

TABLE DES MATIÈRES

	Page
INTRODUCTION.....	1
CHAPITRE 1 PROBLÉMATIQUE ET REVUE DE LITTÉRATURE.....	4
1.1 Problème	4
1.2 Revue de littérature	5
1.2.1 Extraction.....	5
1.2.1.1 Méthode volumétrique	5
1.2.1.2 Méthode géométrique	7
1.2.2 Génération et adaptation	8
1.2.3 Croquis	10
1.2.3.1 Interface.....	10
1.2.3.2 Gestuelle	10
1.2.3.3 Projection de traits.....	11
CHAPITRE 2 MÉTHODOLOGIE	14
2.1 Identification du problème	14
2.2 Composition d'un squelette	14
2.3 Implémentation dans Blender	15
CHAPITRE 3 PRÉSENTATION DE LA MÉTHODE	16
3.1 Adaptation complète.....	16
3.1.1 Survol.....	16
3.1.2 Extraction du graphe topologique	17
3.1.2.1 Filtrage.....	20
3.1.2.2 Filtrage multirésolution	20
3.1.2.3 Symétrie	22
3.1.2.4 Descripteur de forme	23
3.1.2.5 Regroupement	24
3.1.2.6 Métadonnées de symétrie	26
3.1.3 Adaptation.....	27
3.1.3.1 Correspondance multirésolution	27
3.1.3.2 Placement des joints.....	31
3.1.3.3 Orientation.....	34
3.1.3.4 Contrôleurs	36
3.1.4 Résultats	40
3.2 Création et adaptation partielle par croquis.....	44
3.2.1 Survol.....	44
3.2.1.1 Interface.....	44

3.2.2	Implémentation.....	45
3.2.2.1	Gestuelle	45
3.2.2.2	Surimpression de traits	49
3.2.3	Insertion volumétrique	49
3.2.4	Méthodes de génération.....	53
3.2.4.1	Semi-automatique	53
3.2.4.2	Automatique	54
3.2.5	Orientation.....	56
3.2.6	Patron	57
3.2.6.1	Adaptation	58
3.2.6.2	Orientation	58
3.2.6.3	Patron de nom d'entité	58
3.2.7	Résultats	60
CHAPITRE 4 INTERPRÉTATION DES RÉSULTATS.....		63
4.1	Survol	63
4.2	Correspondance entre fonctionnalités et problèmes.....	63
4.2.1	Adaptation complète	63
4.2.1.1	Multirésolution	63
4.2.1.2	Orientation	65
4.2.1.3	Contrôleurs	66
4.2.1.4	Noms des os	66
4.2.2	Adaptation par croquis.....	67
4.2.2.1	Reconnaissance gestuelle	67
4.2.2.2	Insertion volumétrique	67
4.2.2.3	Génération semi-automatisée	67
4.2.2.4	Adaptation de patron	67
4.2.2.5	Entités de nom	68
4.3	Exemple d'utilisation	68
4.4	Performance.....	68
4.4.1	Adaptation complète	68
4.4.1.1	Graphe du maillage.....	69
4.4.1.2	Graphe du squelette	69
4.4.1.3	Adaptation du squelette au maillage.....	69
4.4.1.4	Temps d'exécution total.....	70
4.4.2	Méthode par croquis	70
4.4.2.1	Tracé du croquis	71
4.4.2.2	Reconnaissance gestuelle	71
4.4.2.3	Adaptation de patron	71
CHAPITRE 5 DISCUSSION.....		72
5.1	Survol	72
5.2	Comparaison avec différentes méthodes.....	72
5.2.1	Génération automatisée	72

5.2.2	Adaptation.....	73
5.3	Évaluation auprès des utilisateurs.....	73
	CONCLUSION.....	77
	ANNEXE I COMPOSITION D'UN SQUELETTE.....	80
	ANNEXE II ÉVALUATIONS PAR DES UTILISATEURS	85

LISTE DES TABLEAUX

	Page
Tableau 4.1	Performance du système d'adaptation 70
Tableau 5.1	Comparaison avec les travaux antérieurs 73
Tableau 5.2	Évaluation des utilisateurs : technique et interface 75
Tableau 5.3	Évaluation des utilisateurs : fonctionnalités 75
Tableau II.1	Fiche d'identification 85
Tableau II.2	Questionnaire d'évaluation 88

LISTE DES FIGURES

		Page
Figure 1	Squelette de déformation	1
Figure 2	Blocs d'animation non-linéaire	2
Figure 3	Contrôleurs	2
Figure 1.1	Comparaison : extraction de squelette	6
Figure 1.2	Squelette humanoïde adapté à un beigne	9
Figure 1.3	Déformation rigide par points de contrôle	13
Figure 3.1	Survol.	16
Figure 3.2	Graphe topologique extrait de Buck	17
Figure 3.3	Graphe de squelette avec décalage entre parent et enfant	18
Figure 3.4	Distribution des poids d'un tore	19
Figure 3.5	Graphe topologique avant filtrage	21
Figure 3.6	Filtrage	22
Figure 3.7	Descripteur de forme d'un sous-arbre	23
Figure 3.8	Détection de symétrie	24
Figure 3.9	Détection de symétrie sur un personnage de cartoon	25
Figure 3.10	Symétrie du graphe topologique d'un papillon	26
Figure 3.11	Correspondances multirésolutions	28
Figure 3.12	Étapes d'une correspondance multirésolution	30
Figure 3.13	Placement de joint sur les points de l'arc	32
Figure 3.14	Réorientation des joints par transformation d'adaptation	35
Figure 3.15	Réorientation des joints par plan de rotation	36

Figure 3.16	Liens des contrôleurs dans un squelette de jambe	38
Figure 3.17	Liens des contrôleurs dans un squelette d'insecte	39
Figure 3.18	Doubles liens vs liens uniques	40
Figure 3.19	Rinky et son squelette adapté.....	41
Figure 3.20	Papillon et mouche : adaptation d'un squelette d'insecte générique	42
Figure 3.21	Mancandy : adaptation complète et partielle	43
Figure 3.22	Composition d'un trait.....	45
Figure 3.23	Geste d'insertion	46
Figure 3.24	Geste de coupure	47
Figure 3.25	Geste d'inversion	47
Figure 3.26	Geste de commandes de correction	48
Figure 3.27	Geste de conversion en squelette	48
Figure 3.28	Geste d'effaçage de trait	49
Figure 3.29	Superposition de traits	50
Figure 3.30	Tracé de rayons pour l'insertion volumétrique	51
Figure 3.31	Insertion par couches	52
Figure 3.32	Insertion par objets.....	52
Figure 3.33	Insertion partiellement obscurcie	53
Figure 3.34	Génération automatique adaptative	55
Figure 3.35	Génération automatique par longueur	56
Figure 3.36	Génération automatique constante	57
Figure 3.37	Orientation d'os par vecteur d'orientation de trait	58
Figure 3.38	Adaptation d'un patron	59
Figure 3.39	Patron de doigt avec flexeur	61

Figure 3.40	Création de squelette d'un robot et d'un tatou.....	62
Figure 4.1	Multirésolution : détails superflus.....	64
Figure 4.2	Fusion des niveaux de résolutions.....	65
Figure 5.1	Une main déformée par enveloppe.....	79
Figure I.1	Un squelette simple	80
Figure I.2	Cinématique directe.....	81
Figure I.3	Cinématique inverse	81
Figure I.4	Contrôle de roulement de pied	82
Figure I.5	Contrôle de flexion.....	83
Figure I.6	Étirement et compression.....	84
Figure II.1	Exemples humanoïdes	86
Figure II.2	Exemples non-humoïdes	87

Rapport-Gratuit.com

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

API	<i>application programming interface</i>
FK	<i>forward kinematics</i>
IK	<i>inverse kinematics</i>
LBS	<i>linear blend skinning</i>

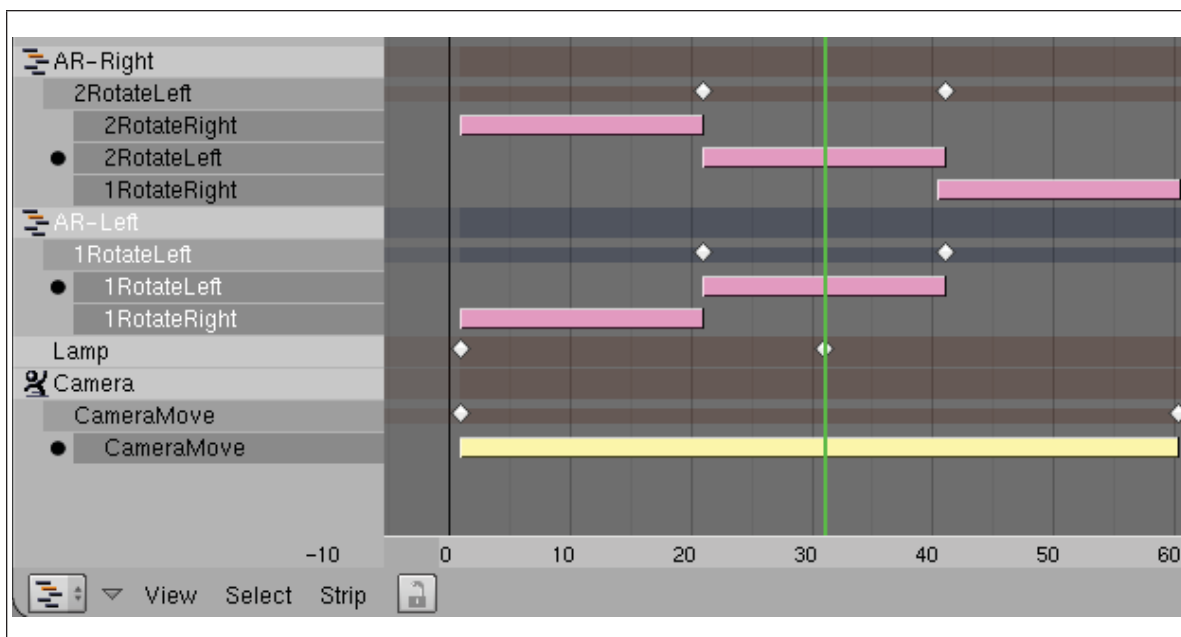


Figure 2 Blender : exemple d'éditeur d'animation non-linéaire.

peuvent travailler plus efficacement sur plusieurs squelettes et peuvent réutiliser des contrôles squelettiques spéciaux (Figure 3) développés sur des projets précédents.

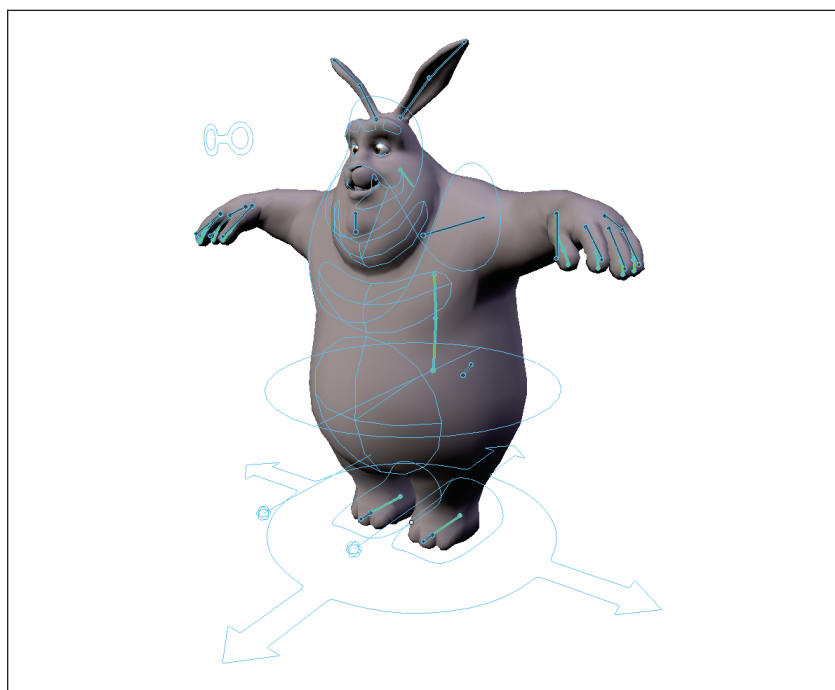


Figure 3 Big Buck Bunny : exemple de contrôleurs.

Adapter automatiquement un squelette existant sur un personnage différent est encore un sujet de recherche relativement jeune et effervescent. Bien que certaines solutions soient capables de traiter des squelettes simples avec peu de joints, ces derniers sont habituellement très loin des squelettes multicouches complexes utilisés par des studios d'animation professionnels. Transférer de tels squelettes manuellement, comme c'est le cas généralement, est une tâche ardue qui, d'après nos discussions avec des professionnels du domaine, peut prendre jusqu'à plusieurs heures.

L'objectif de ce travail de recherche est de développer des techniques permettant une adaptation automatique ou semi-automatique de squelettes préexistants vers un personnage cible dans un milieu professionnel. Le but général étant de réduire le temps nécessaire pour la création ou la réutilisation d'un squelette pour l'animation d'un nouveau personnage. Ceci implique de pouvoir gérer des squelettes assez complexes ainsi que leur hiérarchie de contrôleurs.

De manière générale, nous allons nous limiter aux squelettes autonomes. Ces squelettes seront donc composés uniquement d'os de déformations ou de contrôleurs internes. Cette limite est quand même raisonnable puisque, pour la plupart, les dépendances externes pouvant exister dans un squelette reposent sur la scène où l'animation évolue au lieu de dépendre du personnage lui-même.

CHAPITRE 1

PROBLÉMATIQUE ET REVUE DE LITTÉRATURE

1.1 Problème

Le problème est posé comme suit : Trouver des techniques pour favoriser la réutilisation de squelettes entre différents personnages dans le contexte d'animation 3D de niveau professionnel (composée de squelettes multicouches complexes utilisant des contrôleurs de déformation).

Ce problème est représenté par les objectifs suivants :

- Favoriser et faciliter la réutilisation et l'adaptation de squelettes en totalité ou en partie entre différents personnages.
- Développer une méthode de travail non intrusive qui ne demande pas trop d'informations en dehors de celles déjà présentes dans un squelette standard.

1.2 Revue de littérature

La revue de littérature qui suit permet de donner le cadre théorique de cette recherche en la situant par rapport aux travaux antérieurs connexes, similaires ou précurseurs. Les trois grands sujets traités sont l'extraction de squelettes topologiques, la génération et adaptation de squelettes de déformation et les interfaces de croquis.

1.2.1 Extraction

Extraire un squelette topologique unidimensionnel d'un maillage statique est un problème bien connu pour lequel une grande variété de solutions a été proposée. Ce survol de littérature couvre les travaux les plus représentatifs pour les approches les plus communes. Pour une revue plus en profondeur du domaine, référez vous aux travaux de Cornea et al. (2007).

Les méthodes d'extraction de squelette peuvent être grossièrement séparées en deux grandes catégories dépendant du modèle de représentation spatiale utilisé : un modèle volumétrique basé sur des voxels ou un modèle géométrique basé sur des faces polygonales. La Figure 1.1 compare visuellement les résultats de différentes méthodes.

1.2.1.1 Méthode volumétrique

L'avantage majeur des méthodes basées sur un modèle volumétrique est qu'elles peuvent directement traiter les objets faits de plusieurs surfaces distinctes mais superposées (tel que des vêtements sur un corps). D'un autre côté, des membres minces sont souvent problématiques dû à la limite de taille des voxels. Une limite assez basse pour représenter fidèlement les sections minces d'un objet entraîne des coûts prohibitifs au niveau du temps de calcul. Les algorithmes volumétriques sont habituellement limités aux volumes fermés et tendent à donner des résultats bruités par leur approximation volumétrique. Des approches basées sur une méthode volumétrique utilisent souvent la surface médiane (Wade et Parent, 2000; Baran et Popović, 2007) ou la discrétisation de champs (Brunner et Brunnett, 2008).

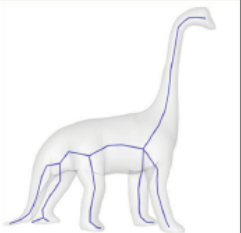
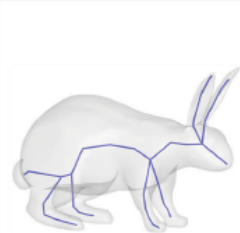

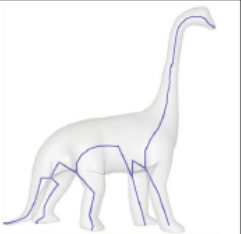
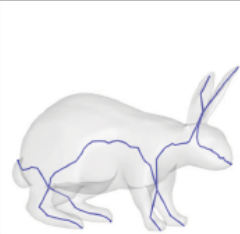

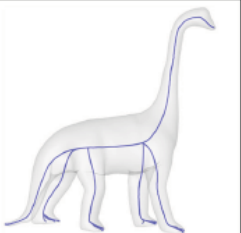
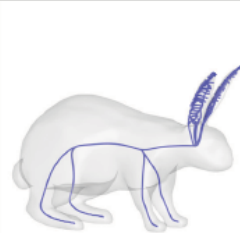


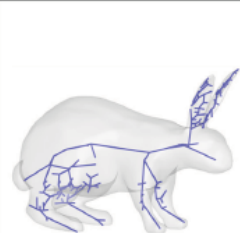

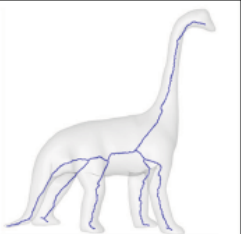
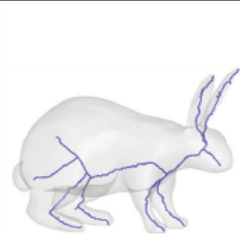
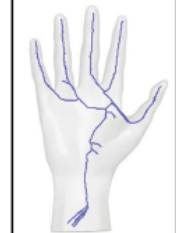
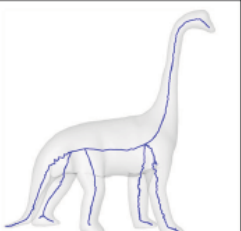


<p>Contraction Laplacien</p> <p>Géométrique</p>			
<p>Graphe de Reeb</p> <p>Géométrique</p>			
<p>Champ de potentiel</p> <p>Volumétrique</p>			
<p>Champ de distance</p> <p>Volumétrique</p>			
<p>Amincissement</p> <p>Volumétrique</p>			
<p>Surface médiane</p> <p>Volumétrique</p>			

Figure 1.1 Comparaison entre différentes méthodes d'extraction de squelette. Figure originale par Au et al. (2008).

1.2.1.2 Méthode géométrique

Les méthodes géométriques, quand à elles, utilisent le maillage de l'objet pour extraire le squelette. De ce fait, elles créent des graphes séparés pour chaque surface distincte. Leurs avantages sont qu'elles sont typiquement plus rapides d'exécution et donnent des résultats plus lisses par rapport aux algorithmes volumétriques. Les algorithmes géométriques peuvent facilement traiter des membres minces sans encourir des coûts additionnels mais produisent parfois de petits arcs supplémentaires qui ne représentent pas des données topologiques importantes et devront être retirés.

Une approche populaire, fortement basée sur la théorie de Morse (Morse, 1932), est de construire un graphe de Reeb (Reeb, 1952) : une structure de données qui représente explicitement la topologie d'une forme. Les graphes de Reeb étant des structures strictement unidimensionnelles, il est important de les combiner avec un algorithme calculant leur projection dans l'espace tridimensionnel. Ceci peut être accompli de différentes façons pendant la génération du graphe (Pascucci et al., 2007), dans une passe de post traitement en utilisant la correspondance entre la géométrie et les arcs du graphe (Aujay et al., 2007) ou encore par la constriction de contours (Tierny et al., 2008), entre autres.

D'autres méthodes, tel que le *fuzzy clustering* (Katz et Tal, 2003), font un usage extensif de l'information géométrique pour premièrement segmenter le maillage en différentes surfaces pour ensuite extraire un squelette à partir de celles-ci.

Une approche plus récente utilisant une méthode géométrique consiste à répéter les contractions de maillage résultant de l'application d'un opérateur de lissage contraint par un Laplacien (Au et al., 2008). Contrairement au graphe de Reeb, cette méthode est peu affectée par le bruit et la déformation de surface et ne dépend pas d'une fonction globale de correspondance $M \rightarrow \mathbb{R}$ entre le maillage et un poids quelconque.

Certaines approches typiquement utilisées avec un modèle volumétrique sont aussi applicables à un modèle géométrique. La fonction de diamètre de volume (Shapira et al., 2008), qui s'apparente à la surface médiane, en est une.

Notre implémentation utilise des graphes de Reeb pour leur simplicité de création. En tant que méthode géométrique, leur habilité à traiter des volumes minces autant que larges est aussi considérée essentielle pour travailler avec des squelettes complexes et fonctionnels.

1.2.2 Génération et adaptation

Générer un squelette de déformation à partir d'un maillage ou en adapter un déjà existant est un sujet relativement bien étudié. Bien que certaines méthodes soient basées sur une définition implicite du squelette, telle que celle utilisée dans le jeu *Spore* (Hecker et al., 2008), cette revue sera limitée aux techniques utilisant un maillage statique en entrée puisqu'elles sont fonctionnellement plus proches de la technique présentée et représentent les cas les plus utiles dans le domaine de l'animation. La génération ou l'adaptation d'un squelette requiert de résoudre plusieurs problèmes similaires (tel que le placement de joints), ainsi, les techniques utilisées sont souvent semblables.

Une méthode récente d'adaptation (Baran et Popović, 2007) est basée sur l'insertion de patron squelettique simple. Les squelettes sont grossiers, sans contrôle sur les membres de plus petites tailles tels que les doigts et sans aucun contrôleur. L'insertion du squelette à l'intérieur du maillage se fait à l'aide de placement de sphère sur la surface médiane de celui-ci. Comme un modèle volumétrique est utilisé, une limite de taille est imposée sur les membres du personnage sous laquelle ils ne sont pas détectés. Par contre, cette technique est bien appropriée dans les cas où un squelette simple est suffisant, tel que l'utilisation par des enfants ou d'autres animateurs amateurs. De plus, la technique ne dépendant pas exclusivement de la topologie du maillage et ainsi permet d'adapter des squelettes à des objets ayant une forme qui ne leur correspond pas directement, comme c'est le cas pour le squelette humanoïde et le beigne à la Figure 1.2.

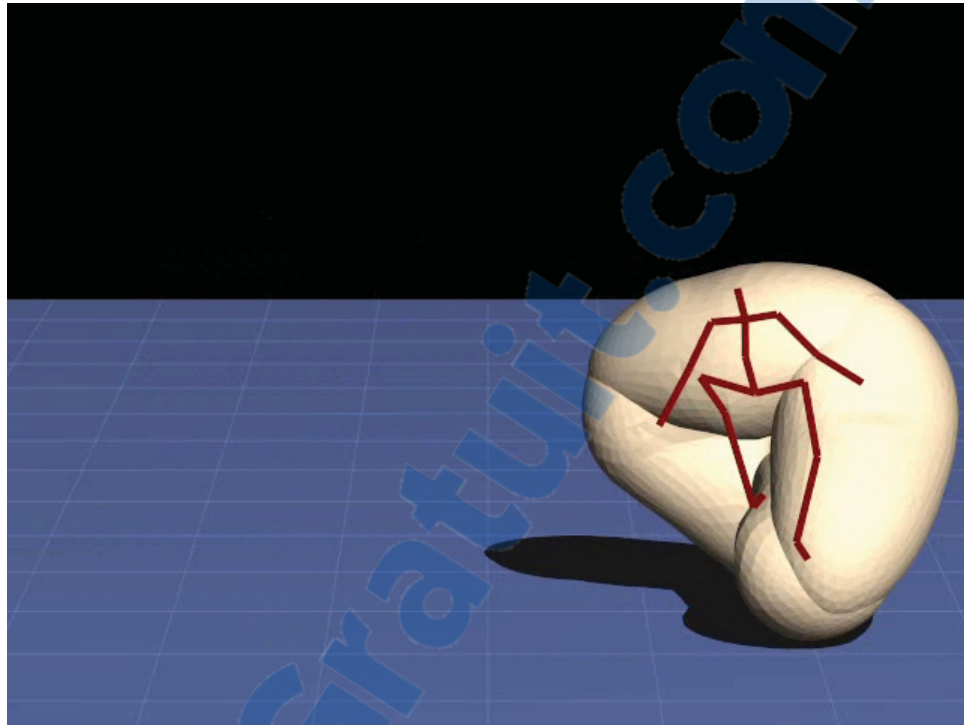


Figure 1.2 Squelette humanoïde adapté à un maillage de beigne. Image provenant du vidéo accompagnant l'article de Baran et Popović (2007).

Certaines méthodes utilisent des connaissances anatomiques (Dellas et al., 2007) pour guider la génération de squelettes fonctionnels. Ces algorithmes sont habituellement limités au squelette humain mais tendent à donner de meilleurs résultats pour ce type particulier de squelette.

D'autres approches (pour ne pas dire la majorité) de génération de squelette de déformation sont basées sur l'extraction du graphe topologique d'un maillage et sur la création d'os sur les arcs résultants. Ces méthodes peuvent être combinées à d'autres pour faire un algorithme spécifique à l'anatomie humaine (Wade et Parent, 2000) ou un algorithme possédant des règles plus génériques pouvant traiter des bipèdes ou des quadrupèdes (Aujay et al., 2007).

Les algorithmes de génération de squelette utilisant des graphes ont habituellement quelques difficultés au niveau de la subdivision automatique d'arcs (pour l'ajout de joints) : genoux et coudes sont typiquement problématiques. Bien que des patrons ou des règles anatomiques peuvent aider à corriger le problème, ils ne le résolvent pas de manière très satisfaisante quand

les proportions des membres sont hors de l'ordinaire (tel que pour un personnage de cartoon, par exemple).

1.2.3 Croquis

Le terme croquis fait référence à un paradigme d'interface personne machine s'éloignant de la traditionnelle combinaison clavier et souris pour se rapprocher le plus possible du *papier et crayon* de tous les jours. Un survol récent du domaine (LaViola, 2007) dresse une liste des avantages et désavantages de ce type d'interface et présente un aperçu des différents problèmes rencontrés lors de l'implémentation d'une telle technique de même qu'une description de certains outils disponibles pour en faire l'implémentation.

Les interfaces de type croquis sont souvent reconnues pour leur affordance familière, pour le sentiment de liberté accru ressenti par leurs utilisateurs ainsi que pour leur expressivité supérieure.

1.2.3.1 Interface

La plupart des interfaces de type croquis se manifestent par la création de traits courbes continus, dont ILoveSketch (Bae et al., 2008), alors que d'autres se limitent à des traits polygonaux, comme l'application de modélisation Chateau (Igarashi et Hughes, 2006).

La superposition, ou surimpression de traits (Bae et al., 2008; Alex, 2005; Baudel, 1994), est aussi utilisée pour permettre aux utilisateurs d'appliquer des correctifs de façon semblable à un dessin avec papier et crayon, en repassant à plusieurs reprises par dessus un trait existant.

1.2.3.2 Gestuelle

Les interfaces de croquis sont souvent accompagnées d'un contrôle par gestuelles. Ces derniers ont fait l'objet de recherches assez complètes depuis plusieurs années, autant au niveau technique qu'au niveau perceptuel.

Du côté théorique, un article récent (Bau et Mackay, 2008) traite du problème de reconnaissance gestuelle par l'aspect d'apprentissage inhérent associé à une interface non-familière. Leur recherche présente un système prédictif et rétroactif permettant une introduction progressive aux différentes fonctionnalités d'un logiciel utilisant ce type de contrôle. Quand à elle, l'application ILoveSketch (Bae et al., 2008) démontre un exemple concret de l'utilisation de gestuelles pour compléter un logiciel de modélisation par croquis.

De notre côté, nous nous sommes limités à un système simple basé sur l'extraction de lignes et des intersections du geste avec les traits existants, sans rétroaction ni prédiction. De telles fonctionnalités pourraient être ajoutées par la suite, mais elles ne se sont pas révélées nécessaires pour rendre les gestuelles utiles.

1.2.3.3 Projection de traits

Dans certaines applications, il peut être utile de projeter un trait dans l'espace tridimensionnel sur un objet ou à l'intérieur de celui-ci. Une technique présentée par Blanco et Oliveira (2008) réalise une insertion des points dans le maillage à l'aide de plans perpendiculaires. Le point d'insertion correspond au centre du contour tracé par les faces intersectant le plan perpendiculaire au trait.

Une autre (Ju et al., 2007) utilise une variation d'une technique classique de rendu par couches pour insérer des traits polygonaux à l'intérieur d'un maillage. Ces traits serviront ensuite à modifier la topologie de ce dernier.

Notre méthode utilise une technique de séparation en couches par tracer de rayons, une implémentation traditionnelle sur le GPU ne donnant pas la précision voulue.

1.2.3.3.1 Pression, inclinaison et roulement

En plus de la simple position du pointeur, une tablette à dessin électronique donne accès à un éventail de données beaucoup plus large qu'une simple souris. De ces informations additionnelles, la pression et l'inclinaison sont les plus communes et les plus étudiées, comme

l'explique une analyse (Zhou et al., 2008) de la précision et du contrôle qu'un utilisateur peut exercer sur la pression et l'inclinaison de son stylet. L'utilisation du roulement (Bi et al., 2008) est un peu plus rare, mais non moins utile.

