



**HAL**  
open science

## Construction automatique d'un squelette pour l'animation de personnages

Grégoire Aujay, Franck Hétroy, Francis Lazarus

► **To cite this version:**

Grégoire Aujay, Franck Hétroy, Francis Lazarus. Construction automatique d'un squelette pour l'animation de personnages. AFIG, Nov 2006, Bordeaux France. inria-00110047v2

**HAL Id: inria-00110047**

**<https://inria.hal.science/inria-00110047v2>**

Submitted on 7 Dec 2006 (v2), last revised 18 Jan 2010 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Construction automatique d'un squelette pour l'animation de personnages

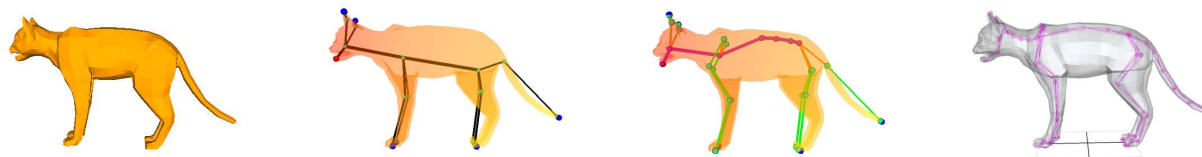
G. Aujay<sup>1</sup>, F. Hétry<sup>1</sup> et F. Lazarus<sup>2</sup>

<sup>1</sup>EVASION-INRIA/INPG, Grenoble

<sup>2</sup>LIS-CNRS, Grenoble

{Gregoire.Aujay,Franck.Hetry}@imag.fr

Francis.Lazarus@lis.inpg.fr



**Figure 1:** De gauche à droite : un modèle de chat (surface triangulée), le squelette calculé : plongement "standard" (non raffiné) et plongement spécifique aux quadrupèdes (non raffiné), un squelette construit à la main par une professionnelle.

---

## Abstract

Nous proposons une méthode permettant de construire automatiquement et rapidement un squelette adapté à l'animation de personnages, bipèdes ou quadrupèdes. Contrairement aux approches précédentes, cette méthode ne repose pas que sur la géométrie du personnage mais aussi sur des informations anatomiques connues a priori. L'utilisation d'un Laplacien nous garantit un squelette sans bruit (c'est-à-dire que les seules extrémités du squelette seront celles sélectionnées au départ). Notre squelette est également capable de capturer la symétrie de la morphologie du modèle, lorsque celle-ci existe. Enfin, nous avons validé quantitativement notre approche en comparant des squelettes de quadrupèdes avec des squelettes construits à la main par une infographiste.

---

## 1. Introduction

### 1.1. Contexte

Afin d'animer un personnage virtuel en trois dimensions, une des techniques les plus couramment utilisées consiste à créer une représentation hiérarchique des articulations principales du modèle, appelée squelette d'animation. Chaque articulation du squelette peut être déplacée individuellement ; le squelette est alors déformé par cinématique directe ou inverse, et le modèle 3D de l'enveloppe externe est ensuite lui-même déformé autour du squelette. La création du squelette est aujourd'hui faite essentiellement manuellement par les infographistes, bien que de nombreuses méthodes de construction automatique aient été proposées [LKA06, LWM\*03, WP02].

En effet, les squelettes obtenus avec les méthodes automatiques ne sont pas adaptés pour l'animation : par exemple,

la colonne vertébrale est systématiquement placée au milieu du tronc, alors qu'anatomiquement elle est plaquée contre le dos. Par ailleurs, les méthodes actuellement existantes souffrent parfois d'autres défauts rendant leur utilisation difficile dans le cadre de l'animation : absence de garantie sur la capture de toutes les caractéristiques morphologiques du modèle, absence de symétrie du squelette alors que la morphologie du modèle en possède, manque de contrôle de l'infographiste sur le résultat, temps de calcul, ...

### 1.2. Bref état de l'art

De nombreux algorithmes ont été proposés pour générer automatiquement des squelettes de formes 3D. Certains donnent des résultats intrinsèquement non exploitables pour l'animation : ainsi la génération de l'axe médian [Blu67]

conduit à un squelette comportant des parties surfaciques, ce qui nécessite un post-traitement pour les éliminer [DS06].

