble le plus fréquemment.

Dans la figure 4.3, les lignes et les colonnes sont les indices de l'image. Par exemple, on peut noter que l'image n°1 a été groupé 21 fois avec l'image n°4, ou encore l'image n°27 n'a été

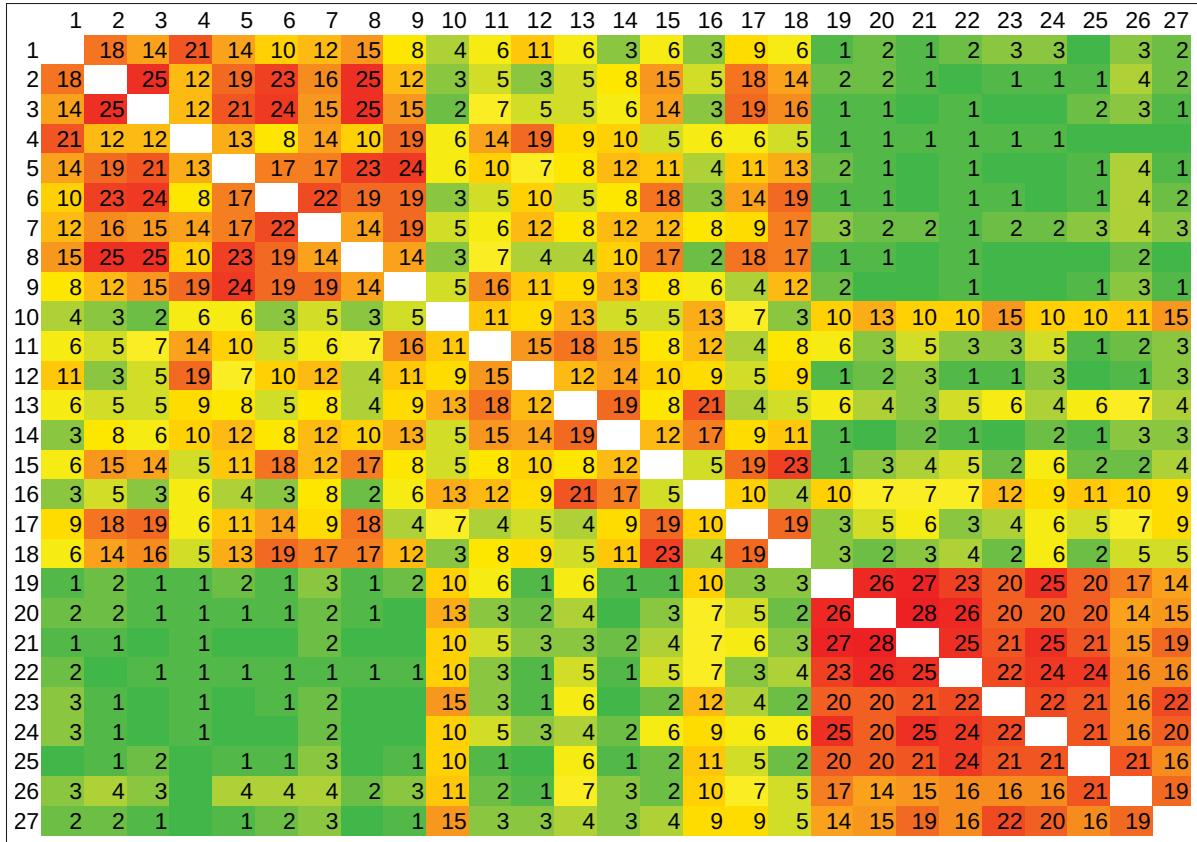


Figure 4.3 Matrice de cooccurrence des images dans un même groupe pour toutes les grammaires.

groupé avec l'image n°3 qu'une seule fois. Cette matrice est aussi produite individuellement pour chacune des grammaires.

Les gradients de couleur facilitent l'évaluation des taux de groupement. Plus la couleur tend vers le rouge, plus deux images ont tendance à être regroupées fréquemment. Réciproquement, la couleur verte indique que les images qui correspondent sont rarement regroupées ensemble.

D'après la matrice obtenue, les regroupements effectués par les participants suivent la logique de la décomposition de l'espace PCA. Trois catégories d'indices ressortent : les indices inférieurs (1 à 9), ceux du milieu (10 à 18) et les indices supérieurs (19 à 27). La figure 4.3 suggère que chacune des images d'une catégorie tend à être regroupée avec une image dans cette même catégorie. Toutefois, les zones du milieu dans la matrice de cooccurrence sont difficiles à inter-

préter de par la disparité des concentrations des regroupements. Ainsi, un rééchantillonnage a été effectué dans les sections ultérieures afin d'atténuer les disparités dans les zones du milieu de la matrice de cooccurrence.

#### 4.4.2 Graphiques sur les regroupements

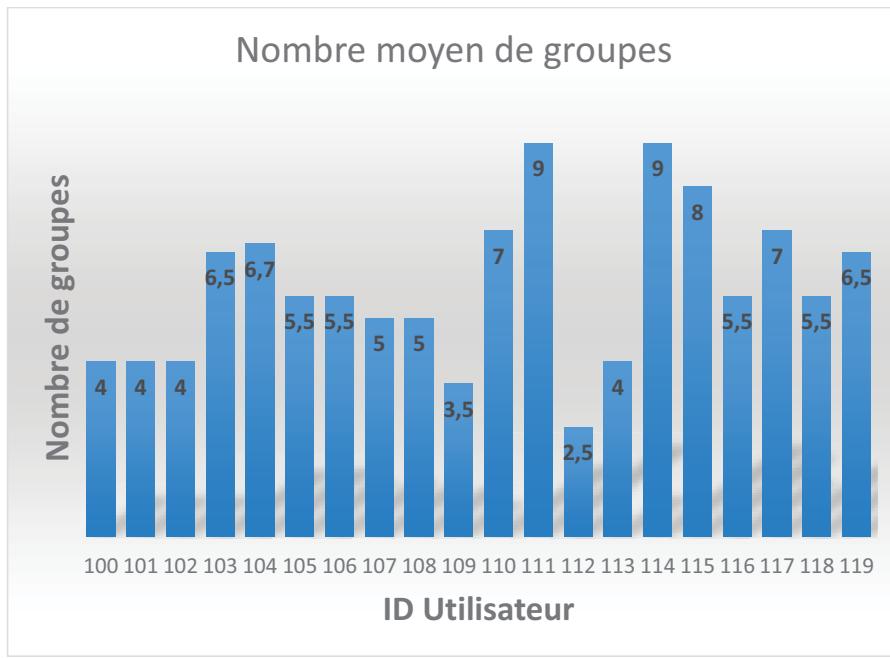
La figure 4.4 montre le nombre moyen de groupes et le nombre moyen d'images par groupe par chaque utilisateur.

La figure 4.5 répertorie les mêmes types d'informations que précédemment, mais cette fois classifiée selon les grammaires utilisées. On peut y lire par exemple que pour la grammaire *honda sympodial*, en moyenne les participants ont formé sept groupes et ont inclus dans chacun des groupes 4.3 images.

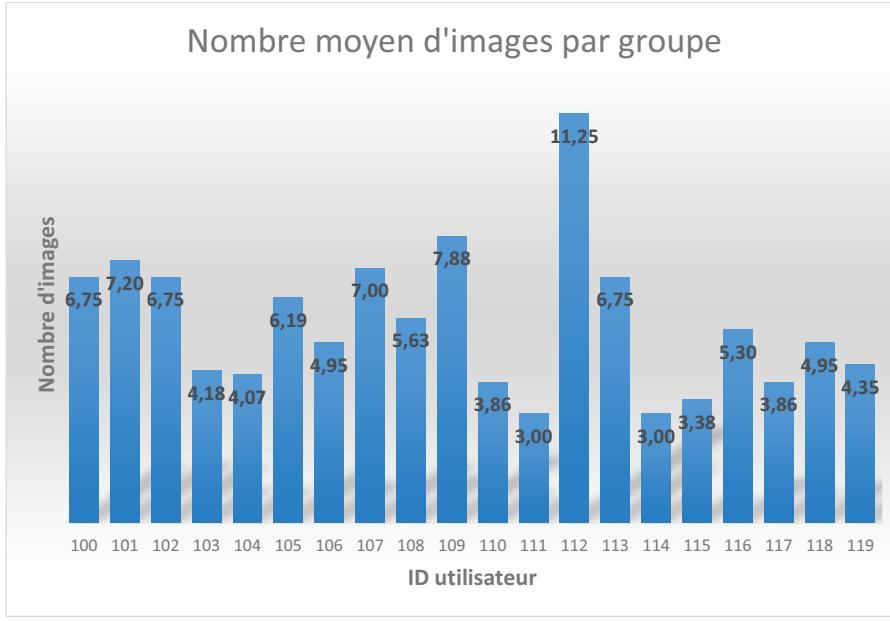
Ainsi, en moyenne, 5.20 groupes ont été formés, et chaque groupe contient 5.97 images. La boîte à moustaches de la figure 4.6 donne davantage d'information sur le regroupement effectué par les utilisateurs pour toutes les grammaires. Les valeurs aux extrémités caractérisent les extrêmes. Par exemple, la boîte de droite indique que le nombre maximal de groupes formés par un utilisateur est de neuf. Les valeurs indiquées par les traits en vert sont les médianes. Respectivement en dessus et en dessous de chacune des valeurs médianes sont les premiers et deuxièmes quartiles. Le quartile consiste en les trois valeurs qui déparentagent les données en quatre parties égales, de façon à ce qu'une partie corresponde au quart de l'échantillon de la population.

#### 4.4.3 Réduction des ambiguïtés dans les matrices de cooccurrence

À titre de rappel, la formule 3.2 donne le nombre initial d'échantillons de points pour une grammaire. Par exemple, on a vu que l'extrait 2.1 a généré  $N_{arbres} = 3^i \times N_{regles} = 3^5 \times 2 = 486$  points. Alors, si on a retenu 27 points après l'analyse en composantes principales, on constate qu'on a éliminé  $486 - 27 = 459$  points. L'approche initiale pour retrouver le point de grammaire le plus proche d'un point d'intérêt consistait à retrouver séquentiellement les points de



(a)



(b)

Figure 4.4 Information de groupement des images selon chaque participant.

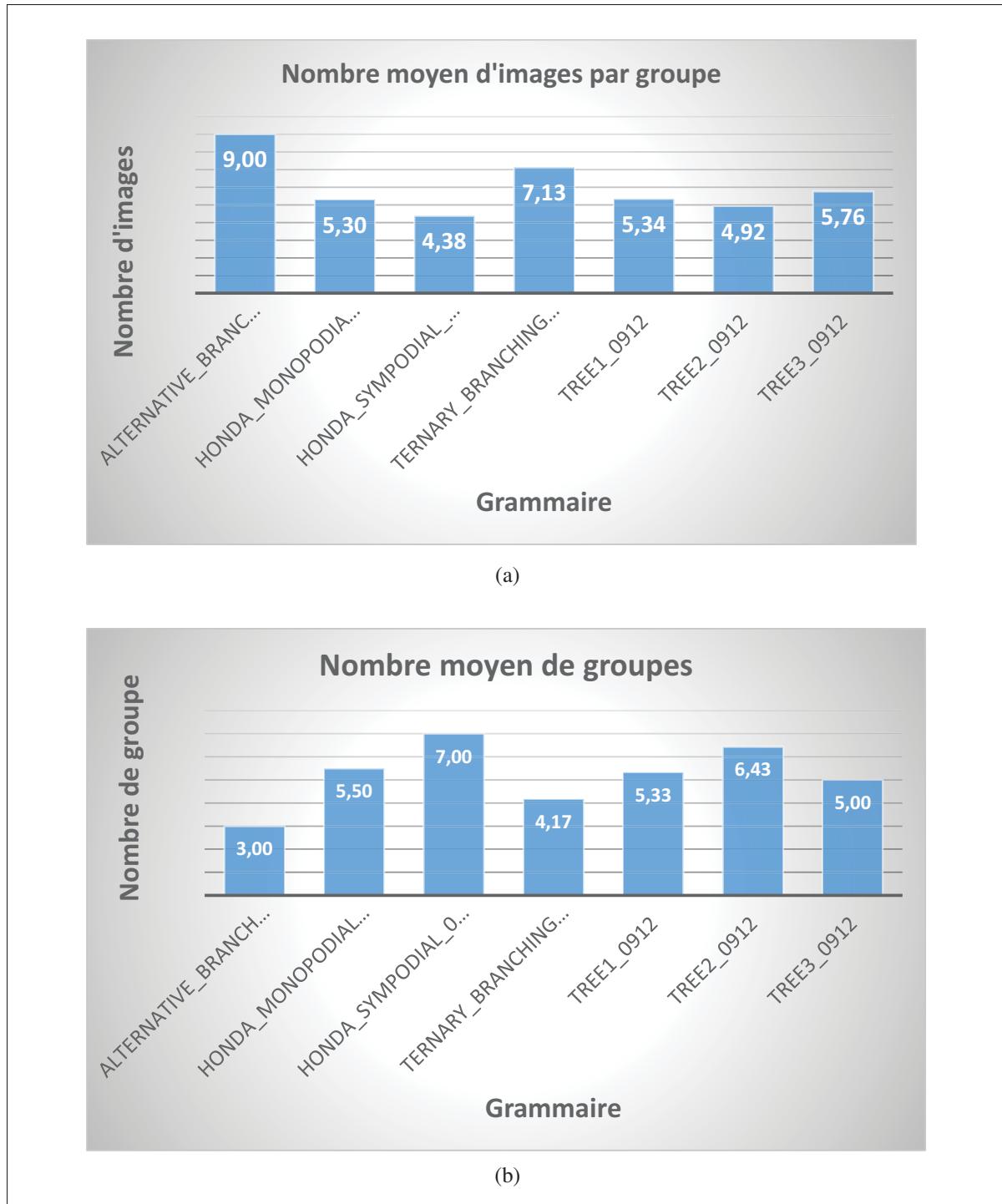


Figure 4.5 Information de groupement des images selon les grammaires.

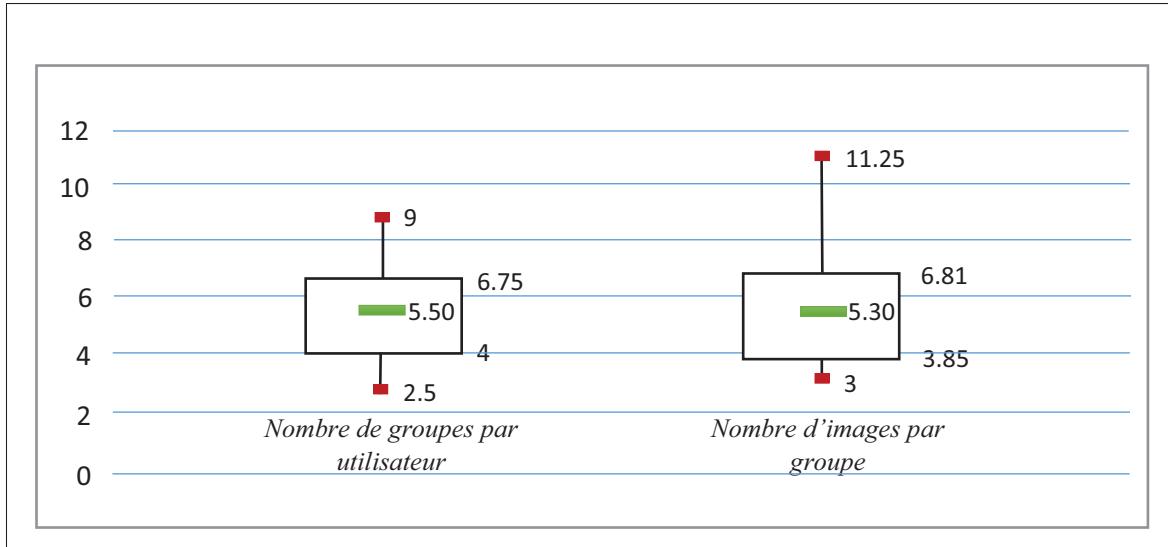


Figure 4.6 Aperçu du regroupement des images.

grammaire les plus proches, en commençant d'abord par le point d'intérêt avec l'identificateur n°1 (cf. figure 4.1). Ensuite, on retrouve le point le plus proche du point d'intérêt n°2 de la boîte, et il en va ainsi jusqu'au point d'intérêt n°27. Cette approche présente l'inconvénient de privilégier les points d'intérêt d'indice inférieur dans le cas où un même point de grammaire est aussi proche du point d'intérêt  $i$  que du point d'intérêt  $i + n$  ( $n > 0$ ). En effet, le point d'intérêt  $i + n$  ne pourra plus sélectionner le même point de grammaire que le point d'intérêt  $i$ . Le problème consiste donc à trouver une autre approche, de façon à ce que l'analyse de données, telles les matrices de regroupement des images, gagne en cohérence.