Par contre, de telles méthodes de contrôle demandent de l'équipement spécial qui n'est pas à la portée de tous. C'est pourquoi notre méthode ne requiert directement aucunes de ces informations propres aux tablettes dans le but de rester compatible avec une simple souris et ainsi rejoindre une plus grande audience. Néanmoins, l'utilisation de l'inclinaison a été considérée pour compléter la technique d'orientation de joints présentée à la Section 3.2.5.

1.2.3.3.2 Modélisation

Les interfaces de croquis se prêtent naturellement aux tâches de modélisation, la similitude avec des croquis exécutés sur papier étant très forte. Cette ressemblance est très frappante dans le cas de ILoveSketch (Bae et al., 2008). D'un autre côté, Château (Igarashi et Hughes, 2006), en se limitant aux traits polygonaux, ressemble plus à la création de plans architecturaux qu'à des dessins à main levée. Finalement, il est impossible de parler de modélisation par croquis sans mentionner Teddy (Igarashi et al., 1999), une des applications de modélisation par croquis les plus connus, ni de Sketchpad (Sutherland, 1963), l'ancêtre de toutes celles-ci.

1.2.3.3.3 Déformation et animation

Des interfaces de croquis sont aussi utilisées en recherche pour des systèmes d'animation et de déformation.

Parmi les systèmes de déformations, certains (Kho et Garland, 2005; Blanco et Oliveira, 2008) appliquent une déformation de la manière suivante :

- Projection d'une courbe sur le maillage
- Déformation à l'aide d'une deuxième courbe, la différence avec la première courbe est propagée au maillage sous-jacent.

Un autre (Zimmermann et al., 2007) déforme un maillage à partir de sa silhouette, extraite en espace écran, et d'une deuxième courbe tracée par l'utilisateur alors que la technique présentée par Igarashi et al. (2005) et illustrée à la Figure 1.3 utilise une cage de déformation interne rigide pour propager la transformation de points de contrôle vers le reste de l'objet.

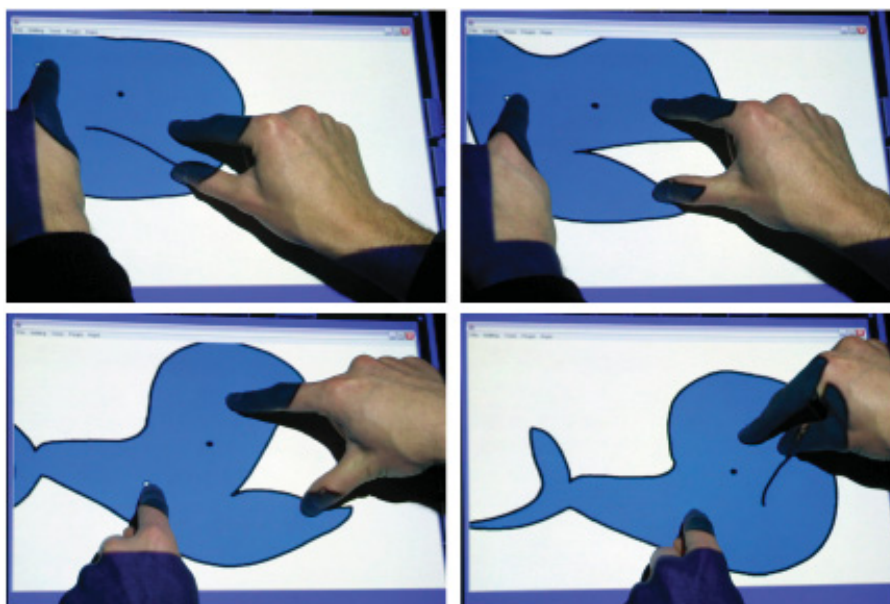


Figure 1.3 “As-rigid-as-possible shape manipulation”.
Image provenant de l'article de Igarashi et al. (2005)

L'approche (Li et al., 2007) la plus semblable à la nôtre utilise une projection volumétrique à l'intérieur du maillage cible du croquis fait par l'utilisateur. Leur technique diverge ensuite puisqu'elle se sert d'un squelette non-rigide alors que la nôtre génère ou adapte un squelette rigide traditionnel.

Le problème avec toutes ces techniques est qu'elles sont assez loin de celles traditionnellement utilisées en industrie. Bien que certaines, comme les splines d'IK, s'en approchent la majeure partie de ces dernières est composée de squelettes hiérarchiques traditionnels. De plus, ces techniques sont pour la plupart de simples techniques de déformation, elles ne prennent pas en compte les variations de temps et ne sont pas bien intégrées aux systèmes d'animation existants. Ceci rend donc les différentes techniques de déformation par croquis difficilement applicables au contexte de productions professionnelles d'animation.

CHAPITRE 2

MÉTHODOLOGIE

2.1 Identification du problème

L'identification du problème s'est faite à travers plusieurs conversations avec des professionnels de l'industrie de l'animation ainsi que par des discussions entre utilisateurs experts et développeurs du logiciel Blender (Blender-Foundation, 2009), le tout sur une période de plusieurs mois. Durant ces discussions sur l'animation de personnage en générale, la réutilisation est un sujet qui revenait avec persistance. Bien que ce soit un problème assez bien défini et étudié en ce qui a trait à la réutilisation de blocs d'animation, le problème d'adaptation et de réutilisation de squelettes entre différents personnages similaires était un peu nébuleux, jugé par plusieurs plus facile à laisser résoudre manuellement par un expert.

2.2 Composition d'un squelette

Tout d'abord, il est important de définir ce qui compose un squelette. Cette section consiste en un survol assez large de la question, avec suffisamment de détails pour permettre une compréhension des sections à venir. Pour des explications plus en profondeur, voir l'Annexe I.

Un squelette est largement défini comme une hiérarchie de joints et d'os. Lorsqu'un os bouge, ses enfants hiérarchiques suivent. Les os ont une orientation et une taille, chaque chaîne d'os correspondant grosso modo à un membre.

Pour la suite, les os d'un squelette seront catégorisés selon leur fonction :

Déformation : Ces os définissent la déformation de l'objet dans le temps et l'espace. Différentes méthodes de transfert entre transformation des os et déformation d'un maillage existent, la plus courante étant le *Linear Blend Skinning* (LBS) (Magenat-Thalmann et al., 1988) qui assigne à chaque sommet un poids correspondant à chaque os, le résultat final étant la somme pondérée normalisée des transformations.

Contrôle : Les os de contrôle ont deux grands rôles : ils peuvent servir à diriger la transformation d'autres os (contrôleurs eux-mêmes ou de déformation) ou bien ils peuvent avoir un rôle purement hiérarchique (définir un parent commun à deux chaînes d'os de déformation, par exemple).

2.3 Implémentation dans Blender

Les techniques présentées dans ce document ont été implémentées dans le logiciel Blender 3D (Blender-Foundation, 2009), une suite complète de modélisation, animation, rendu, montage et simulation temps réel 3D. Blender est sous licence GPL (Free Software Foundation, 2007).

Ce choix offre des avantages certains :

- Accès au code source : L'implémentation n'est pas limitée à un API d'extension.
- Fonctionnalités préexistantes : Par rapport à une implémentation autonome, le logiciel est déjà très riche en fonctionnalité, ce qui permet de reproduire complètement le pipeline d'un utilisateur typique.
- Expériences préalables : L'auteur de ce mémoire est un utilisateur du logiciel depuis 2001 et un développeur depuis 2003.
- Réseaux : Contacts préalables avec des utilisateurs professionnels.
- Portabilité : Disponible sur plusieurs plateformes : Windows, OS X, Linux, Solaris, Irix, ...

D'un autre côté, ce logiciel est moins utilisé en industrie, ce qui rend la tâche de trouver des utilisateurs experts professionnels un peu plus difficile.

CHAPITRE 3

PRÉSENTATION DE LA MÉTHODE

3.1 Adaptation complète

La première technique présentée est utilisée pour faire une adaptation complète d'un squelette vers un maillage cible. Ainsi, elle permet de directement réutiliser un squelette existant sur un nouveau personnage. Cette technique a été publiée dans le cadre de la conférence *Graphics Interface 2009* (Poirier et Paquette, 2009).

3.1.1 Survol

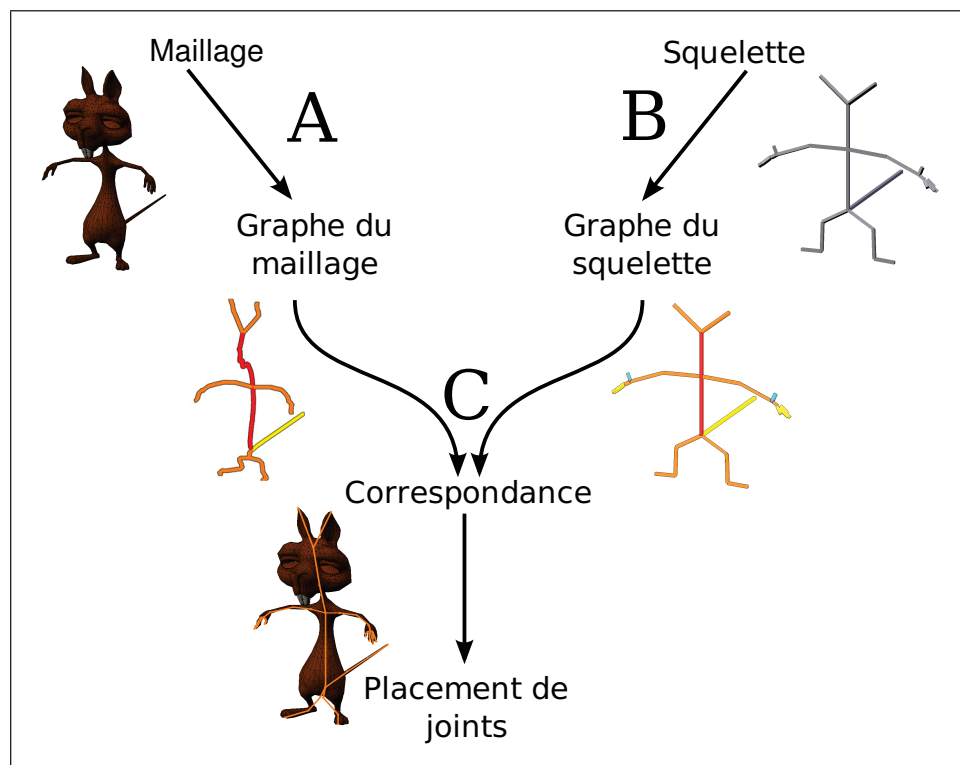


Figure 3.1 Survol.

Voici un résumé de l'algorithme présenté, aussi illustré dans la Figure 3.1.

A. Extraire le graphe topologique multirésolution du maillage

- Générer un graphe topologique à partir du maillage
 - Filtrer et construire la structure multirésolution
 - Détecter et marquer les métadonnées de symétrie
- B. Extraire le graphe du squelette
- Générer un graphe topologique à partir du squelette
 - Détecter et marquer les symétries
 - Lier les contrôleurs au graphe
- C. Adapter le squelette au maillage
- Faire correspondre les arcs entre le graphe du squelette et le graphe multirésolution du maillage
 - Positionner les joints internes, équilibrés entre la forme du maillage et la forme original du squelette
 - Propager la transformation du squelette vers les contrôleurs qui y sont liés

3.1.2 Extraction du graphe topologique

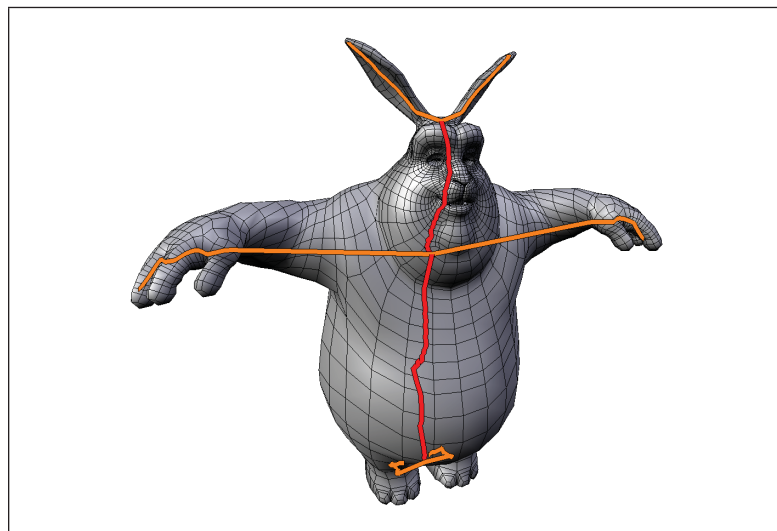


Figure 3.2 Graphe topologique extrait de Buck, le lapin du court métrage *Big Buck Bunny* (2008).

Générer un graphe topologique unidimensionnel à partir du squelette de déformation reçu en entrée est très simple puisque celui-ci est déjà un graphe. Néanmoins, le graphe de maillage et celui du squelette devront quand même partager un noeud racine qui sera utilisé pour démarrer la méthode de correspondance. L'utilisateur identifie ce noeud sur le maillage par la sélection d'un sommet et sur le squelette par la sélection d'un joint. Pour la suite du document, ce noeud sera appelé le *noeud de tête*. Celui-ci doit se trouver sur l'axe de symétrie principale puisqu'il servira à des fins d'identification de symétrie et autre à plusieurs étapes subséquentes. Le noeud de tête est typiquement sélectionné sur la tête du personnage.

Du côté du squelette, le seul traitement direct requis consiste à convertir en arcs de graphe les décalages entre joints pouvant exister entre un os et son parent. Les contrôleurs (Section 3.1.3.4) sont aussi liés aux os de déformation à cette étape de l'algorithme.

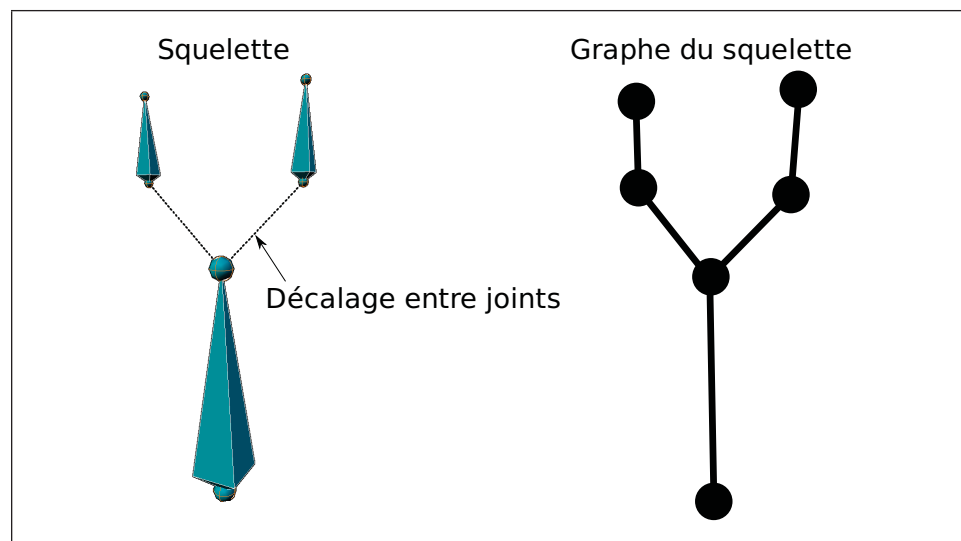


Figure 3.3 Graphe de squelette avec décalage entre parent et enfant.

Générer le graphe topologique à partir du maillage 3D est beaucoup plus complexe. Une façon de faire est de contracter le maillage sur lui-même, remplaçant les membres tridimensionnels (doigts, oreilles, bras, jambes, queue, torse, . . .) par des courbes qui suivent la géométrie. Dans le but d'extraire ce graphe, la méthode proposée calcule un graphe de Reeb. Pour la création du graphe, l'utilisateur sélectionne tout d'abord un sommet qui correspond au noeud de tête. À partir de ce dernier, des poids sont calculés pour chaque noeud du graphe en utilisant une fonc-

tion harmonique basée sur les distances géodésiques aux extrémités (Aujay et al., 2007). La figure 3.4 est un exemple de distribution de poids sur un tore, chaque anneau représentant une valeur unique. L'algorithme de génération de graphe est inspiré de Pascucci et al. (2007) avec des modifications mineures pour mieux traiter les maillages de plus faible résolution (leur algorithme est mieux adapté aux résultats de numérisation 3D avec une densité de polygones très élevée). Le graphe topologique est projeté dans l'espace tridimensionnel le long du maillage contracté. Les Figures 3.1 et 3.2 montrent des exemples de graphes de Reeb et leur projection.

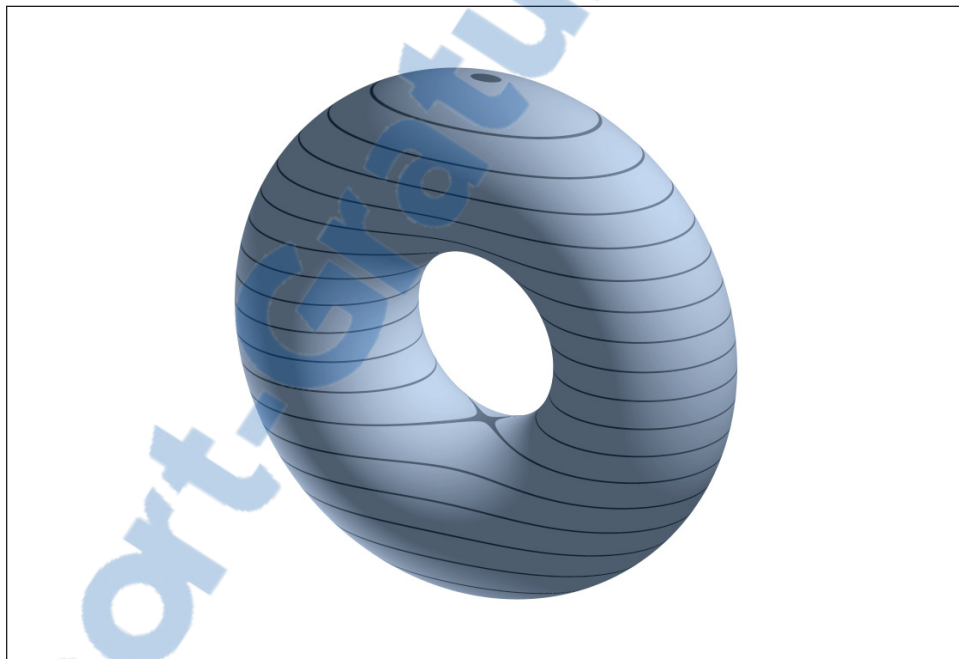


Figure 3.4 Illustration de la distribution des poids pour le graphe de Reeb d'un tore. Dans ce cas simple, le poids correspond à la coordonnée sur l'axe des Z de chaque sommet.

L'algorithme d'adaptation ne dépend pas directement des propriétés des graphes de Reeb, un autre algorithme de graphe topologique qui préserve correctement tous les membres d'un personnage serait également utilisable. Ainsi, ce procédé ne sera pas expliqué en détails puisque les références indiquées sont suffisantes à l'implémentation de cette partie de la méthode.

3.1.2.1 Filtrage

Le graphe topologique du maillage possède typiquement trop d'arcs pour l'adaptation du squelette, puisque des arcs sont générés pour la plupart des caractéristiques topologiques, petites ou grandes, comme on peut le voir à la Figure 3.5. Il sera donc filtré pour retirer les arcs représentant des caractéristiques indésirables ou non requises. Cette étape est seulement applicable au graphe du maillage, puisque l'adaptation a besoin de tous les noeuds et arcs du graphe du squelette de déformation. Comme dans plusieurs méthodes existantes de filtrage de graphes (Aujay et al., 2007), il est important de retirer les arcs les plus petits d'abord pour préserver le plus possible les formes importantes du maillage. L'attribut de filtrage utilisé est la longueur des arcs le long de leur projection 3D ; les arcs qui sont plus courts qu'une limite contrôlable sont retirés. Bien que différents attributs auraient pu être utilisés, tel que la différence de poids entre les noeuds des arcs, il a été déterminé expérimentalement que l'utilisation de la longueur des arcs donne un filtrage progressif qui suit mieux l'importance spatiale du maillage, spécialement en ce qui a trait aux arcs qui courbent dans l'espace 3D. Cette dernière propriété du filtrage par longueur est importante pour la prochaine étape de l'algorithme.

3.1.2.2 Filtrage multirésolution

Le filtrage multirésolution retire l'obligation d'avoir à modifier une limite de filtrage globale jusqu'à ce que la forme du graphe corresponde au squelette. Plus important encore, il permet à l'algorithme de correspondance de traiter avec succès des cas où une seule limite de filtrage donnerait des résultats incorrects. Cette situation est discutée plus en profondeur à la Section 3.1.3.1.

La technique utilisée ici est différente du graphe de Reeb multirésolution présenté par Hilaga et al. (2001) et de la hiérarchie de persistance de Pascucci et al. (2007) sur un point majeur : l'interconnexion des membres d'un personnage requiert que les noeuds du graphe, présents à plus d'un niveau de résolution, restent fixes dans l'espace 3D.

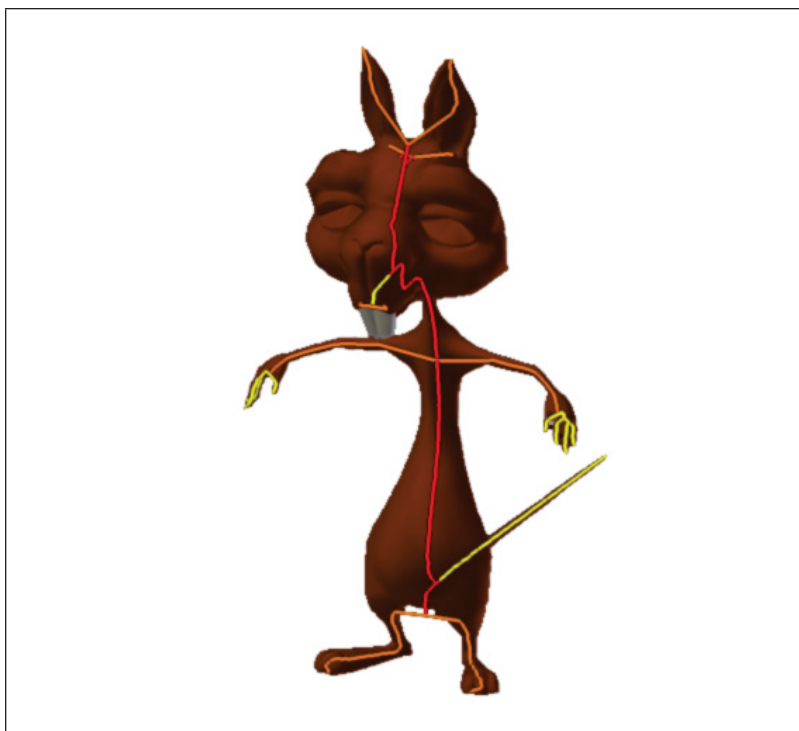


Figure 3.5 Graphe topologique de l'écureuil avant filtrage.

Cette restriction sur le déplacement des noeuds est conservée par la méthode de fusion et élagage suivante, illustrée à la Figure 3.6.

- Les arcs internes (en rouge) sont fusionnés au noeud le plus près du *noeud de tête*.
- Les arcs externes (en vert) sont élagués.
- Les arcs connectés entre eux par un noeud de degré 2 (en orange) sont fusionnés pour conserver la taille du graphe.

Le filtrage multirésolution est accompli par copies successives du graphe de Reeb haute résolution initiale. Par la suite, chaque niveau de résolution est filtré avec une limite sur la longueur des arcs augmentant progressivement, résultant en un graphe grossièrement détaillé au niveau le plus bas alors que chaque niveau additionnel ajoute de plus en plus de détails. Les noeuds et arcs sont liés à leurs homologues de niveau de résolution inférieur et supérieur pour faciliter la navigation entre les différents niveaux du graphe.

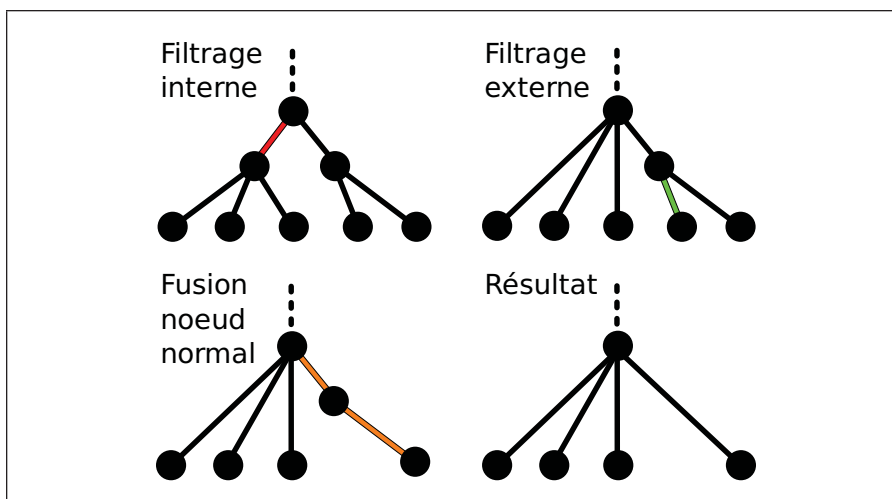


Figure 3.6 Filtrage des arcs internes et externes d'un sous-arbre.

La limite de filtrage varie linéairement entre zéro (aucun filtrage) et une valeur sélectionnée par l'utilisateur. Une valeur typique de 10% de la taille du graphe a été déterminée expérimentalement. L'implémentation utilise un nombre fixe (10) de niveau de résolution qui a été suffisant pour tous les cas de tests présentés.

3.1.2.3 Symétrie

La détection de symétrie est applicable autant au graphe du maillage qu'à celui du squelette. Pour celui du maillage, cette étape est réalisée indépendamment à chaque niveau de résolution. L'approche proposée est inspirée par la comparaison simple de profondeur de sous-arbre et de degré de noeud (Aujay et al., 2007), mais est rendue plus robuste par la comparaison partielle de la forme du sous-arbre à l'aide d'un descripteur de forme (Section 3.1.2.4). Comme pour leur méthode, l'utilisateur doit indiquer un noeud de tête qui repose sur l'axe de symétrie principale. La méthode proposée n'est pas trop sensible au positionnement de ce noeud, mais il doit être grossièrement positionné sur l'axe de symétrie géométrique du personnage pour que l'algorithme identifie correctement les symétries géométriques. Pour un personnage humanoïde, tous les noeuds du dessus de la tête sont habituellement adéquats pour cette fin. Après la sélection du noeud de tête, l'algorithme descend récursivement, groupant les sous-arbres de forme similaire en groupes de symétrie, continuant récursivement sur tous les noeuds enfants. Un *ni-*

veau de symétrie est aussi assigné à chaque arc, ce niveau correspondant à leur éloignement le long du graphe par rapport à l'axe de symétrie principale ; 1 étant sur l'axe principal, la valeur augmentant pour chaque symétrie secondaire qui doit être traversée pour rejoindre l'axe principal. Cette méthode simple est basée sur les suppositions suivantes : (a) le graphe est en fait un arbre, ainsi tous les arcs sont orientés en s'éloignant du *noeud de tête* et (b) le descripteur de forme de deux sous-arbres symétriques est équivalent. De telles restrictions sont faciles à imposer et sont monnaie courante en industrie.

3.1.2.4 Descripteur de forme

Pour que la détection de symétrie et la correspondance avec le squelette soient efficaces et robustes, un *descripteur de forme* est proposé. Le descripteur de forme d'un sous-arbre est défini comme le nombre de noeuds à chaque niveau de profondeur à partir de la racine du sous-arbre. Un noeud avec deux enfants a donc comme descripteur de forme $\{1, 2\}$ (Figure 3.7 à gauche). Si un des enfants en a aussi deux lui-même alors que l'autre en a trois, le descripteur devient $\{1, 2, 5\}$ (Figure 3.7 à droite). Le descripteur de forme est défini pour un noeud mais aussi pour un arc. Dans ce cas, il est égal au descripteur de son noeud le plus bas (en s'éloignant de la racine de l'arbre).

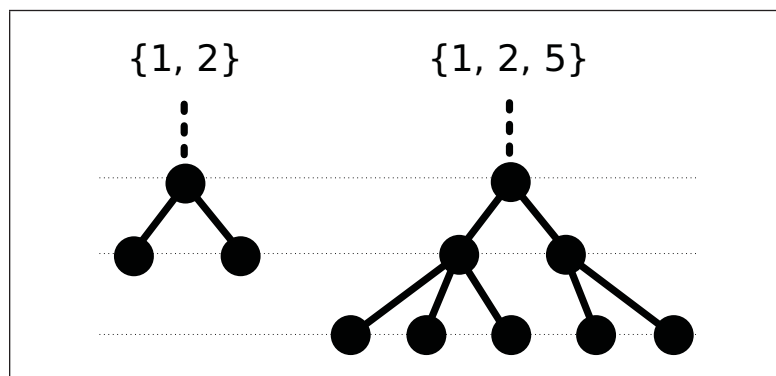


Figure 3.7 Descripteur de forme d'un sous-arbre.