Parmi les techniques fournissant un squelette dont la structure est un graphe, certaines utilisent une voxélisation du modèle [WP02]. Elles nécessitent un temps de calcul important, et la taille des caractéristiques prises en compte par le squelette dépend de la taille des voxels.

Les autres, plus nombreuses, travaillent sur la surface du modèle, souvent représentée sous forme de maillage. Le calcul automatique d'un squelette a de nombreuses applications : indexation d'un modèle dans une base de données, comparaison de formes, segmentation,... [LWM\*03] propose une méthode originale de construction d'un squelette pour l'animation, utilisant un champ de forces répulsives pour placer les nœuds du squelette : les positions d'équilibre d'une particule chargée dans un champ électrique généré par une densité de charge située à la surface du modèle donnent l'emplacement de ces nœuds. Cette méthode est lente, et ne garantit pas que le résultat capturera toutes les caractéristiques du modèle. [LKA06] propose de calculer simultanément un squelette multi-résolution et une décomposition hiérarchique du modèle étudié. Comme les approches précédentes, ce travail ne prend en compte que la géométrie du modèle, et génère des squelettes anatomiquement peu réalistes.

Notre approche repose sur la construction d'un graphe des contours (aussi appelé graphe de Reeb) associé à la surface du modèle. Cet outil repose sur des propriétés mathématiques prouvant qu'il capture, dans une certaine mesure, la morphologie du modèle [BMMP03]. Les nœuds du graphe correspondent aux points critiques d'une fonction définie sur la surface étudiée. Plusieurs fonctions ont été proposées : distance à un point source [LV99], intégrale de distances à des points sources [HSKK01], fonction de paramétrisation à partir de points caractéristiques [TVD06], solution de l'équation de Laplace [NGH04]. Cette dernière est intéressante car elle garantit l'absence de points critiques non désirés dans le graphe ; c'est pour cette raison que nous avons choisi de l'utiliser.

### 1.3. Contributions

Nous proposons une méthode de construction automatique de squelettes pour l'animation. Les squelettes créés vérifient les propriétés suivantes :

- ils correspondent à des graphes et sont hiérarchiques ;
- leurs extrémités correspondent à celles imposées par l'utilisateur, et uniquement à celles-ci (i.e., ils ne sont pas bruités) ;
- ils sont symétriques lorsque la morphologie du modèle l'est également ;
- ils sont facilement éditables et manipulables (déplacement de nœuds, subdivision d'arêtes, ...).

Dans le cas de modèles de bipèdes et de quadrupèdes, nous proposons un placement spécifique correspondant mieux à l'anatomie des modèles, que nous avons validé quantitativement.

La construction de ces squelettes est automatique mais peut être facilement configurée. Le temps de calcul est de l'ordre de quelques secondes sur un PC standard, pour des modèles de plusieurs dizaines de milliers de faces. Notons enfin que la détection du caractère bipède ou quadrupède d'un modèle peut être faite automatiquement.

### 1.4. Aperçu de la méthode

Notre algorithme construit un squelette de personnage (représenté sous forme de surface triangulée) à partir d'un sommet sélectionné sur la tête du personnage par l'utilisateur. Ce sommet sera une extrémité du squelette ; l'utilisateur peut sélectionner manuellement les autres extrémités, ou laisser cette tâche à l'algorithme. Savoir que le sommet de départ est situé sur la tête du personnage apporte des informations sémantiques sur le squelette : l'algorithme pourra automatiquement déterminer quelle arête du squelette correspond à quelle patte ou jambe du personnage, et ainsi le déplacer et le modifier pour qu'il corresponde au mieux à un squelette de bipède ou de quadrupède.

La suite de ce papier présente les étapes successives de l'algorithme :

1. à partir du sommet de départ, création d'un graphe des contours d'une fonction harmonique définie sur le modèle et filtrage de ce graphe (section 2) ;
2. plongement de ce graphe dans  $\mathbb{R}^3$  et détection de symétries (section 3) ;
3. dans le cas des bipèdes et des quadrupèdes, modification du plongement afin d'obtenir un squelette plus conforme aux attentes des infographistes (section 4).

La section 5 présente quelques résultats, ainsi que la validation quantitative effectuée.