Plusieurs approches de mise en correspondance des points d'intérêt et des points de grammaire PCA ont été tentées. La mesure d'efficacité de ces approches consiste en un calcul de la somme des distances entre le point d'intérêt et le point de grammaire le plus proche calculé. La figure 4.7 est un exemple en 3D de mise en correspondance des points de grammaire pris parmi les grammaires soumises aux participants. Les points en rouge sont actuellement les points d'intérêt, en outre, ce sont les points dont on essaie de trouver les voisins les plus proches parmi les points de grammaire. Les points en vert sont les points de grammaire retrouvés, dont les images d'arbre 3D correspondantes ont été présentées aux participants. Dans cette figure, la méthode initiale qui privilégie les points d'indice inférieur produit une distance totale de

1 590.59. La nouvelle approche de mise en correspondance consiste à recalculer toutes les distances entre les 27 points d'intérêts de la boîte et les 27 points images, puis reformer les paires pour la correspondance en priorisant les distances minimales. Toujours dans la figure 4.7, les anciennes correspondances sont montrées en pointillés gris, tandis que le nouveau calcul est en pointillé orangé. La distance pour cette méthode est 1 539.53. En appliquant cette nouvelle méthode de correspondance, le gain en distance en moyenne, pour toutes les grammaires confondues, est de 3.12 %.

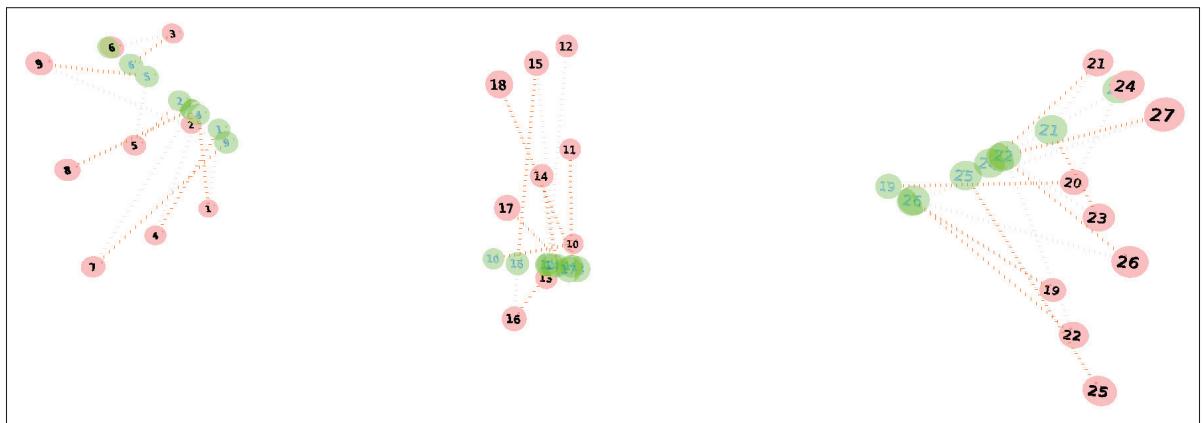


Figure 4.7 Correspondance des points d'intérêts englobante PCA avec les points de grammaire les plus proches.

Une nouvelle matrice de cooccurrence de groupement est produite en utilisant cette nouvelle approche de mise en correspondance des points images avec les points d'intérêt de la boîte. Cette matrice est reportée à la figure 4.8 avec des gradients de couleur. L'objectif est d'obtenir moins de disparité notamment dans les zones du milieu dans la matrice de cooccurrence. La réduction des disparités signifie plus de cohérence dans le groupement. Elle témoigne de la diminution de l'aspect aléatoire des regroupements effectués par les participants. Le cas idéal pour nous est d'avoir une zone de forte concentration des regroupements, intégralement séparés des zones de faible concentration. Dans la figure 4.8, une amélioration des zones de concentration au centre de la matrice est constatée. Elle indique que les points de grammaire se rapportant aux indices de milieu sont mieux regroupés dans cette nouvelle approche.

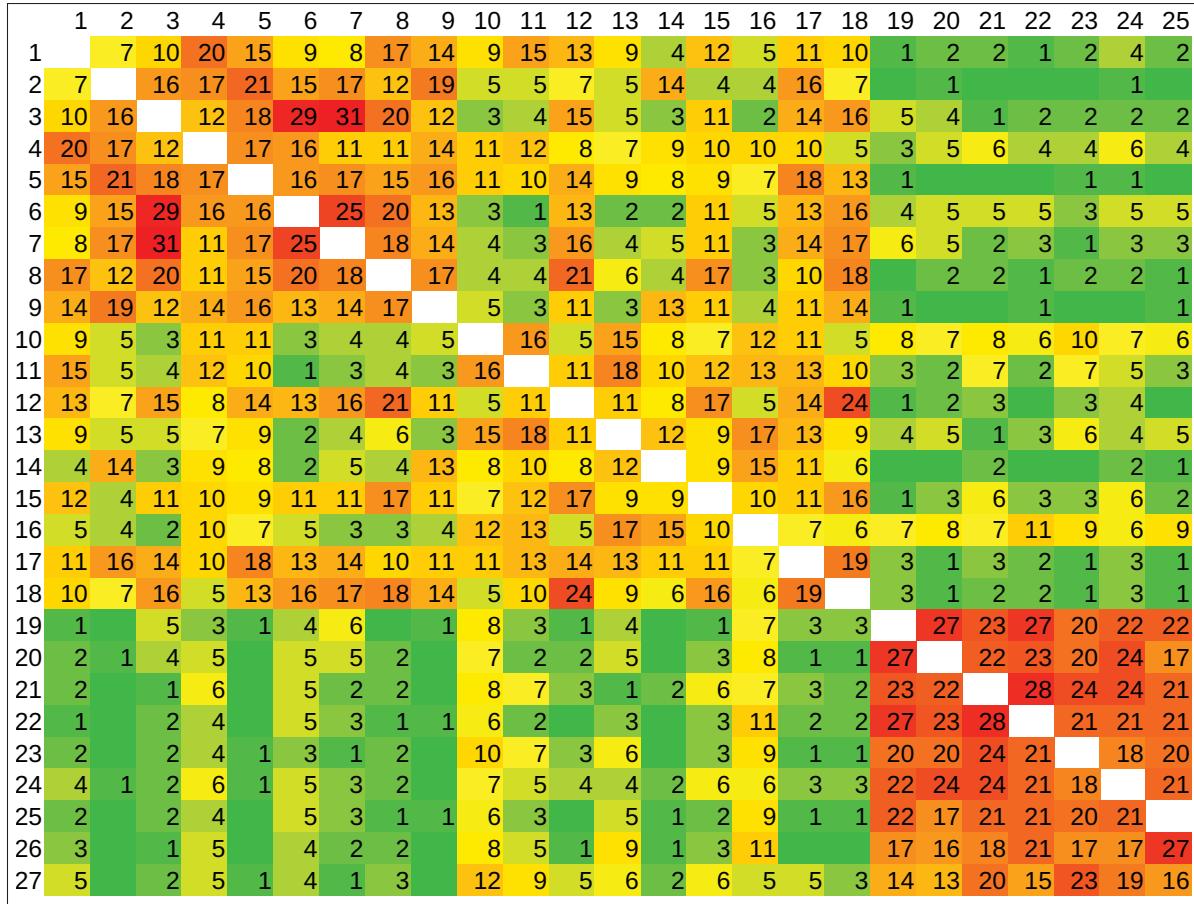


Figure 4.8 Matrice de cooccurrence des images en utilisant une nouvelle approche de correspondance.

D'autre part, les nouvelles mises en correspondances des points réels de l'espace PCA et des points d'intérêt de la boîte auraient pu être effectuées sur les  $N_{arbres}$  au lieu de seulement les 27 points extraits des  $N_{arbres}$ . De cette façon, les images d'arbre 3D proposées aux participants diffèreraient à un certain degré des images actuellement proposées, avec des résultats probablement plus intelligibles. Les 27 images évaluées par les participants ont été toutefois obtenues à partir des 27 points réels calculés avec la méthode initiale. C'est la raison pour laquelle les nouvelles mises en correspondance se sont appuyées sur les 27 points préalablement évalués par les participants. Nous avons considéré plusieurs approches de mise en correspondance, et celle donnant la distance totale la plus petite a été retenue. Cette dernière approche consiste à prioriser le rapprochement du point de grammaire avec le point d'intérêt dont la distance mi-

nimale est la plus petite, et itérativement, on va rechercher pour les couples de points restants la nouvelle distance minimale. Un gain de 2.96 % a été enregistré par rapport à la méthode originale.

## 4.5 Pertinence des images

Un des objectifs poursuivis lors de l'exécution des expérimentations est de pouvoir établir un lien concret entre un point de grammaire de l'espace PCA et la pertinence de l'image qui y correspond. Cette pertinence est une variable indépendante de type catégorique, représentée à l'échelle de Likert. À des fins d'analyse statistique, ces échelles sont traduites selon les conventions suivantes :

- La catégorie *Mauvaise* est attribuée la note 1/4 ou 0.25 ;
- La catégorie *Passable* est attribuée la note 2/4 ou 0.50 ;
- La catégorie *Bonne* est attribuée la note 3/4 ou 0.75 ;
- La catégorie *Excellente* est attribuée la note 4/4 ou 1.00.

Le tableau 4.3 résume pour chaque grammaire les notes reçues pour chacun des points de grammaire. Pour obtenir ce tableau, chaque observation du participant portant sur la pertinence d'un groupe est transmise à chacune des images à l'intérieur de ce groupe. Ainsi dans le tableau, la pertinence d'une image est la pertinence du groupe auquel elle appartient. À travers des tests statistiques, nous allons établir s'il y a une corrélation entre les indices des points images (et donc leur position dans l'espace PCA), et la pertinence attribuée par les participants. Les mêmes tests sont aussi effectués pour savoir si le type de grammaire a une influence sur la pertinence des images.

### 4.5.1 Choix de l'outil pour les tests statistiques

L'analyse de variance à un facteur, ou ANOVA (pour *Analysis of Variance*), est utilisée pour déterminer s'il y a une différence statistique significative entre les moyennes de plusieurs variables indépendantes. Cet outil de test s'applique sur une population à distribution normale.

Tableau 4.3 Pertinence de chaque image des grammaires *Alternative Branch(AB)*, *Ternary Branching (TB)*, *Honda Monopodial (HM)*, *Honda Sympodial (HS)*, *Tree1 (T1)*, *Tree2 (T2)*, *Tree3 (T3)*, selon les participants.

	AB	TB	HM	HS	T1	T2	T3
1	2.00	1.83	3.00	3.00	2.67	2.79	3.00
2	2.00	2.67	3.00	2.67	2.67	2.79	3.43
3	2.00	2.67	3.00	2.67	2.67	2.71	3.43
4	2.00	1.83	3.00	3.00	2.67	2.93	2.71
5	2.00	2.67	3.00	2.67	3.00	2.86	2.71
6	2.00	2.50	2.50	2.67	2.67	2.79	3.43
7	2.00	3.00	3.00	2.67	2.67	2.93	2.71
8	2.00	2.67	3.00	2.00	3.00	2.86	3.43
9	2.00	2.67	3.00	2.67	2.67	2.64	2.57
10	2.00	3.17	2.50	3.00	2.67	3.14	2.86
11	1.50	2.67	3.00	3.00	2.67	2.79	3.00
12	1.50	1.83	3.00	2.67	3.00	2.86	3.00
13	2.00	2.67	3.00	3.00	3.00	3.07	2.86
14	1.50	2.50	3.00	2.33	3.00	2.93	2.71
15	1.50	3.00	3.00	2.67	3.00	2.93	3.43
16	2.00	3.17	3.00	2.67	2.00	2.93	3.14
17	1.50	3.17	3.00	2.33	2.00	2.64	3.43
18	1.50	2.83	3.00	2.33	2.67	2.93	3.43
19	2.00	2.83	3.50	3.00	3.00	3.43	3.29
20	2.00	3.17	3.50	3.00	3.00	3.43	3.00
21	1.50	3.17	3.50	3.00	3.00	3.43	3.43
22	2.00	3.00	3.50	3.00	3.00	3.43	3.43
23	2.00	3.17	2.50	3.00	3.00	3.21	3.29
24	1.50	3.17	3.50	3.00	3.00	3.57	3.14
25	2.00	3.17	3.50	3.00	3.00	3.29	3.29
26	2.00	3.00	3.50	3.00	3.00	3.29	3.14
27	1.50	3.17	3.50	3.00	3.00	3.21	3.43

L'hypothèse nulle dans les tests ANOVA est que les ensembles d'observation proviennent d'une même distribution. Si cette hypothèse n'est pas vérifiée, l'hypothèse alternative est constatée. L'hypothèse alternative consiste en des distributions différentes. La règle de décision est telle que l'on rejette l'hypothèse nulle si le test statistique (F) est plus grand que la valeur critique de F ( $F_{crit}$ ).

#### 4.5.2 Analyse statistique d'interdépendance des grammaires

En suivant le principe exposé dans la section précédente, on cherche à établir si la pertinence des images décèle une corrélation forte avec les grammaires. Pour cela, il est posé comme hypothèse nulle que les moyennes de pertinence des images pour toutes les grammaires sont égales. Le tableau 4.4 récapitule l'analyse de variance effectuée sur les échantillons collectés dans le tableau 4.3. D'après les calculs,  $F = 58.20$  et  $F_{crit} = 2.149$ , donc  $F > F_{crit}$ . Par conséquent, l'hypothèse nulle est rejetée. Il y a eu une différence statistique significative entre les grammaires comme établi par l'analyse de variance à un facteur. Au moins, une grammaire vient d'une distribution différente des autres grammaires. La pertinence des images est de ce fait en corrélation avec le type de grammaire.

Tableau 4.4 Test ANOVA sur la corrélation entre les grammaires et la pertinence des images.

$F$	$F_{crit}$
58.20	2.149

#### 4.5.3 Analyse statistique d'interdépendance des points de grammaire

Une analyse de variance sur l'interdépendance des points de grammaire détermine si un point de grammaire en particulier se distingue en termes de pertinence. Le même tableau 4.3 est repris en procédant avec les lignes. Le principe est toujours similaire au précédent, et le sommaire des calculs est donné dans le tableau 4.5. Comme hypothèse nulle, on pose que chaque point image possède une même distribution. Cette fois, on a  $F = 0.730$  et  $F_{crit} = 1.564$ . Ainsi,  $F < F_{crit}$ , l'hypothèse nulle est donc vérifiée. Il n'existe donc pas de différence statistique significative sur la pertinence des points de grammaire. On peut en déduire que tous les points de grammaire de l'espace PCA sont semblables en terme de pertinence.

Tableau 4.5 Test ANOVA sur la corrélation entre les points images et la pertinence des images.

$F$	$F_{crit}$
0.730	1.564

#### 4.5.4 Analyse statistique des différences entre les points de grammaire

Toujours avec l'idée de procéder au même type d'analyse, l'ensemble de données basé sur l'appréciation des participants concernant les différences de chaque point de grammaire est dressé dans le tableau 4.6. La variable de différence pour un point de grammaire est, dans le questionnaire proposé aux participants, le niveau de différence attribué par le participant au groupe dans lequel figure l'image concernée. Ainsi, si un groupe est par exemple jugé « *excellent* » en termes de différence par rapport aux autres groupes, alors chaque point de grammaire de ce groupe hérite de cette observation. Pour chaque grammaire, une moyenne de différence est alors calculée suivant ce principe pour chaque image.

Dans le premier test, l'hypothèse nulle est que toutes les grammaires possèdent une moyenne de différence égale. Le tableau 4.7 résume les calculs sur l'analyse de variance. On a  $F = 13.361$  et  $F_{crit} = 2.149$ , donc  $F > F_{crit}$ . L'hypothèse nulle est rejetée. On peut en conclure qu'au moins une des grammaires est contrastée en critère de différence par rapport aux grammaires restantes.