Bien que cette méthode ne fasse aucune distinction en ce qui a trait à comment les enfants les plus loins sont connectés à la racine, elle s'est montrée assez robuste dans tous les cas d'utilisation. De plus, son utilisation est équivalente en complexité de calcul à la méthode de

profondeur et degré (Aujay et al., 2007) tout en étant un meilleur indicateur de similarité. La Figure 3.8 illustre un cas où le descripteur de forme identifie correctement une symétrie alors que la méthode de comparaison de profondeur et degré échoue. La Figure 3.9 montre l'application du descripteur de forme pour une détection de symétrie sur un personnage de cartoon.

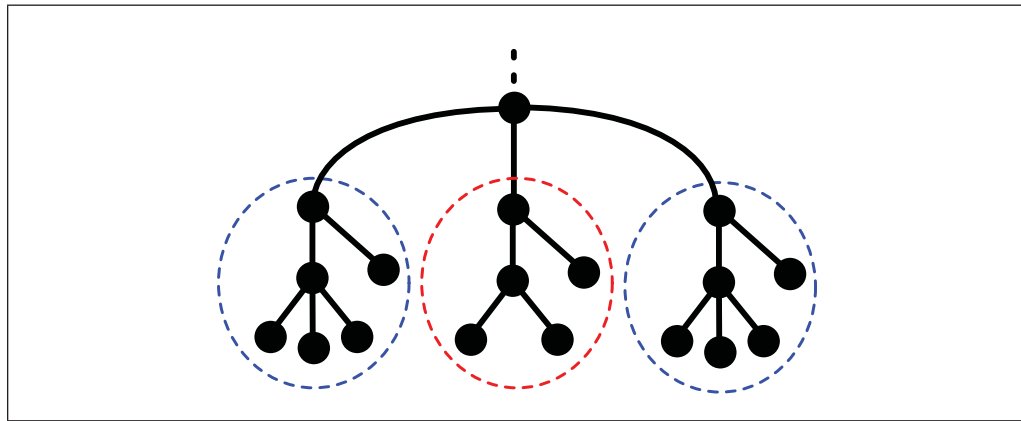


Figure 3.8 Détection de symétrie par comparaison de descripteur de forme. Les deux sous-arbres de gauche et de droite ont la même forme $\{1, 2, 3\}$ alors que celui du milieu est seul avec $\{1, 2, 2\}$. En utilisant la méthode de profondeur et degré, les trois sous-arbres auraient une profondeur de 3 et un degré de 2.

3.1.2.5 Regroupement

Comme le descripteur de forme se limite aux symétries topologiques, notre implémentation resubdivise les groupes d'arcs symétriques à l'aide d'une limite C de variation de longueur. Des sous-groupes sont créés pour s'assurer que la différence de longueur entre l'arc le plus grand et le plus court est moins de C . Ceci permet aux différents arcs ayant des descripteurs de formes semblables, mais de longueurs différentes, d'être correctement séparés en groupes symétriques, tel qu'illustré à la Figure 3.10.

Le regroupement par la longueur pourrait aussi être amélioré par une méthode combinatoire de tests de symétrie axiale et radiale. Si un groupe de sous-arbres topologiquement symétriques n'est pas géométriquement symétrique, les différents sous-groupes possibles seraient testés

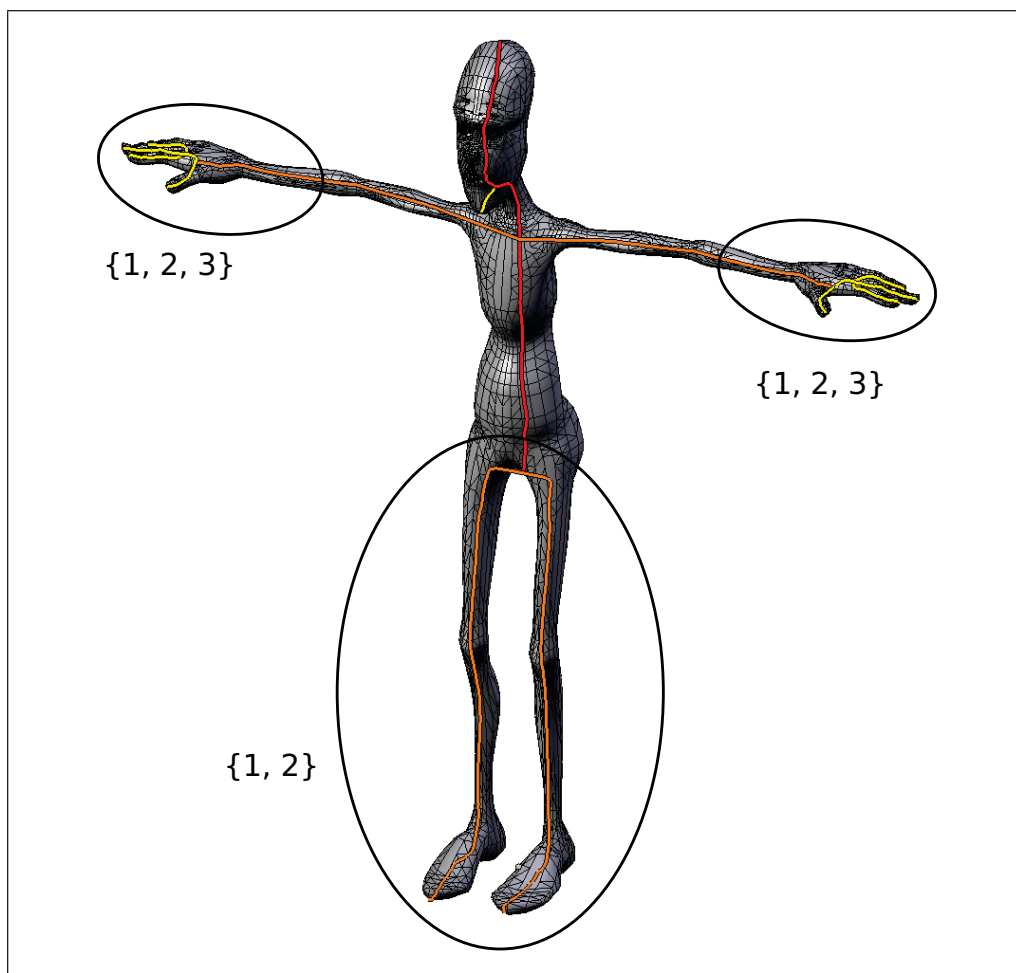


Figure 3.9 Détection de symétrie par comparaison de descripteur de forme sur personnage de cartoon. Si le personnage avait eu des orteils, la méthode de profondeur et degré n'aurait pas détecté la symétrie correctement.

pour une symétrie axiale ou radiale possible. Cette méthode serait utile, par exemple, dans le cas où la queue et les jambes d'un personnage sont de la même longueur ; seules les jambes vont partager une symétrie axiale géométrique. Néanmoins, le regroupement par longueur simple a été suffisant pour correctement subdiviser les groupes de symétrie topologique en symétrie géométrique dans tous les cas de tests.

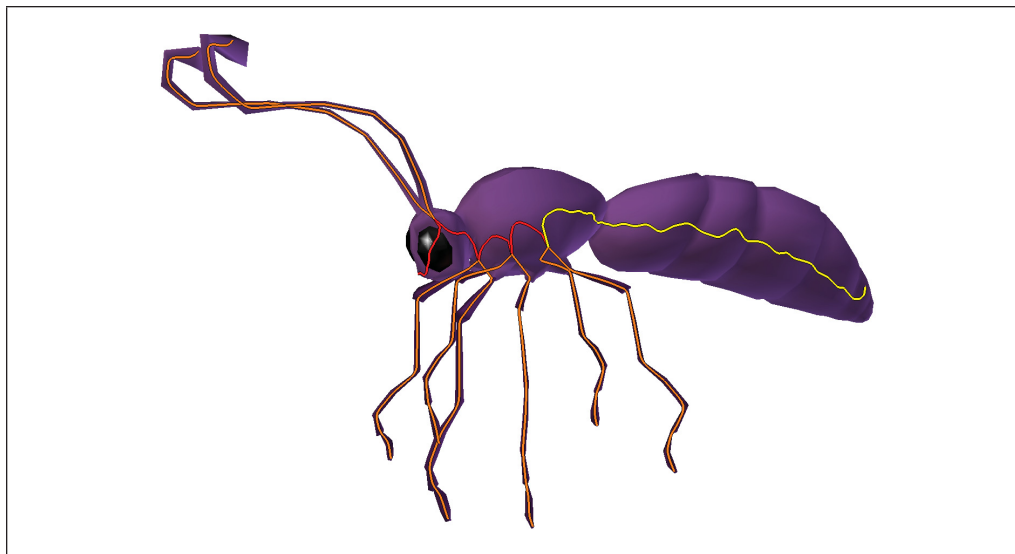


Figure 3.10 Papillon (en mauve) et son graphe topologique. Les deux pattes postérieures et la queue ont le même descripteur de forme, mais, puisque la différence de longueur est plus grande que la limite, les deux pattes sont groupées en symétrie axiale alors que la queue est laissée par elle-même.

3.1.2.6 Métadonnées de symétrie

Pour une adaptation finale plus stable et consistante, l'approche proposée introduit le concept de métadonnées de symétrie. Les arcs faisant partie d'une symétrie axiale sont automatiquement marqués comme *droite* et *gauche* alors que ceux faisant partie d'un groupe de symétrie radiale sont triés par longueur et numérotés. Le type de symétrie est aussi exprimé comme une métadonnée comme le démontre la Figure 3.10 : *symétrie primaire* en rouge, *symétries axiales* en orange et *aucune symétrie* en jaune. Ces métadonnées seront utilisées durant le procédé d'adaptation pour aider à garder la correspondance avec des blocs d'animation existants. La métadonnée d'orientation (droite ou gauche) est calculée à partir de l'axe principal (X, Y ou Z) le plus près du vecteur reliant deux joints pour former un membre. Le membre du côté positif de l'axe est marqué comme *gauche* alors que celui du côté négatif est marqué comme *droite*. Le côté ainsi assigné peut ne pas correspondre à sa position réelle, mais ceci importe peu tant que ce côté est consistant entre le squelette source et le maillage destination. De cette façon, les jambes, bras, oreilles, ... du maillage et du squelette seront positionnés de manière consistante et appropriée. Toutefois, si l'orientation globale du maillage et du squelette est inversée (par

exemple, si le maillage fait face à l'axe des Y alors que le squelette pointe vers -Y), les données d'orientation le seront aussi, puisque notre implémentation n'assume rien par rapport à l'orientation originale des personnages. Si besoin est, l'utilisateur pourrait donner cette orientation sous la forme de l'axe vers lequel pointe le personnage, mais il est relativement aussi simple de demander aux utilisateurs de s'assurer que le maillage et le squelette aient grosso modo la même orientation.

3.1.3 Adaptation

Pour résoudre le problème d'adaptation, nous proposons une méthode de correspondance multirésolution couplée à une méthode de positionnement de joints. Comparativement aux méthodes précédentes d'adaptation de squelette (principalement Baran et Popović (2007)), la méthode proposée est plus robuste. De plus, ces premières sont restreintes aux squelettes de déformation alors que la méthode proposée prend aussi en compte les contrôleurs. Les Sections 3.1.3.1 à 3.1.3.3 décrivent la méthode robuste d'adaptation de squelettes de déformation alors que la Section 3.1.3.4 traite de l'adaptation des contrôleurs.

3.1.3.1 Correspondance multirésolution

La première étape du processus d'adaptation est d'établir une correspondance entre les arcs du graphe du squelette et ceux du graphe multirésolution du maillage. La topologie du graphe du maillage contient typiquement plusieurs arcs représentant des détails superflus non présents dans le squelette qui doit être adapté. Tel que mentionné à la Section 3.1.2.2, identifier les arcs utiles est une tâche difficile. Par exemple, adapter les squelettes de formes différentes de la Figure 3.11 requiert que la méthode de correspondance garde un sous-ensemble des arcs du maillage différent à chaque fois. La Section 4.2.1.1 présente d'autres cas où une correspondance multirésolution est requise. Le troisième squelette de la Figure 3.11 est un exemple où la méthode doit garder les petits arcs à la base de la forme en "T" tout en retirant ceux plus grands au sommet de la forme.

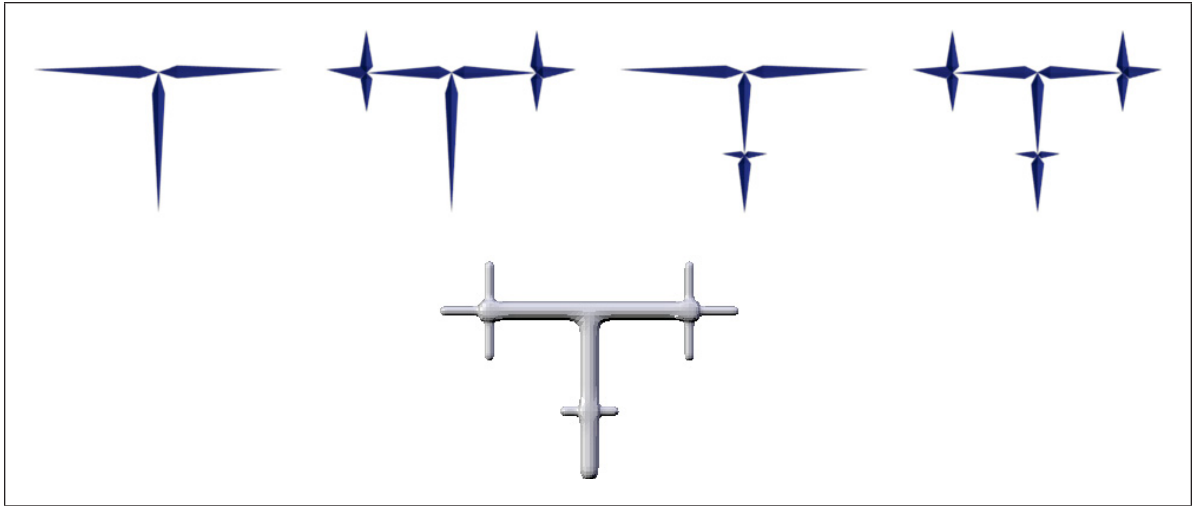


Figure 3.11 La correspondance multirésolution est efficace et versatile. Différents squelettes (en haut) ont tous été adaptés au même maillage (en bas).

L'algorithme de correspondance compare les arcs en utilisant leur niveau de symétrie, leurs métadonnées ainsi que leur descripteur de forme. Alors que le descripteur de forme complet était utilisé pour la détection de symétrie, la correspondance n'utilise que les deux premiers nombres du descripteur. Ceci est appelé le *descripteur de forme local* (*FormeL* dans l'Algorithme 1) et donne l'information requise pour faire correspondre un arc de squelette à l'arc du maillage correspondant.

Voici un résumé de cet algorithme :

– **En partant du noeud de tête (en bleu)**

- 1 Pour tous les arcs connectés au noeud courant
 - 1.1 Faire correspondre un arc du squelette à un arc (non-utilisé) du maillage en utilisant les niveaux et métadonnées de symétrie.
 - 1.2 Parcourir la hiérarchie multirésolution de l'arc du maillage à des niveaux de plus en plus fins dans le but de trouver un descripteur de forme local équivalent.
 - 1.3 Le prochain noeud est le suivant le long de l'arc.

- 1.4 Parcourir la hiérarchie multirésolution du noeud du maillage à des niveaux de plus en plus grossiers pour trouver le niveau le moins détaillé qui a un descripteur de forme local équivalent.
- 1.5 Récursivement établir la correspondance à partir de ce noeud (étape 1).

L'algorithme 1 présente la méthode de correspondance en détails alors que la Figure 3.12 en présente un exemple.

Algorithme 1 : Correspondance($\text{noeud}_M, \text{noeud}_S, \text{arc}_S$)

```

/* Soit un arc du squelette ( $\text{arc}_S$ ) connecté à un noeud ( $\text{noeud}_S$ ), trouver un arc du maillage
( $\text{arc}_M$ ) correspondant connecté à un noeud ( $\text{noeud}_M$ ). */
foreach  $\text{arc}_M$  inutilisé do
  if  $\text{SymMéta}(\text{arc}_M) = \text{SymMéta}(\text{arc}_S)$  then
    /* correspondance de niveau de symétrie et de métadonnées. */
    while  $\text{FormeL}(\text{arc}_S) \neq \text{FormeL}(\text{arc}_M)$  do
      |  $\text{arc}_M \leftarrow \text{arc}_M$  à résolution plus fine
    if  $\text{FormeL}(\text{arc}_S) = \text{FormeL}(\text{arc}_M)$  then
      /* correspondance de niveau de symétrie et de métadonnées + descripteur local de
      forme. */
      noeud $_M \leftarrow$  nouveau noeud connecté par  $\text{arc}_M$ 
      noeud $_S \leftarrow$  nouveau noeud connecté par  $\text{arc}_S$ 
      marquer  $\text{arc}_M$  (à toutes résolutions) et  $\text{arc}_S$  comme utilisés
      noeud $_{CM} \leftarrow$  noeud $_M$  à résolution plus grossière
      while  $\text{FormeL}(\text{noeud}_S) = \text{FormeL}(\text{noeud}_{CM})$  do
        | noeud $_M \leftarrow$  noeud $_{CM}$ 
        | noeud $_{CM} \leftarrow$  noeud $_M$  à résolution plus grossière
      foreach  $\text{arc}_S$  inutilisé connecté au noeud $_S$  do
        | Correspondance( $\text{noeud}_M, \text{noeud}_S, \text{arc}_S$ )

```

L'utilisation du descripteur de forme local permet de faire correspondre différents niveaux de résolution du graphe topologique du maillage pour chaque arc du graphe de squelette. Tel que mentionné précédemment, dans le troisième exemple de la Figure 3.11, les deux plus petits arcs du bas doivent être conservés alors que les plus longs du haut doivent être retirés. C'est un cas où une limite unique de filtrage ne donnerait pas un résultat acceptable alors que la correspondance multirésolution le gère correctement. La fonction *Correspondance* est premièrement appelée avec, comme paramètres, les noeuds de tête préalablement identifiés par l'utilisateur sur le squelette et le maillage et poursuivra aux noeuds connectés à ces derniers.

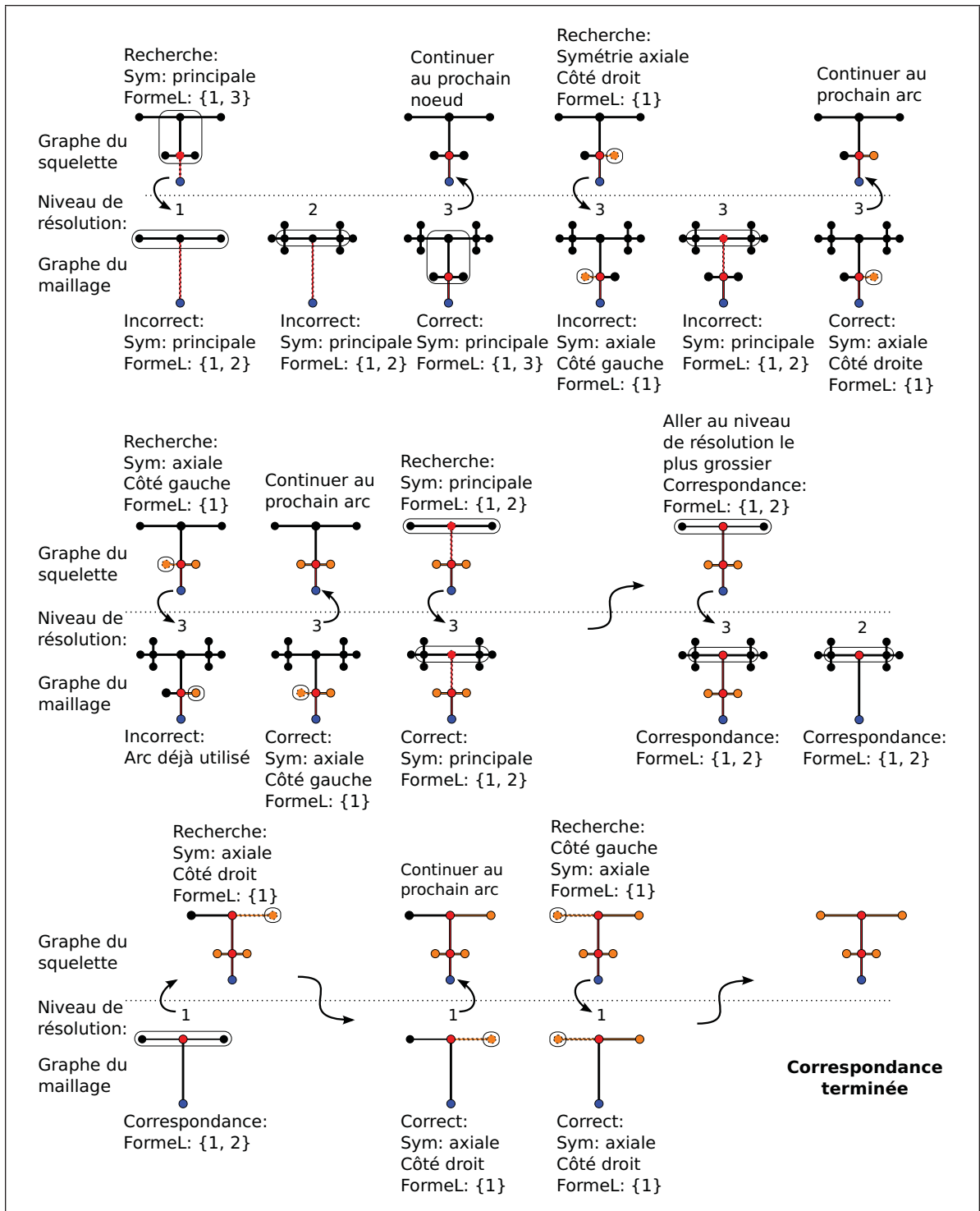


Figure 3.12 Étapes d'une correspondance multirésolution. Noeud racine en bleu. Axes de symétrie colorés lorsqu'une correspondance est trouvée.

Cette méthode de correspondance traite correctement les cas où le graphe du maillage ou le graphe du squelette ont plus d'arcs l'un que l'autre. La correspondance multirésolution s'occupe du cas d'un graphe de maillage ayant plus d'arcs que nécessaire, tel qu'illustré à la Figure 3.11. De plus, l'algorithme de correspondance peut très facilement détecter et automatiquement retirer les arcs de squelette superflus, comme les antennes de l'insecte à la Figure 3.20 ou le cinquième doigt du squelette humain à la Figure 3.21. Néanmoins, cette méthode de correspondance de sous-graphes limite le processus d'adaptation aux situations où la topologie du maillage est grossièrement équivalente à celle du squelette. Bien que ce soit souvent le cas, cela est moins générique que la méthode utilisée par Baran et Popović (2007) qui peut adapter un squelette humanoïde simple à un beigne, par exemple.

3.1.3.2 Placement des joints

La méthode de correspondance construit des paires entre les arcs du maillage et du squelette. Après que celles-ci sont bien établies, les positions des joints du squelette doivent être déterminées. Ce positionnement est trivial pour les arcs du squelette composé d'un seul os : les extrémités de l'os sont repositionnées à la première et à la dernière position sur l'arc du maillage projeté dans l'espace tridimensionnel. Le repositionnement est beaucoup plus complexe pour les arcs du squelette constitués de deux os ou plus, telles que les pattes de l'insecte à la Figure 3.20 où quatre joints doivent être positionnés. En considérant que de tels arcs du squelette sont associés un pour un à des arcs du maillage, les os correspondants seront repositionnés dans l'espace le long des positions tridimensionnelles des arcs du maillage.

Soit un arc du squelette avec t os et $t + 1$ joints et un arc du maillage ayant une projection tridimensionnelle composée d'une liste ordonnée de points $V = [v_1, v_2, \dots, v_m]$. Le premier joint est positionné à $v_{k_1} = v_1$. Chacun des $t - 1$ joints internes sera assigné à un index $k_i \in \{1, \dots, m\}$ correspondant à une position de la projection v_{k_i} du maillage. Finalement, le dernier joint sera positionné à $v_{k_{t+1}} = v_m$. La Figure 3.13 illustre le repositionnement possible d'un arc du squelette sur un arc du maillage. Les positions des joints internes $v_{k_2}, v_{k_3}, \dots, v_{k_t}$ sont choisies pour minimiser la fonction de coût suivante :

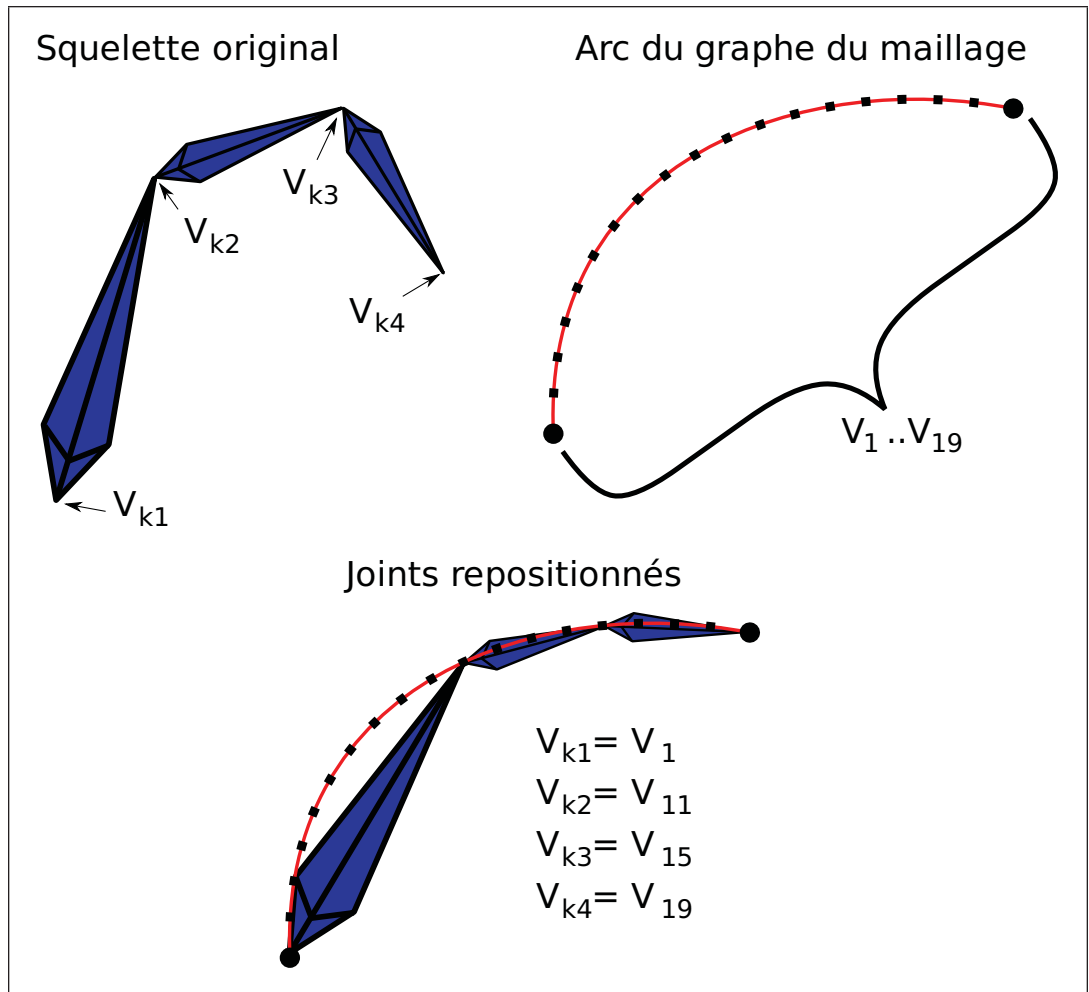


Figure 3.13 Placement de joint sur les points de l'arc.

$$\text{coût} = \sum_{i=1}^{t-1} \underbrace{\gamma_{\theta} |\theta_i|}_{\text{angle}} + \sum_{i=1}^t \left[\underbrace{\gamma_l |l_i|^2}_{\text{longueur}} + \underbrace{\gamma_x x_i}_{\text{distance}} \right] \quad (3.1)$$

$\theta_i \rightarrow$ angle au joint i – angle initial au joint i
 $l_i \rightarrow \frac{\text{longueur de l'os } i - \text{longueur initiale de l'os } i}{\text{longueur initial de l'os } i}$
 $x_i \rightarrow$ distance maximum entre l'os i et la projection de l'arc du maillage V aux indices comprises entre k_i et k_{i+1}

Les paramètres γ , définis par l'utilisateur, déterminent si le résultat devrait s'influencer plus de la forme originale du squelette (paramètres γ_θ et γ_l) ou de la forme du maillage (paramètre γ_x). Le terme de variation d'angles aux joints $\sum_{i=1}^{t-1} \gamma_\theta |\theta_i|$ pénalise les changements au niveau des angles entre chaque os du squelette. Le terme de variation de longueur (longueur de l'os i - longueur initial de l'os i) n'est pas très utile par lui-même puisqu'un même changement sera significatif sur un os de courte longueur alors qu'il serait négligeable sur un os de grande longueur. C'est pourquoi une mesure relative est utilisée : La variation de longueur en temps que ratio de la longueur originale $\left(\frac{\text{longueur de l'os } i - \text{longueur initiale de l'os } i}{\text{longueur initiale de l'os } i} \right)$. En utilisant le ratio directement, il est difficile de proprement ajuster le paramètre γ_l pour certains maillages, comme celui de la Figure 3.20(c). Le ratio au carré donne une amplification qui est utile lorsque la longueur des os du squelette original et du squelette adapté diffère de manière significative. Le terme de distance $\sum_{i=1}^t \gamma_x x_i$ pénalise un positionnement des os qui s'éloigne de la forme du maillage. Ceci est quantifié par la distance maximale entre un os et la projection de l'arc du maillage correspondant.