## 2. Calcul et filtrage du graphe des contours

### 2.1. Construction d'un graphe des contours

Le graphe de Reeb [Ree46] et le graphe des contours sont des outils mathématiques permettant de capturer, dans une certaine mesure, la morphologie d'une surface. Un graphe de Reeb est défini de la manière suivante :

**Définition 1 (Graphe de Reeb)** Soit  $f: M \rightarrow \mathbb{R}$  une fonction définie sur une surface  $M$  lisse et compacte. Le *graphe de Reeb* de  $f$  sur  $M$  est l'espace quotient de  $M$  par la relation d'équivalence  $\sim$  définie par :

$$x_1 \sim x_2 \Leftrightarrow \begin{cases} f(x_1) = f(x_2) \\ x_1 \text{ et } x_2 \text{ sont dans la même composante} \\ \text{connexe de } f^{-1}(f(x_1)) \end{cases}$$

Concrètement, le graphe de Reeb est composé de nœuds, correspondant aux points critiques de  $f$  (c'est-à-dire aux points pour lesquels le gradient de  $f$  s'annule), et d'arêtes, correspondant aux composantes connexes de  $M$  reliant ces points critiques. Notons que les nœuds du graphe sont soit de valence 1 (nœuds extrémités), soit de valence supérieure ou égale à 3. Le *graphe des contours* correspond à la même définition, mais avec  $M$  une surface triangulée.

**Définition 2 (Squelette)** Un *squelette* est le plongement dans  $\mathbb{R}^3$  d'un graphe des contours, c'est-à-dire l'association de coordonnées 3D à tout nœud du graphe.

Le calcul d'un graphe des contours peut se faire en temps  $O(n \log n)$ , avec  $n$  le nombre d'arêtes de la surface triangulée [CMEH\*04].

Le choix de la fonction  $f$  dépend de l'application visée ; en ce qui nous concerne, nous avons repris l'idée de Ni, Garland et Hart [NGH04] d'utiliser une fonction harmonique, qui vérifie des propriétés intéressantes.

## 2.2. Fonction harmonique

**Définition 3 (Fonction harmonique)** Soit  $M$  une surface compacte. Une fonction  $f: M \rightarrow \mathbb{R}$  est dite *harmonique* si elle vérifie l'équation de Laplace :  $\forall P \in M, \Delta f(P) = 0$ .

**Propriété 4** Une fonction harmonique vérifie les propriétés suivantes :

- la valeur moyenne de la fonction sur un petit disque est égale à la valeur au centre du disque (propriété de la moyenne) ;
- elle ne possède pas d'extrema locaux hors de la frontière de  $M$ .

**Définition 5 (Equation de Laplace avec conditions de Dirichlet non-homogènes)** Soit  $M$  une surface compacte,  $\Gamma$  sa frontière et  $g$  une fonction définie sur  $\Gamma$ . On définit le système suivant :

$$\begin{cases} \Delta f(P) = 0 & \forall P \in M \\ f(P) = g(P) & \forall P \in \Gamma \end{cases} \quad (1)$$

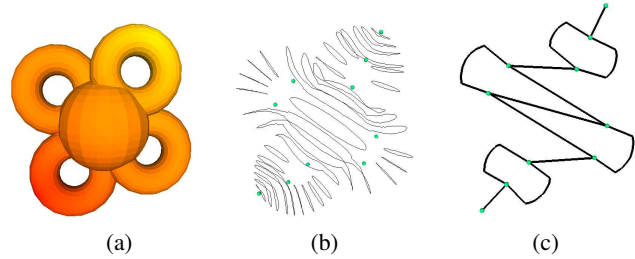
**Propriété 6** Le système (1) admet une solution unique. On définit alors  $f_{g,\Gamma}: M \rightarrow \mathbb{R}$  la fonction solution de ce système.

La fonction que nous utilisons pour définir notre graphe des contours est cette fonction  $f_{g,\Gamma}$ , avec  $\Gamma$  l'ensemble des extrémités souhaitées du graphe. Celles-ci peuvent être soit sélectionnées manuellement par l'utilisateur, soit calculées automatiquement (voir notamment [DKG05, TVD06]). Néanmoins, dans tous les cas, nous imposons à l'utilisateur de choisir manuellement une extrémité, que nous appelons *sommet de départ*, au milieu de la tête du personnage, et nous posons  $g(P) = 0$  pour ce sommet. Dans le cas d'une sélection manuelle, la valeur de  $g$  pour les autres extrémités correspondra à la distance de celles-ci au sommet de départ,

calculée selon l'algorithme de Dijkstra. Dans le cas d'un calcul automatique des extrémités, la valeur de  $g$  peut également être calculée selon l'algorithme de Dijkstra, ou plus simplement grâce à la fonction qui permet de déterminer ces extrémités (par exemple, une solution de l'équation de Poisson pour [DKG05]).