Parallèlement, à l'aide du tableau 4.6, on voudrait aussi savoir à l'aide d'une analyse de variance si parmi les points de grammaire, un élément en particulier se distingue des points de grammaire restants. On pose comme hypothèse nulle que les différences pour tous les points de grammaire sont égales. Le tableau 4.8 synthétise le résultat des calculs. D'après les calculs,  $F = 2.397$  et  $F_{crit} = 1.564$ , donc  $F > F_{crit}$ . Nous rejetons encore l'hypothèse nulle. Au moins un point de grammaire possède une moyenne de différence distincte des autres points de grammaire.

Tableau 4.6 Les différences entre les groupes projeté à chaque image pour chacune des grammaires *Alternative Branch(AB)*, *Ternary Branching (TB)*, *Honda Monopodial (HM)*, *Honda Sympodial (HS)*, *Tree1 (T1)*, *Tree2 (T2)*, *Tree3 (T3)*.

	<i>AB</i>	<i>TB</i>	<i>HM</i>	<i>HS</i>	<i>T1</i>	<i>T2</i>	<i>T3</i>
1	3.500	3.167	2.500	3.667	2.333	2.286	2.571
2	3.500	1.833	2.500	2.000	2.333	2.214	3.286
3	3.500	1.833	2.500	2.000	2.333	2.429	3.286
4	3.500	3.167	2.500	3.667	2.333	2.357	2.429
5	3.500	1.833	2.500	2.000	2.000	2.357	2.571
6	3.500	1.833	2.000	2.000	2.333	2.214	3.286
7	3.500	2.167	2.500	2.333	2.333	2.000	2.429
8	3.500	1.833	2.500	2.333	2.000	2.214	3.000
9	3.500	1.833	2.500	2.000	2.333	2.214	2.571
10	4.000	2.333	2.000	3.667	2.333	3.286	2.286
11	3.000	1.833	2.500	3.667	2.667	2.714	2.286
12	3.000	3.167	2.500	2.667	2.333	2.286	2.429
13	4.000	1.667	2.500	3.000	2.333	2.714	2.429
14	3.000	1.833	2.500	2.333	2.333	2.143	2.429
15	3.000	2.333	2.500	2.333	2.333	2.429	3.000
16	4.000	2.000	2.500	2.667	2.667	2.643	2.429
17	3.000	2.000	2.500	2.333	2.667	2.429	3.000
18	3.000	2.500	2.500	2.333	2.333	2.143	3.000
19	4.000	2.500	3.000	3.667	4.000	3.786	3.143
20	4.000	2.333	3.500	3.667	4.000	3.786	2.714
21	3.000	2.333	3.000	3.667	4.000	3.786	3.143
22	4.000	2.333	3.500	2.667	3.667	3.786	3.143
23	4.000	2.333	2.000	2.667	3.667	3.643	2.857
24	3.000	2.333	3.000	3.333	3.667	3.786	2.857
25	4.000	2.333	3.500	2.333	3.667	3.571	3.143
26	4.000	2.167	3.000	2.667	3.667	3.500	2.571
27	3.000	2.000	3.500	2.667	3.667	3.714	2.714

Tableau 4.7 Test ANOVA sur la corrélation entre les grammaires et la différence des images.

<i>F</i>	<i>F crit</i>
13.361	2.149

Tableau 4.8 Test ANOVA sur la corrélation entre les points de grammaire et la différence des images.

<i>F</i>	<i>F crit</i>
2.397	1.564

#### 4.5.5 Interprétation des tests

Notons dès le départ que pour que l'approche choisie pour générer des variétés d'arbres soit jugée comme robuste, il faudrait d'emblée que toutes les grammaires se comportent d'une manière semblable. C'est-à-dire que toutes les grammaires seraient identiques aussi bien en terme de pertinence des images que de différence entre les groupes d'image. Le résultat de test le plus souhaitable serait donc d'avoir une distribution égale de ces deux critères. De cette façon, nous aurions une approche qui serait adaptée à tout type de grammaire. Autrement dit, la génération de modèles d'arbre 3D par une analyse en composantes principales sur les métriques pourrait effectivement générer des objets 3D pertinents et sensiblement variés.

Les tests ANOVA à un facteur ont fait ressortir qu'une ou plusieurs grammaires présentent une distribution de pertinence différente des grammaires restantes. Ce résultat peut se traduire par une certaine inconstance des modèles d'arbre 3D générés, dépendamment des grammaires entrées par l'utilisateur. On a constaté aussi d'après les tests de différence entre groupes qu'au moins une des grammaires peut produire des ensembles d'images qui se distinguent dans les critères de différence, par rapport à l'ensemble des grammaires. Ce résultat indique qu'il faudrait s'attendre à générer, pour certaines grammaires, des modèles récurrents d'arbre 3D avec moins de variétés par rapport à d'autres grammaires.

Dans un second temps, le choix des points d'intérêt dans l'espace PCA doit aussi être validé par des résultats statistiques qui pourraient supporter ces choix. Pour que le choix de ces 27 points soit considéré comme convenable, il faudrait qu'il y ait des images (parmi les 27) qui possèdent une moyenne de différence qui se distingue des images restantes. Si de telles images existent, les points d'intérêt correspondant à ces images peuvent faire l'objet d'une étude plus poussée, par exemple en analysant les qualificatifs qui sont attribués par les participants à ces points.

Nous avons pu valider dans les tests précédents que les pertinences des images possèdent une distribution égale. Autrement dit, chacune des 27 images, et donc des 27 points de grammaire, peut produire des arbres pertinents. Ce résultat supporte la méthodologie adoptée pour le choix

des points d'intérêt. D'autre part, les tests ont également indiqué qu'au moins une image relative à un point de grammaire possède une moyenne de différence distincte par rapport aux autres points de grammaire. Cette observation est également positive quant aux choix des méthodes de génération de modèles d'arbre 3D.

## 4.6 Traitements sur les informations d'attribut

L'objectif de cette étape est de pouvoir extraire des informations utiles et fiables pour ultérieurement élaborer un système capable de réagir à des directives subjectives. Dans cette section, nous exposons les démarches pour éliminer le maximum d'informations inconsistantes et retenir les informations les plus utiles. Le traitement sur les informations d'attribut a été accompli grâce à la contribution de Clément-Corriveau (2016), un étudiant en maîtrise du laboratoire multimédia. Son travail avait permis d'éliminer des ensembles d'attributs qui s'avéraient être inexploitables pour l'objectif de régénération que nous poursuivons. L'auteur de ce mémoire, quant à lui, avait concentré son effort sur le traitement des qualificatifs retenus, notamment en effectuant plusieurs analyses graphiques de ces qualificatifs comme nous allons le voir dans les sections qui vont suivre.

### 4.6.1 Collecte des qualificatifs

La dernière partie du questionnaire fourni aux participants leur demande de produire un qualificatif pour chaque groupe. Le qualificatif d'un groupe est un mot ou un ensemble de mots qui peut décrire les caractéristiques du groupe. Un qualificatif sous forme d'ensemble de mots est composé d'une phrase nominale, comme dans « *branches obliques et longues* », ou parfois avec un superlatif comme « *angles plus fermés* ». Dans notre étude, la notion de mot n'est pas prise en compte c'est-à-dire qu'un qualificatif comportant par exemple trois mots sera uniquement traité comme un seul qualificatif.

Chaque participant ayant créé en moyenne 5.20 groupes par grammaire. Plusieurs participants, dépendamment de la nature des arbres qui leur sont présentés, ne se sont pas limités à un seul

qualificatif pour chaque groupe. Un participant a ainsi utilisé en moyenne 2.60 qualificatifs par groupe. Le nombre total d'observations manifestées est de 271, réparties en deux langues (le français et l'anglais). En regroupant les observations redondantes, on arrive à 185 qualificatifs uniques. Statistiquement, chaque utilisateur a rapporté en moyenne 38.71 qualificatifs par grammaire. L'élimination des qualificatifs est un travail ardu qui repose sur les travaux de Clément-Corriveau (2016). Partant initialement à 185 qualificatifs, Clément-Corriveau a pu réduire ce nombre à 10 qualificatifs en utilisant les méthodes suivantes :

- Seuls sont retenus les qualificatifs dont la pertinence des images est soit « *Bonne* » soit « *Excellente* » ;
- Les qualificatifs dont les occurrences ont été élevées dans les observations des participants sont prioritaires ;
- La langue est uniformisée grâce à des consultations auprès de plusieurs étudiants du laboratoire sur la traduction des termes anglais en français ;
- Des recherches de synonymes sont effectuées au sein de l'ensemble de tous les qualificatifs avec l'aide de plusieurs étudiants du laboratoire, alors les données statistiques et les informations d'image des qualificatifs synonymes sont regroupées.

Les qualificatifs qui sont retenus sont ensuite intégrés dans l'interface de l'application (figure 4.9) dans le but de regénérer des parties de l'arbre en considérant les contraintes imposées par le qualificatif choisi par l'utilisateur.

#### 4.6.2 Visualisation graphique des informations

Les qualificatifs donnés par les participants à un groupe sont donnés à toutes les images membres de ce groupe, donc par extension à leur point de grammaire dans cet espace. Pour la visualisation, on retient uniquement les points de grammaire dont les qualificatifs correspondants figurent dans la liste des 10 qualificatifs qui ont été retenus. Ensuite, une boîte englobante est calculée avec la perspective de pouvoir associer un qualificatif à une zone délimitée dans l'espace PCA. Si la collision entre les zones, c'est-à-dire les boîtes englobantes des qualificatifs, est relativement élevée, il sera difficile de discerner les qualificatifs. La figure 4.10

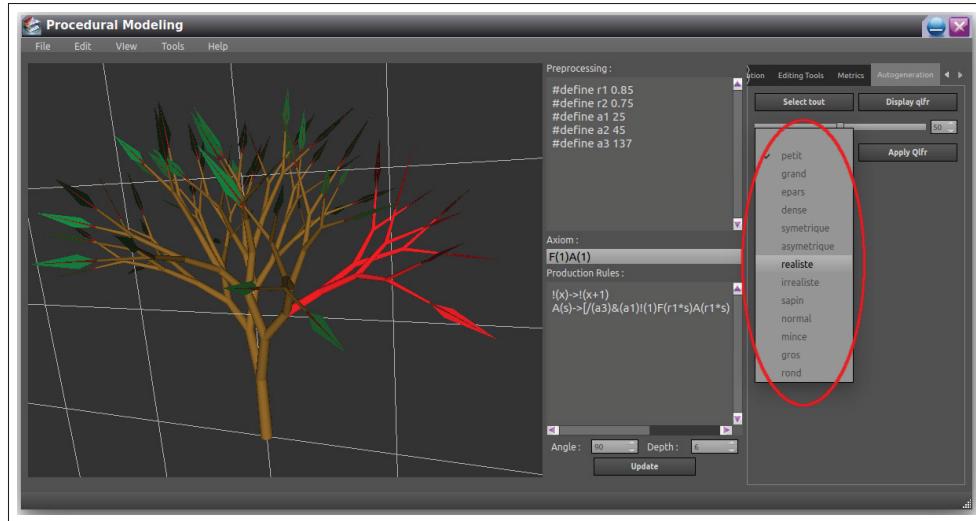


Figure 4.9 Ajout de qualificatifs sur l’interface utilisateur, ici encerclé en rouge.

montre par exemple la collision entre les boîtes englobantes des qualificatifs « *petit* », « *grand* », « *réaliste* » et « *rond* ». Dans cette figure, si la case relative au qualificatif est cochée, sa boîte englobante est alors dessinée. On peut alors constater que dans cette figure, la boîte englobante pour les deux qualificatifs « *petit* » et « *grand* » est assez large et peut facilement entrer en collision avec d’autres qualificatifs.

#### 4.6.3 Collisions des qualificatifs dans l'espace PCA

Pour deux qualificatifs donnés, chaque point du premier faisant partie de la boîte englobante du second qualificatif est compté comme une collision. Le pourcentage de collision entre deux qualificatifs est donné par le rapport entre le nombre de points du premier à l’intérieur de la boîte du second qualificatif par le nombre total de points du premier qualificatif. Le résultat sous forme d’une matrice est présenté à la figure 4.11 avec des surbrillances en rouge si la collision est élevée et en vert si elle est moindre. En procédant ainsi, nous avons pu calculer qu’il y a une moyenne de collision de 66.04 %.

La boîte peut être encore affinée en enlevant un pourcentage de points (7 %) situés aux extrémités de la boîte, l’objectif étant toujours d’atténuer les collisions entre les boîtes englobantes. Cette méthode a permis de réduire la moyenne de collision de 66.04 % à 60.83 %.

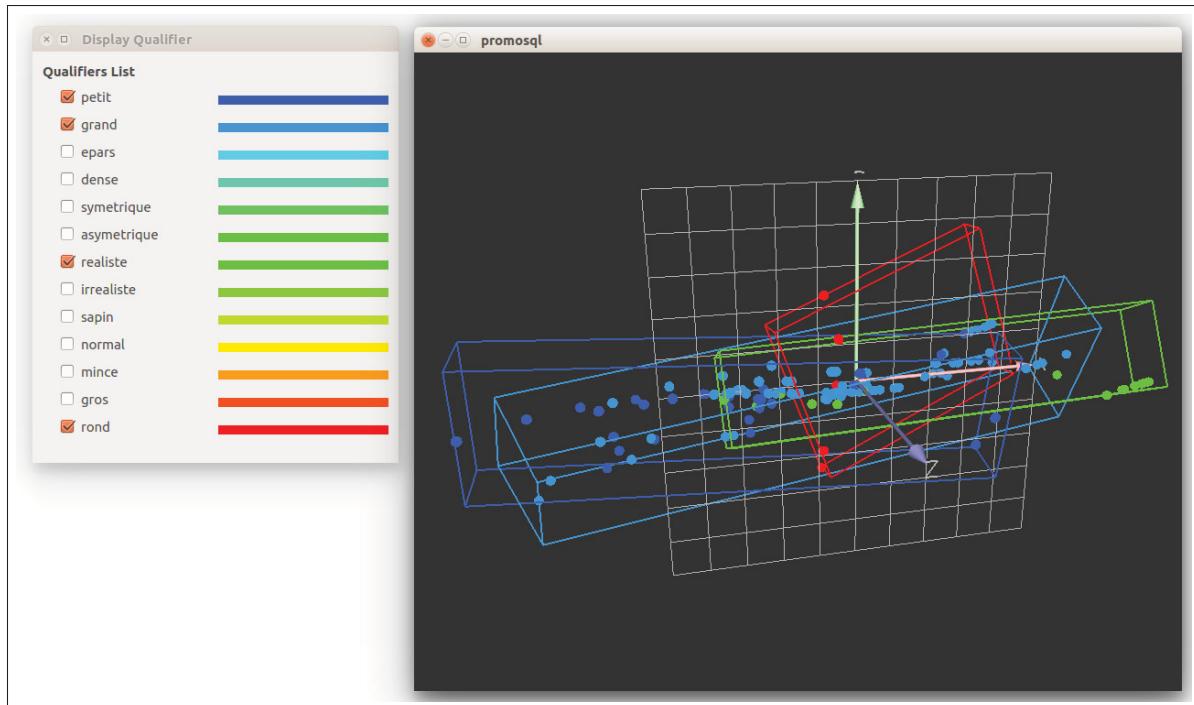


Figure 4.10 Outil de visualisation 3D des boîtes englobantes pour chaque qualificatif.

À partir de ce stade, pour regénérer un arbre 3D en tout ou en partie, le centre de la boîte englobante est utilisé comme référence pour désigner quelle combinaison de métriques utiliser lorsque l'utilisateur requiert une régénération.

#### 4.7 Rendus des arbres regénérés à l'aide des qualificatifs

Dans la figure 4.12, nous choisissons de régénérer les branches identifiées en surbrillance rouge. Les arbres à gauche (4.12c et 4.12e) sont les régénéérations partielles de l'arbre *Honda Monopodial* (4.12a) tandis que ceux à droite (4.12d et 4.12f) sont les régénéérations de *Honda Sympodial* (4.12b).