Des restrictions logiques sont aussi prises en compte par l'algorithme de résolution pour prévenir les retours en arrière et les os de longueur nulle ($i < j \rightarrow k_i < k_j$). Le problème est résolu par la fonction résoudreJointsInterne (Algorithme 2) qui trouve la meilleure position des joints internes à partir de la position des deux joints précédents et du nombre de joints restants. Résoudre le problème revient à appeler résoudreJointsInterne(1, 1, $t - 1$).

Comparativement, la méthode utilisée par Baran et Popović (Baran et Popović, 2007) utilise aussi des pénalités de longueur et de distance des os, mais repose sur l'orientation de ceux-ci

Algorithme 2 : résoudreJointsInterne(k_p, k_c, j) $k_p, k_c \in \{1, \dots, m-1\}$: index de position des joints précédent et courent $j \in \{0, \dots, t-1\}$: nombre de joints internes restant**if** $j > 0$ **then**résultat.coût $\leftarrow \infty$ **foreach** $k_n \leftarrow$ tous les indices de position pour le prochain joint **do**coût \leftarrow CoûtAngle (k_p, k_c, k_n)// $\gamma_\theta |\theta_i|$ + CoûtLongueur (k_c, k_n)// $\gamma_l |l_i|^2$ + CoûtDistance (k_c, k_n)// $\gamma_x x_i$ **if** coût \geq résultat.coût **then**

┌ /* coût trop élevé, récursion inutile */

elseenfant \leftarrow résoudreJointsInterne ($k_c, k_n, j-1$)coût \leftarrow coût + enfant.coût**if** coût $<$ résultat.coût **then**┌ résultat.coût \leftarrow coût┌ résultat.positions \leftarrow [k_n : enfant.positions]**else** /* $j = 0$ */┌ $k_n \leftarrow m$ // dernier joint sur l'arc┌ résultat.coût \leftarrow CoûtAngle (k_p, k_c, k_n)// $\gamma_\theta |\theta_i|$ ┌ + CoûtLongueur (k_c, k_n)// $\gamma_l |l_i|^2$ ┌ + CoûtDistance (k_c, k_n)// $\gamma_x x_i$ **return** résultat

à la place de l'angle aux joints, ce qui la rend sensible à l'orientation des membres dans le maillage ciblé, comme le passage d'une orientation verticale à une orientation horizontale des bras à la Figure 3.21.

3.1.3.3 Orientation

L'orientation des os d'un squelette est aussi, sinon plus, importante que leur position. Cette orientation définit les axes de rotation d'un joint et est donc primordiale pour toute animation d'un squelette. Les joints doivent avoir une orientation compatible au squelette original non seulement pour la réutilisation de blocs d'animation mais aussi parce que des contraintes de rotation peuvent affecter ceux-ci. Le genou d'un humain ne pouvant généralement que plier sur un axe, ceci est représenté par une contrainte de rotation guidant la déformation du squelette, si

l'axe de contrainte n'est pas orienté de la même façon après adaptation, le genou pourrait être forcé à plier de côté au lieu d'avant en arrière.

La plupart des techniques antérieures d'adaptation ou de génération de squelette ne gèrent pas l'orientation des os, préférant traiter ceux-ci comme de simples droites tridimensionnelles. L'information d'orientation est donc perdue ou pire, inconstante.

Le problème est donc de préserver l'orientation des os pendant une adaptation de squelette. Pour ce faire, différentes méthodes sont proposées :

- a. Appliquer la rotation correspondante au déplacement des os à l'orientation de ceux-ci. En utilisant les positions originales et adaptées des deux joints constituant un os, calculer un axe ainsi qu'un angle de rotation et appliquer cette rotation à l'orientation originale de l'os, comme le montre la Figure 3.14

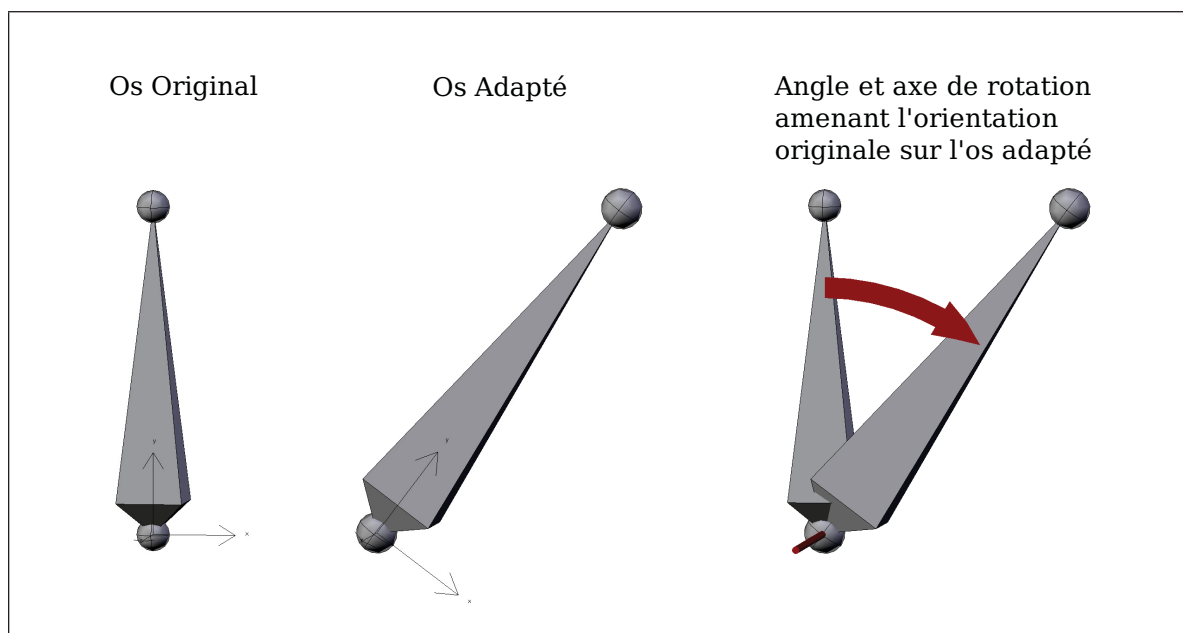


Figure 3.14 Orientation des joints adaptés à l'aide de la rotation correspondante à la transformation d'adaptation entre l'os original et adapté.

- b. Pour les joints déjà fléchis dans la pose de départ, calculer l'orientation par rapport au plan de flexion et rapporter celle-ci sur le plan de flexion après adaptation. La Figure 3.15

illustre cette méthode avec le plan de flexion en vert et sa normale en rouge. On peut voir que le joint original et le joint final sont correctement orientés à 45 degré par rapport au plan.

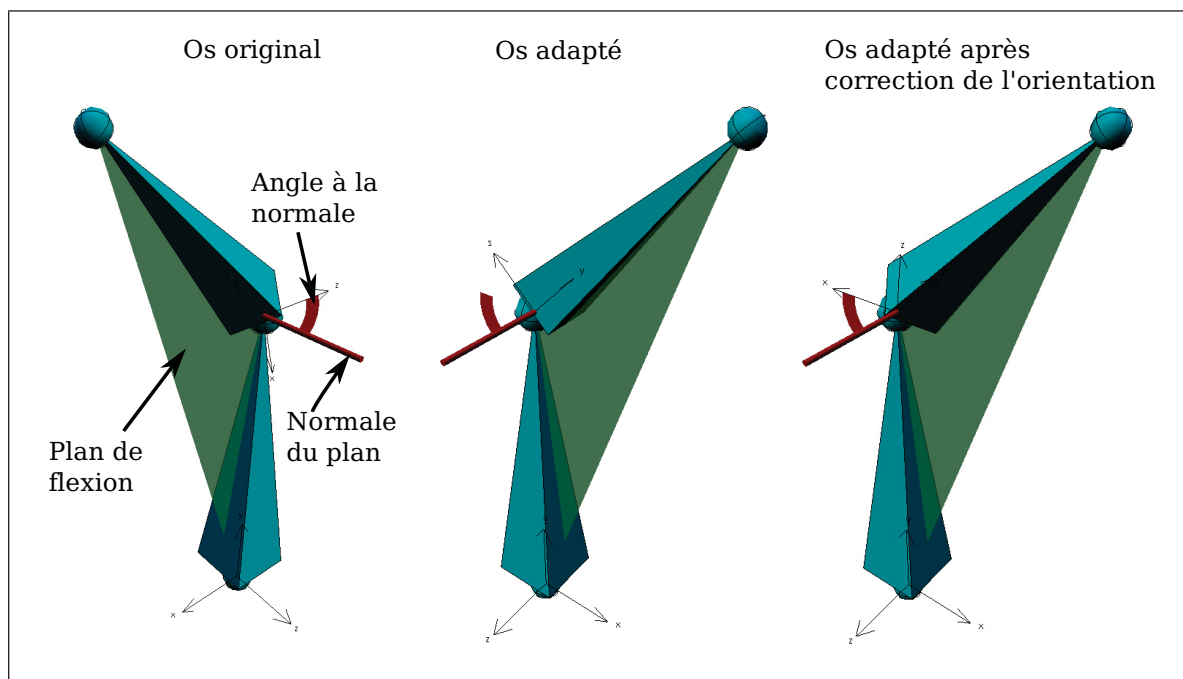


Figure 3.15 Orientation des joints adaptés à l'aide du plan de rotation original et résultant.

En théorie, la première méthode minimise les différences avec l'orientation globale initiale des joints du squelette alors que la deuxième minimise les différences d'orientations locales à chaque plan de rotation.

En pratique, la deuxième méthode donne des résultats plus consistants avec l'*intention* associée à l'orientation d'un joint en conservant les axes de rotations de ce dernier alors que la première donne des résultats visuellement plus proches de l'orientation originale, correspondant aux rotations qui emmènent le squelette original sur le nouveau personnage.

3.1.3.4 Contrôleurs

Un contrôleur est défini comme un os non-déformant qui est utilisé pour transformer des os de déformation, pour ajouter un parent commun à des chaînes d'os déconnectées ou autres raisons

similaires. Ces types d'os sont très fréquents dans les squelettes d'animation professionnelle parce qu'ils permettent aux utilisateurs de créer des hiérarchies plus flexibles, des actions plus facilement contrôlables (flexions des doigts, roulement du pied, anti-patinage, ...) et sont donc souvent considérés comme indispensables à une animation de qualité. La plupart des approches précédentes au problème d'adaptation ne prennent aucunement en compte les contrôleurs, ce qui rend leur utilisation en industrie très limitée.

3.1.3.4.1 Liens

Les contrôleurs doivent aussi être repositionnés, mais le fait que leur position est rarement dictée par la forme du personnage (comme c'est le cas pour les os de déformation) pose un nouveau problème. L'approche utilisée est de lier chaque contrôleur à un os de déformation avant la phase d'adaptation et de propager la transformation résultante de l'os de déformation aux contrôleurs associés. Lors de la création des liens, la méthode considère d'abord les os de déformation qui sont directement contraints par le contrôleur. Si aucun os de déformation n'est contraint au contrôleur, celui-ci sera lié à son parent hiérarchique, si présent, sinon à un de ses enfants.

Les os de déformations sont donc choisis selon les priorités suivantes :

- a. Cible de contraintes : Contrôleurs servant pour de la cinématique inverse (IK), comme cible de pôle, comme contrôle d'actions et autres. Ces contrôleurs sont liés à l'os de déformation qui reçoit la contrainte tel que démontré à la Figure 3.16.
- b. Os parent : Contrôleurs ayant un parent hiérarchique, tels que ceux dans des chaînes de contrôleurs. Si le parent est un contrôleur lui-même, la méthode suit la chaîne jusqu'à ce qu'elle trouve un contrôleur lié à un os de déformation. Celui-ci est choisi comme lien pour le contrôleur courant tel que démontré à la Figure 3.17.
- c. Os enfant : Contrôleurs ayant un ou des enfants hiérarchiques, tels que ceux utilisés comme racine. Comme ces contrôleurs peuvent avoir plusieurs enfants, il est important de sélectionner celui qui est le plus significatif. Le contrôleur est donc lié à son enfant

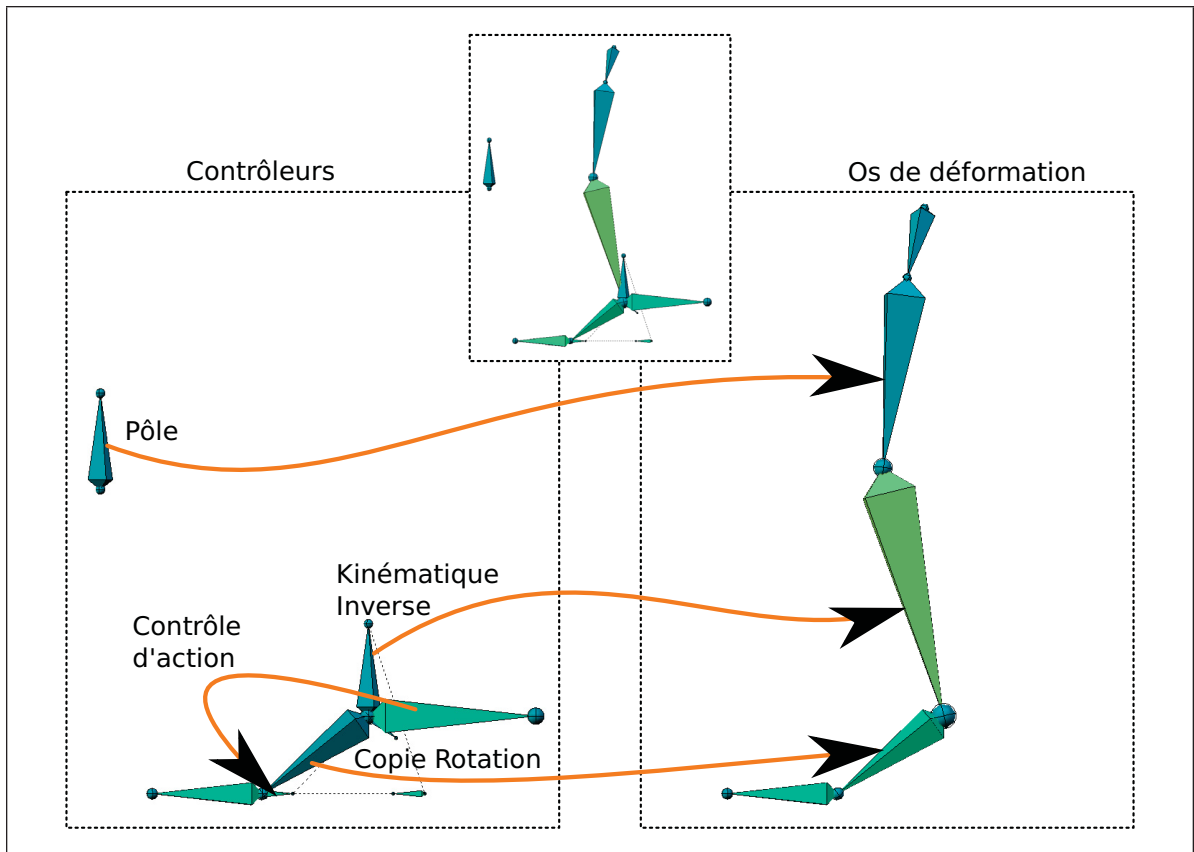


Figure 3.16 Liens des contrôleurs dans un squelette de jambe.
(certains liens ne sont pas montrés)

de plus bas niveau de symétrie, celui qui est donc le plus proche de l'axe de symétrie principale. Ceci lie la racine de la colonne vertébrale à cette dernière et non aux jambes ou à la queue qui sont aussi ses enfants tel que démontré à la Figure 3.17.

Lors de cette étape, un écart de translation entre le contrôleur et l'os de déformation est calculé et servira plus tard pendant le repositionnement.

3.1.3.4.2 Doubles liens

Dans certain cas, la position d'un contrôleur peut dépendre de deux os différents. Par exemple, un contrôleur peut servir à distancer deux joints différents d'un squelette. Dans ce cas, il est primordial que les deux joints de l'os de contrôle soient aussi positionnés sur les mêmes joints de déformation après adaptation. Ces cas sont identifiés en testant chaque joint correspondant

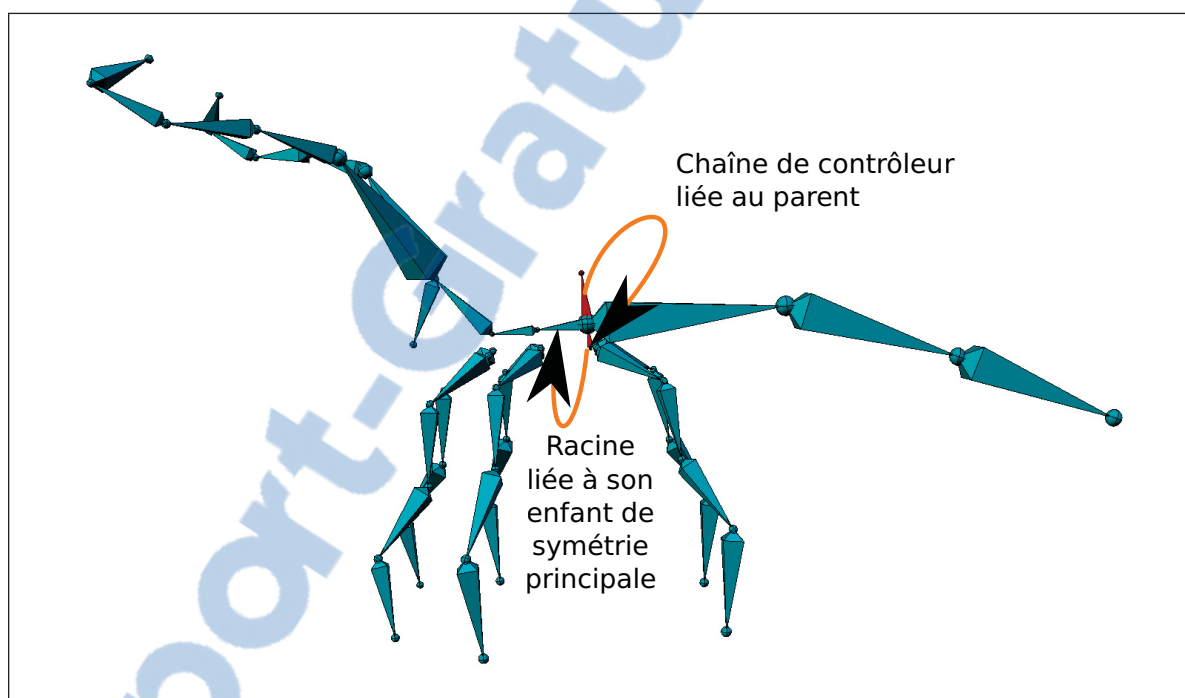


Figure 3.17 Liens des contrôleurs dans un squelette d'insecte.

au bout d'un os de contrôle pour voir s'il est superposé à un joint de déformation. Si c'est le cas, lors de l'adaptation, le contrôleur sera d'abord repositionné selon son lien primaire (décrit plus tôt) et ensuite réorienté et rallongé ou raccourci pour superposer le deuxième joint au lien secondaire.

La Figure 3.18 montre l'adaptation d'un squelette de doigt avec flexeur IK avec lien de contrôleur unique et lien double. Dans le cas du lien unique, le contrôleur ne dépend que du premier os du doigt, alors il est adapté relativement à celui-ci seulement et ne pointe plus vers le dernier joint du doigt. L'application d'un double lien corrige la situation.

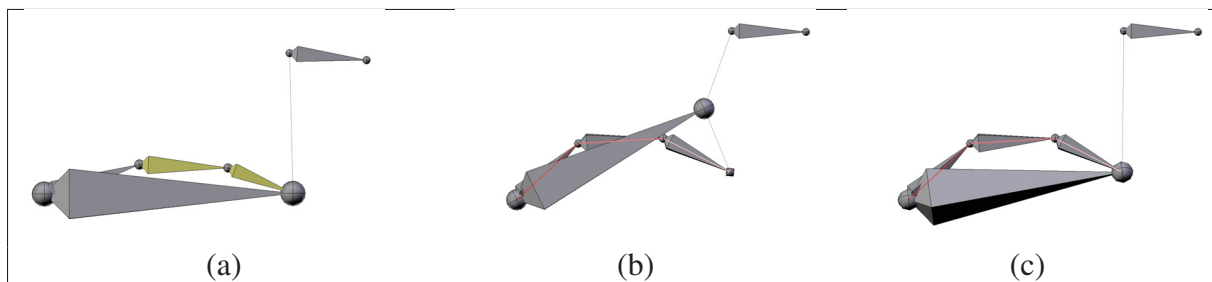


Figure 3.18 (a) Squelette de doigt avec flexeur IK. (b) Squelette adapté avec lien unique. (c) Squelette adapté avec lien double.

3.1.3.4.3 Repositionnement

Les contrôleurs sont repositionnés en dérivant une rotation et un facteur de redimensionnement à partir de la transformation subie par l'os de déformation auquel ils sont liés. La translation calculée précédemment et la taille du contrôleur sont mises à l'échelle et l'os lui-même subit une rotation par rapport à l'extrémité de son os de déformation.

Cette technique est générique et peut être jumelée à d'autres méthodes de transformation de squelette, que ce soit une autre méthode d'adaptation ou encore pour automatiquement ajuster les contrôleurs lorsqu'un artiste apporte des modifications au squelette de déformation.

3.1.4 Résultats

Les figures suivantes montrent des résultats de l'algorithme d'adaptation :

- Un squelette de bipède avec système de transition entre cinématique directe (FK) et inverse (IK) adapté à un écureuil (Figure 3.19).



Figure 3.19 Rinky de *Big Buck Bunny* (2008) et l'adaptation d'un squelette avec système de transition de cinématique directe et inverse.

- Un squelette d'insecte générique adapté à un maillage de papillon et de mouche (Figure 3.20) pour démontrer l'algorithme de placement de joints pour les membres à plusieurs articulations. Dans ces deux cas, les fonctions de pénalité donnaient plus de poids à la forme du maillage qu'au squelette original.
- Un squelette d'humain complet et complexe (IK, contrainte de pôle, contrôle de roulement de pied et autres contrôleurs) adapté à un personnage de cartoon, Figure 3.21(a), de même qu'une adaptation partielle pour le bras, Figure 3.21(b), et la jambe, Figure 3.21(c). Dans ces cas, le placement de joints est plus précis quand les membres sont quelques peu fléchis puisque la fonction de pénalité de distance pousse la position du joint près du point de flexion. Ce squelette humanoïde est considéré comme complexe dû à ces nombreux membres et contrôleurs. De plus, ce squelette nous a été fourni par un studio d'animation professionnel et a été utilisé dans des courts métrages d'animation ainsi que dans des publicités télévisées.

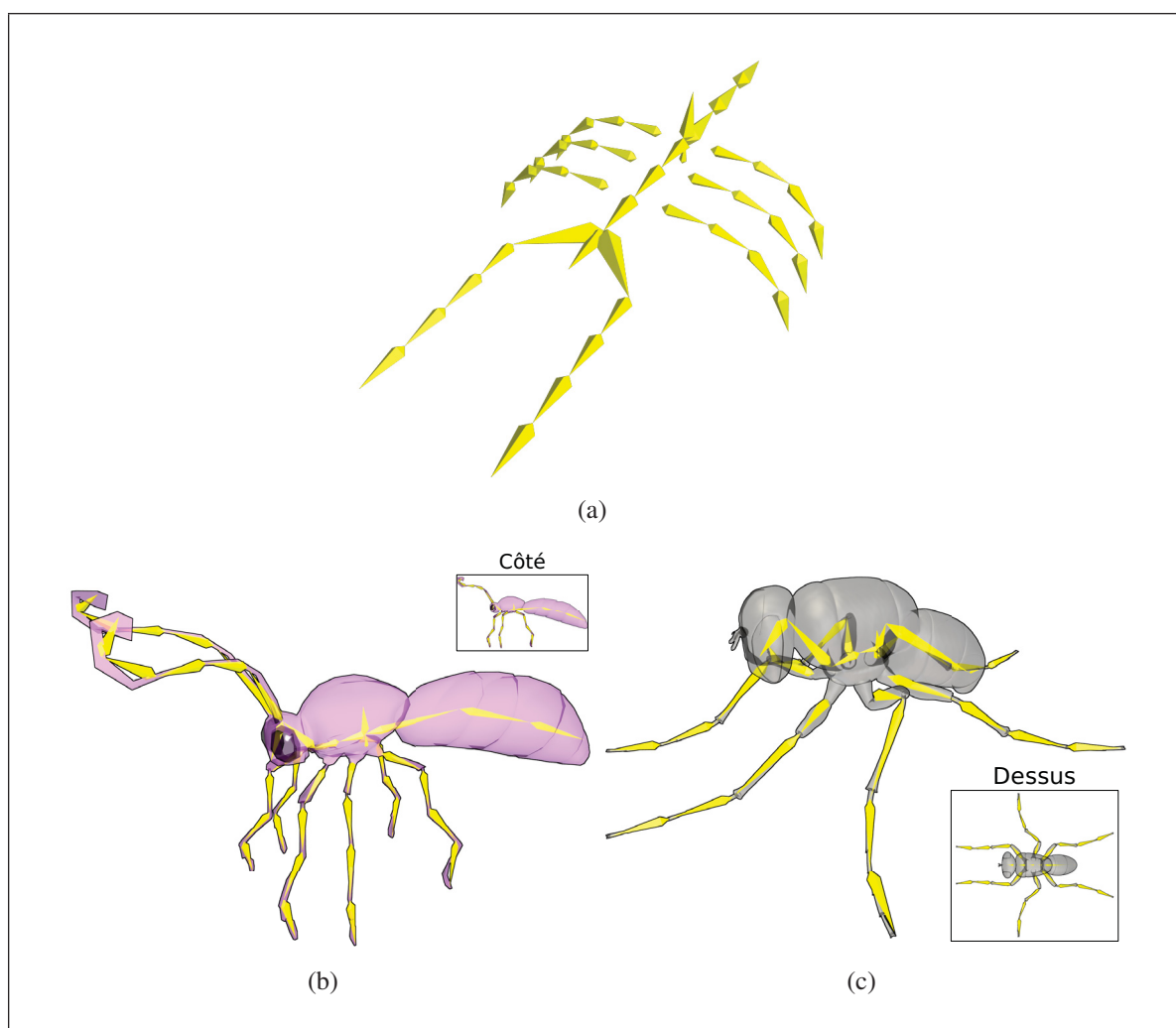
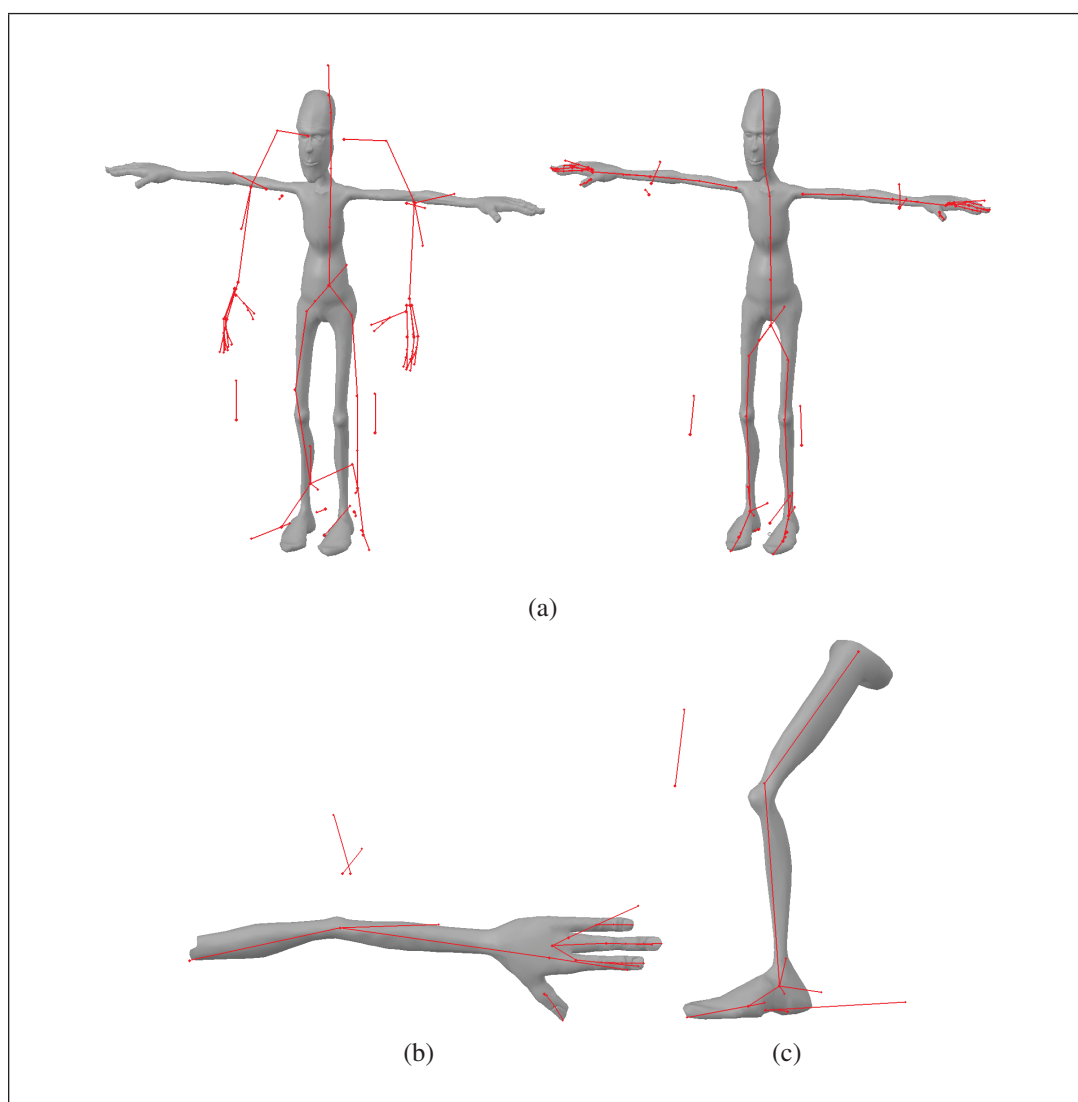


Figure 3.20 (a) Le squelette d'insecte générique utilisé dans les deux cas.
(b) Maillage d'un papillon et squelette adapté.
(c) Maillage d'une mouche et squelette adapté.