La résolution numérique du système (1) se fait avec une méthode aux éléments finis de type  $\mathbb{P}_1$  (triangles, fonctions linéaires) [Cha04].

La figure 2 montre la fonction et le graphe des contours calculés sur une surface synthétique, pour laquelle un seul maximum a été sélectionné (manuellement).



**Figure 2:** (a) Surface étudiée (b) Quelques lignes de niveau et les points critiques de  $f$  (c) Graphe des contours de  $f$  sur la surface.

## 2.3. Filtrage du graphe

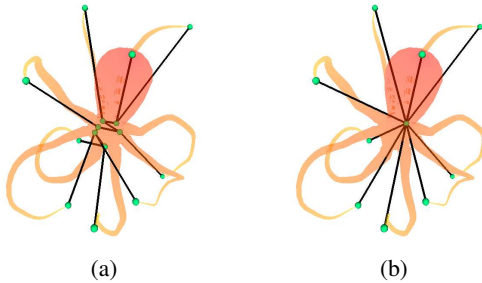
Nous proposons deux types de filtrage du graphe des contours obtenu. Le premier a pour objectif de retranscrire la symétrie de la morphologie du modèle, et correspond à la suppression d'arêtes "intérieures" au graphe (c'est-à-dire d'arêtes dont aucun des deux sommets n'est une extrémité du graphe). Le second permet d'éliminer certaines extrémités qui auraient été calculées automatiquement puis jugées inappropriées, et correspond donc à la suppression d'arêtes dont un des deux sommets est une extrémité du graphe.

Afin de filtrer les arêtes, on associe à chacune d'entre elles un poids (normalisé entre 0 et 1) : soit  $v_1 v_2$  une arête du graphe  $G$ , on pose  $\delta(v_1 v_2) = \frac{|f(v_1) - f(v_2)|}{|\max_{v_i \in G} f(v_i) - \min_{v_i \in G} f(v_i)|}$ . Le

filtrage des arêtes du graphe se fait par ordre de poids croissant.

Nous souhaiterions obtenir un graphe symétrique quand le modèle l'est, ou même quand la *morphologie* du modèle l'est (un modèle de personnage ayant deux jambes de même taille et deux bras de même taille est considéré comme morphologiquement symétrique, même si les deux bras ou les deux jambes ne sont pas dans la même position). Cependant, l'algorithme de construction d'un graphe des contours utilisé [CMEH\*04] ne donne pas, par construction, un graphe symétrique : au lieu d'avoir un unique nœud correspondant

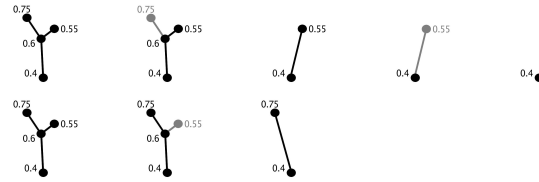
à une symétrie de la morphologie, l'algorithme en construit plusieurs, de valence au plus 4 (voir figure 3 (a)). Or comme le sommet de départ est choisi de manière centrale par rapport à la morphologie du modèle, deux nœuds du graphe voisins et correspondant à deux points critiques de  $f$  voisins sur la surface auront des valeurs très proches pour  $f$ , et donc l'arête reliant ces deux nœuds aura un poids très faible. Nous introduisons donc un paramètre  $\sigma$ , et nous contractons toutes les arêtes "intérieures" au graphe pour lesquelles le poids est inférieur à  $\sigma$ . Cette opération très simple nous permet de capturer au mieux la symétrie de la morphologie du modèle, puisqu'ainsi les multiples nœuds correspondant à une symétrie de la morphologie seront fusionnés en un unique nœud, comme l'illustre la figure 3 (b) pour une symétrie complexe.



**Figure 3:** (a) Graphe des contours obtenu pour un modèle de pieuvre présentant une symétrie axiale (i.e. les bases des tentacules sont disposées régulièrement par rapport au corps de l'animal) (b) Filtrage avec  $\sigma = 0.007$  : la valence du nœud central du graphe est égale à 9.