Qualifiers	petit	grand	epars	dense	symetrique	asymetrique	realiste	irrealiste	sapin	normal	mince	gros	rond	
petit	78.57 %	58.82 %	81.18 %	81.63 %	83.33 %	78.87 %	96.30 %	92.31 %	71.43 %	97.92 %	11.11 %	52.94 %		
grand	60.47 %	94.12 %	71.76 %	89.80 %	93.33 %	90.14 %	70.37 %	100.00 %	78.57 %	83.33 %	50.00 %	82.35 %		
epars	34.88 %	31.25 %		21.18 %	30.61 %	46.67 %	30.99 %	22.22 %	23.08 %	0.00 %	16.67 %	50.00 %	29.41 %	
dense	81.40 %	90.18 %	88.24 %		93.88 %	100.00 %	87.32 %	96.30 %	92.31 %	71.43 %	97.92 %	38.89 %	94.12 %	
symetrique	74.42 %	96.43 %	94.12 %	82.35 %		96.67 %	91.55 %	96.30 %	100.00 %	64.29 %	97.92 %	50.00 %	76.47 %	
asymetrique	51.16 %	83.04 %	70.59 %	68.24 %	77.55 %		83.10 %	66.67 %	88.46 %	57.14 %	79.17 %	33.33 %	64.71 %	
realiste	46.51 %	83.04 %	88.24 %	54.12 %	67.35 %		80.00 %	51.85 %	73.08 %	78.57 %	60.42 %	72.22 %	76.47 %	
irrealiste	72.09 %	75.89 %	52.94 %	77.65 %	79.59 %	56.67 %	73.24 %		80.77 %	57.14 %	97.92 %	22.22 %	64.71 %	
sapin	30.23 %	51.79 %	29.41 %	27.06 %	59.18 %	46.67 %	15.49 %	51.85 %		50.00 %	60.42 %	16.67 %	11.76 %	
normal	67.44 %	78.57 %	76.47 %	65.88 %	67.35 %		73.33 %	85.92 %	62.96 %	73.08 %		68.75 %	33.33 %	70.59 %
mince	48.84 %	48.21 %	29.41 %	62.35 %	59.18 %		40.00 %	29.58 %	85.19 %	38.46 %	28.57 %		0.00 %	17.65 %
gros	4.65 %	29.46 %	52.94 %	7.06 %	18.37 %		40.00 %	29.58 %	7.41 %	19.23 %	28.57 %	0.00 %		23.53 %
rond	20.93 %	48.21 %	52.94 %	51.76 %	28.57 %		20.00 %	61.97 %	22.22 %	30.77 %	21.43 %	27.08 %	44.44 %	

Figure 4.11 Collisions entre les boîtes englobantes des qualificatifs.

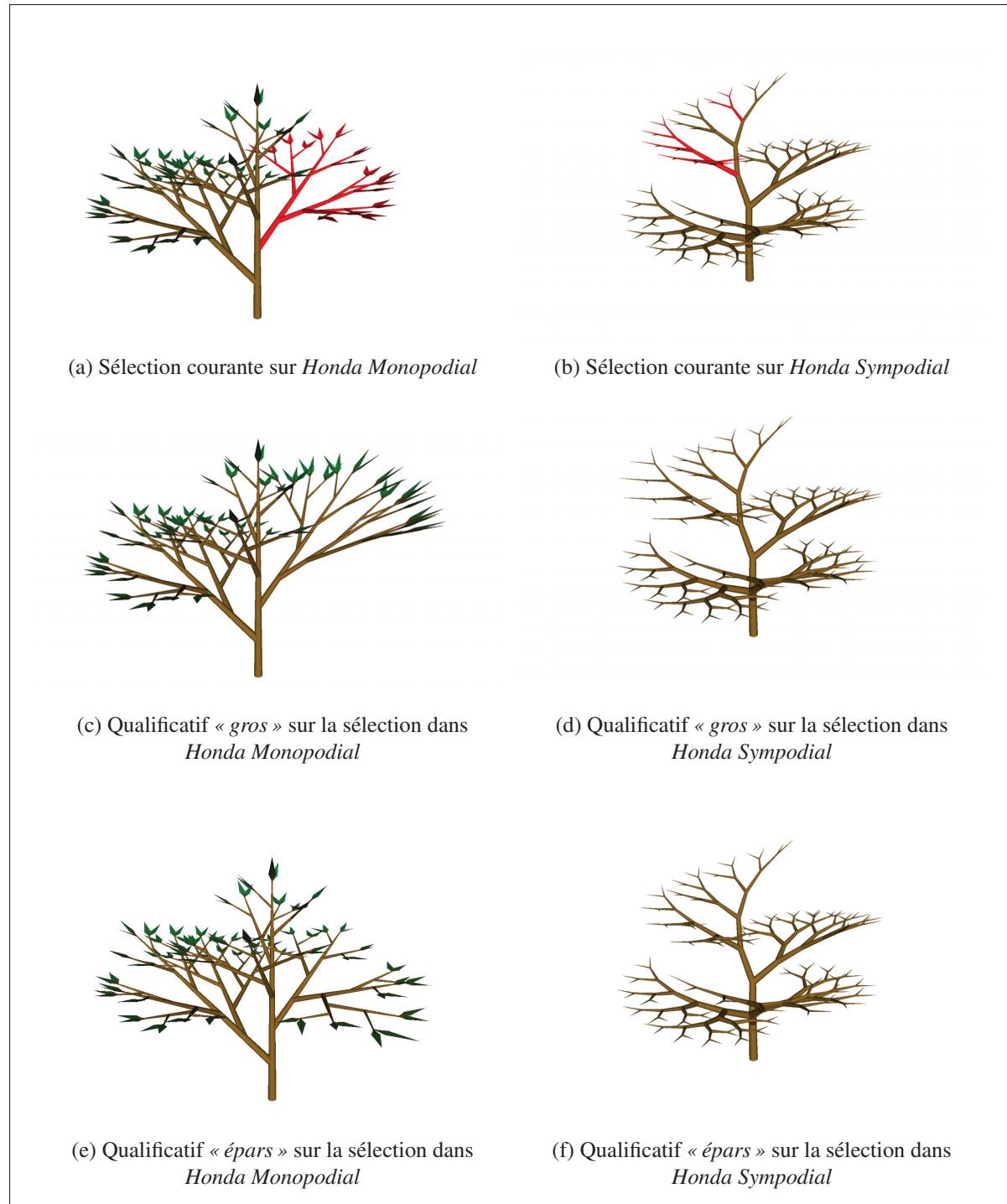


Figure 4.12 Regénération de la sélection courante des deux grammaires *Honda Monopodial* et *Honda Sympodial* en utilisant les qualificatifs.

## CHAPITRE 5

### SYNTHÈSE DES RÉSULTATS

Les approches que nous avons suggérées ont été implémentées dans un logiciel. Le logiciel comprend les fonctionnalités essentielles pour effectuer des regénérations d'arbre en 3D. Il prend en entrée une grammaire L-système, et peut afficher plusieurs sorties incluant des rendus en 3D de l'arbre, et d'autres informations telles que l'allure du graphe orienté ou encore la chaîne courante.

#### 5.1 Implémentation du système

Le système est développé dans le langage C++, avec les versions C++98 et C++11. La plateforme d'exécution se trouve sur un processeur Intel® Core™ i7-4500 CPU à 1.80 GHz. L'ordinateur est équipé d'une mémoire de 8 Gb avec une carte graphique Intel®HD Graphics 4400.

##### 5.1.1 Technologies utilisées

Le logiciel de modélisation procédurale a été développé à l'aide de Qt SDK et peut fonctionner sous Microsoft Windows® et Linux. Le code source est réparti en 215 fichiers avec 31 157 lignes de codes comme rapportés dans le tableau 5.1.

Tableau 5.1 Décompte des fichiers et lignes de code source du projet.

Langages	Fichiers	Blancs	Commentaires	Codes
C++	90	3 020	1 959	23 397
Entêtes C/C++	112	1 204	927	5 320
Qt	8	0	0	2 042
Projet Qt	2	20	5	329
CSS	1	7	0	49
Bourne Shell	1	0	5	19
Fichier Ressource Windows	1	0	0	1
Total	215	4 251	2 896	31 157

La librairie *OpenGL* en version 2.1 a été utilisée pour prendre en charge toutes les fonctions de dessin 3D. Les fenêtres de vue et l’interfaçage avec les périphériques d’entrée et sortie ont été plus commodes avec l’utilisation de la librairie *libQGLViewer v2.3.6*. Les librairies suivantes ont également été utilisées :

- *Boost v1.55.0* pour les conteneurs associatifs et les pointeurs intelligents ;
- *CGLA* pour les routines d’algèbre linéaire ;
- *maxflow-v3.01* pour calculer la coupe minimum d’un graphe.

Le déroulement de la coupe minimum du graphe, vu dans le chapitre 3 sur la regénération, commence par la sélection utilisateur initiale et aboutit à la regénération de sous-partie de branches. Un système de visualisation du déroulement de la coupe a été intégré au logiciel, en utilisant la méthode de Clément-Corriveau (2016), dans le but de valider graphiquement les stratégies de regénération choisies. *ShowGraph* est l’outil choisi pour visualiser les graphes orientés. Il prend en entrée des fichiers XML décrivant la requête de génération de graphe orienté. L’aperçu d’un graphe orienté ouvert dans *ShowGraph* est donné à la figure 5.1. Sur fond de couleur bleue on a les nœuds qui appartiennent à la source, et en rouge les nœuds qui appartiennent au drain. On déduit de ces jeux de couleur la coupe minimum retenue par l’algorithme.

Le traitement des informations reçues des participants est fait conjointement par un utilitaire spécialement développé à des fins d’analyse, ainsi qu’avec des tableurs dont Microsoft Excel et LibreOffice Calc. Le stockage des données initiales a été fait en utilisant SQLite.

### 5.1.2 Temps d’exécution

Le tableau 5.2 rapporte les coûts des sept grammaires utilisées, en mettant en avant le nombre de caractères dans la chaîne finale et le nombre de dérivations effectuées dans la grammaire. Nous pouvons aussi voir que le temps d’exécution et la consommation en mémoire sont du même ordre de grandeur pour toutes les grammaires.

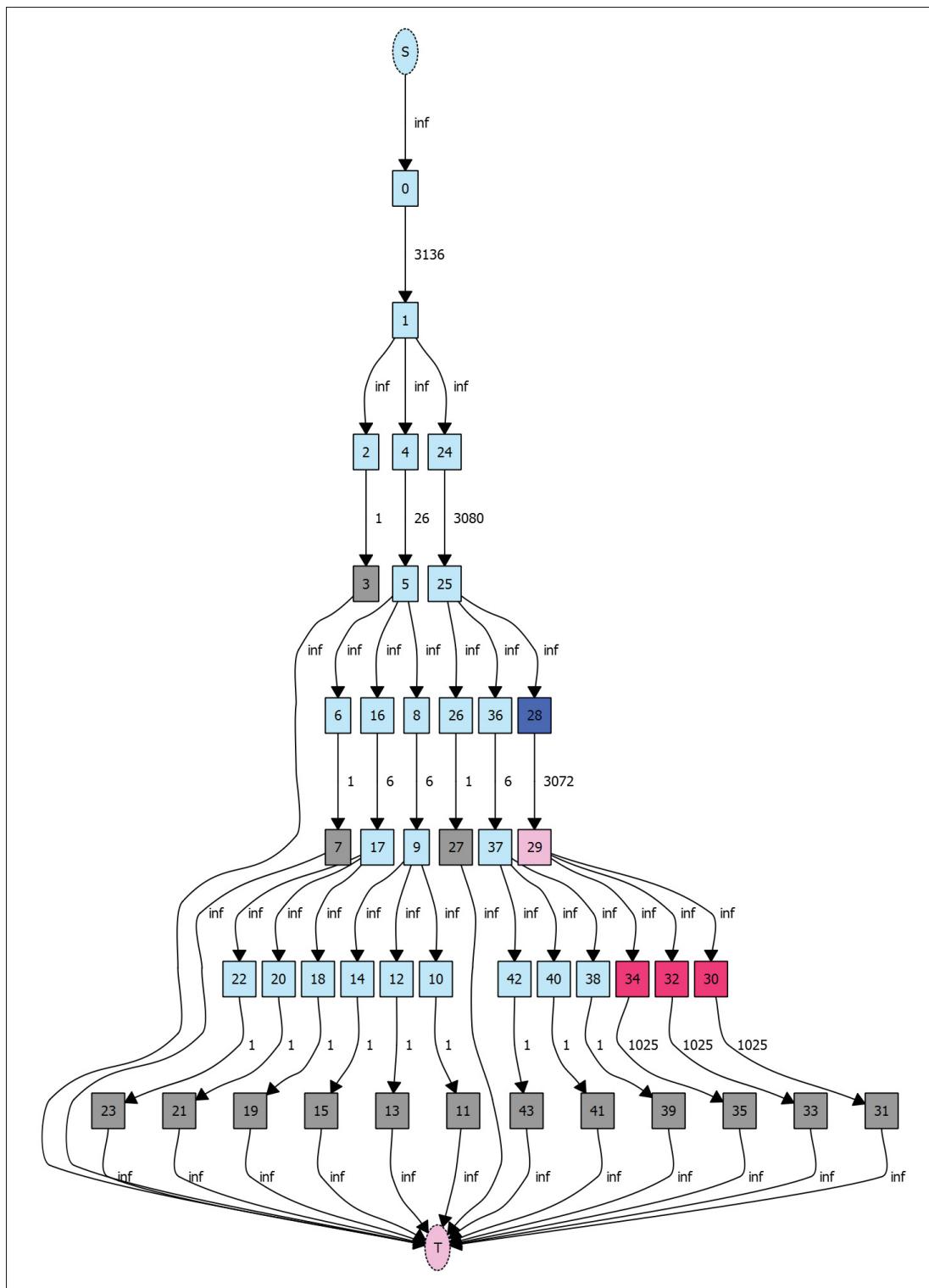


Figure 5.1 Visualisation du déroulement de la coupe minimum dans *ShowGraph*.

Tableau 5.2 Résumé des coûts des sept grammaires utilisées en temps d'exécution et en consommation mémoire.

Grammaires	Nombre de règles	Nombre de dérivations	Longueur de la chaîne (nombre de caractères)	Temps d'exécution (en sec)	Consommation mémoire (en Mo)
Alternative Branch.	5	5	16 806	1.91	42.4
Ternary Branch.	3	5	10 576	2.97	39.1
Honda Monopodial	3	7	6 160	1.55	38.0
Honda Sympodial	2	7	7 124	2.01	36.2
Tree 1	2	7	8 594	1.61	40.0
Tree 2	2	6	3 751	1.39	39.2
Tree 3	4	6	10 884	2.24	40.6

L'analyse en composantes principales a permis de générer, comme rapportée à l'équation 3.2,  $3^m \times N_{règles}$  combinaisons de grammaires ;  $m$  étant le nombre de paramètres à varier dans la grammaire et  $N_{règles}$  le nombre de règles. Le tableau 5.3 offre en résumé le temps d'exécution des analyses en composante principale pour chacune des grammaires. Ce tableau contient une estimation du temps d'exécution en multipliant le temps d'exécution d'une grammaire (tableau 5.2) par le nombre d'arbres à générer. Sont compris dans cette estimation : le temps pour générer et dessiner chaque arbre, évaluer les métriques et calculer les PCAs. Les optimisations opérées pour arriver en dessous des estimations consistent en des réductions des opérations d'entrée-sortie, notamment le dessin en 3D dans la scène qui ne sera pas affiché et visualisé par l'utilisateur. Il en résulte que le temps d'exécution réel des analyses en composante principale est de très loin inférieur à l'estimation, comme vu dans le tableau 5.3. Le gain de temps moyen pour les sept grammaires est de 96.89 %.

Pour ce qui est de la régénération, le temps d'exécution est similaire à celui de l'exécution d'une grammaire. La moyenne de temps d'exécution des sept grammaires est de 1.95 seconde, selon le tableau 5.2. L'opération supplémentaire la plus importante dans les régénérations est celle de la manipulation du flux d'entrée-sortie consistant en un fichier XML, nécessaire pour le calcul de la coupe minimum. Le calcul de la coupe minimum se fait en quelques millisecondes, négligeable par rapport aux autres opérations.

Tableau 5.3 Informations de coût d'exécution des PCAs sur les sept grammaires.

Grammaires	Nombre de paramètres à varier	Nombre d'arbres à générer	Estimation du temps d'exécution PCA (en sec)	Temps d'exécution réel PCA (en sec)	Consommation mémoire (en Mo)
Alternative Branch.	5	1 215	2 308	199	4 400
Ternary Branch.	5	729	2 165	58	1 400
Honda Monopodial	6	2 187	3 389	65	1 600
Honda Sympodial	5	486	975	18	540
Tree 1	5	486	782	21	590
Tree 2	5	486	676	12	365
Tree 3	5	972	2 168	48	1 200

## 5.2 Conception du logiciel

La figure 5.2 illustre d'une manière globale le déroulement de la lecture d'un ensemble de grammaire. Au niveau de la conception du système, la classe de scène constitue une intermédiaire entre les instances de L-systèmes et la fenêtre de vue (figure 5.3). Les L-systèmes paramétriques comportent toutes les fonctionnalités des L-systèmes de base.