**Figure 3.21 (a) Personnage de cartoon avec squelette adapté.
 (b) Bras et squelette adapté : IK, pôle et contrôles de flexion de doigts.
 (c) Jambe et squelette adapté : IK, pôle et contrôle de roulement de pied.**

3.2 Création et adaptation partielle par croquis

Bien que la méthode d'adaptation complète soit robuste et pratique, plusieurs cas bénéficieraient d'une méthode plus flexible de création et d'adaptation de squelette. Cette section présente une méthode de création et d'adaptation partielle basée sur une interface de croquis qui délaisse le traitement de squelette complet pour permettre aux utilisateurs une fluidité et flexibilité de création de squelette accrue.

3.2.1 Survol

Voici un survol de la méthode :

A. Création du croquis par l'utilisateur

- Trait continu et ligne polygonale
- Modification par gestuelle et par surimpression

B.1 Création des os sur les traits du croquis

- Ajout des joints aux subdivisions automatiques et manuelles des traits.
- Orientation des os

OU

B.2 Adaptation d'un patron os sur les traits du croquis

- Génération d'un patron à partir d'un squelette
- Liaison des contrôleurs au patron
- Adaptation du patron aux traits

Cette méthode est proposée plus en tant qu'outil de création de squelette que comme un simple outil d'adaptation.

3.2.1.1 Interface

La métaphore d'interface utilisée est celle d'une tablette à dessin. L'utilisateur trace différentes lignes courbes continues ou polygonales. Il peut modifier ces courbes à l'aide de tracés ges-

tuels (Section 3.2.2.1) permettant de rajouter des points aux courbes, d'en lisser une section sélectionnée, d'en couper une partie, . . . L'utilisateur peut aussi modifier l'allure d'une courbe en la retraçant en surimpression (Section 3.2.2.2).

3.2.2 Implémentation

Les traits sont représentés par une liste ordonnée de points. Chaque point possède une coordonnée dans l'espace 3D, un vecteur de direction représente l'orientation de la vue lors du tracé ainsi qu'un type de point (pour différencier entre les points formant une ligne polygonale et ceux faisant partie d'un trait courbe continu). La direction d'un trait est donc implicite dans l'ordre des points de la liste. La Figure 3.22 montre les différentes données composantes d'un trait.

Pour simplifier la géométrie des traits continus, souvent trop précise à cause de la fréquence d'échantillonnage, un filtre de Chaikin inverse (Hassan et Dodgson, 2005) est appliqué. Ce filtre a pour effet de réduire le nombre de points dans une courbe tout en conservant son allure générale.

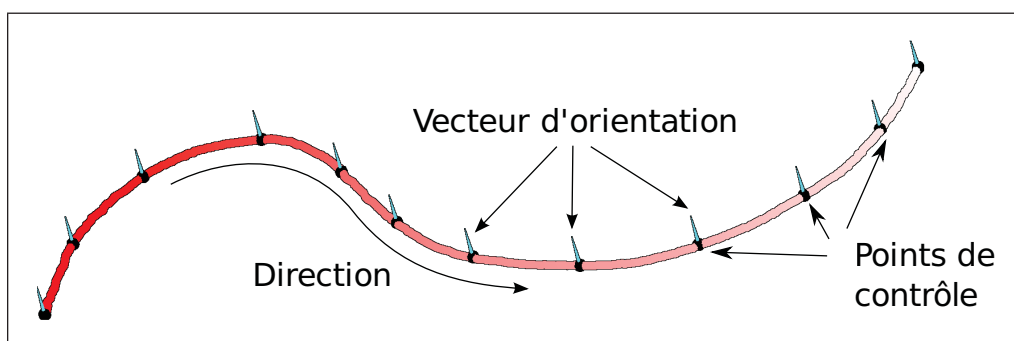


Figure 3.22 Composition d'un trait : liste ordonnée de points.

3.2.2.1 Gestuelle

La reconnaissance de gestuelle se fait à un niveau très abstrait. Tout d'abord, l'utilisateur trace un trait continu à l'écran, de la même façon qu'un trait régulier mais l'outil est invoqué différemment (avec Shift-clique ou clique de droite, par exemple). Ensuite, l'algorithme extrait

une série d'informations qui servira à identifier la fonction gestuelle. Les informations extraites pour l'identification du geste sont les suivantes :

- Segmentations : découper le geste en segments de droite
- Intersections avec des traits : produire une liste de points d'intersections entre le geste et les traits à l'écran
- Intersections avec lui-même : produire une liste de points d'intersections du geste sur lui-même.

Les gestes suivants sont disponibles dans l'application, les Figure 3.23 à 3.28 en montrent l'application (gestes en bleu, traits en rouge, le dégradé de couleurs foncé à pâle indique la direction d'un trait).

- Insertion d'un point exact : Figure 3.23

Le geste contient un segment qui intersecte un ou plusieurs traits une seule fois. Les points exacts sont ajoutés aux intersections.

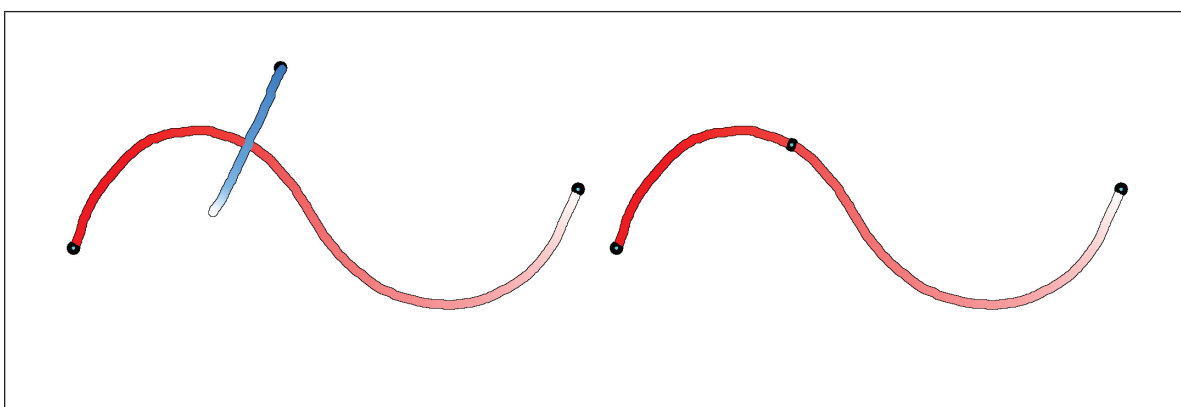


Figure 3.23 Geste d'insertion et son résultat.

- Coupure de traits : Figure 3.24

Le geste contient deux segments formant un L et intersecte un ou plusieurs traits une seule fois. Les sections de traits du côté de la patte du L (le deuxième segment de droite) sont retirées et un point exact est ajouté aux intersections.

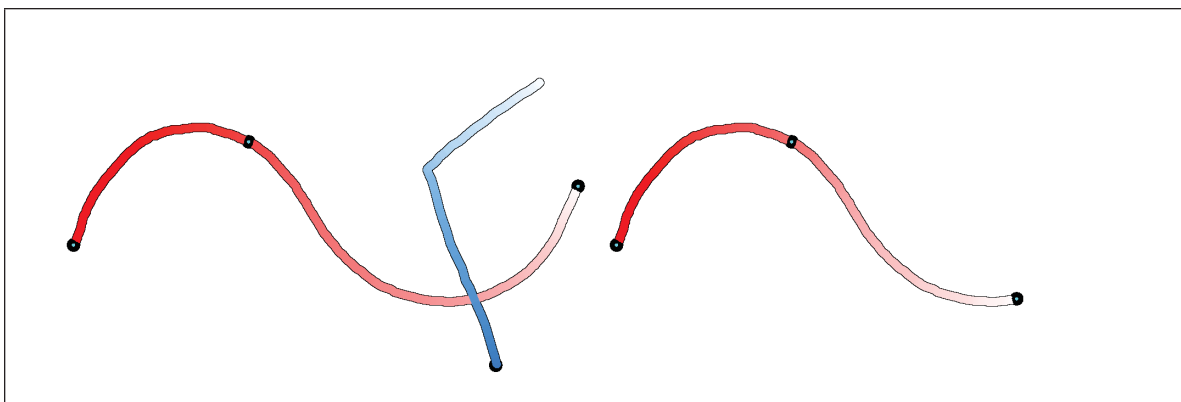


Figure 3.24 Geste de coupure et son résultat.

- Inversion de la direction du trait : Figure 3.25

Le geste comporte plus de quatre segments et intersecte un ou plusieurs traits en deux points. Les listes de points sont inversées, ce qui inverse la direction d'un trait.

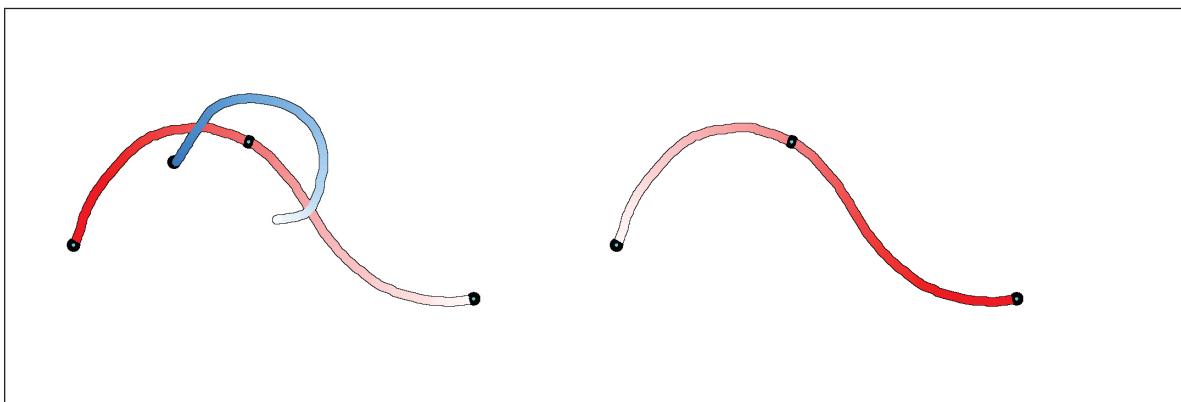


Figure 3.25 Geste d'inversion et son résultat.

- Commandes de correction (aplatir, redresser, polygonaliser) : Figure 3.26

Le geste comporte plus de quatre segments, intersecte un ou plusieurs traits en deux points et se termine en queue de cochon (une boucle en plusieurs segments et intersection sur lui-même). Une commande de correction est appliquée aux sections de traits entre les intersections.

- Conversion en squelette : Figure 3.27

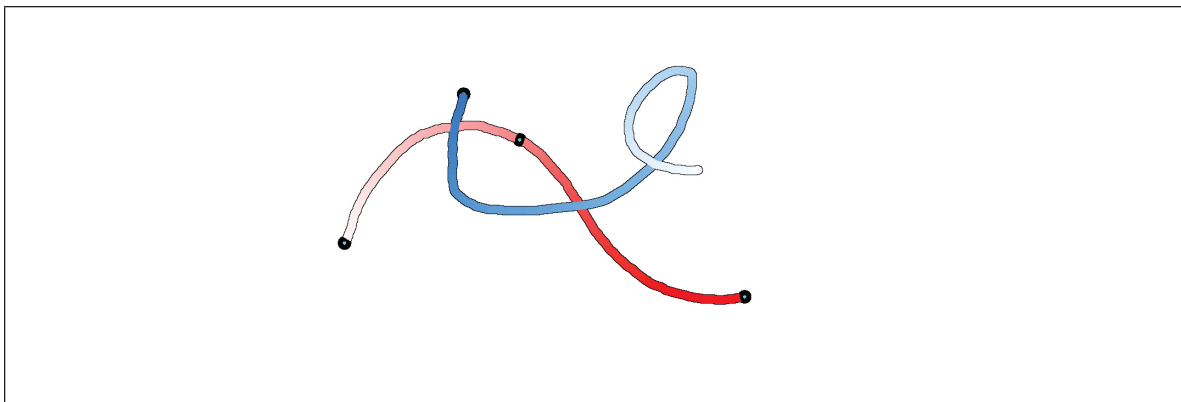


Figure 3.26 Geste de commandes de correction.

Le geste comporte trois segments qui s'intersectent en un point. Les traits sélectionnés sont convertis en squelette selon la méthode choisie (décrites aux Sections 3.2.4 à 3.2.6).

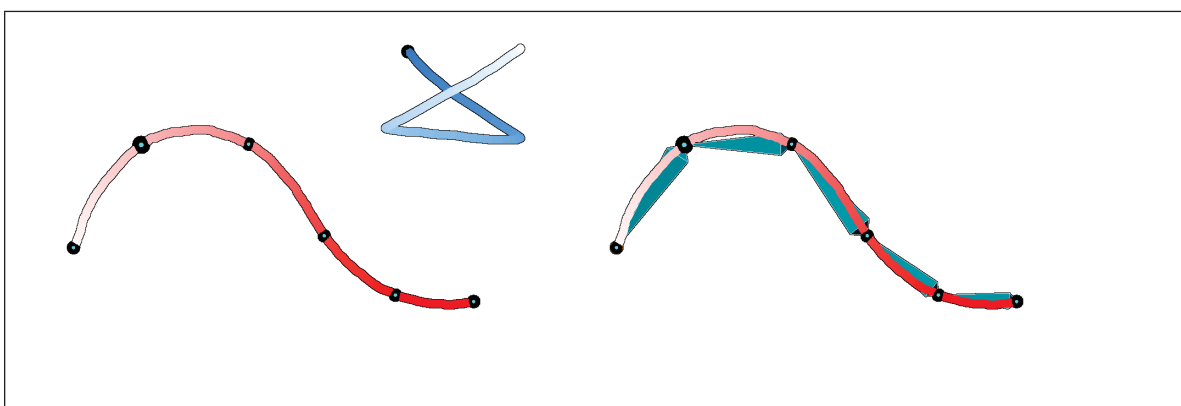


Figure 3.27 Geste de conversion en squelette et son résultat.

– Effacer des traits : Figure 3.28

Le geste comporte deux segments et intersecte un ou plusieurs traits en deux points. Les traits sont retirés du dessin.

Les commandes de corrections permettent d'appliquer différents correctifs sur une section de trait choisie par l'utilisateur lors de la gestuelle.

– Aplatisir : Corrige la position des points sélectionnés pour que la profondeur le long du vecteur d'orientation varie linéairement entre le premier et dernier point.

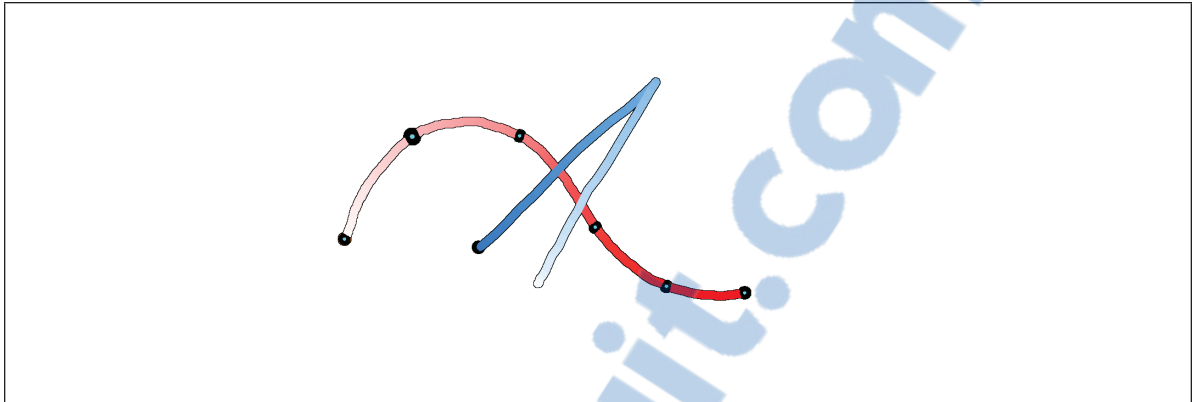


Figure 3.28 Geste d'effaçage de trait.

- Redresser : Corrige la position des points sélectionnés pour que leur position varie linéairement entre le premier et dernier point.
- Polygonaliser : En partant du premier point exact avant la section sélectionnée jusqu'au premier point exact suivant, retirer tous les points continus.

3.2.2.2 Surimpression de traits

La technique de surimpression de traits permet d'appliquer des corrections au tracé en redessinant par-dessus un trait existant. Ceci est similaire au dessin traditionnel où un artiste peut repasser sur une ligne plusieurs fois pour en préciser la forme. La Figure 3.29 démontre cette technique.

3.2.3 Insertion volumétrique

La méthode d'insertion volumétrique est basée sur une variante de la technique traditionnelle de segmentation par profondeur, le découpage d'une scène en différentes couches de profondeur. Comme les méthodes matérielles (implémentées sur une carte graphique) sont souvent relativement imprécises dû à la nature du tampon de profondeur, une méthode logicielle par tracé de rayons est utilisée.

La technique fonctionne en lançant un rayon le long du vecteur de la caméra sous le pointeur de la souris entre les plans rapproché et éloigné. Une liste est construite avec les coordonnées

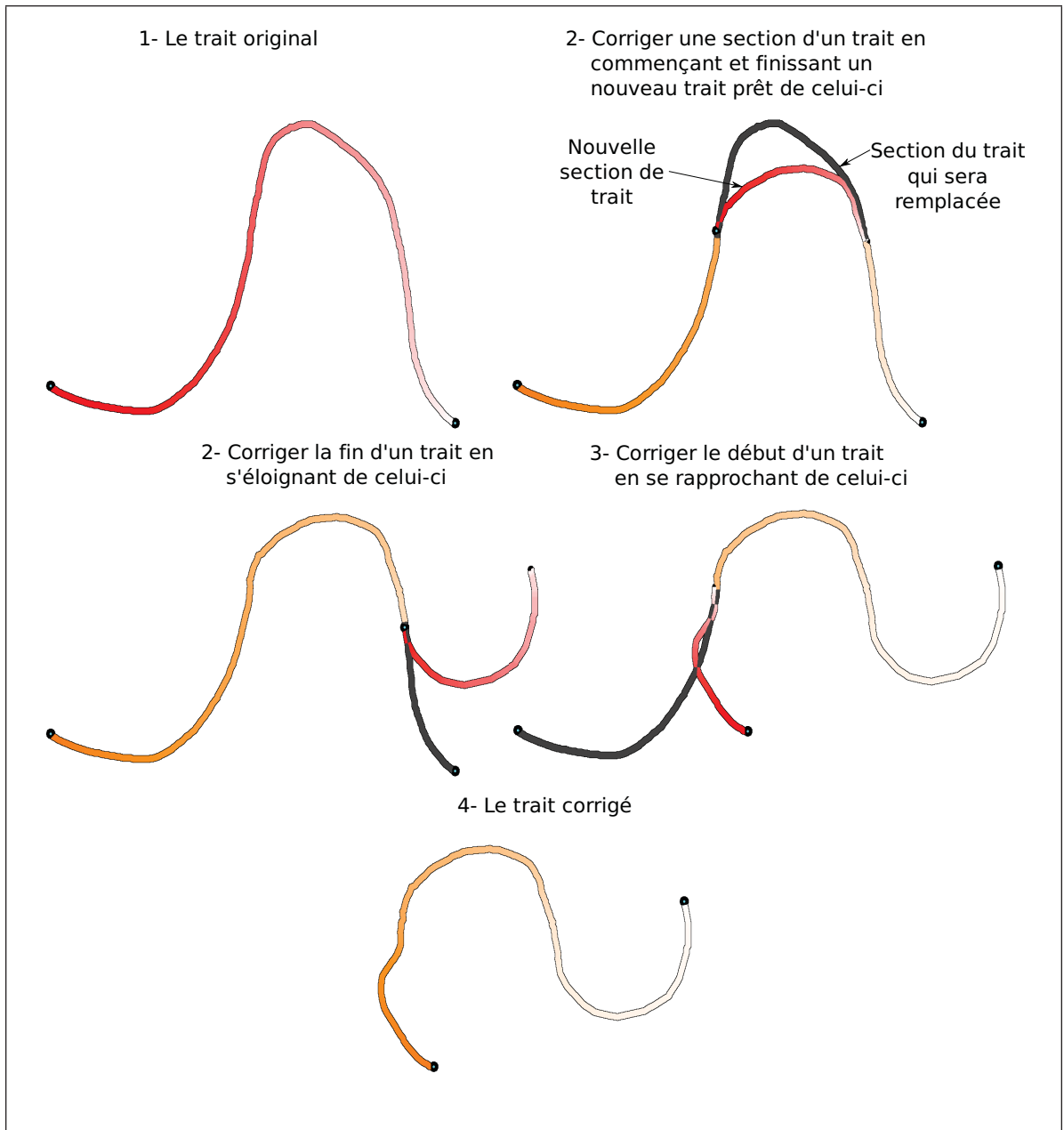


Figure 3.29 Superposition d'un nouveau trait pour corriger un trait existant.

d'intersection, la profondeur de celles-ci par rapport au plan rapproché et l'identificateur de l'objet intersecté. Cette liste est triée en ordre croissant de profondeur. Par la suite, dépendant du mode, faire des paires avec chaque intersection successive d'un même objet ou avec la première et la dernière intersection de chaque objet. Le centre de chaque paire en profondeur

et en coordonnées spatiales correspond au centre du volume sous ce rayon. La Figure 3.30 contient un exemple de ce processus.

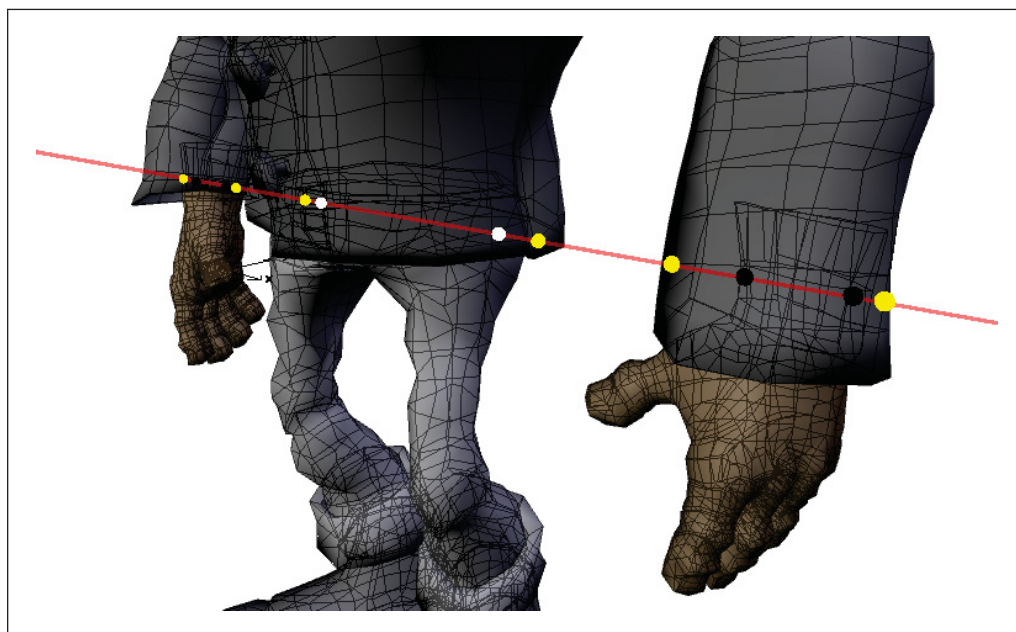


Figure 3.30 Tracé de rayons pour l'insertion volumétrique. Les couleurs représentent l'identificateur d'objets.

Les identificateurs d'objets sont utilisés pour ignorer les volumes internes tel que les bras dans un manteau comme le montre la Figure 3.31 ou pour trouver le milieu de chaque pièce atomique comme c'est le cas à la Figure 3.32.

De plus, la procédure d'insertion permet d'insérer un trait continu dans des volumes partiellement obscurcis en choisissant toujours le centre du volume le plus près du point précédent en ce qui a trait à la profondeur du point par rapport à la caméra. Dans le cas où plusieurs volumes sont superposés comme à la Figure 3.33, l'utilisateur peut donc continuer un trait passant sous un autre volume en étant assuré que la fonction d'insertion ne sautera pas au volume le plus proche de la caméra lorsqu'il veut dessiner sous celui-ci.

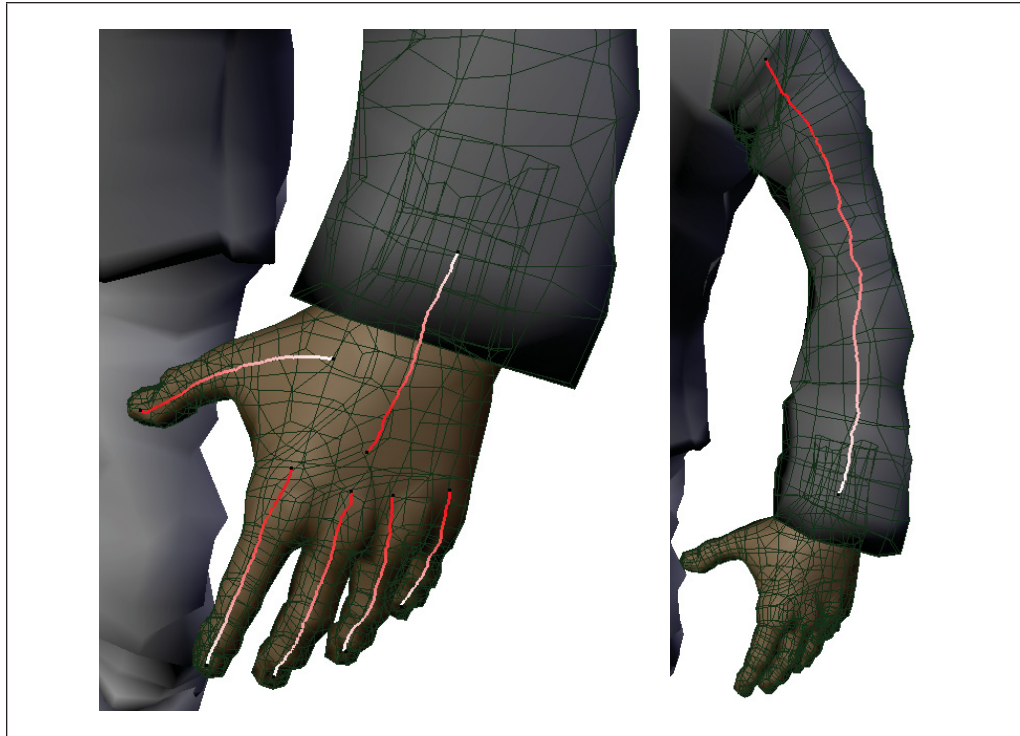


Figure 3.31 Insertions par couches dans un objet organique pour ignorer les volumes internes.

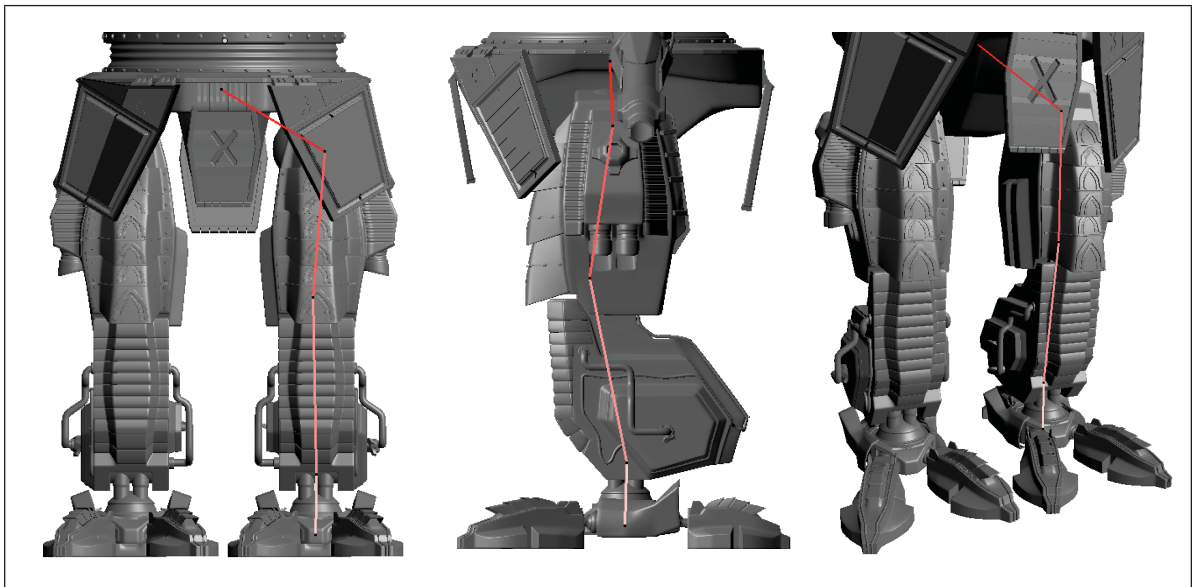


Figure 3.32 Insertions par objets dans un objet mécanique pour ignorer les détails superposés dans chaque pièce.