Les méthodes existantes pour calculer automatiquement les extrémités du graphe (hormis le sommet de départ choisi manuellement) [DKG05, TVD06] peuvent engendrer des solutions non désirées. De plus, la résolution numérique de l'équation de Laplace peut également engendrer ce type de bruit. Afin de remédier à ce problème, nous proposons de supprimer les arêtes du graphe dont un des sommets est une extrémité du graphe et dont le poids est inférieur à un seuil  $\beta$ , ainsi que leurs sommets (afin de garder uniquement, pour l'instant, des nœuds de valence 1 ou supérieure ou égale à 3). Comme l'illustre la figure 4, il est primordial de supprimer les arêtes selon l'ordre croissant de leur poids, afin de ne pas supprimer des arêtes qui ne devraient pas l'être.

Les deux paramètres proposés  $\sigma$  et  $\beta$  peuvent être ajustés par l'utilisateur, mais peuvent également être calculés automatiquement de manière statistique : en effet, le poids des arêtes "inutiles" est généralement très faible par rapport aux autres, et celles-ci peuvent donc être clairement distinguées des autres arêtes. Notons également que la suppression successive de toutes les arêtes du graphe des contours par ordre de poids croissant conduit à la création d'un graphe multirésolution.



**Figure 4:** En haut, étapes successives de l'algorithme avec  $\beta = 0.18$  et sans trier par poids croissant : le graphe se réduit à un nœud. En bas, la même chose mais en triant : il reste une arête.

### 3. Du graphe au squelette : plongement et détection de symétrie

Pour obtenir un squelette, il faut plonger les nœuds du graphe dans  $\mathbb{R}^3$ , c'est-à-dire associer des coordonnées 3D à chacun d'entre eux. Pour cela, nous nous sommes inspirés de la méthode proposée par [LV99].

#### 3.1. Plongement des nœuds du graphe des contours dans $\mathbb{R}^3$

Soit  $a$  une valeur régulière de la fonction  $f$  (c'est-à-dire une valeur ne correspondant pas à un nœud du graphe des contours de  $f$ ), soit  $f^{-1}(a)$  sa ligne de niveau, et soit  $C$  une composante connexe de  $f^{-1}(a)$ .  $C$  est une courbe simple fermée composée de segments de droites dont les extrémités (situées sur les arêtes de la triangulation) seront notées  $p_1, p_2, \dots, p_k$ .

**Définition 7 (Centre d'un contour)** Le centre de  $C$ , noté  $centre(C)$ , est défini comme le barycentre de ses segments : en posant  $p_{k+1} = p_1$ , on obtient

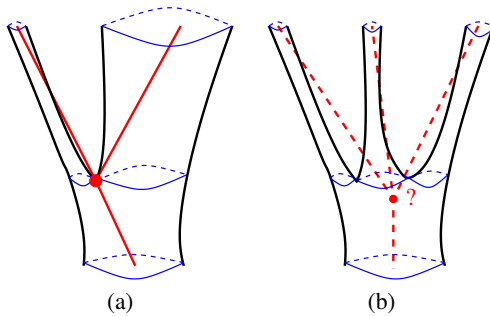
$$centre(C) = \frac{\sum_{i=1}^k \|p_i p_{i+1}\| \frac{p_i + p_{i+1}}{2}}{\sum_{i=1}^k \|p_i p_{i+1}\|}$$

Le barycentre des segments est un choix plus judicieux que le barycentre des points  $p_i$  car le résultat est indépendant du niveau de discrétisation de la surface : on évite ainsi les artefacts pour des maillages non uniformes.

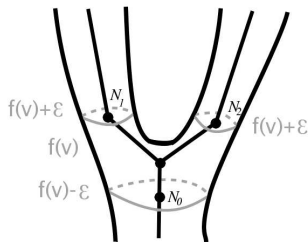
Soit  $V$  une extrémité du graphe des contours :  $V$  correspond à un sommet  $v$  de la triangulation dont la valeur pour  $f$  est extrémale. Le plongement de  $V$  dans  $\mathbb{R}^3$  peut se faire naturellement en ce sommet  $v$ . Soit  $V$  un nœud interne (i.e. non extrémité) du graphe :  $V$  est donc un nœud de valence supérieure ou égale à 3.  $V$  correspond à un sommet  $v$  de la triangulation appelé *selle*, pour lequel le nombre de composantes connexes de  $f^{-1}(f(x))$  est modifié. Plonger  $V$  en  $v$  pourrait conduire à une position peu représentative de la zone du modèle à laquelle correspond ce nœud (voir figure 5 (a)), et peut même être ambigu, par exemple