La figure 5.4 montre une image de la fenêtre principale du logiciel. La conception de l'application est illustrée dans la figure 5.5. On peut voir dans cette dernière que la gestion de la scène 3D (classe *Scene*) est gérée par l'outil de visualisation (classe *Viewer*). La fenêtre principale communique également avec la grammaire L-système initiale à travers la classe *PLSystemParametric*. L'arbre de dérivation pour un L-système est converti en un graphe orienté à travers un objet *PGraphCreator*. Cet objet entre en action dès la réception des sélections de l'utilisateur.

Le traitement du qualificatif choisi par l'utilisateur est également amorcé depuis la fenêtre principale comme montrée dans la figure 5.6 qui représente le diagramme de séquence de l'application du qualificatif. Chaque qualificatif est relié à une métrique, donc à une combinaison de variation de paramètres préalablement calculés. Cette combinaison est stockée dans l'interface *PPreproc* de la figure.

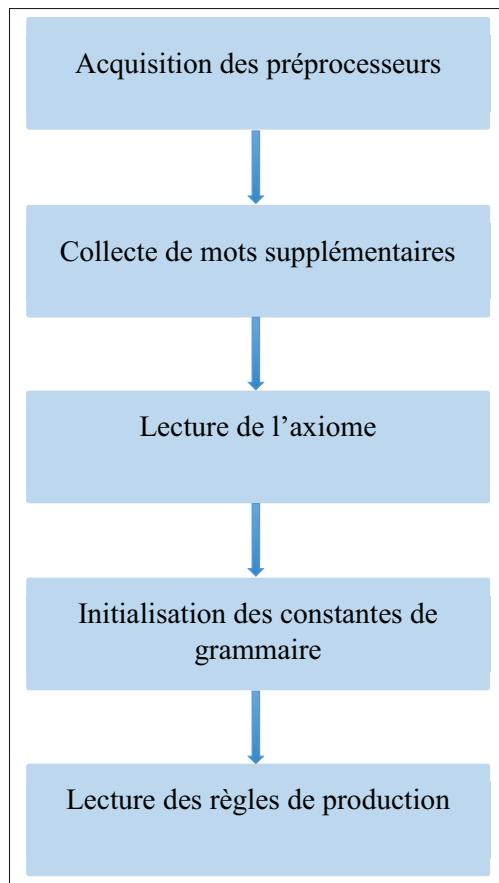


Figure 5.2 Pipeline de lecture de la grammaire.

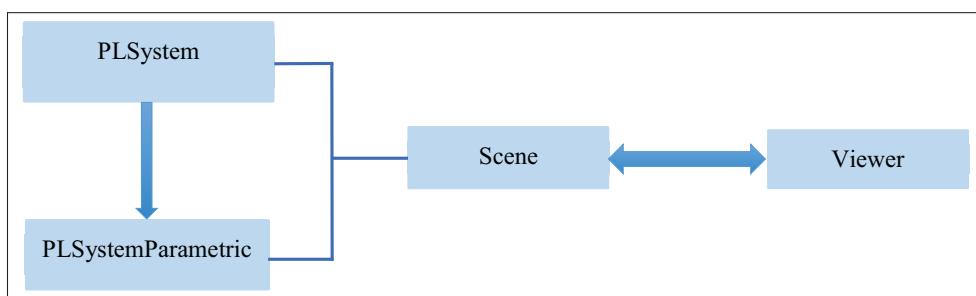


Figure 5.3 Diagramme de communication entre la vue et le L-système.

Le projet comprend une structure modèle/vue omniprésente dans l'environnement de développement *Qt*. La vue (ex : *MainWindow*, *Viewer...*) consiste à acquérir, modifier et afficher les données. Les modèles renferment les données qui sont destinées pour la vue.

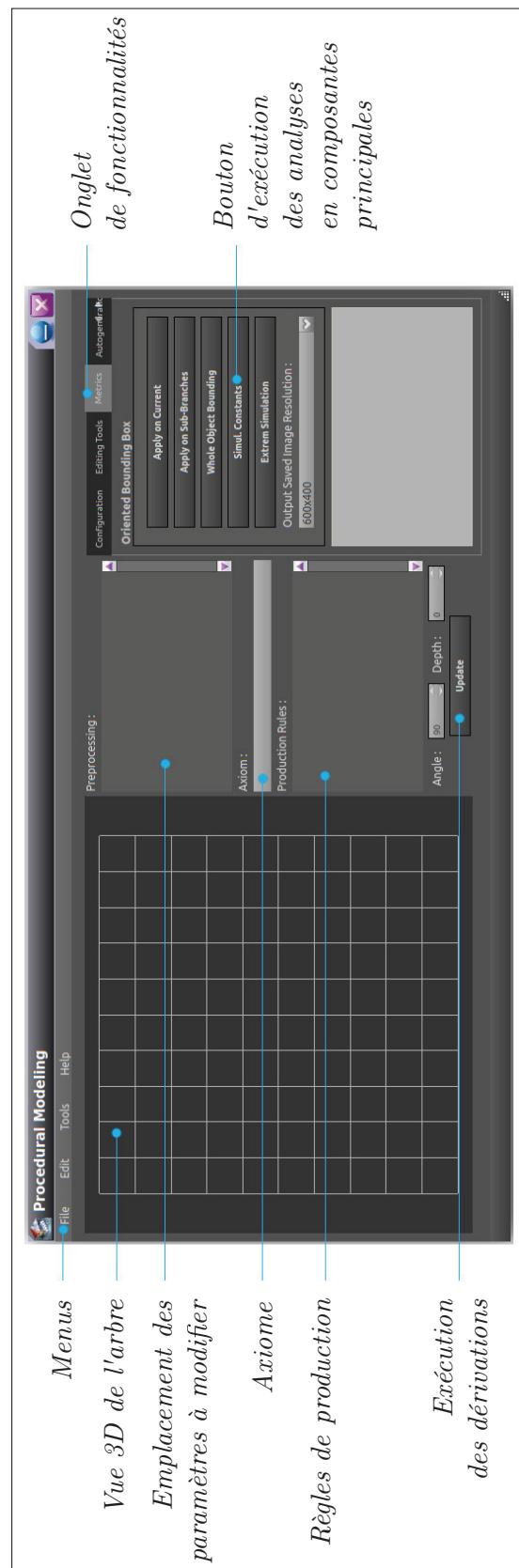


Figure 5.4 Aperçu et brève description de la fenêtre principale du logiciel.

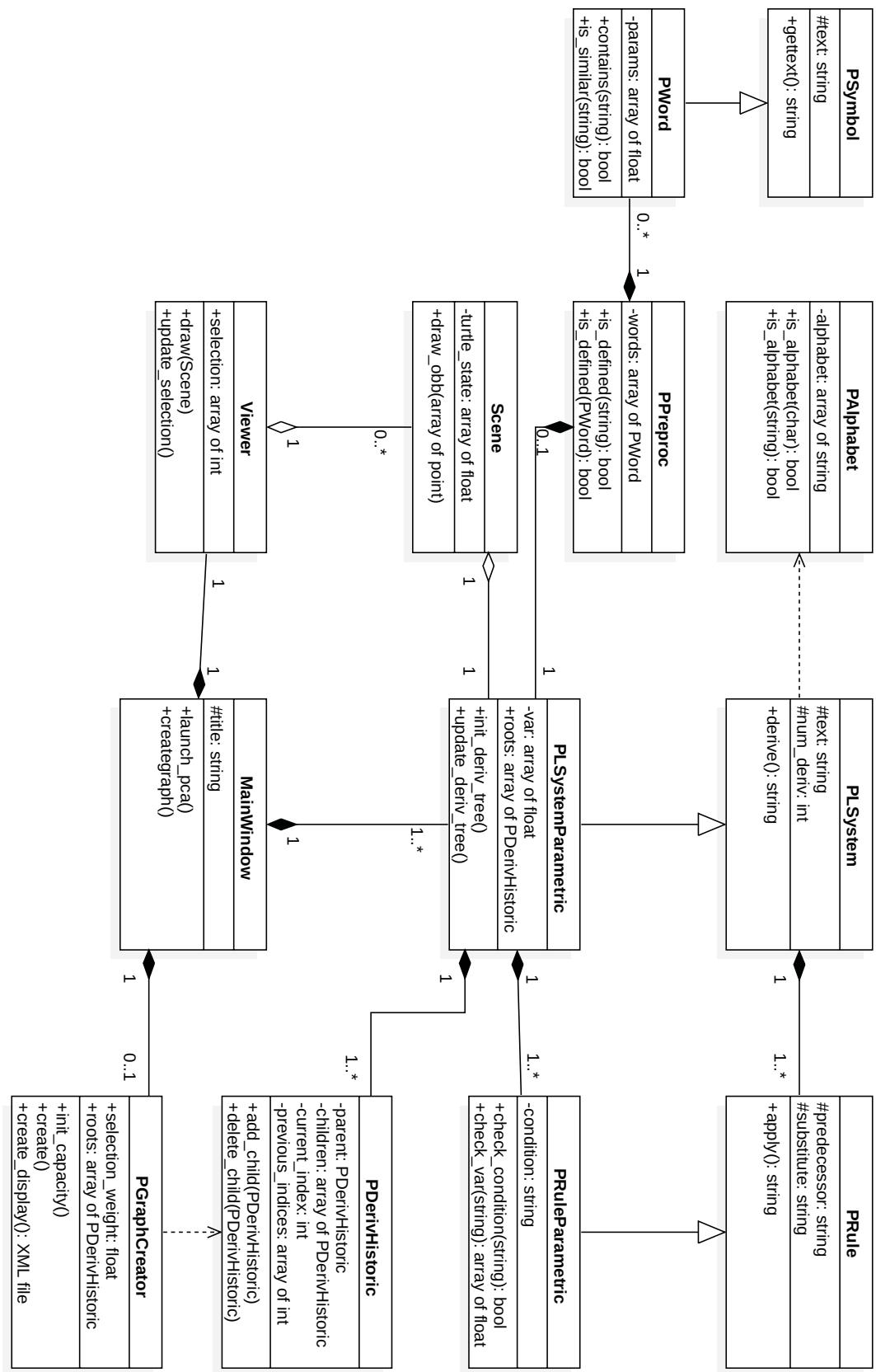


Figure 5.5 Diagramme de classe du système.

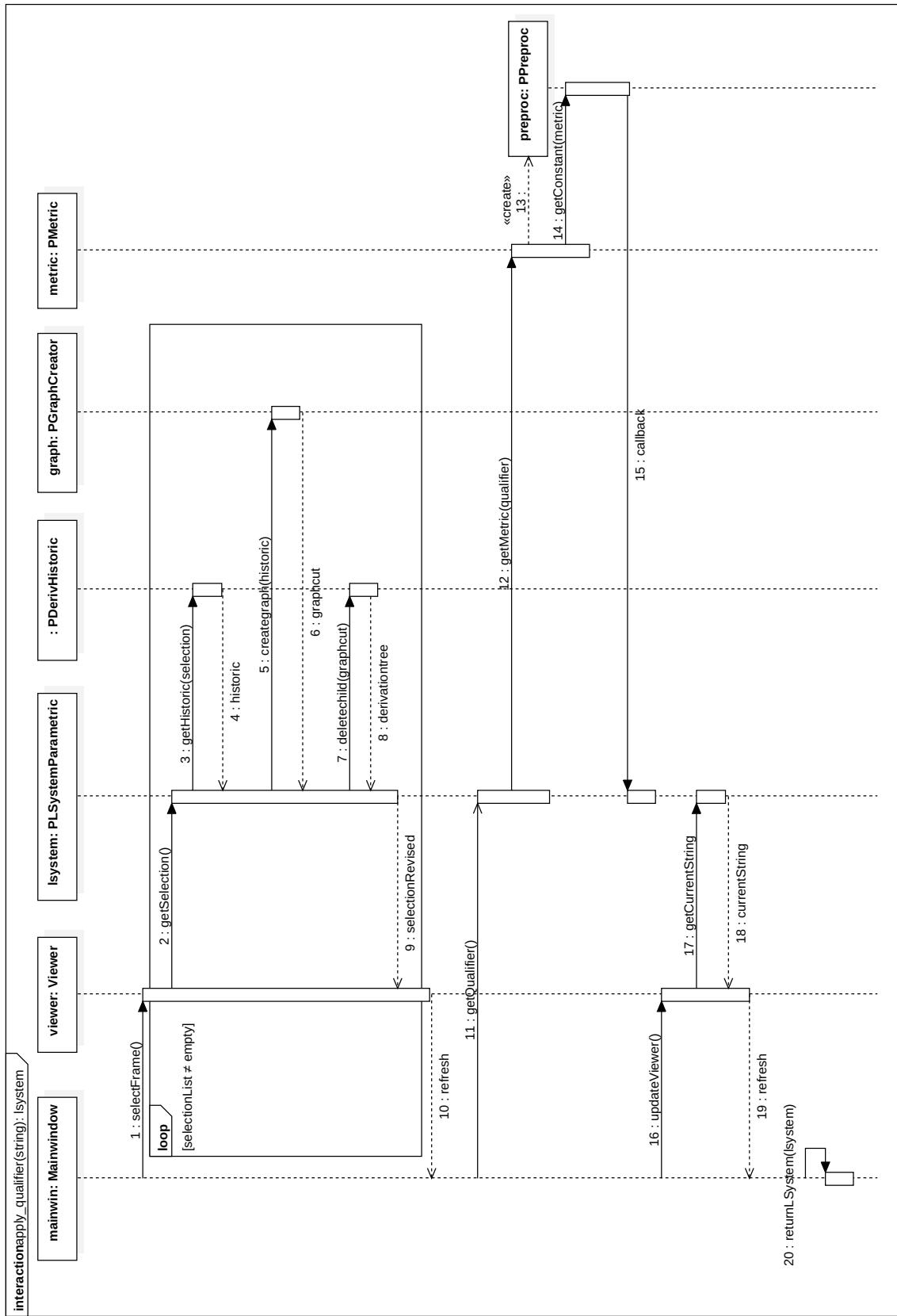


Figure 5.6 Diagramme de séquence portant sur la régénération par qualificatif.

Le patron médiateur, représenté dans le code par la classe *PMediator*, encapsule l’interaction des objets. Il sert à pallier au couplage rigide entre les classes, c’est-à-dire, les référencements directs et explicites entre deux objets. Une des fonctions du médiateur dans le cadre du système est d’agir interface entre la scène (classe *Scene*), la vue (classe *Viewer*) et le L-système (*PLSystemParametric*).

Les expressions arithmétiques sont analysées dans les méthodes à l’intérieur de la classe *PRuleParametric*. Deux emplacements sont à traiter pour une règle procédurale paramétrique : d’abord dans les conditions, et ensuite dans les mots de l’alphabet à la fois du prédécesseur et du successeur.

Lors de la lecture de la chaîne de caractères des règles procédurales, les expressions dans les paramètres du mot se rattachent aux algèbres générales, tandis que les expressions dans les conditions peuvent contenir des expressions booléennes en plus des algèbres générales. Dans la conception du système, tous les objets impliquant l’utilisation de ces expressions dérivent d’une classe *PAlgebra*.

Les opérateurs booléens renferment tout ce qui s’intéresse aux variables et fonctions logiques. Au départ, les paramètres d’entrée des opérateurs booléens sont les résultats (booléens) d’une comparaison d’expression arithmétique, qui sont encapsulés dans *PComparisonOperator*. *PComparisonOperator* comprend également un membre à gauche et un membre à droite, étant donné qu’il compare deux éléments. Les combinaisons de ces résultats s’imbriquent par des fonctions booléennes dont la fonction *ET*, notée par `&&`, et la fonction *OU*, notée par `||`.

Les préprocesseurs consistent en des collectes des constantes de la grammaire. Rappelons que les L-systèmes paramétriques permettent de définir des constantes en utilisant la directive `#define`. Les noms ainsi que les valeurs des constantes sont stockés deux fois : la première en tant qu’élément préprocesseur au même titre que d’autres directives, et une seconde fois en tant que variable numérique qui sera consultée avec les paramètres des mots de l’alphabet du L-système. Ces constantes sont passées à la méthode `PRuleParametric::compute()` en complément avec d’autres variables liées aux mots courants.