Figure 3.33 Insertions dans des volumes partiellement obscurcis.

3.2.4 Méthodes de génération

La seconde étape est de convertir les traits du croquis en un squelette. À cette fin, une méthode semi-automatique simple est proposée de même que plusieurs méthodes automatiques de différents niveaux de complexité. Les méthodes automatiques sont utiles pour les membres ayant des joints à intervalles réguliers (subdivision basée sur la longueur ou à nombre constant) ainsi que pour ceux ayant plus de joints dans les sections à grande courbure (subdivision adaptative).

3.2.4.1 Semi-automatique

Chaque point exact correspond à un joint. L'utilisateur peut ajouter des joints avec le geste d'*insertion*.

3.2.4.2 Automatique

La génération automatique se fait en parcourant la courbe du trait et en ajoutant des joints selon différentes formules mathématiques. Soit la liste de points d'une courbe $V = [v_1, v_2, \dots, v_m]$, pour un joint existant à v_i , un nouveau joint sera ajouté à v_j selon les conditions décrites dans les sections suivantes.

3.2.4.2.1 Adaptative

La règle de subdivision adaptative ajoute des joints quand le segment de courbe dépasse une limite de colinéarité (mesuré par la variance de la distribution des points de la courbe).

$$\sigma(v_i..v_j) \geq T \quad (3.2)$$

Ceci permet de concentrer l'ajout de joints dans les zones à plus haute courbure où ils seront présumément plus utiles, comme le montre la Figure 3.34.

3.2.4.2.2 Limite de longueur

La règle de subdivision par limite de longueur ajoute un joint quand la distance entre deux points de courbe dépasse une limite de longueur spécifiée par l'utilisateur.

$$|v_i - v_j| \geq L \quad (3.3)$$

Ceci permet de subdiviser plusieurs traits en os de mêmes longueurs, comme le montre la Figure 3.35.

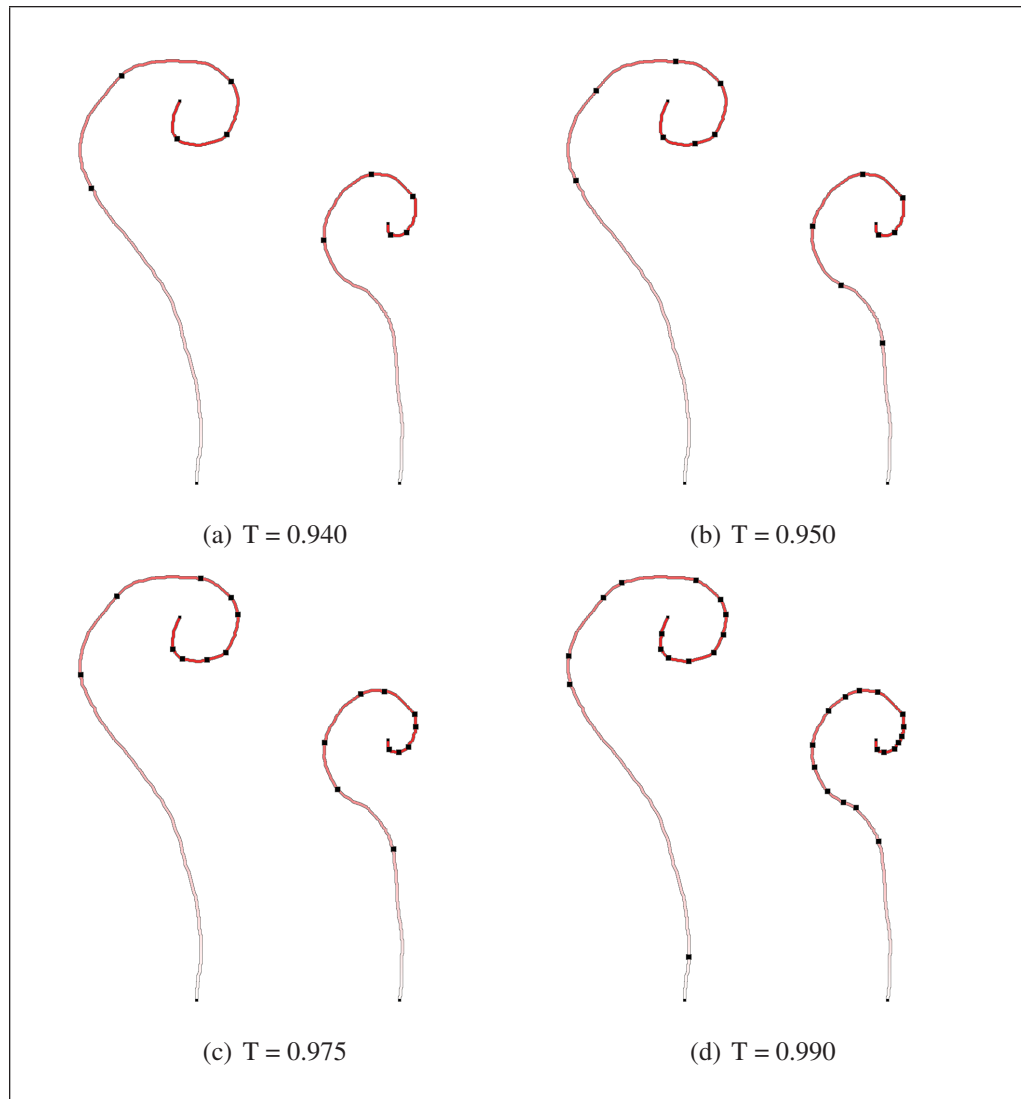


Figure 3.34 Génération automatique adaptative avec différentes tolérances.

3.2.4.2.3 Nombre constant

La règle de subdivision par nombre constant ajoute un joint pour que la courbe soit divisée en un nombre prédéfini de segments de même longueur.

$$\sum_{k=i}^{j-1} (|v_k - v_{k+1}|) \geq \sum_{k=1}^m (|v_k - v_{k+1}|) / N \quad (3.4)$$

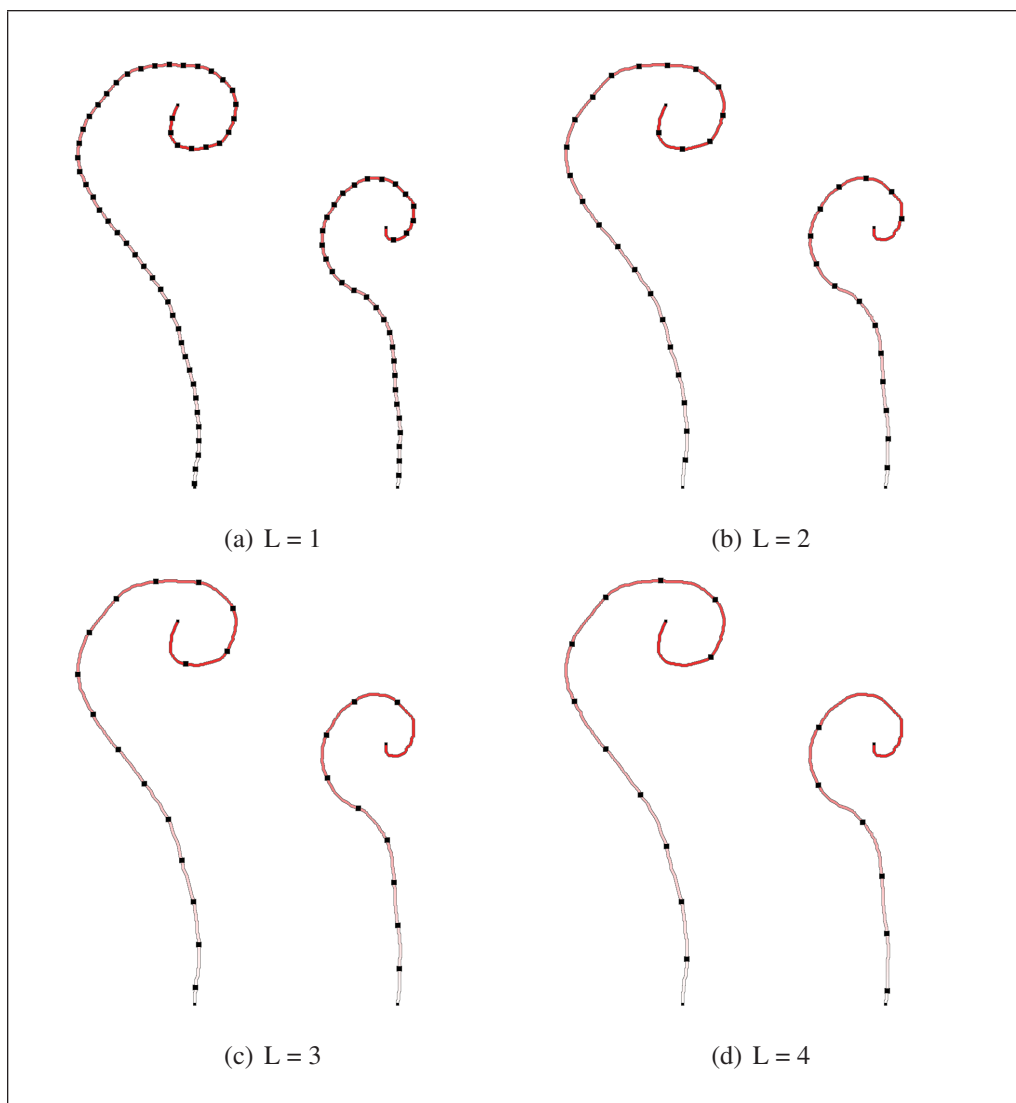


Figure 3.35 Génération automatique par longueur avec différentes valeurs.

Ceci permet de subdiviser plusieurs traits de longueurs différentes en un même nombre d'os, comme le montre la Figure 3.36.

3.2.5 Orientation

Lors du tracé, le vecteur normal au plan de projection est assigné comme vecteur d'orientation pour chaque point d'un trait. Ce vecteur est constant le long de chaque section continue d'un trait mais peut varier entre chaque point polygonal puisque l'utilisateur peut faire bouger la caméra de la vue 3D entre le tracé de chacun de ces points. Ces vecteurs sont par la suite

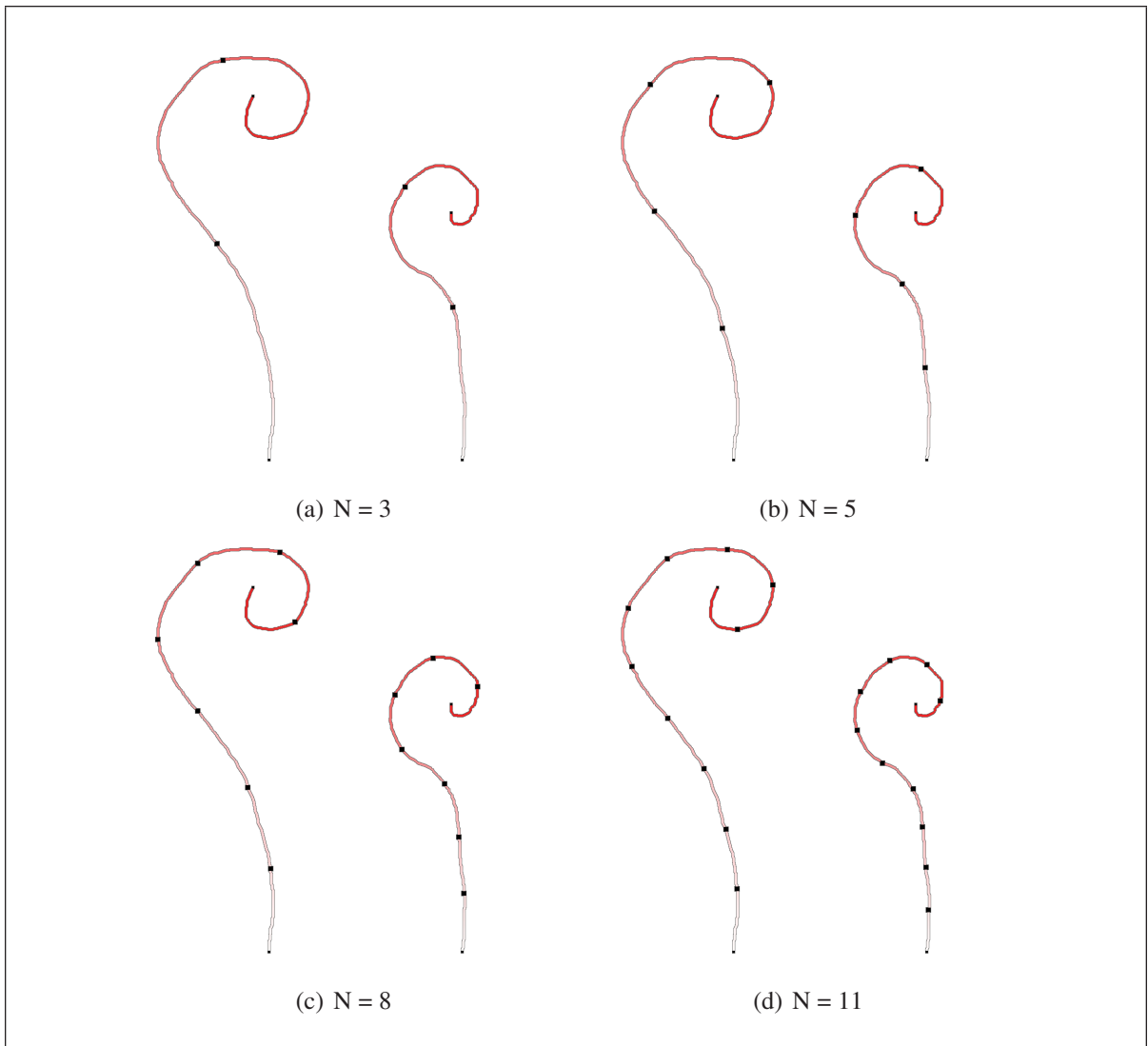


Figure 3.36 Génération automatique par nombre constant avec différents nombres d'os.

utilisés lors de la création des os pour déterminer leur orientation. Le vecteur d'orientation à la racine (le premier joint) d'un os est projeté dans un plan perpendiculaire à ce dernier. L'os subit par la suite une rotation sur son axe longitudinale pour l'aligner avec cette projection, comme dans l'exemple à la Figure 3.37.

3.2.6 Patron

En plus de servir de guide aux méthodes de création automatique, un trait peut aussi servir à adapter un patron de squelette. Un patron peut être un objet séparé ou une chaîne d'os sélec-

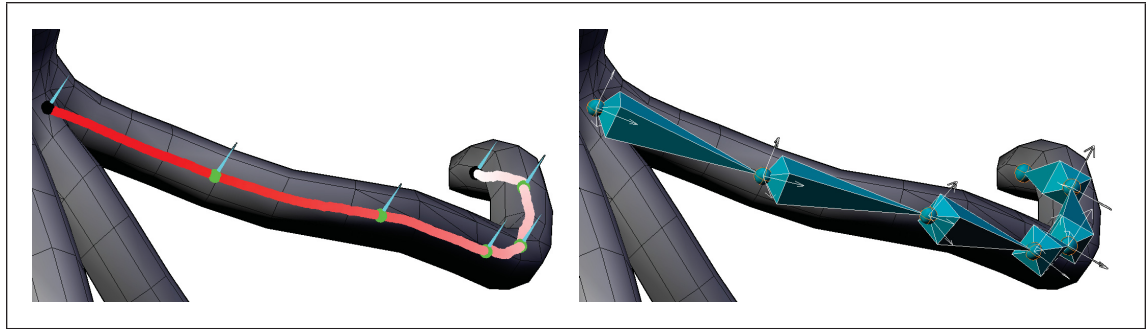


Figure 3.37 L'orientation d'une chaîne d'os lors d'une génération de squelette est déterminée par le vecteur d'orientation du trait (en cyan).

tionnée dans le squelette courant. Il peut contenir aussi bien des os de déformation que des contrôleurs. Contrairement à la technique d'adaptation complète, les patrons utilisés sont limités à de simples chaînes, c'est-à-dire une séquence de joints connectés linéairement entre eux (jambe, bras, doigt, ...). La Figure 3.38 montre un patron de jambe, le trait effectué par un utilisateur et l'adaptation résultante sur une patte de souris.

3.2.6.1 Adaptation

La méthode d'adaptation décrite précédemment (Section 3.1.3) est utilisée pour adapter une copie du patron sur le tracé de l'utilisateur.

3.2.6.2 Orientation

Dans le cas d'un patron, la méthode de réorientation par plan de rotation (présentée à la Section 3.1.3.3) est utilisée. Le premier joint de la chaîne n'étant pas associé à un plan de rotation, le vecteur d'orientation de la courbe est utilisé.

3.2.6.3 Patron de nom d'entité

Pour faciliter la réutilisation d'un même patron dans un squelette, une simple technique de modification automatique de nom d'os a été implémentée. La technique consiste à remplacer différentes entités textuelles spécifiques, présentes dans les noms des os du patron, par d'autres spécifiées automatiquement ou manuellement par l'utilisateur.

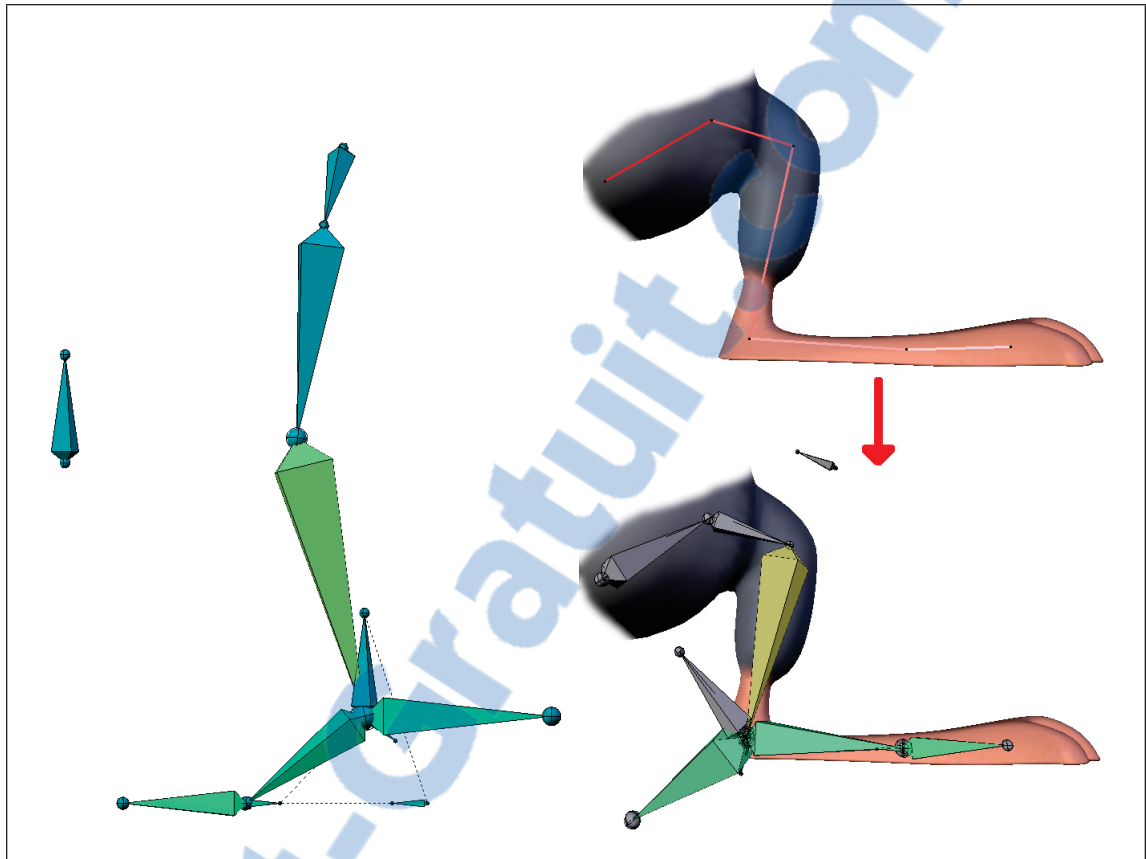


Figure 3.38 Patron de jambe et son adaptation sur une souris.

Deux types d'entités sont supportés :

- a. Numérique (%N)
- b. Axial (%S)

Une option d'automatisation laisse la méthode de remplacement choisir le texte à ajouter dépendant de certains critères.

3.2.6.3.1 Entité numérique

L'entité numérique permet de faire plusieurs copies d'un patron tout en gardant une consistance dans les noms de chaque copie d'os en ajoutant un chiffre là où l'utilisateur le désire.

Exemple : *doigt.%N* devient *doigt.1*, *doigt.2* et ainsi de suite.

L'option d'automatisation incrémente la valeur de remplacement après chaque application d'un patron.

3.2.6.3.2 Entité axiale

L'entité axiale permet de bien nommer les os d'un patron dépendant du côté où ils sont localisés (gauche ou droite) en ajoutant le suffixe approprié.

Exemple : *jambe.%S* devient *jambe.R* ou *jambe.L* dépendant de la position du membre dans l'espace.

L'option d'automatisation détecte automatiquement si le patron ajouté est à droite ou à gauche. L'algorithme de détection assume que le personnage est dans le plan $X \circ Z$, fait face dans la direction de l'axe $-Y$ et est centré par rapport à l'axe X local. Les membres dans l'espace $X > 0$ correspondent donc à la gauche du personnage alors que ceux dans l'espace $X < 0$ correspondent à la droite.

3.2.7 Résultats

Les figures suivantes montrent des résultats de l'algorithme de création et adaptation par croquis :

- Un patron de jambe avec contrôle de roulement de pied et cinématique inverse (Figure 3.38).
- Un patron de doigt avec contrôleur de flexion adapté à un maillage de main squelettique à haute densité de polygones à la Figure 3.39.
- Un squelette tracé sur un maillage de robot avec plusieurs couches de détails (Figure 3.40.a) L'insertion volumétrique inter-objet est utilisée pour trouver le centre réel de chaque pièce et non le centre des panneaux et tubes sur celles-ci.

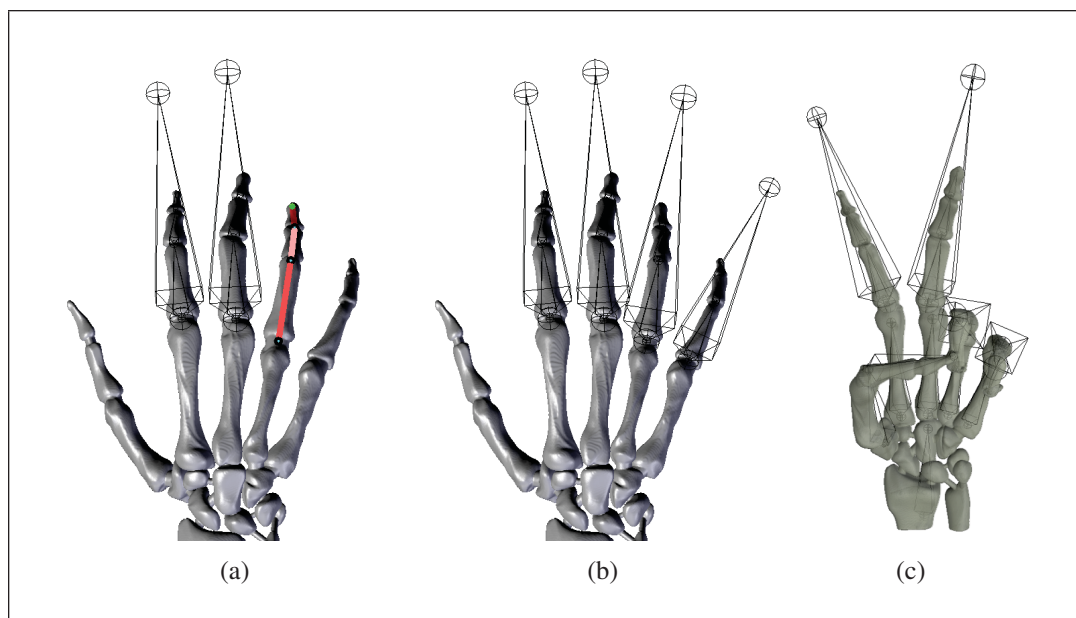


Figure 3.39 Patron de doigt avec flexeur adapté sur une main, (a) et (b) Processus d'adaptation, (c) Résultat après déformation.

- Un squelette tracé sur le tatou, un personnage assez fréquent dans les publications portant sur l'animation. La Figure 3.40.b montre le tracé d'un côté et le squelette généré de l'autre.

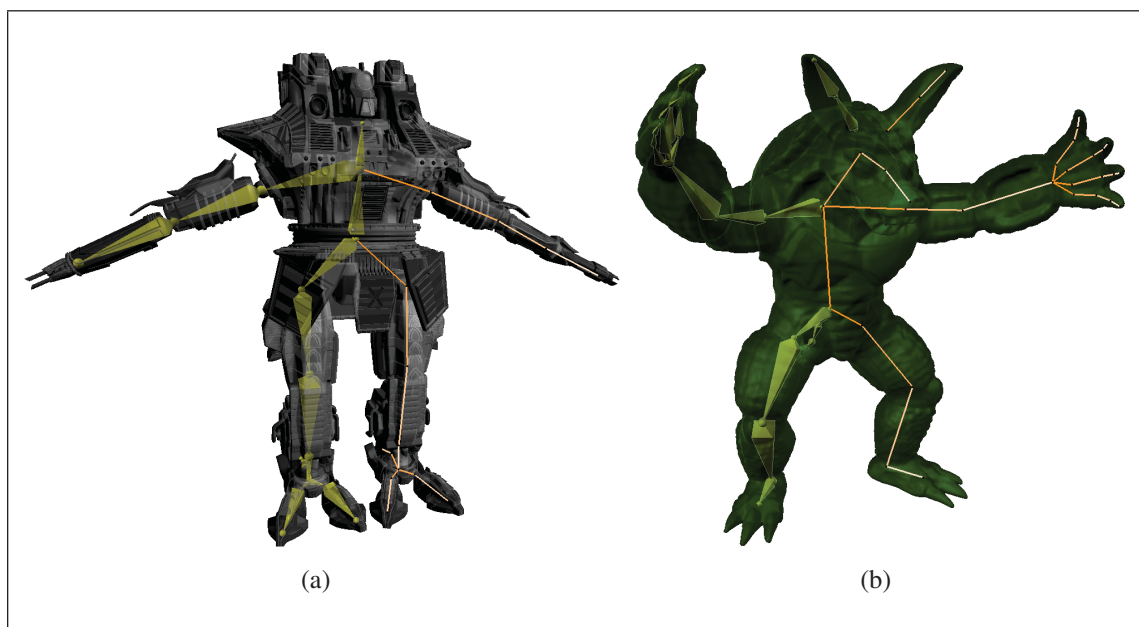


Figure 3.40 (a) Tracé et squelette sur un maillage de robot avec plusieurs couches de détails, (b) Tracé et squelette d'un tatou, un personnage classique du domaine de la recherche.

CHAPITRE 4

INTERPRÉTATION DES RÉSULTATS

4.1 Survol

Cette section contient l'analyse des techniques présentées au chapitre précédent. Premièrement, une revue du problème explique comment les éléments clés des techniques d'adaptation complètes et partielles par croquis résolvent les différentes composantes du problème. Deuxièmement, une analyse théorique et pratique de la complexité des algorithmes permet de considérer leur application à des situations réelles.

4.2 Correspondance entre fonctionnalités et problèmes

Les prochaines sections établissent des liens entre les différentes fonctionnalités des techniques proposées et des problèmes réels rencontrés lors de l'adaptation ou la création d'un squelette.

4.2.1 Adaptation complète

La technique d'adaptation correspond au problème de réutilisation de squelette en entier comme, par exemple, passer d'un personnage de géant à un nain.

4.2.1.1 Multirésolution

Une approche multirésolution au niveau du filtrage du graphe topologique et de l'algorithme de correspondance permet d'accomplir différentes tâches jugées essentielles.

Premièrement, cette méthode rend possible l'adaptation d'un squelette à un personnage contenant plus de détails que celui-ci. Par exemple, adapter un squelette d'humain vers un personnage de cartoon avec de grandes oreilles comme à la Figure 4.1. La technique multirésolution permet d'ignorer les parties du maillage qui ne correspondent à aucune partie du squelette.

Deuxièmement, les membres superflus du squelette, ceux n'ayant pas trouvé correspondance dans le graphe du maillage, peuvent facilement être éliminés. À remarquer que cette méthode est surtout utile pour les membres situés aux extrémités du squelette, c'est-à-dire les arcs terminaux du graphe. Par exemple, adapter un squelette d'humain normal avec une main à cinq doigts vers un personnage de cartoon à quatre doigts par main n'échoue pas à cause du doigt supplémentaire, tel que c'est le cas pour le personnage à la Figure 4.1.