lorsque  $V$  correspond à la division d'une composante connexe de  $f^{-1}(f(x))$  en plus de deux composantes (voir figure 5 (b)). Nous avons donc choisi une autre solution : nous créons virtuellement de nouveaux nœuds  $N_i$  dans le graphe des contours, de valence égale à 2, et correspondant à chacune des composantes connexes des valeurs  $f^{-1}(f(v) - \varepsilon)$  et  $f^{-1}(f(v) + \varepsilon)$ , pour un  $\varepsilon$  très petit (qui peut par exemple être défini comme inférieur au  $\min_{v_i \text{ voisin de } v} |f(v) - f(v_i)|$ ).

Chaque nœud  $N_i$  est plongé dans  $\mathbb{R}^3$  au centre de sa composante connexe, d'après la définition 7 (voir figure 6). Le nœud  $V$  est ensuite identifié à l'un des nœuds  $N_i$ . Le choix de ce nœud parmi tous les nœuds  $N_i$  se fait grâce à une détection de symétrie dans le graphe, comme expliqué dans la sous-section suivante. Dans le cas où aucune symétrie n'est détectée, le nœud est plongé en l'isobarycentre des plongements des nœuds  $N_i$ .



**Figure 5:** Le plongement d'un nœud du graphe des contours au sommet de la triangulation correspondant n'est pas forcément judicieux (a) ni même toujours possible (b).



**Figure 6:** Insertion (virtuelle) et plongement des nœuds  $N_i$  pour le plongement d'un point selle  $V$ .

Notons que nous n'avons aucune garantie sur le fait que le squelette ainsi créé est bien situé à l'intérieur du modèle : ceci n'est pas notre but, contrairement par exemple à [LKA06], car les squelettes construits par des infographistes ne le sont pas nécessairement non plus.

### 3.2. Détection de symétries

Nous proposons une heuristique permettant, dans la plupart des cas, de détecter un axe de symétrie dans le graphe des

contours précédemment construit, éventuellement augmenté des nœuds virtuels introduits pour le plongement des nœuds internes du graphe. Ce problème correspond à rechercher un automorphisme de graphe laissant invariant un ensemble connexe d'arêtes du graphe. Cette recherche sur un graphe quelconque et sans indication est un problème très difficile et très coûteux en temps de calcul. L'algorithme que nous proposons ne garantit pas un résultat pour un graphe quelconque, mais fournit une solution valide sous les hypothèses suivantes :

1. le sommet de départ  $s$ , pris sur la tête du personnage, correspond à un nœud  $S_0$  sur l'axe de symétrie du graphe ;
2. le graphe des contours ne possède pas de cycle (c'est donc un arbre) ;
3. deux sous-arbres de l'arbre des contours sont isomorphes s'ils ont la même profondeur et si leurs racines sont de degrés égaux.

L'hypothèse 1 est réaliste dans la mesure où nous ne nous intéressons pas à la symétrie exacte du modèle, mais plutôt à la symétrie de sa morphologie. Les hypothèses 2 et 3 sont la plupart du temps vérifiées dans le cas de modèles de personnages, car les graphes calculés sont généralement simples. D'autres conditions peuvent également être ajoutées à l'hypothèse 3 (les fils des racines sont de degrés égaux, etc.).

$S_0$  étant une extrémité de l'arbre des contours, nous la considérons comme la racine de l'arbre. Ceci nous donne une première arête  $A_0$  se trouvant sur l'axe de symétrie du graphe, d'après l'hypothèse 1. L'algorithme suivant décrit comment trouver les autres arêtes de l'axe de symétrie.