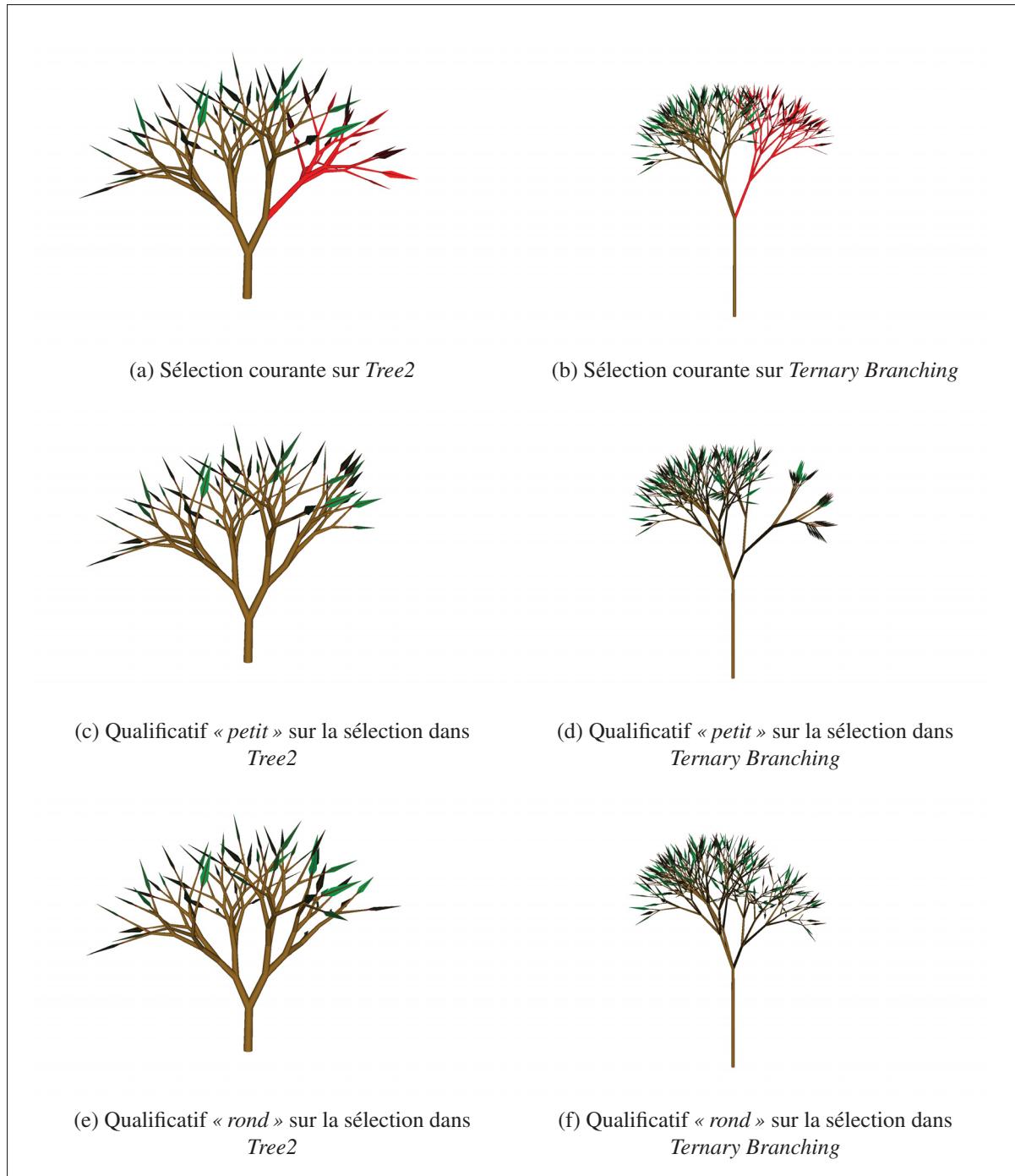


Figure 5.7 Regénération de la sélection courante des deux grammaires *Tree2* et *Ternary Branching* en utilisant les qualificatifs.

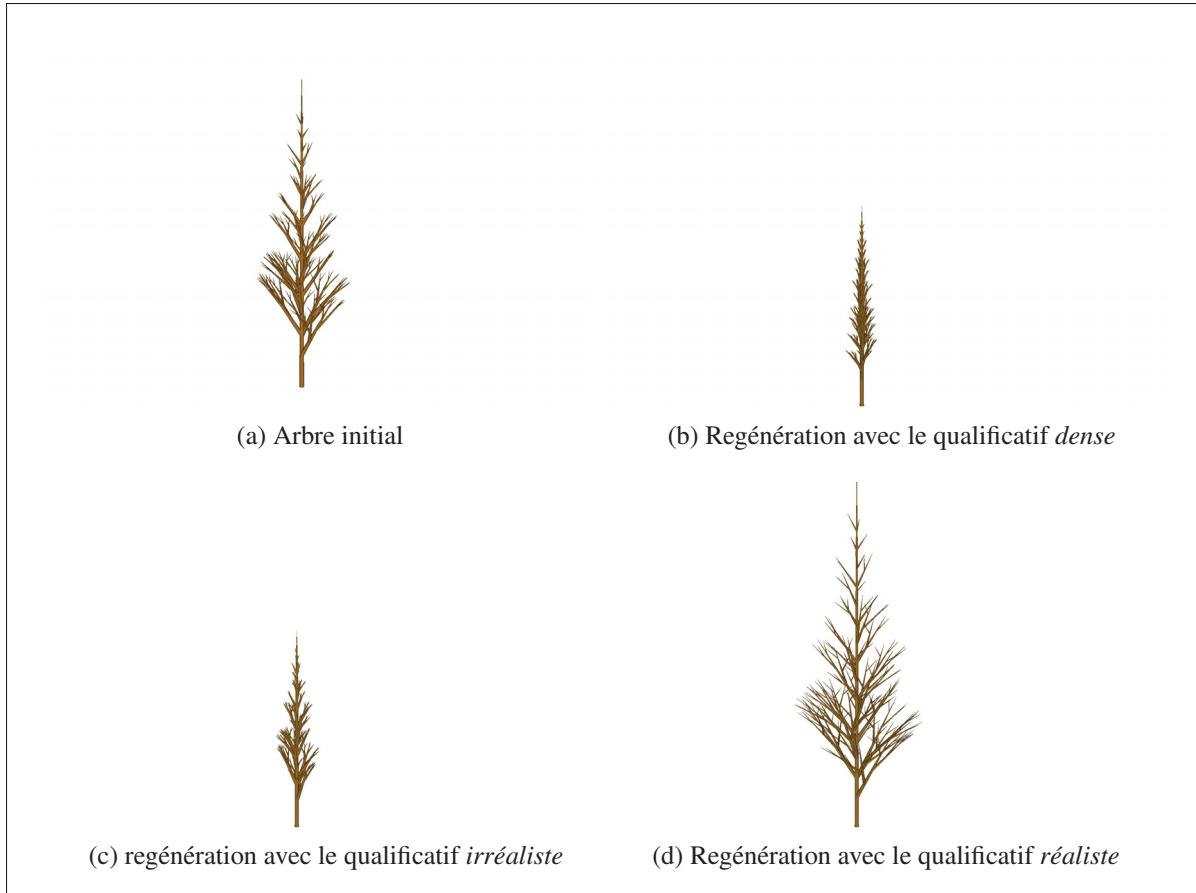


Figure 5.8 Regénération intégrale de l'arbre de la grammaire *Tree3*.

Les opérations arithmétiques ayant lieu dans les L-systèmes paramétriques sont des arithmétiques élémentaires avec les opérations traditionnelles dont l'addition, la soustraction, la multiplication et enfin la division. L'évaluation des expressions arithmétiques consiste à vérifier si la comparaison entre deux expressions est vérifiée, c'est-à-dire correcte (*vraie*) ou non (*faux*). Pour plusieurs conditions dans la règle, on effectue alors des opérations booléennes sur l'ensemble des conditions. Le système intègre des fonctionnalités qui évaluent des expressions booléennes. Le format de l'entrée reçue par le système est toujours une chaîne de caractères.

Ces opérations interviennent toutes dans l'évaluation des expressions des conditions, et sont développées encore plus dans l'annexe II de ce document.

### 5.3 Résultats et discussions

Les deux grammaires *Tree2* et *Ternary Branching* sont mises côte à côte dans la figure 5.7. Les qualificatifs utilisés sont « *petit* » pour l’arbre (c) et (d), puis « *rond* » pour l’arbre (e) et (f). La figure 5.8 est un aperçu de l’arbre *Tree3* et ses régénéérations intégrales avec les qualificatifs « *dense* » (b), « *irréaliste* » (c) et « *réaliste* » (d). La mesure de différence entre les arbres à travers l’analyse en composantes principales sur les métriques produit des variétés intéressantes (figures 3.5 et 5.8). Les approches utilisées pour obtenir des diversités de modèle sont novatrices mais présentent aussi les limites.

#### Points forts

Les démarches utilisées pour créer des types d’arbres à partir d’une grammaire initiale présentent l’aspect le plus avantageux de pouvoir contrôler l’évolution de l’arbre. L’ajout et la suppression d’éléments dans la sélection se font intuitivement en utilisant le clavier et la souris. Les gradients de couleurs lors d’une sélection simple permettent d’avoir une idée sur les voisinages des branches par rapport à son arbre de dérivation. De plus, la coupe de graphe orienté permet une inférence intelligente sur la sélection de l’utilisateur, de manière à prioriser l’ensemble et pénaliser les sélections isolées. La méthode conçue est hautement interactive et peut offrir un contrôle avancé sur le modèle final de l’arbre à générer.

Un autre point fort de l’approche est sa capacité à sélectionner parmi une multitude de modèles d’arbre ceux dont l’apparence visuelle peut se démarquer. Nous avons pu voir depuis les expérimentations effectuées avec des participants qu’en moyenne, plus de cinq groupes d’arbres différents peuvent être créés à partir de l’arbre de la grammaire initiale. Ces cinq groupes sont tous formés à partir des 27 images sélectionnées par l’approche.

L’utilisation de qualificatifs dans la modélisation peut assister les utilisateurs sur la nature des arbres ou branches qui seront générés. Le système renferme dix qualificatifs pour guider l’utilisateur.

Un avantage important de nos démarches est finalement le temps d'exécution des créations, puis des regénérations totales et partielles de l'arbre. L'analyse en composantes principales est la partie la plus coûteuse en termes de processeur et temps d'exécution, mais elle s'effectue également en dessous des barres des 5 minutes pour générer un nombre typique d'arbres pouvant aller de 500 jusqu'à 2 000 arbres. D'autant plus, cette analyse en composantes principales est effectuée pour les tests soumis aux participants, et de ce fait n'entrera plus dans le processus de régénération proprement dit. L'utilisateur ne verra donc aucune longue opération. Lorsque l'analyse en composantes principales est effectuée, tous les types de régénération sont complétés en moins de deux secondes. Ce temps inclut le calcul des coupes de graphe orienté pour la sélection, et l'application des nouveaux paramètres dans les règles selon les qualificatifs choisis par l'utilisateur.

## Limitations

Bien que les qualificatifs puissent être un bon assistant pour l'infographiste, une description subjective de l'arbre peut être un sujet assez vaste pour n'être décrit qu'à travers 10 qualificatifs. Les participants ont donné un total de 186 qualificatifs uniques dans le cadre des expérimentations. Ceci nous donne une idée de la multiplicité des qualificatifs qu'un arbre peut avoir. Le nombre de qualificatifs proposés dans notre méthode peut alors devenir une limite contraignante pour les utilisateurs.

Dans cette même optique, les qualificatifs peuvent produire des arbres ne correspondant pas à la représentation du qualificatif choisi. Par exemple dans la figure 5.7f, le qualificatif « *rond* » est appelé pour générer une partie des branches de l'arbre *Ternary Branching*. Cependant, il devient difficile de relier visuellement le terme « *rond* » aux branches nouvellement formées par la régénération. On a pu constater que plusieurs points dans l'espace PCA peuvent correspondre au même qualificatif, et que ces espaces peuvent entrer en collision pour plusieurs qualificatifs. Ce décalage entre l'aspect visuel et la signification du qualificatif est engendré par les différences obtenues dans les vecteurs caractéristiques, puisque l'analyse en composantes

principales a été calculée indépendamment pour chaque grammaire. L'espace propre peut en effet varier d'une grammaire à une autre.

Une autre barrière pour l'exploitation de notre démarche est le format d'une grammaire L-système qui nécessite une connaissance préalable du fonctionnement des L-systèmes. Toutefois, l'utilisateur n'a à manipuler les grammaires qu'au début. Par la suite, les regénérations ne nécessitent pas que l'utilisateur manipule la grammaire.

Enfin, la méthode de génération de modèle d'arbre par analyse en composantes principales n'est pas à l'abri de la conception d'un arbre ultimement irréaliste, ce qui peut être incongrue pour l'artiste modeleur. La figure 5.7d montre un arbre qui illustre bien ce cas. On peut constater dans cette figure, générée en utilisant le qualificatif « *petit* », que les éléments de l'arbre sont entremêlés et sont graphiquement invraisemblables.

### **Perspectives et recommandations**

Bien que la méthode présente des atouts intéressants, il y a encore beaucoup de place à l'amélioration dans le but de parfaire encore les résultats.

L'ajout de métriques est intéressante dans le but d'obtenir encore plus de variétés d'arbres dans les résultats. Entre autres, on pourrait exploiter la colonisation de l'espace dans l'arbre. Cette métrique évaluerait un taux de convergence et de contact entre les branches dans l'arbre. Une autre métrique qu'on peut également considérer serait la compétition pour la lumière. On pourrait déjà mesurer dans une première phase la quantité de lumière que la surface en dessous de l'arbre peut accumuler. Ensuite, la même quantité peut être calculée mais sur les éléments individuels de l'arbre.

La prise en charge des grammaires stochastiques et avec contextes est aussi une avenue intéressante pour le projet. Ces grammaires permettent de choisir les règles procédurales à appliquer, selon les probabilités d'application de ces règles, et aussi selon les conditions des voisinages de

l'élément sélectionné par l'utilisateur. Ces aspects des L-systèmes peuvent être exploités pour obtenir des modèles d'arbres variés.

Ensuite, la catégorisation des qualificatifs peut accélérer le processus de génération de l'arbre. La morphologie de l'arbre pourrait par exemple être dissociée de sa taille. De cette façon, le qualificatif « *grand* » n'entrerait pas en conflit avec le qualificatif « *dense* » ou encore « *réaliste* ». Cette catégorisation serait indiquée à mesure que de nouveaux termes s'invitent dans les qualificatifs du système.

Une autre avenue à explorer sur les qualificatifs serait de conduire une nouvelle étude qui validerait si les arbres générés par les qualificatifs correspondent bien au sens des qualificatifs. Il serait intéressant de faire valider par une nouvelle expérimentation où les participants classeront des ensembles d'arbre 3D selon les qualificatifs. Cette expérimentation aurait pour but d'estimer l'efficacité des approches par mise en correspondance de l'espace PCA avec les qualificatifs.

Finalement, le calcul des métriques pourrait également être effectué sur un nombre limité de dérivation, au lieu de l'arbre au complet comme il a été suggéré dans notre méthode. Les mesures effectués localement peuvent en réalité produire des effets qui seront transparents pour une régénération complète. Une boîte englobante sur l'arbre complet peut par exemple ignorer totalement une modification locale à une branche, alors que cette modification peut être prise en compte si la mesure s'effectue sur une partie de l'arbre. Le calcul des métriques sur une ou plusieurs dérivations prend tout son intérêt dans ce type de cas.

## CONCLUSION

Ce projet a eu comme objectif de proposer des solutions à la création de modèles procéduraux d'arbre à partir d'un arbre initial. L'approche suggérée dans le cadre de ce mémoire traite des grammaires L-systèmes reçues en entrée. Des mécanismes permettent de sélectionner un sous-ensemble de l'arbre et de regénérer ce dernier. L'apparence de l'arbre 3D résultant est influencée par des ajustements automatiques des paramètres du L-système. Pour identifier les modifications les plus pertinentes de l'arbre, des centaines d'arbres ont été générés afin d'évaluer des métriques. Un sous-ensemble restreint de ces arbres a été retenu et présenté à des participants afin de distinguer un ensemble encore plus réduit de modifications intéressantes de paramètres.

Les modifications de l'arbre reposent sur une structure appelée arbre de dérivation. Lorsque l'utilisateur requiert une régénération ou une suppression d'un élément de l'arbre 3D, le mot correspondant est identifié, et une suppression récursive des nœuds et feuilles de l'arbre de dérivation est opérée.