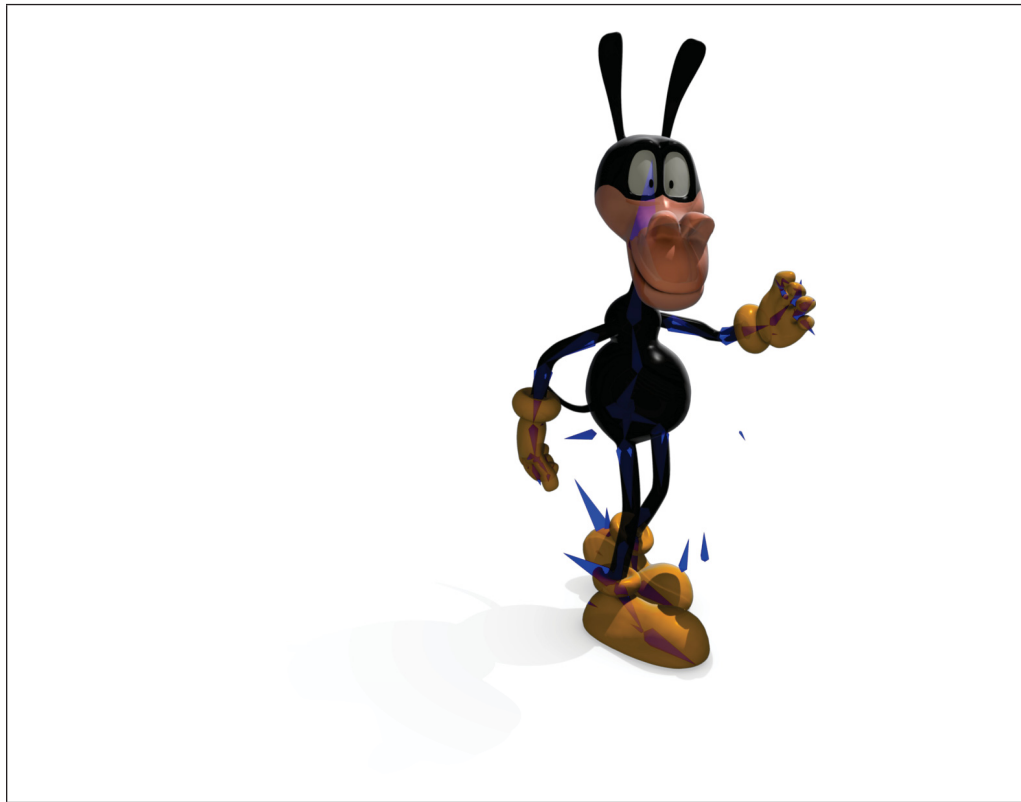


Figure 4.1 L'approche multirésolution permet d'ignorer les oreilles, absentes du squelette, de même que le cinquième doigt de chaque main, absent du maillage.

Le chemin parcouru par l'algorithme d'adaptation à travers le graphe multirésolution correspond à une fusion de différentes sections du graphe à différents niveaux de résolution. La Figure 4.2 montre les deux niveaux de résolution utilisés par le processus ainsi que la fusion résultante. Le niveau de résolution plus fin est utilisé pour faire apparaître les doigts alors que la résolution la plus grossière est utilisée pour le reste du corps.

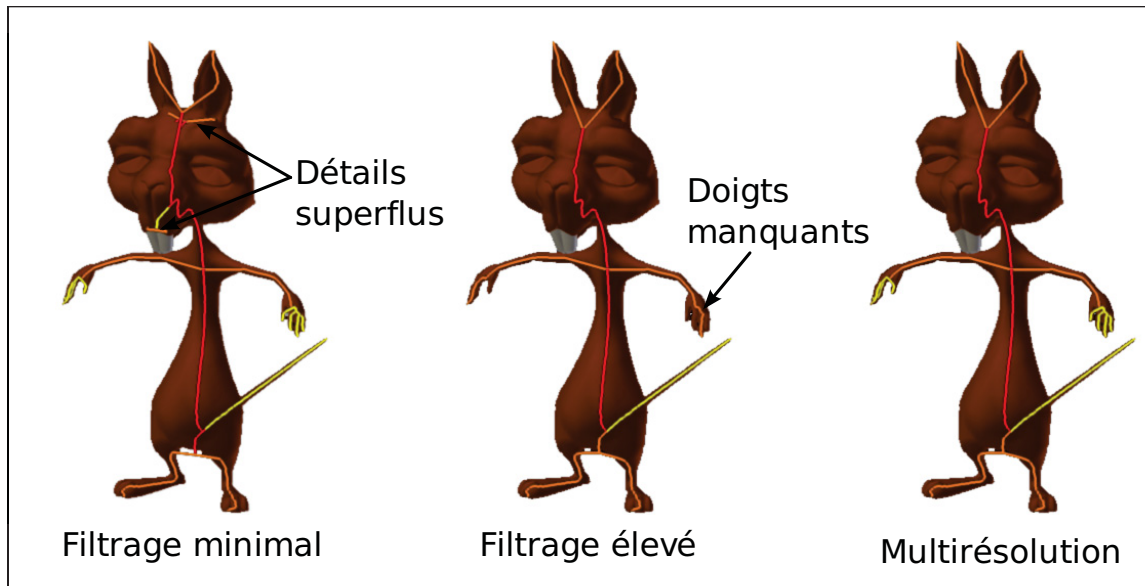


Figure 4.2 (de gauche à droite) Les deux niveaux de résolution utilisés pour l'adaptation du squelette d'écureuil ainsi que le graphe fusionné résultant contenant exactement les détails requis.

4.2.1.2 Orientation

L'orientation des joints d'un squelette est un des principaux attributs de celui-ci. Lors de l'adaptation d'un squelette vers un nouveau personnage, il est primordial de conserver une orientation compatible à chaque joint. Par exemple, si les genoux du personnage original pliaient sur leur axe des X local, les genoux du squelette adapté se doivent de plier sur ces mêmes axes pour qu'un animateur, familier avec l'original, ne soit pas surpris par un changement d'axe de rotation.

De plus, une orientation compatible au niveau des joints est aussi très importante pour tout ce qui a trait aux contraintes et limitations. Une contrainte qui force un os à pointer vers une cible aura l'air incongru si son orientation est modifiée drastiquement. De la même façon, une chaîne d'os avec limites de rotation (comme un bras robotique, par exemple) est grandement affectée par une mauvaise adaptation d'orientation aux joints. À cette fin, deux techniques ont été présentées à la Section 3.1.3.3.

Tout d'abord, la méthode basée sur la conservation des orientations par rapport au plan de rotation d'un joint donne de bons résultats dans la plupart des cas où les joints sont déjà fléchis sur le squelette original. Des joints fléchis aident aussi à avoir une bonne déformation de maillage, alors ce n'est pas une caractéristique rare sur un squelette de qualité. Les contrôleurs de roulements de pied et de flexion de doigts utilisés par les squelettes des Figures 3.21 et 4.1 dépendent fortement sur une orientation particulière au niveau des os, leur bon fonctionnement pointe donc vers la validité de cette méthode.

Par contre, cette technique n'est pas applicable aux joints de début de chaînes, où un plan de rotation est indéfini. Dans ces cas, la méthode par rotation correspondante à la transformation donne généralement des résultats acceptables quand la transformation entre le squelette original et son adaptation sont minimales, surtout quand les deux font face dans la même direction. Dans un cas extrême, si le personnage cible et le squelette à adapter sont dos à dos, les transformations découlant de l'adaptation des joints ne reflèteront aucunement la rotation de 180° que le squelette aurait dû subir.

4.2.1.3 Contrôleurs

Les contrôleurs sont essentiels aux squelettes d'animations professionnels, qu'ils servent à compléter la hiérarchie de joints de déformation ou à contrôler les transformations de ceux-ci. La Section 3.1.3.4 présente les différents cas de contrôleurs et la méthode développée pour en faire une adaptation adéquate.

La technique présentée est une des premières à faire une gestion saine des contrôleurs et donc rend la technique d'adaptation utilisable en industrie.

4.2.1.4 Noms des os

Les noms des os sont importants pour la correspondance avec les blocs d'animations et pour la symétrie sémantique entre les différents membres d'un personnage. La technique d'adaptation complète garde les noms originaux, ce qui est le comportement désiré.

4.2.2 Adaptation par croquis

L'adaptation par croquis corrige le manque de flexibilité de la méthode d'adaptation complète tout en étant plus précise pour le positionnement des joints en se basant sur des traits tracés par l'utilisateur. Bien que cette technique soit limitée aux chaînes simples, cette limitation est acceptable face aux avantages qu'elle offre : une interface plus naturelle et fluide.

4.2.2.1 Reconnaissance gestuelle

La reconnaissance de gestes est un principe d'interface très bien adapté aux tablettes à dessin. De ce fait, l'utilisation de cette technique permet une interface naturellement mieux adaptée à ce périphérique qu'une suite de raccourcis au clavier ou de boutons à l'écran, par exemple.

4.2.2.2 Insertion volumétrique

L'insertion volumétrique est une façon simple de positionner les joints d'un squelette à l'intérieur du maillage, où ceux-ci sont les plus efficaces pour une déformation sans artefact. Les deux façons de considérer les identificateurs d'objets (inter objet ou intra objet) couvrent la plupart des cas possibles pour l'obtention du résultat voulu.

4.2.2.3 Génération semi-automatisée

La technique de génération semi-automatisée présentée à la Section 3.2.4 permet de rapidement créer un squelette de base avec un placement de joints manuel ou avec des règles de subdivision précises spécifiées par l'utilisateur.

4.2.2.4 Adaptation de patron

L'adaptation de patron permet la réutilisation de parties de squelettes complexes dans le but de construire un squelette complet semblable aux squelettes existants.

4.2.2.5 Entités de nom

Les entités de nom permettent l'utilisation du même patron plusieurs fois dans un même squelette sans risquer un conflit de nom. L'automatisation de ces entités donne des résultats logiques sur plusieurs copies d'un patron, quel qu'en soit le nombre ou le côté du personnage où elles sont appliquées. La possibilité de définir manuellement le contenu des entités donne la flexibilité voulue pour les cas non couverts par l'automatisation.

4.3 Exemple d'utilisation

Le système de création et d'adaptation de squelettes par croquis est présentement utilisé dans le cadre de la production d'un court métrage (Tube Team, 2009). Le directeur et chef de projet pour Tube a été grandement impliqué dans la création de cette technique, donnant des commentaires et suggestions constructives à chaque étape de la recherche. Récemment, il a publié un vidéo démontrant l'utilisation de l'adaptation par croquis couplé à un outil de gestion de contraintes d'orientation. Ce vidéo est disponible en ligne à l'adresse suivante :

<http://tube.freefac.org/post/autorig-script-for-tube>

4.4 Performance

Cette section traite de l'aspect performance des algorithmes présentés plus tôt. Les performances sont analysées autant par complexité théorique (à l'aide de la notation grand O) que par des temps de calculs réels sur des exemples concrets.

4.4.1 Adaptation complète

Voici tout d'abord une analyse de la complexité théorique de l'algorithme d'adaptation complète de squelette. Celle-ci sera suivie d'une table de résultats montrant les temps pour les exemples présentés précédemment.

4.4.1.1 Graphe du maillage

Le calcul des poids sur le maillage est effectué à l'aide de l'algorithme de Dijkstra (résolution du plus court chemin dans un graphe). La complexité de cette opération est $O(a \log s)$ où s est le nombre de sommets et a le nombre d'arêtes dans le maillage.

La limite minimale théorique pour l'algorithme de création de graphe de Reeb est $O(s \log s)$. Les filtres post traitement (fusion des différents sous graphes, filtre multirésolution, ...) sont au pire $O(n^2)$ pour n le nombre de noeuds dans le graphe topologique, ce qui, excluant les cas dégénérés, est toujours plus rapide que la création du graphe.

4.4.1.2 Graphe du squelette

La création du graphe du squelette est en $O(j)$ pour j le nombre de joints alors que son post traitement (fusion des arcs, liens des contrôleurs, ...) est en $O(j^2)$. Dû au fait que le nombre de joints est significativement plus petit que l'ensemble de données représentant le maillage, cette étape est insignifiante en temps de calcul par rapport aux autres : le cas de test le plus lent est celui du squelette humain complet prenant seulement 2 ms.

4.4.1.3 Adaptation du squelette au maillage

La procédure de correspondance est bornée maximale en $O(ak\Delta_M)$ pour a le nombre d'arcs dans le graphe du squelette, k le nombre de niveaux de résolution et Δ_M le degré maximal pour un noeud du graphe topologique multirésolution du maillage.

Le placement de joints est un problème combinatoire ayant une complexité potentielle de $O(m^t)$ pour t le nombre de joints à placer et m le nombre de points à considérer. L'application de mémoïsation (une technique de programmation dynamique (Wikipedia, 2009b)) réduit le problème à un remplissage de tableau de données en $O(tm^2)$ de complexité temporelle et spatiale, échangeant une plus grande consommation de mémoire pour un temps de calcul plus court.

De plus, le placement de joints pour différents arcs est un problème intrinsèquement parallélisable et a été implémenté pour prendre complètement avantage des processeurs multi-coeurs d'aujourd'hui.

4.4.1.4 Temps d'exécution total

Les éléments qui ont le plus grand impact sur le temps de calcul sont les paramètres d'entrée. Pour un squelette avec beaucoup de joints internes, le temps de calcul est dominé par le placement de joints, alors que de larges maillages auront une plus longue phase de création de graphe topologique multirésolution (voir Table 4.1). Les maillages et les squelettes utilisés sont de qualité supérieure, ayant été utilisés dans des productions d'animations professionnelles (courts métrages, publicités pour la télévision ...). Les performances sur des personnages à grand nombre de polygones sont comparables aux solutions antérieures (Baran et Popović, 2007) tout en étant beaucoup plus rapides dans d'autres cas et en offrant des résultats sans précédant en ce qui a trait à la complexité des squelettes et à l'adaptation des contrôleurs.

Tableau 4.1 Performance (Core™2 Duo 2.4GHz / 4 GB RAM)

Maillages	Tailles		Temps de calcul		Total
	Faces	Joints	Graphe	Adaptation	
Papillon	2,428	51	0.3s	1.5s	1.8s
Mouche	8,834	41	1.2s	4.5s	6.0s
Rinky	8,562	42	1.9s	0.4s	2.7s
Mancandy	27,232	136	7.5s	23.8s	32.6s

4.4.2 Méthode par croquis

Au niveau de la méthode de création et d'adaptation par croquis, les algorithmes utilisés ne sont généralement pas très complexes. Le but étant, évidemment, de garder l'utilisation de cette technique le plus rapide possible.

4.4.2.1 Tracé du croquis

L'étape la plus complexe au niveau du tracé est l'insertion volumétrique. En théorie, celle-ci est en $O(f)$ pour f le nombre de face des maillages à analyser, mais des structures accélératrices (tels que des octree, BVH ou KD-tree) ou un simple élagage par volumes englobant, comme c'est le cas dans notre implémentation, coupe grandement dans la complexité réelle du problème. Dans tous les tests, la fonction d'insertion était assez rapide pour être utilisée en temps réel lors d'un tracé, même avec un maillage à haute densité de polygones.

4.4.2.2 Reconnaissance gestuelle

La reconnaissance gestuelle est faite en trois étapes. La première est une segmentation du tracé du geste en $O(p)$ pour p le nombre de points du geste. La deuxième étape est un calcul d'intersections avec les traits existants pour une complexité de $O(pt)$ (t le nombre de points des traits existants). La dernière étape est un calcul d'intersections du geste avec lui-même en $O(p^2)$. Ces variables étant relativement petites, la détection de geste s'effectue toujours en temps infinitésimal.

4.4.2.3 Adaptation de patron

L'adaptation de patron réutilise la fonction de positionnement de joints de la technique d'adaptation complète. La complexité est donc la même que celle décrite à la Section 4.4.1.3 à la différence que le trait fait par l'utilisateur (plus particulièrement les points 3D qu'il contient) est utilisé à la place des arcs topologiques du maillage.

CHAPITRE 5

DISCUSSION

5.1 Survol

Cette section contient une comparaison des techniques présentées avec des travaux de recherches antérieures comparables ainsi qu'une analyse des résultats de l'évaluation de la technique par croquis auprès d'utilisateurs experts.

5.2 Comparaison avec différentes méthodes

Le Tableau 5.1 contient une comparaison point par point entre la méthode d'adaptation de squelette présentée précédemment (Section 3.1) et d'autres basées sur des travaux de recherches antérieurs. Cette comparaison démontre que notre méthode, bien que non topologiquement indépendante comme celle de Baran et Popović (2007), est plus apte à être utilisée dans le contexte de l'animation de personnage professionnelle, spécialement grâce à l'utilisation d'un système multirésolution et du bon traitement de contrôleurs.

5.2.1 Génération automatisée

Les méthodes de génération automatisée sont limitées dans leur application par le peu d'information disponible pour guider la création de joints. Soit celles-ci sont basées uniquement sur le maillage, ce qui est problématique lorsqu'il n'a pas d'indication géométrique (tels que des fléchissements aux joints), soit elles font appel à des informations externes : base de connaissances anatomiques (Dellas et al., 2007) ou patrons simples de bipèdes ou quadrupèdes (Aujay et al., 2007). Le problème de ces approches est qu'elles limitent l'application de la technique à des situations connues alors que la technique d'adaptation complète, présentée précédemment, est applicable à tous les cas où un squelette préexistant semblable est disponible, peu importe si celui-ci correspond à ceux utilisés lors de la recherche. De plus, les algorithmes de génération

Tableau 5.1 Comparaison avec les travaux antérieurs

	(Baran et Popović, 2007)	(Aujay et al., 2007)	Méthode proposée
Graphe topologique	volumétrique	géométrique	géométrique
Complexité des squelettes	simple	patron bipède et quadrupède	squelettes d'animation professionnels
Contrôleurs	non	non	oui
Placement des joints	poids fixes	patrons rigides	poids contrôlables et balancés
Filtrage	n/a	filtre simple	multirésolution
Indépendance topologique	complètement	non	par multirésolution
Indépendance d'orientation	non	oui	oui

ne servent habituellement qu'à créer un squelette de déformation, les utilisateurs doivent par la suite rajouter les contrôleurs par eux-mêmes.

5.2.2 Adaptation

Du côté des algorithmes d'adaptation, les recherches antérieures sont assez peu nombreuses. Une des plus récentes (Baran et Popović, 2007) ne traite que des squelettes simples sans détails aux extrémités (aucun doigt par exemple) et sans contrôleurs. D'autre part, leur algorithme est capable d'adapter un squelette à un maillage sans aucun lien topologique (un squelette humain vers un maillage de beigne par exemple) alors que l'algorithme présenté plus tôt dans ce document se limite aux sources topologiquement semblables. Par contre, cette indépendance topologique coûte cher au niveau de la complexité et des temps de calcul.

5.3 Évaluation auprès des utilisateurs

La méthode d'adaptation par croquis a été évaluée par un groupe d'utilisateurs. Ce groupe était composé de deux utilisateurs experts et d'un novice. Les deux utilisateurs expert sont des professionnels du domaine avec de l'expérience en animation de personnages dans plusieurs

logiciels, incluant Blender. L'utilisateur novice vient de finir sa formation en infographie et ne connaît Blender que très peu.

Voici en quoi consistait sommairement l'évaluation, voir l'Annexe II pour plus de détails.

- Une fiche d'identification à remplir
- Une explication sommaire du logiciel
- Quelques fichiers d'exemples
- Une liste de tâches suggérées qui touchent à un maximum de cas d'utilisation possibles
- Un questionnaire d'évaluation du logiciel à remplir après avoir effectué les tâches

Le Tableau 5.2 présente les réponses des utilisateurs aux questions portant sur la technique en générale ainsi que sur son interface personne-machine. En générale, les utilisateurs ont trouvé l'interface familière et intuitive. Ils ont aussi tous exprimé le désir d'utiliser un système de création et d'adaptation de squelettes semblable pour de futures tâches de création de squelettes, autant dans leurs réponses quantitatives que dans leurs commentaires qualitatifs. Quant à lui, le Tableau 5.3 présente les réponses des utilisateurs aux questions portant sur l'utilité des différentes fonctionnalités de la méthode.

Le questionnaire demandait aussi d'évaluer le temps gagné par l'utilisation de la technique par rapport aux techniques de création de squelettes traditionnelles. Dans le cas d'un personnage où il est impossible de réutiliser des patrons de squelettes, les réponses variaient entre 5% et 30% de temps en moins, alors que pour un personnage pour lequel des patrons sont utilisables, le temps gagné estimé varie entre 10% et 50%. À noter que tous les utilisateurs s'entendaient pour dire que la technique introduite devrait réduire de façon significative le temps nécessaire à la création du squelette d'un nouveau personnage. De plus, les estimations les plus basses des deux cas (5% et 10%) étaient accompagnées de suggestions précises pour rendre la technique plus utile et plus rapide.

Tableau 5.2 Évaluation de la technique et de son interface auprès d'utilisateurs

Critères	Utilisateurs		
	A	B	C
La technique est intuitive	4	3	4
L'interface est familière	3	4	4
Le système répond bien aux actions de l'utilisateur	3	4	3
Vous êtes confortables avec l'utilisation du système	3	3	5
Le système est facile d'utilisation	3	3	4
Vous êtes satisfaits du système	4	3	4
L'interface est fluide	3	4	3
Vous voudriez utiliser ce système dans le future	4	4	5

Les utilisateurs devaient évaluer leur accord ou désaccord avec chaque critère selon l'échelle suivante : (pas du tout) 1 .. 2 .. 3 .. 4 .. 5 (complètement)

Tableau 5.3 Évaluation de l'utilité des différentes fonctionnalités auprès d'utilisateurs

Fonctionnalités	Utilisateurs		
	A	B	C
Traits continus	2	2	2
Traits polygonaux	4	5	5
Interface gestuelle	3	5	3
Insertion volumétrique	4	5	5
Contraintes aux joints existants	4	4	NA
Subdivision par longueur	NA	3	3
Subdivision adaptative	NA	2	5
Subdivision fixe	NA	NA	2
Adaptation de patron	NA	4	4
Patron de nom	NA	5	5
Noms automatiques (S et N)	5	5	4
Surimpression	NA	4	NA

Les utilisateurs devaient évaluer l'utilité de chaque fonctionnalité selon l'échelle suivante : (pas du tout utile) 1 .. 2 .. 3 .. 4 .. 5 (essentielle) ou NA s'ils n'avaient pas utilisé la fonctionnalité en question.

De façon générale, les utilisateurs ont apprécié la technique et ont offert des commentaires constructifs pour l'améliorer, tel que des suggestions d'interface pour un meilleur accès aux patron ainsi qu'une rétroaction plus complète et informative lors du tracé de ceux-ci.

CONCLUSION

Problème

Le problème initial était celui-ci : Trouver des techniques pour favoriser la réutilisation de squelettes entre différents personnages dans le contexte d'animation 3D de niveau professionnel (composée de squelettes multicouches complexes utilisant des contrôleurs de déformation).

Solution

Les solutions apportées consistent en une technique d'adaptation complète par correspondance entre le graphe topologique du squelette et le graphe topologique multirésolution du maillage cible, ainsi qu'une technique de création semi-automatique et adaptation de patron par croquis avec insertion volumétrique. Dans les deux cas, les contrôleurs sont gérés correctement et les algorithmes sont assez rapides pour une utilisation sur des squelettes et personnages professionnels.

Limitations

Pour l'instant, la méthode est limitée aux squelettes autonomes (où la déformation et les contrôles sont des os du squelette). Ceci exclu les squelettes utilisant des volumes ou des courbes de déformation ainsi que ceux qui se servent d'objets externes comme contrôleurs. Cette limite en est une d'implémentation. Pour les contrôleurs externes, et la plupart des cas de volumes de déformation, la méthode proposée pour le traitement de contrôleurs ferait très bien l'affaire. Dans le cas des courbes de déformation, elles pourraient être découpées en segments linéaires et traitées comme une chaîne de joints.

Travaux futurs

Pour l'adaptation complète : utiliser le personnage initial en plus du squelette. Associer le squelette original au graphe topologique du maillage et établir des règles de positionnement

(par exemple, le joint des épaules peut être un peu plus haut que le graphe topologique du maillage).

Au niveau de la création automatique de joints le long d'une courbe, un nouveau critère d'ajout pourrait être la variance du volume du maillage le long de la courbe. Les joints étant souvent positionnés au maximum ou minimum d'épaisseur d'un membre, cette information pourrait être exploitée pour le positionnement.

Un système de règles de positionnement pourrait aussi être utilisé pour établir la position ou l'orientation de certains contrôleurs ou os de déformation selon des critères établis par l'utilisateur. Voici deux critères suggérés par un utilisateur :

- Pôle de chaîne IK toujours aligné sur l'axe des Z
- Contrôle de rotation du talon aligné en hauteur avec l'avant du pied

En ce qui a trait à la méthode de création et d'adaptation par croquis, la technique d'insertion volumétrique peut très bien aussi servir à estimer le rayon de l'enveloppe d'un os, pour une déformation par enveloppe au lieu d'un LBS (voir Annexe I et Figure 5.1) ou pour la génération d'une cage à faible résolution de polygones. De telles cages sont utilisées de plus en plus souvent comme intermédiaire de déformation entre un maillage à haute densité et un squelette et pourraient très bien être générées avec une technique de surface à distance constante (Johnson et Cohen (2009) par exemple) au lieu d'être faites à la main.

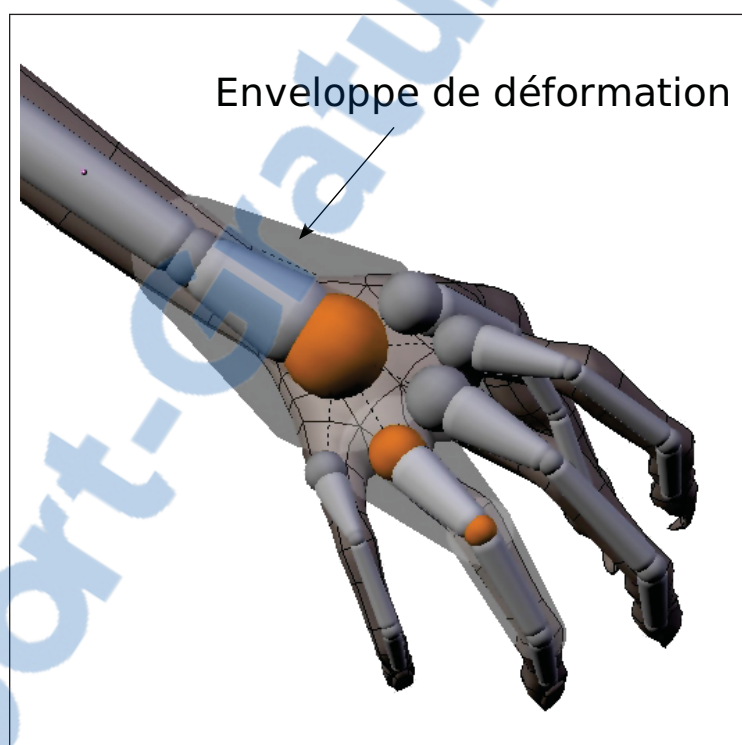


Figure 5.1 Une main déformée par enveloppe.

ANNEXE I

COMPOSITION D'UN SQUELETTE

Un squelette typique est composé d'une hiérarchie de joints liés entre eux par des relations de parent-enfant. Deux joints reliés entre eux forment un os. La Figure I.1 montre un squelette simple formant une hiérarchie de joints et d'os.

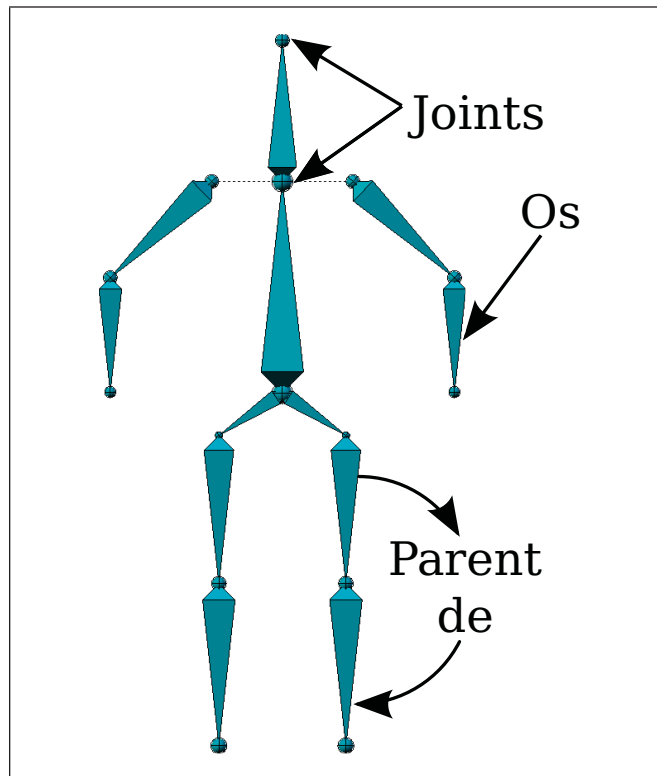


Figure I.1 Un squelette simple.