---

#### Algorithm 1 Detection de symétrie

---

**Require:**  $A_0 = \langle S_0, S_1 \rangle$

**Ensure:** marque toutes les arêtes se trouvant sur l'axe de symétrie

$A = \langle S, S' \rangle \leftarrow A_0 = \langle S_0, S_1 \rangle$

**while**  $A \neq \emptyset$  **do**

marquer  $A$  comme appartenant à l'axe de symétrie

$\{E_0, E_1, \dots, E_k\}$  est l'ensemble des arêtes  $E_j = \langle S', P_j \rangle$  incidentes à  $S'$  et telles que  $E_j \neq A$

$\Delta_j \leftarrow$  l'arbre dont  $P_j$  est la racine, privé du sous arbre contenant  $E_j$

$U_l \leftarrow$  ensemble d'arbres  $\Delta_j$  isomorphes d'après l'hypothèse 3

**if**  $\exists! U_l$  tel que  $\#U_l = 1$  **then**

$A \leftarrow E_j, j$  tel que  $U_l = \{\Delta_j\}$

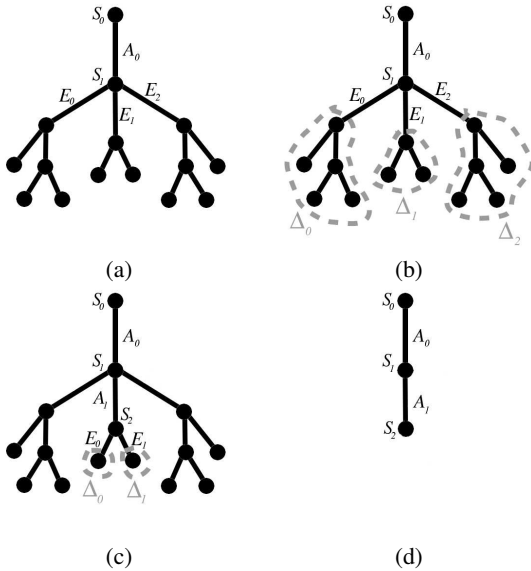
**else**

$A \leftarrow \emptyset$

**end if**

**end while**

---



**Figure 7:** (a) Initialisation (b) Première itération : création des  $\Delta_j$  ;  $\Delta_0$  et  $\Delta_2$  sont égaux, donc  $\Delta_1$  est le seul candidat :  $E_1$  est sur l'axe de symétrie (c) Deuxième itération,  $\Delta_0$  et  $\Delta_1$  sont égaux, il n'y a pas de candidat donc l'algorithme s'arrête (d) Axe de symétrie détecté.

L'algorithme s'arrête lorsque l'on n'obtient pas un unique ensemble  $U_l$  de cardinal égal à 1. En effet si  $U_l$  n'a qu'un élément alors celui-ci ne peut pas avoir d'image par la symétrie, à part lui-même : il est sur l'axe de symétrie. Et s'il existe plusieurs ou aucun ensemble de cardinal égal à 1, alors on ne peut pas trouver une nouvelle arête sur l'axe de symétrie, donc l'algorithme s'arrête.

Il est intéressant de noter que les ensembles  $U_l$  peuvent contenir plus de 2 éléments, dans ce cas on ne détecte plus simplement une symétrie axiale (invariance par rotation d'angle  $\pi$ ), mais une invariance par rotation d'angle  $\frac{2\pi}{\#U_l}$ . C'est le cas pour la pieuvre (figure 3) : on obtient un ensemble  $\#U_0 = \text{nombre de tentacules} = 8$  qui donne ce que l'on peut appeler une invariance du graphe par rotation d'angle  $\frac{\pi}{4}$  et d'axe l'arête correspondant au corps de l'animal.

Cet algorithme est utilisable pour trouver des symétries non principales, en l'exécutant sur les sous-parties du graphe qui ne sont pas sur l'axe principal. On peut ainsi obtenir une hiérarchie de symétries, comme [SKS06].

### 3.3. Retour sur le plongement des nœuds du graphe

Une fois que l'axe de symétrie du graphe augmenté est connu, on peut choisir, pour chaque nœud interne du graphe appartenant à cet axe de symétrie, le nœud virtuel à utiliser pour son plongement. Dans le cas où un nœud interne du graphe des contours n'est pas sur l'axe de symétrie, il est

possible de relancer l'algorithme de détection de symétrie avec un sous-graphe contenant ce nœud.