Les paramètres sont modifiés dans le but de produire des grammaires avec des combinaisons de paramètres qui auront un impact visuel sur le modèle procédural 3D. Chaque arbre 3D généré est mesuré sur une base de 10 métriques afin d'évaluer la différence avec l'arbre initial fourni par l'utilisateur. Une analyse en composantes principales est pratiquée sur les métriques de l'ensemble des arbres générés. Les trois vecteurs caractéristiques les plus importants sont retenus pour choisir 27 points dans ce repère qui correspondent à des arbres dont la différence par rapport à l'arbre initial serait considérable. Les 27 arbres générés sont présentés à des participants pour des expérimentations. Les participants émettent leurs observations sur la pertinence des arbres générés ainsi que les différences au sein de ces derniers. Les participants sont également invités à proposer selon leurs appréciations des qualificatifs sur les arbres générés. Des analyses sont réalisées afin de pouvoir réutiliser les qualificatifs proposés par les participants.

Les régénérations peuvent concerner plusieurs mots en même temps. Tel est le cas lors des multisélections. La méthode traite les multisélections avec un mécanisme de conversion de

l’arbre de dérivation en un graphe orienté pour ensuite y effectuer une coupe minimale. Cette conversion consiste à traduire chaque nœud et arête de l’arbre de dérivation en un ensemble de nœud et arête dans un graphe orienté. Les feuilles de l’arbre de dérivation sont connectées du côté du drain, tandis que la racine va à la source. L’orientation dans le graphe va dans le même sens que dans l’arbre de dérivation : du nœud parent au nœud enfant. Une attribution de capacité s’ensuit dans le graphe dans le but de donner plus de poids aux sélections où plusieurs enfants d’un même parent sont sélectionnés. La décision d’entreprendre ou non une régénération sur une feuille repose sur le calcul de la coupe minimale dans le graphe orienté.

Ce mémoire est une avancée dans le domaine de la modélisation procédurale appliquée aux arbres. Les approches proposées automatisent sous plusieurs aspects la production de modèles d’arbre 3D grâce à l’exploitation de techniques de sélection intelligente et à l’intégration des métriques pour mesurer la différence entre les arbres. Cette façon de procéder a pu générer à partir d’une grammaire plusieurs types d’arbres, sans une intervention intensive de la part de l’artiste modeleur. L’étude empirique sur des sujets humains a permis d’évaluer la pertinence des arbres, mais aussi de tirer avantage des qualificatifs. De plus, chaque arbre généré et différent de l’arbre initial peut être proposé aux artistes grâce à une interface utilisateur comportant des vignettes qui permettent d’avoir un aperçu du résultat.

Les démarches choisies dans le cadre de la génération de modèles procéduraux d’arbre vont désormais pousser les acteurs du domaine à aborder le problème sous deux angles importants. Le premier point consiste en un mécanisme de sélection d’une structure organisée en arborescence, qui est l’essence même de la modélisation procédurale. Le second point est la génération automatique d’objets 3D à l’aide d’une analyse statistique sur leurs propriétés géométriques et morphologiques.

Enfin, le projet peut encore connaître des améliorations notamment à travers le support d’autres types de grammaires telles les grammaires de forme. Aussi, les mesures de différence entre les arbres générés peuvent être perfectionnées en prenant en compte la colonisation de l’espace et la compétition pour la lumière au sein des éléments de l’arbre afin d’identifier davantage

les modèles d'arbres différents. Ces nouvelles idées vont aider à atténuer la complexité de la création d'arbres par la modélisation procédurale en introduisant plus d'arbres aux utilisateurs.



## ANNEXE I

### SYMBOLES ET INTERPRÉTATION DE LA TORTUE

Symbol	Interpretation
F	Avancer en dessinant une ligne
f	Avancer sans dessiner une ligne
+	Tourner à gauche
-	Tourner à droite
^	Tanguer en haut
&	Tanguer en bas
\	Rouler à gauche
/	Rouler à droite
	Tourner autour
\$	Tourner la tortue à la verticale
[	Commencer une branche
]	Compléter une branche



## ANNEXE II

### ALGÈBRE SUR LES CONDITIONS DES RÈGLES PROCÉDURALES

#### 1. Opérateurs arithmétiques

Les opérations arithmétiques agissent sur deux termes pour la facilité d'implémentation. Ainsi, tout objet de la classe *PArithmetic* contient obligatoirement un membre à gauche et un membre à droite pour pouvoir exécuter une opération. Les membres peuvent-être des variables *PVariable*, des numériques *PNumericValue* mais aussi peuvent-être de la classe *PArithmetic* elle-même qui dérive d'une même classe *PNumeric*. L'instanciation d'un objet pendant la lecture de caractères va ainsi dépendre du type de caractère que l'on reçoit à l'entrée. Par exemple, les types numériques vont engendrer la création d'un objet *PNumericValue*, tandis que la lecture d'un alphanumérique va créer un objet *PVariable*. En conclusion, pour une expression arithmétique telle présentée à la figure II-1, l'évaluation commence à partir des feuilles.

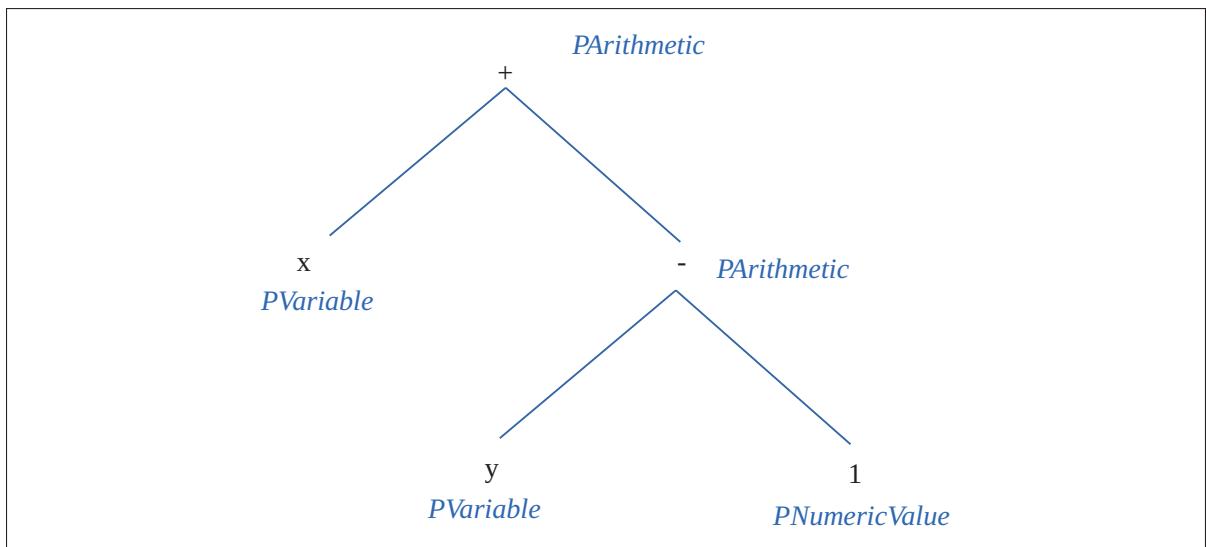


Figure-II-1 Organisation en arborescence du calcul arithmétique dans le système.  
Tous les nœuds qui ne sont pas des feuilles comprennent obligatoirement deux membres.

## 2. Opérateurs booléens

Pour les facilités d'implémentation, les opérateurs booléens encapsulés dans la classe *PBooleanOperator* comprennent deux membres obligatoires. Cependant, la fonction *NON* est une opération à une seule entrée. Elle est prise en charge dans notre système, mais ne crée aucune structure de donnée. Pour gérer cette fonction, la négation ou l'exécution de la fonction *NON* est effectuée à l'interne. Plus concrètement, chaque type de données qui retourne un booléen, doit *savoir* à travers un attribut interne s'il doit retourner sa négation au lieu de sa valeur. Chaque lecture du symbole « ! » initiera cet attribut interne de négation à *TRUE*.

## 3. Gestion des priorités

Toujours pour des facilités d'implémentation, les parenthèses dans les expressions arithmétiques sont obligatoires. À partir de là, toutes les lois régissant l'algèbre classique sont maintenues. L'expression qui se trouve dans les parenthèses les plus rentrés est prioritaire. Le système comprend une organisation en pile des expressions qui se basent sur le fait que toutes les structures de données sont à deux membres. Toute ouverture d'une parenthèse initie la préparation d'un bloc pour stocker des informations algébriques notamment des membres, tandis que la fermeture d'une parenthèse initie l'exécution d'une opération sur les membres.

## ANNEXE III

### L-SYSTÈMES STOCHASTIQUES ET AVEC CONTEXTES

#### 1. L-systèmes stochastiques

Une même grammaire va générer un arbre rigoureusement identique. Plongée dans un même milieu, une grammaire entraîne une apparition d'un motif récurrent évident. Les L-systèmes stochastiques permettent une variation en introduisant des paramètres aléatoires dans l'application des productions. Plus formellement, un L-système stochastique est défini par un quadruplet  $G_\pi = < V, \omega, P, \pi >$  où  $\pi$  est une fonction prenant sa valeur dans  $(0, 1]$ , et est appelée *distribution de probabilité*. Cette valeur, si différente de 1, est écrite au début de la règle de production, comme dans l'exemple suivant :

.33  $A(x) \rightarrow A(x+1)$

Ici, la règle sera utilisée avec une probabilité de 33 %.

Pour un même mot prédécesseur, la somme des paramètres stochastiques doit être égale à 1. L'extrait III.1 présente un exemple de grammaire stochastique. Les paramètres des règles, spécifiés en début de ligne dans les règles, sont  $\rho_1 = 0.25$  (ou 25 %) et  $\rho_2 = 0.75$  (ou 75 %) pour respectivement la première et la seconde règle.

#### Extrait III.1 Exemple type de grammaire stochastique

```
#define c1 0.5
#define c2 60.0
n:      5
w:      Branche(5,10)
p:
.25 Branche(x,y):* -> Branche(x*c1,y+5)[&(c2)Feuille(y)]
/(120)[&(c2)Feuille(1)]
.75 Branche(x,y):* -> !(30)[+(c2)$Branche((x*c1),y)]
```

La sélection de la règle qui sera appliquée est un processus simple qui consiste à choisir parmi toutes les règles qui comportent le mot courant. Un tirage au sort est effectué avec une équ-probabilité si aucun paramètre stochastique n'est assigné.

## 2. L-système avec contexte

Les contextes fournissent un cadre pour un mot d'adopter une action par rapport à son environnement. L'application d'une production peut en effet dépendre du précédent et du successeur. Ce concept permet d'imiter les interactions entre les différentes parties des plantes. On utilise les symboles « < » et « > » pour signifier respectivement le contexte à gauche et à droite suivant la représentation suivante :  $c_l < a > c_r \rightarrow \chi$ . Dans cette notation, on applique la production si et seulement si le mot  $a$  est précédée par  $c_l$  et est suivie par  $c_r$ . On assume que les règles de production avec contexte précèdent celles sans contexte. En conséquence, si un mot admet une règle avec et sans contexte, celle avec contexte s'applique en priorité.

Les contextes de voisinage servent de condition à l'application d'une règle. À chaque règle de production munie d'un contexte, un objet *PContext* est créé. Ce dernier contient les routines de validation d'une règle par rapport aux contextes. Le processus se déroule en deux parties :

- l'analyse de la règle où on garde en mémoire les mots à gauche et à droite,
- l'analyse de la chaîne courante où s'effectue à chaque itération une fouille des voisinages du mot courant.

Pour un contexte d'ordre  $N$ , l'application parcourt les  $N$  mots à gauche et les  $N$  mots à droite de lu mot courant.

## ANNEXE IV

### ARCHIVAGE DES INFORMATIONS DE DÉRIVATIONS

L'objectif quant à la conception d'une structure de données d'un état d'historique est de pouvoir identifier de façon unique l'origine d'un mot. Ainsi pour un mot donné, on souhaiterait savoir dans un premier temps son indice dans la chaîne courante noté *PDerivHistoric :: idx\_into\_deriv*. Cet indice est son rang, qui correspond au nombre de caractères qui le précèdent dans la chaîne courante.

Un même type de propriété est utilisé, *PDerivHistoric::idx\_into\_gen*, pour désigner l'indice du mot dans la chaîne précédente, c'est-à-dire, avant l'exécution de la dernière itération de la dérivation. Dans le cas où le mot n'existe qu'après la dernière itération, nous assignons une valeur négative à cette propriété, et nous retenons en conséquence des propriétés qui sont des informations de racine.

Le numéro, dans la liste de règles qui a généré le mot, est stocké dans la propriété *PDerivHistoric :: rule\_gen\_num*. Ensuite, l'origine de lu mot dans la règle procédurale est conservée dans *PDerivHistoric :: idx\_within\_rule*.



## ANNEXE V

### QUESTIONNAIRE AUX PARTICIPANTS DES EXPÉRIMENTATIONS

#### 1. Projet « Analyse des variations de l'apparence d'arbres virtuels »

Remplir ce questionnaire en lien avec les groupes d'images que vous venez de former. Pour chaque question, il se peut que vous ayez identifié moins de groupes qu'il y en a sur le questionnaire. Ne considérez alors que les numéros de groupes qui sont pertinents.

#### 2. Question 1

Dans le contexte de la modélisation d'arbres 3D, indiquez si vous trouvez que les groupes que vous avez identifiés seraient utiles.

Encerclez la réponse choisie.

Groupe	1	2	3	4	5	6	7
Excellent							
Bon							
Passable							
Mauvais							

#### 3. Question 2

Qualifiez la différence d'un groupe par rapport aux autres groupes.

Encerclez la réponse choisie.

Groupe	1	2	3	4	5	6	7
	Excellent						
	Bon						
	Passable						
	Mauvais						

#### 4. Question 3

Qualifiez l'image représentative que vous avez choisie pour chaque groupe. Il n'est pas nécessaire de répondre si le groupe ne comporte qu'une seule image.

Encernez la réponse choisie.

Groupe	1	2	3	4	5	6	7
	Excellent						
	Bon						
	Passable						
	Mauvais						

#### 5. Question 4

Trouvez un qualificatif pour chaque groupe. Le qualificatif choisi devrait être utile pour la modélisation d'arbres virtuels.

Groupe      Qualificatif

1

2

3

4

5

6

7

## ANNEXE VI

### CALCUL DES MÉTRIQUES POUR LES ANALYSES EN COMPOSANTES PRINCIPALES

#### 1. Combinaisons de constante

Le nombre de combinaisons qui sera pris en compte dans les PCAs est donné par la formule :

$$N_{arbres} = 3^m \times N_{regles} \quad (\text{A VI-1})$$

comme exposé dans le chapitre 4 sur la mesure de différence des arbres.

#### 2. Centrage des données et calcul des covariances

Une grammaire est formée à partir de chaque combinaison de constantes constituée précédemment. Une méthode de calcul de différence est implémentée dans les objets métriques. L'ensemble des valeurs qui en ressort constituera les entrées pour l'analyse en composantes principales.

Pour une entrée en vecteur colonne :

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}$$

la matrice de covariance  $\Sigma$  est une matrice dont l'entrée  $(i, j)$  est la covariance :

$$\Sigma_{i,j} = cov(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)]$$

Le centrage des données consiste à soustraire la moyenne sur chaque dimension ou colonne de la population.

#### 3. Décomposition en éléments propres

La matrice  $V$  qui diagonalise la matrice de covariance  $C$  forme les vecteurs propres :

$$V^{-1}CV = D$$

$D$  est une matrice diagonale qui contient les valeurs propres de  $C$ . La décomposition en éléments propres est un problème classique dont plusieurs solutions ont été apportées. Un calcul des éléments propres à l'aide de l'outil MATLAB peut s'effectuer avec l'instruction :

$$[V, D] = eig(comat)$$

avec  $comat$  la matrice de covariance,  $V$  les vecteurs propres et  $D$  les valeurs propres.

Tableau-A VI-1 Résultat de la décomposition en éléments propres appliquée sur les différences de métriques de toutes les grammaires

Vecteurs propres										
[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	
0.00	-0.04	0.01	-0.30	0.39	0.01	0.03	0.61	-0.35	0.51	
0.00	0.07	0.00	0.70	-0.50	0.15	0.09	0.27	-0.20	0.33	
0.00	-0.41	0.73	-0.13	-0.15	0.06	0.00	-0.32	-0.38	0.13	
0.00	0.20	-0.39	-0.47	-0.39	0.47	0.17	-0.26	-0.33	0.11	
0.58	0.07	-0.11	0.23	0.32	-0.20	0.52	-0.28	-0.30	0.00	
0.00	0.00	0.00	-0.16	-0.34	-0.41	0.09	0.40	-0.42	-0.59	
0.00	0.00	0.01	0.27	0.38	0.60	-0.29	0.07	-0.35	-0.45	
-0.56	-0.01	-0.31	0.18	0.17	-0.39	-0.22	-0.37	-0.42	0.15	
0.56	0.20	-0.04	-0.06	-0.14	-0.19	-0.74	-0.08	-0.12	0.15	
0.19	-0.86	-0.45	0.02	-0.09	0.04	-0.08	0.05	0.05	0.00	
Valeurs propres										
0.00	0.05	0.07	1.42	2.03	21.08	77.70	149.29	787.34	3,791.88	

## ANNEXE VII

### ANALYSE COMPLÉMENTAIRE SUR LA COUPE DE GRAPHE

Un cas de sélection de la moitié des feuilles se trouve dans la figure VII-1. Encore une fois, la coupe se veut prudente, bien que nous aimerions avoir une option de coupe sur  $\Gamma_3$  qui a la moitié des feuilles rattachées sélectionnées.