1 Cinématique

Le type de cinématique définit comment les chaînes de joints sont animées. Deux cinématiques existent, la cinématique directe et inverse, les deux étant utiles dans différents cas d'utilisation.

1.1 Cinématique directe

Chaque joint de la chaîne est modifié par rotation successive. La pose finale est le résultat de ces rotations comme le montre la Figure I.2. Dans la plupart des cas, il est difficile de positionner précisément le bout de la chaîne dans l'espace 3D, par contre la rotation de chaque joint est explicitement définie par l'utilisateur.

De plus, lors d'une animation, il est presque impossible de garder le bout de la chaîne fixe dans l'espace (un pied au sol dans un cycle de marche par exemple).

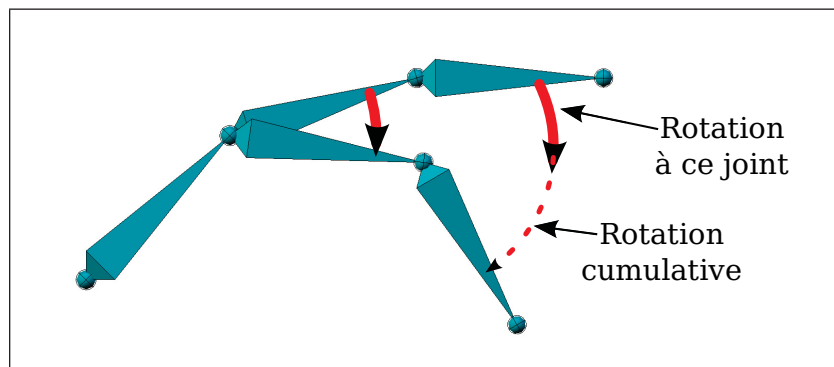


Figure I.2 Chaîne d'os modifiée par cinématique directe.

1.2 Cinématique inverse

La position du bout de la chaîne est définie par la position d'une *cible*. Un système de résolution de contrainte fait subir des rotations aux joints de la chaîne pour que le bout soit le plus près possible de la cible tout en minimisant les rotations par rapport à la pose précédente, comme le montre la Figure I.3.

Dans la plupart des cas, il est difficile de définir précisément la position de chaque joint le long de la chaîne. Par contre, lors d'une animation, la position du bout de la chaîne restera fixée à la cible dans les limites du possible.

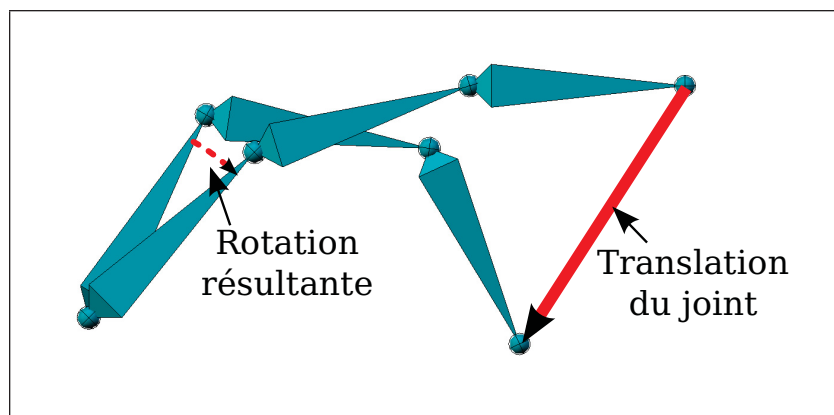


Figure I.3 Chaîne d'os modifiée par cinématique inverse.

2 Déformation

La méthode de déformation la plus utilisée est le *Linear Blend Skinning* (LBS). C'est une technique qui assigne à chaque sommet d'un maillage un poids correspondant à chaque os. Si v_j est la position du sommet j , T^i est la transformation du $i^{\text{ème}}$ os et w_j^i est le poids du $i^{\text{ème}}$ os au $j^{\text{ème}}$ sommet, LBS donne la position du sommet transformé j comme $\sum_i w_j^i T^i(v_j)$.

3 Contrôleurs

Les contrôleurs sont des os qui n'affectent pas directement la déformation du maillage. Ils peuvent servir de simple support hiérarchique (établir une racine commune à deux chaînes d'os de déformation, ...) ou à contrôler la transformation d'un ou des os de déformation. Ce deuxième cas d'utilisation est le plus utile parce qu'il permet de simplifier de beaucoup l'animation d'un squelette.

Les prochaines sections contiennent différents usages courant de contrôleurs.

3.1 Racine

Les contrôleurs servant de racine permettent d'avoir un parent hiérarchique commun pour plusieurs chaînes d'os de déformation. Par exemple, un squelette humain traditionnel aura deux chaînes orientées vers le bas (les jambes) et une orienté vers le haut (le haut du corps). Ces trois chaînes sont habituellement les enfants d'un contrôleur correspondant au bassin. Le contrôle de bassin peut donc être bougé et autant les jambes que le haut du corps suivront le mouvement.

3.2 Roulement du pied

Un contrôleur de roulement de pied permet de faire rouler celui-ci vers l'arrière (talon au sol, orteils levées) et vers l'avant (orteils au sol, talon levé) en un mouvement continu, comme le démontre la Figure I.4.

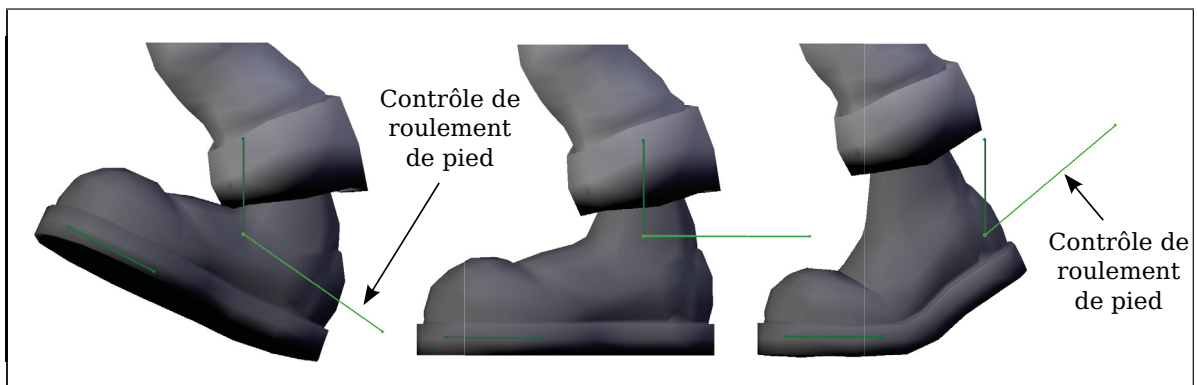


Figure I.4 Contrôle de roulement de pied.

3.3 Transition de cinématique

Un système de contrôle de transition de cinématique est un système complexe de contrôleurs qui permet d'animer une chaîne d'os (souvent un bras) autant avec une cinématique directe qu'inverse et offre aussi une manière d'effectuer des transitions graduelles entre ces deux modes.

Généralement, ceci est accompli avec au moins deux chaînes de contrôleurs correspondant directement à la chaîne de déformation. Une de ces deux chaînes est contrôlée par cinématique directe alors que l'autre l'est par cinématique inverse. La chaîne de déformation est alors configurée pour copier les rotations d'une ou l'autre des deux chaînes de contrôle en totalité ou en partie (pour permettre une transition). Généralement, un contrôle supplémentaire permet de faire la transition entre copier l'une ou l'autre des cinématiques.

3.4 Flexion

Un contrôle de flexion permet de fléchir plusieurs joints en même temps. Ce type de contrôleur est particulièrement utilisé pour des doigts (Figure I.5).

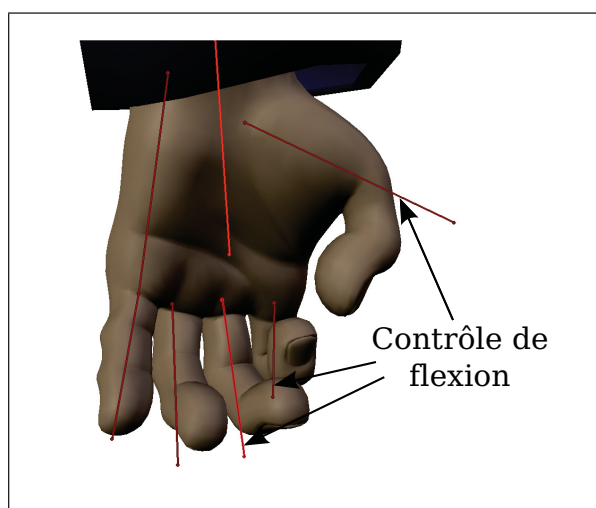


Figure I.5 Contrôle de flexion.

3.5 Étirement et compression

L'étirement et la compression (traduction libre de *Stretch and Squash*) est une technique d'animation traditionnelle applicable aux personnages ou objets en mouvement. Le personnage ou l'objet en question est comprimé ou étiré pour accentuer les mouvements et donner une apparence réelle et solide. La Figure I.6 montre son application sur une balle qui rebondit. Le livre *The Illusion of Life* de Disney (Thomas et Johnston, 1995) traite de ce sujet et de plusieurs

autres en lien à l'animation traditionnelle. Plusieurs des sujets traités sont maintenant aussi appliqués à l'animation 3D.

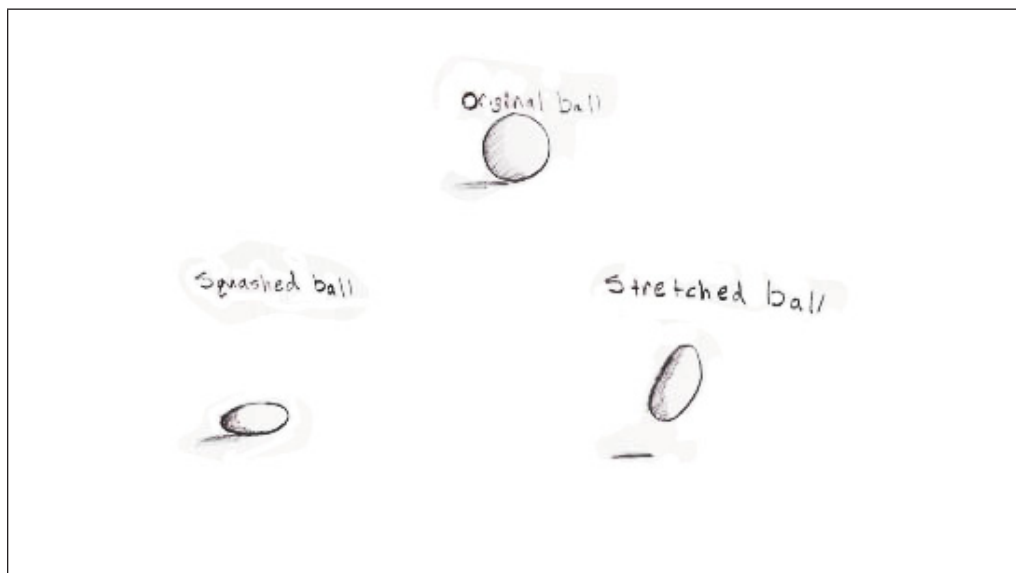


Figure I.6 Exemple d'étirement et compression d'une balle bondissante.

Il est possible d'automatiser cet effet dans un squelette à l'aide de contrôleurs réglant l'étirement ou la compression des os de déformation dépendant de leur position par rapport à une référence neutre.

ANNEXE II

ÉVALUATIONS PAR DES UTILISATEURS

1 Matériel envoyé aux utilisateurs

Voici en quoi consistait le matériel envoyé aux utilisateurs évaluateurs.

- Une fiche d’identification à remplir
- Une explication sommaire du logiciel
- Quelques fichiers d’exemples
- Une liste de tâches suggérées qui touchent à un maximum de cas d’utilisation possibles
- Un questionnaire d’évaluation du logiciel à remplir après avoir effectué les tâches

À noter que toutes les informations et textes envoyés aux participants étaient en langue anglaise, la majorité de ces derniers étant anglophones.

Tableau II.1: Fiche d’identification

Name	
Gender	
Age (or age range)	
Occupation/Field of expertise	
Do you have access/experience with a graphic tablet (such as wacom) with pen input ?	YES / NO
Have you used 3D rigging tools before ?	YES / NO
If YES :	
How many years/projects have you worked as a rigger ?	
Do you have experience with more than one rigging software/tool ?	YES / NO
List them	
Have you used Blender before ?	YES / NO
If YES :	
Do you use Blender for professional work ?	YES / NO
Do you have experience with the rigging tools in Blender ?	YES / NO
Is rigging an important part of your experience with Blender ?	YES / NO
Have you noticed any problems or difficulties with the rigging user interfaces that you have used before ? These might be general problems in many rigging tools, or problems specific to one tool. Do you have any suggestions or ideas on how they might be improved ?	

1.1 Fiche d'identification

La fiche d'identification contenait les questions présentées au Tableau II.1. Celles-ci permettaient de mieux cerner les habiletés, connaissances et expériences des utilisateurs.

1.2 Explication du logiciel

L'explication du logiciel est contenue sur une page du wiki officiel de Blender.

<http://wiki.blender.org/index.php/User:Theeth/etch-a-ton>

1.3 Fichiers d'exemples

Les fichiers d'exemples inclus avec le formulaire d'évaluation contenaient quatre personnages humanoïdes (Figure II.1), trois personnages non-humoïdes (Figure II.2) ainsi qu'un fichier contenant des patrons de jambes et de doigts.

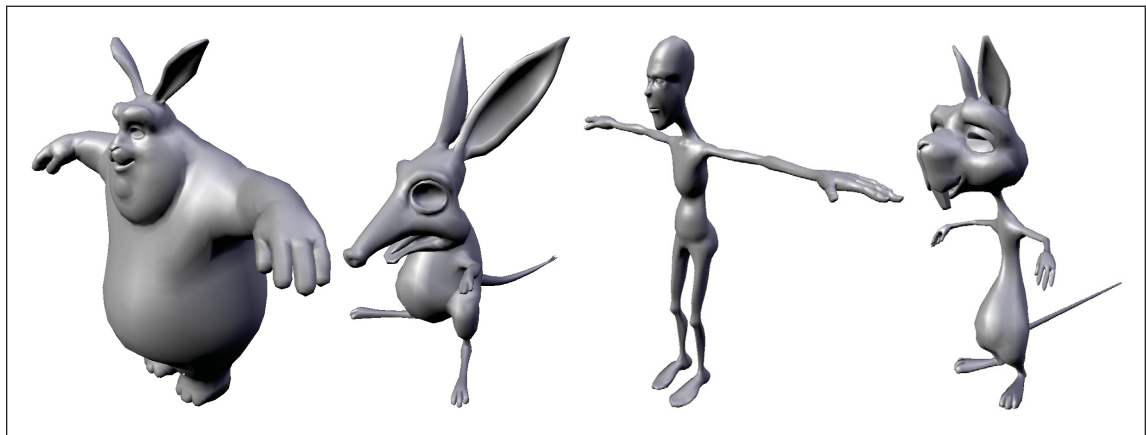


Figure II.1 Personnages exemples humanoïdes envoyés aux utilisateurs.

1.4 Liste de tâches

Voici la liste de tâches présentées aux utilisateurs.

- a. Creation : Try creating rigs for different models using the sketch conversion methods (length, correlation, fixed).

Some models are provided if you want to use them or use your owns.

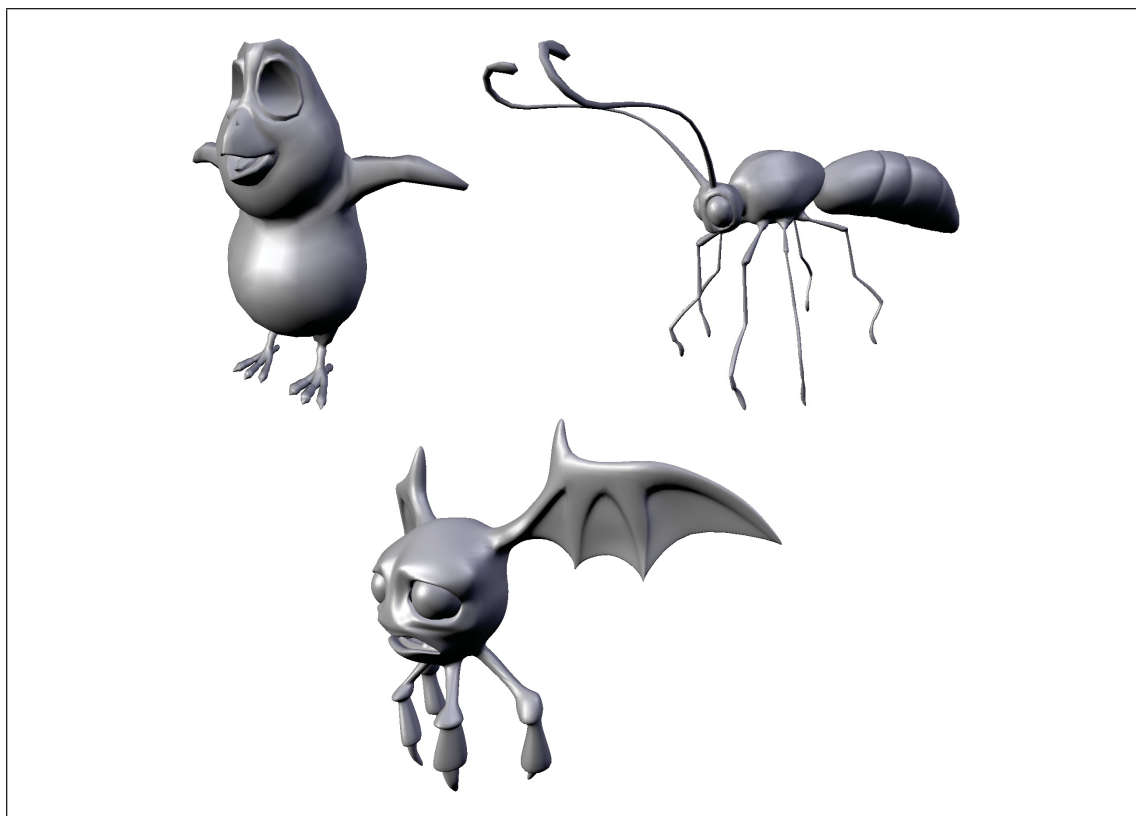


Figure II.2 Personnages exemples non-humanoïdes envoyés aux utilisateurs.

- Humanoid : The file human.blend contains four different scenes with humanoid/bipedal models
 - Non-humanoid : The file non_human.blend contains three different scenes with non humanoid characters
- b. Template Usage Templates for an arm, finger and leg are available in templates.blend. Append them in the file you want to work with.

Some models are provided if you want to use them or use your owns.

- Humanoid : The file human.blend contains four different scenes with humanoid/bipedal models
 - Non-humanoid : The file non_human.blend contains three different scenes with non humanoid characters
- c. Template Creation
- Simple : Try creating a simple armature template : single deform bone chain, maybe some control bones (IK solver, pole, ...).

- Templating Names : Try making a template with templating names (&N and &S). This enables you to use a single template to quickly create correctly named copies of a single limb (like fingers, tentacles, ...). See the TemplateFinger armature with the provided templates for a good example.

1.5 Questionnaire d'évaluation

Le Tableau II.2 contient les questions d'évaluation du logiciel posées aux utilisateurs après leur période d'essai.

Tableau II.2: Questionnaire d'évaluation

(You are free to elaborate on any questions where you'd feel this would help enhance the experience/features)

Did you find the new rigging techniques intuitive ?

(not at all) 1 .. 2 .. 3 .. 4 .. 5 (completely)

Did you find the sketching interface familiar ?

(not at all) 1 .. 2 .. 3 .. 4 .. 5 (completely)

Did you find that the software gave enough feedback in response to your actions ?

(not at all) 1 .. 2 .. 3 .. 4 .. 5 (completely)

During the scripted tasks, were you able to accomplish what you set out to do ?

(not at all) 1 .. 2 .. 3 .. 4 .. 5 (completely)

What was the most challenging aspect(s) of the tasks (check all appropriate)

Adapting your process to the new technique _____

Results were often unexpected _____

Sketching interface was not precise enough _____

Other : _____

What operational mode did you feel most comfortable with / was the most useful ?

Normal mode with gestures _____

Quick mode with instant conversion _____

How would you rate the usefulness of the bone creation methods (length / correlation / fixed) ?

(not at all) 1 .. 2 .. 3 .. 4 .. 5 (completely)

How would you rate the usefulness of the template method (retarget) ?

(not at all) 1 .. 2 .. 3 .. 4 .. 5 (completely)

When rigging a new character, do you think you would be able to complete the rigging faster with the new techniques ?

YES / NO

If yes, what percentage of time do you think you would save ?

Percentage time saved _____

If no, explain why ?

When rigging a new character for which you wouldn't be able to reuse existing parts, do you think you would be able to complete the rigging faster with the new techniques ?

	YES / NO
If yes, what percentage of time do you think you would you save ?	
Percentage time saved	___
If no, explain why ?	
Rate the usefulness of each of these features on a scale of 1 (not useful at all) to 5 (very useful/essential). Use NA if you didn't use the feature.	
Continuous strokes (draw)	___
Poly line (clicks)	___
Gestures (add joint, trim, delete, convert)	___
Embedding sketch inside geometry (Shift)	___
Snapping to joints (Ctrl)	___
The different bone creation methods :	
Length	___
Correlation	___
Fixed	___
Retarget	___
Template renaming (&N, &S)	___
Autonaming (automatic side and increment)	___
Other :	___
Rate each of the following on a scale of 1 (not at all) to 5 (very much)	
Now that you've had a first experience with the new rigging technique	
Do you feel comfortable using it ?	___
Did you find it easy to operate ?	___
How satisfied were you with the new technique ?	___
Would you agree that the sketching interface is "fluid" ?	___
Would you like to use this technique in the future ?	___
Did you use the gesture interface ? Do you think it is useful ?	
	YES / NO
If you had to perform similar tasks in the future, would you want to use the new rigging techniques (in addition to existing methods) ?	
	YES / NO
If not, what methods would you prefer and why ?	
Do you have any suggestions to improve the new rigging techniques (as far as results goes/user experience/...)?	
	YES / NO
Any other comments ?	

LISTE DES RÉFÉRENCES

- Alex, J. (2005). *Hybrid sketching : a new middle ground between 2- and 3-d*. PhD thesis, Cambridge, MA, USA. Supervisor-Julie Dorsey and Supervisor-William L. Porter.
- Au, O. K.-C., Tai, C.-L., Chu, H.-K., Cohen-Or, D., et Lee, T.-Y. (2008). Skeleton extraction by mesh contraction. *ACM Transactions on Graphics*, 27(3).
- Aujay, G., Hétry, F., Lazarus, F., et Depraz, C. (2007). Harmonic skeleton for realistic character animation. *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 151–160, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- Bae, S.-H., Balakrishnan, R., et Singh, K. (2008). Ilovesketch : as-natural-as-possible sketching system for creating 3d curve models. *UIST '08 : Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 151–160, New York, NY, USA. ACM.
- Baran, I. et Popović, J. (2007). Automatic rigging and animation of 3D characters. *ACM Transactions on Graphics*, 26(3) :72.
- Bau, O. et Mackay, W. E. (2008). Octopocus : a dynamic guide for learning gesture-based command sets. *UIST '08 : Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 37–46, New York, NY, USA. ACM.
- Baudel, T. (1994). A mark-based interaction paradigm for free-hand drawing. *UIST '94 : Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 185–192, New York, NY, USA. ACM.
- Bi, X., Moscovich, T., Ramos, G., Balakrishnan, R., et Hinckley, K. (2008). An exploration of pen rolling for pen-based interaction. *UIST '08 : Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 191–200, New York, NY, USA. ACM.

- Blanco, F. R. et Oliveira, M. M. (2008). Instant mesh deformation. *I3D '08 : Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pages 71–78, New York, NY, USA. ACM.
- Brunner, D. et Brunnett, G. (2008). Fast force field approximation and its application to skeletonization of discrete 3D objects. *Computer Graphics Forum, Vol. 27, No. 2 (2008), Eurographics 2008*, pages 261–270.
- Cornea, N. D., Silver, D., et Min, P. (2007). Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13(3) :530–548. Member-Deborah Silver.
- Dellas, F., Moccozet, L., Magnenat-Thalmann, N., Mortara, M., Patané, G., Spagnuolo, M., et Falcidieno, B. (2007). Knowledge-based extraction of control skeletons for animation. *SMI '07 : Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007*, pages 51–60.
- Hassan, M. F. et Dodgson, N. A. (2005). Reverse subdivision, in. *Sabin (Eds.), Advances in Multiresolution for Geometric Modelling*, pages 271–283. Springer.
- Hecker, C., Raabe, B., Enslow, R. W., DeWeese, J., Maynard, J., et van Prooijen, K. (2008). Real-time motion retargeting to highly varied user-created morphologies. *SIGGRAPH '08 : ACM SIGGRAPH 2008 papers*, pages 1–11, New York, NY, USA. ACM.
- Hilaga, M., Shinagawa, Y., Kohmura, T., et Kunii, T. L. (2001). Topology matching for fully automatic similarity estimation of 3D shapes. *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212, New York, NY, USA. ACM.
- Igarashi, T. et Hughes, J. F. (2006). A suggestive interface for 3d drawing. *SIGGRAPH '06 : ACM SIGGRAPH 2006 Courses*, page 10, New York, NY, USA. ACM.
- Igarashi, T., Matsuoka, S., et Tanaka, H. (1999). Teddy : a sketching interface for 3d freeform design. *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics*

- and interactive techniques*, pages 409–416, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- Igarashi, T., Moscovich, T., et Hughes, J. F. (2005). As-rigid-as-possible shape manipulation. *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 1134–1141, New York, NY, USA. ACM.
- Johnson, D. E. et Cohen, E. (2009). Computing surface offsets and bisectors using a sampled constraint solver. *GI '09 : Proceedings of Graphics Interface 2009*, pages 31–37, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.
- Ju, T., Zhou, Q.-Y., et Hu, S.-M. (2007). Editing the topology of 3d models by sketching. *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, page 42, New York, NY, USA. ACM.
- Katz, S. et Tal, A. (2003). Hierarchical mesh decomposition using fuzzy clustering and cuts. *SIGGRAPH '03 : ACM SIGGRAPH 2003 Papers*, pages 954–961, New York, NY, USA. ACM.
- Kho, Y. et Garland, M. (2005). Sketching mesh deformations. *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 934–934, New York, NY, USA. ACM.
- LaViola, J. J. (2007). Sketching and gestures 101. *SIGGRAPH '07 : ACM SIGGRAPH 2007 courses*, page 2, New York, NY, USA. ACM.
- Li, X., Fei, G., et Zhu, Y. (2007). Ismas : An interactive sketch modeling and animation system. *Edutainment*, pages 687–700.
- Magenat-Thalmann, N., Laperrière, R., et Thalmann, D. (1988). Joint-dependent local deformations for hand animation and object grasping. *Proceedings on Graphics interface '88*, pages 26–33, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.
- Blender-Foundation (2008). Big buck bunny. <http://www.bigbuckbunny.org/>.
- Blender-Foundation (2009). Blender 3D. <http://www.blender.org>.

- Free Software Foundation (2007). Gnu general public license.
- Tube Team (2009). Tube movie project. <http://tube.freefac.org/>.
- Morse, M. (1932). *The calculus of variations in the large*, volume 18 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, RI.
- Pascucci, V., Scorzelli, G., Bremer, P.-T., et Mascarenhas, A. (2007). Robust on-line computation of Reeb graphs : simplicity and speed. *ACM Transactions on Graphics*, 26(3) :58.
- Poirier, M. et Paquette, E. (2009). Rig retargeting for 3d animation. *GI '09 : Proceedings of Graphics Interface 2009*, pages 103–110, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.
- Reeb, G. (1952). *Sur certaines propriétés topologiques des variétés feuilletées*. *Actualités Sci. Ind.*, no. 1183. Hermann & Cie., Paris. *Publ. Inst. Math. Univ. Strasbourg* 11, pp. 5–89, 155–156.
- Shapira, L., Shamir, A., et Cohen-Or, D. (2008). Consistent mesh partitioning and skeletonization using the shape diameter function. *The Visual Computer*, 24(4) :249–259.
- Sutherland, I. E. (1963). Sketchpad : a man-machine graphical communication system. *AFIPS '63 (Spring) : Proceedings of the May 21-23, 1963, spring joint computer conference*, pages 329–346, New York, NY, USA. ACM.
- Thomas, F. et Johnston, O. (1995). *The illusion of life : Disney animation*.
- Tierny, J., Vandeborre, J.-P., et Daoudi, M. (2008). Enhancing 3D mesh topological skeletons with discrete contour constrictions. *The Visual Computer*, 24(3) :155–172.
- Wade, L. et Parent, R. E. (2000). Fast, fully-automated generation of control skeletons for use in animation. *CA '00 : Proceedings of the Computer Animation*, page 164. IEEE Computer Society.
- Wikipedia (2009b). Memoization — wikipedia, the free encyclopedia. [Online ; accessed 15-June-2009].

Zhou, X., Ren, X., et Hui, Y. (2008). An empirical comparison of pen pressure and pen tilt input techniques. *ISPA '08 : Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications*, pages 982–989, Washington, DC, USA. IEEE Computer Society.

Zimmermann, J., Nealen, A., et Alexa, M. (2007). Silsketch : automated sketch-based editing of surface meshes. *SBIM '07 : Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling*, pages 23–30, New York, NY, USA. ACM.