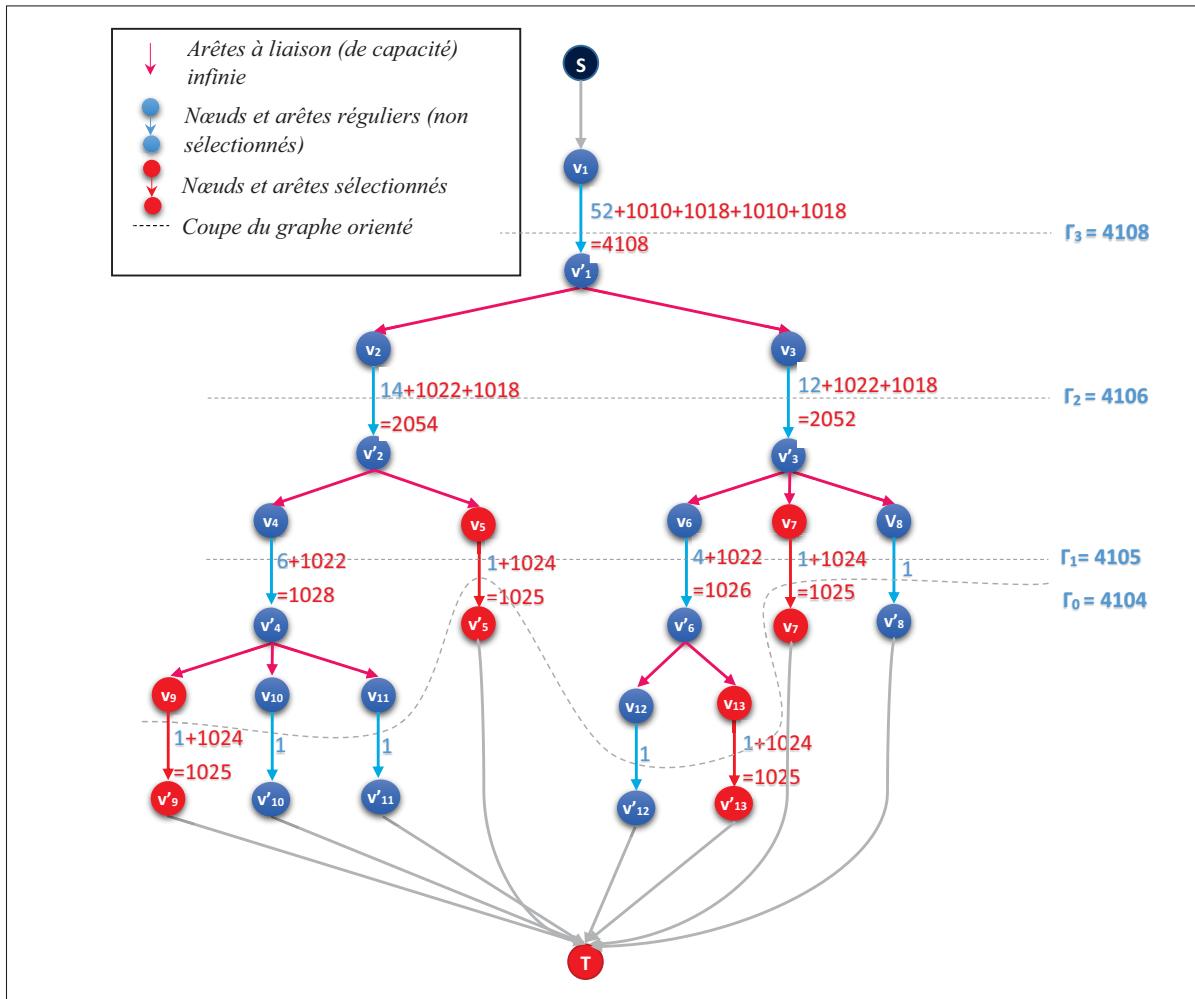


Figure-A VII-1 Cas de sélection de la moitié des feuilles dans un graphe plus complexe.

Dans la figure VII-2, nous pouvons observer qu'un ajout de sélection par rapport à la configuration précédente a tout de suite dirigé la coupe minimale vers le niveau  $k = 3$ .

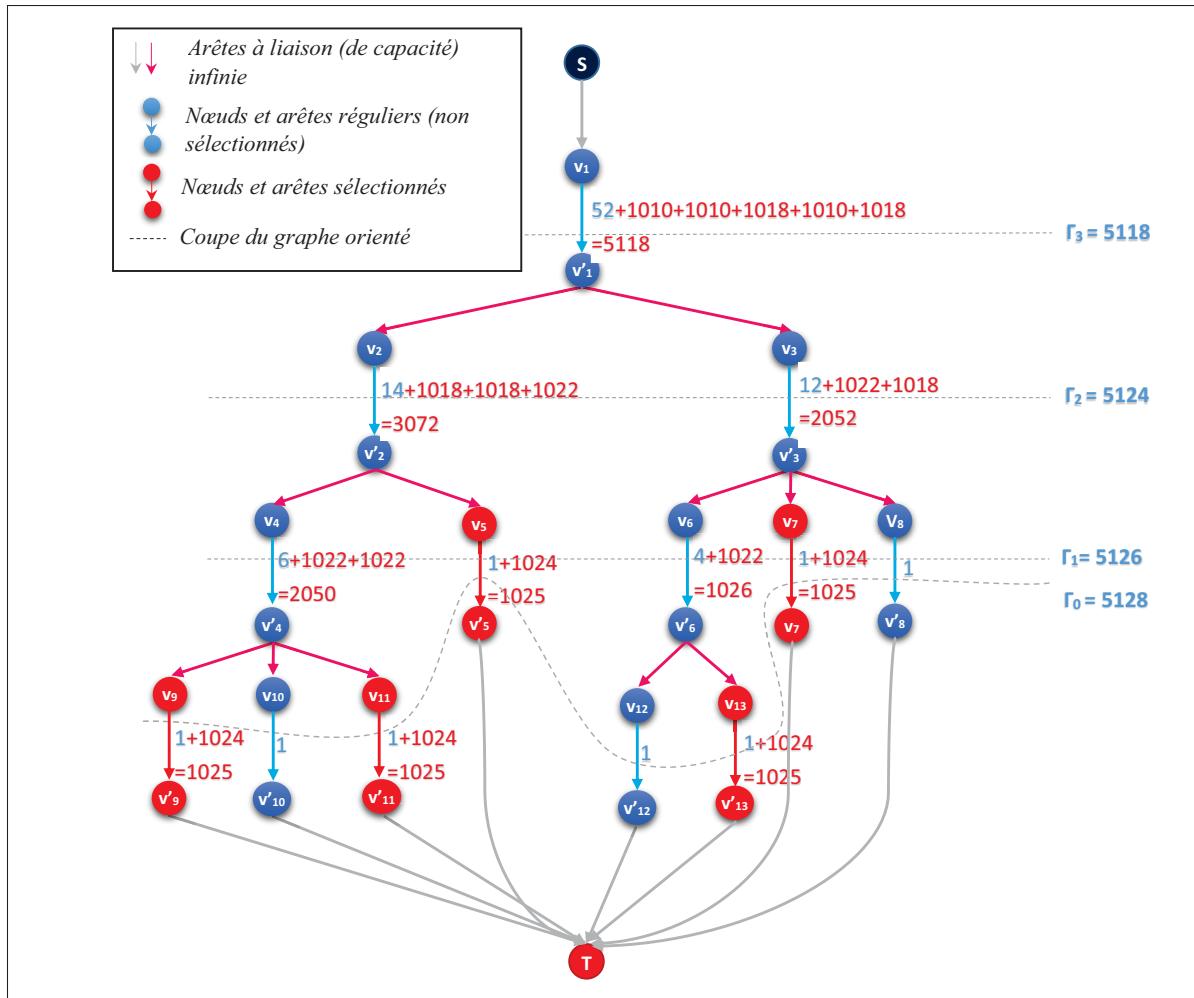


Figure-A VII-2 Sélection aléatoire dans un graphe plus complexe.

## BIBLIOGRAPHIE

- Aliaga, D., Rosen, P. & Bekins, D. (2007). Style grammars for interactive visualization of architecture. *IEEE Transactions on Visualization and Computer Graphics*, 13(4), 786–797.
- Beneš, B., Št'ava, O., Měch, R. & Miller, G. (2011). Guided procedural modeling. *Computer Graphics Forum*, 30(2), 325–334.
- Biggers, K. & Keyser, J. (2011). Inference-based procedural modeling of solids. *Computer-Aided Design*, 43(11), 1391–1401.
- Birch, P. J., Browne, S. P., Jennings, V. J., Day, A. M. & Arnold, D. B. (2001). Rapid procedural-modelling of architectural structures. In *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage*, pp. 187–196.
- Bokeloh, M., Wand, M. & Seidel, H.-P. (2010). A connection between partial symmetry and inverse procedural modeling. *ACM Trans. Graph.*, 29(4), 104 :1–10.
- Boykov, Y. & Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1), 1124–1137.
- Chen, G., Esch, G., Wonka, P., Müller, P. & Zhang, E. (2008). Interactive procedural street modeling. *ACM Trans. Graph.*, 27(3), 103 :1–10.
- Clément-Corriveau, N. (2016). *Modélisation procédurale par l-système : Interagir, regénérer et visualiser*. École de technologie supérieure.
- Gagnon, J. & Paquette, E. (2011). Procedural and interactive icicle modeling. *The Visual Computer*, 27(6-8), 451–461.
- Galin, E., Peytavie, A., Maréchal, N. & Guérin, E. (2010). Procedural generation of roads. *Computer Graphics Forum*, 29(2), 429–438.
- Kelly, T. & Wonka, P. (2011). Interactive architectural modeling with procedural extrusions. *ACM Trans. Graph.*, 30(2), 14 :1–15.
- Kolmogorov, V. & Zabih, R. (2001). Computing visual correspondence with occlusions using graph cuts. In *Proceedings of Eighth IEEE International Conference on Computer Vision*, 2, 508–515.
- Krecklau, L. & Kobbelt, L. (2012). Interactive modeling by procedural high-level primitives. *Computers & Graphics*, 36(5), 376–386.
- Lindenmayer, A. (1968). Mathematical models for cellular interactions in development i. filaments with one-sided inputs. *Journal of theoretical biology*, 18(3), 280–299.

- Lintermann, B. & Deussen, O. (1999). Interactive modeling of plants. *IEEE Transactions on Computer Graphics and Applications*, 19(1), 56–65.
- Lipp, M., Wonka, P. & Wimmer, M. (2008). Interactive visual editing of grammars for procedural architecture. *ACM Trans. Graph.*, 27(3), 102 :1–10.
- Liu, Y., Xu, C., Pan, Z. & Pan, Y. (2006). Semantic modeling for ancient architecture of digital heritage. *Computers & Graphics*, 30(5), 800–814.
- Livny, Y., Pirk, S., Cheng, Z., Yan, F., Deussen, O., Cohen-Or, D. & Chen, B. (2011). Texture-lobes for tree modelling. *ACM Trans. Graph.*, 30(4), 53 :1–10.
- Longay, S., Runions, A., Boudon, F. & Prusinkiewicz, P. (2012). Treesketch : Interactive procedural modeling of trees on a tablet. *In Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, pp. 107–120.
- Müller, P., Wonka, P., Haegler, S., Ulmer, A. & Van Gool, L. (2006). Procedural modeling of buildings. *ACM Trans. Graph.*, 25(3), 614–623.
- Müller, P., Zeng, G., Wonka, P. & Van Gool, L. (2007). Image-based procedural modeling of facades. *ACM Trans. Graph.*, 26(3), 85 :1–10.
- Oppenheimer, P. E. (1986). Real time design and animation of fractal plants and trees. *In Proceedings of SIGGRAPH 86, Annual Conference Series*, 20(4), 55–64.
- Palubicki, W., Horel, K., Longay, S., Runions, A., Lane, B., Měch, R. & Prusinkiewicz, P. (2009). Self-organizing tree models for image synthesis. *ACM Trans. Graph.*, 28(3), 58 :1–10.
- Parish, Y. I. H. & Müller, P. (2001). Procedural modeling of cities. *In Proceedings of SIGGRAPH 01, Annual Conference Series*, pp. 301–308.
- Prusinkiewicz, P. (1986). Graphical applications of l-systems. *In Proceedings of Graphics Interface '86*, 86, 247–253.
- Prusinkiewicz, P. & de Reuille, P. B. (2010). Constraints of space in plant development. *Journal of experimental botany*, 61(8), 2117–2129.
- Prusinkiewicz, P., Lindenmayer, A. & Hanan, J. (1990). *The algorithmic beauty of plants*. Springer-Verlag.
- Prusinkiewicz, P., James, M. & Měch, R. (1994). Synthetic topiary. *In Proceedings of SIGGRAPH 94, Annual Conference Series*, pp. 351–358.
- Prusinkiewicz, P., Hammel, M., Hanan, J. & Mech, R. (1996). L-systems : from the theory to visual models of plants. *In Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences*, 3, 1–32.

- Rasmussen, J. L. (1989). Analysis of likert-scale data : A reinterpretation of gregoire and driver. *Psychological Bulletin*, 105(1), 167–170.
- Reeves, W. T. (1983). Particle systems – a technique for modeling a class of fuzzy objects. *ACM Trans. Graph.*, 2(2), 91–108.
- Runions, A., Lane, B. & Prusinkiewicz, P. (2007). Modeling trees with a space colonization algorithm. In *Proceedings of the Third Eurographics conference on Natural Phenomena*, pp. 63–70.
- Smelik, R. M., Tutenel, T., de Kraker, K. J. & Bidarra, R. (2011). A declarative approach to procedural modeling of virtual worlds. *Computers & Graphics*, 35(2), 352–363.
- Smith, A. R. (1984). Plants, fractals, and formal languages. In *Proceedings of SIGGRAPH 84, Annual Conference Series*, 18(3), 1–10.
- Št'ava, O., Beneš, B., Měch, R., Aliaga, D. G. & Krištof, P. (2010). Inverse procedural modeling by automatic generation of l-systems. *Computer Graphics Forum*, 29(2), 665–674.
- Št'ava, O., Kratt, J., Said, M. A. M., Neubert, B., Mech, R., Benes, B. & Deussen, O. (2012). Plastic trees : interactive self-adapting botanical tree models. *ACM Trans. Graph.*, 31(4), 50 :1–10.
- Št'ava, O., Pirk, S., Kratt, J., Chen, B., Mech, R., Deussen, O. & Benes, B. (2014). Inverse procedural modelling of trees. *Computer Graphics Forum*, 33(6), 118–131.
- Stiny, G. & Gips, J. (1971). Shape grammars and the generative specification of painting and sculpture. In *Proceedings of International Federation for Information Processing Congress 71*, 2(3), 1460–1465.
- Talton, J. O., Lou, Y., Lesser, S., Duke, J., Měch, R. & Koltun, V. (2011). Metropolis procedural modeling. *ACM Trans. Graph.*, 30(2), 11 :1–14.
- Watson, B., Müller, P., Veryovka, O., Fuller, A., Wonka, P. & Sexton, C. (2008). Procedural urban modeling in practice. *IEEE Transactions on Computer Graphics and Applications*, 28(3), 18–26.
- Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E. & Quan, L. (2008). Image-based facade modeling. *ACM Trans. Graph.*, 27(5), 161 :1–10.
- Xu, L. & Mould, D. (2012). A procedural method for irregular tree models. *Computers & Graphics*, 36(8), 1036–1047.
- Yong, L., Mingmin, Z., Yunliang, J. & Haiying, Z. (2012). Improving procedural modeling with semantics in digital architectural heritage. *Computers & Graphics*, 36(3), 178–184.