Soit  $V$  un nœud interne du graphe des contours, se trouvant sur l'axe de symétrie. Soient  $N_1, \dots, N_k$  les nœuds virtuels introduits pour le plongement de  $V$ . Deux cas de figures sont envisageables :

- il existe un seul  $N_i$  qui est aussi sur l'axe de symétrie, ce qui signifie que  $V$  est à une extrémité de l'axe de symétrie. Nous choisissons alors de plonger  $V$  au même endroit que  $N_i$  ;
- il existe deux nœuds  $N_i$  et  $N_j$  sur l'axe de symétrie. Nous choisissons alors le nœud  $N_j$  tel que la valeur pour  $f$  associée à son contour est plus petite que celle du contour de  $N_i$  (en d'autres termes,  $N_i$  est plus proche du nœud correspondant au sommet de départ que  $N_j$ ).

Les nœuds virtuels  $N_i$  non présents sur l'axe de symétrie sont insérés dans le graphe, et reliés de manière rigide au nœud  $V$  : ils correspondront notamment aux épaules et aux hanches des personnages.

### 3.4. Hiérarchie et subdivision des arêtes du squelette

Une fois qu'un axe de symétrie du graphe a été détecté, il est très facile d'indexer les nœuds de ce graphe selon une hiérarchie : le nœud principal sera un nœud de cet axe de symétrie, qui peut être choisi suivant différents critères (position centrale sur l'axe de symétrie, ou au contraire nœud le plus éloigné du nœud correspondant au sommet de départ, etc.). Le reste de la hiérarchie dérive du choix du nœud principal.

Une arête du squelette peut facilement être subdivisée : le nœud introduit (de valence 2 dans le graphe) est plongé au centre de son contour. Le choix de la valeur de  $f$  correspondant à ce nœud peut suivre plusieurs heuristiques :

- sélection par l'utilisateur ;
- valeur moyenne entre les valeurs correspondant aux deux extrémités de l'arête ;
- valeur correspondant à une caractéristique géométrique particulière sur le modèle (par exemple minimum local de la courbure gaussienne, comme proposé par [TVD06]) ;
- etc.

Notons simplement que l'information des valeurs de  $f$  peut être très utile pour ce problème.

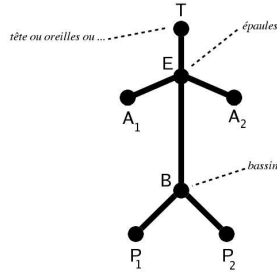
### 4. Plongement spécifique aux bipèdes et aux quadrupèdes

Nous proposons dans cette section une amélioration du plongement, destinée à obtenir dans le cas des bipèdes et des quadrupèdes un squelette plus proche de celui que construirait manuellement un professionnel. Pour cela, nous utilisons des connaissances *a priori* sur le squelette de ces types de personnages.



#### 4.1. Détection du caractère bipède ou quadrupède

Il est possible de distinguer directement sur le squelette construit précédemment s'il s'agit d'un bipède ou d'un quadrupède. En effet, nous connaissons l'extrémité du squelette correspondant à la tête du personnage. Dans le cas d'un bipède ou d'un quadrupède, l'axe de symétrie du graphe, détecté précédemment, correspondra au cou, à la colonne vertébrale et éventuellement à la queue du modèle. En parcourant cet axe de symétrie à partir de la tête du personnage, nous supposons que le dernier nœud rencontré correspond à son bassin, et l'avant-dernier à ses épaules (voir figure 8 – on peut éventuellement avoir un nœud supplémentaire, par exemple pour les oreilles).



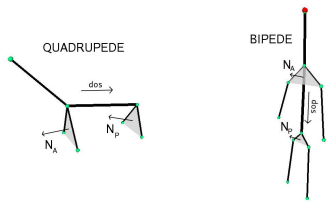
**Figure 8:** Squelette type d'un bipède ou d'un quadrupède.

Notons :

- $\vec{dos} = \frac{\vec{EB}}{\|\vec{EB}\|}$  le vecteur normé correspondant à la direction de la colonne vertébrale, orientée de l'avant vers l'arrière du personnage ;
- $\vec{N}_P$  un vecteur unitaire normal au triangle  $BP_1P_2$  ;
- $\vec{N}_A$  un vecteur unitaire normal au triangle  $EA_1A_2$ .

D'après les caractéristiques morphologiques de chaque espèce on a :

- pour un quadrupède,  $|\vec{N}_P \cdot \vec{dos}| \approx 1$  et  $|\vec{N}_A \cdot \vec{dos}| \approx 1$  ;
- pour un bipède,  $|\vec{N}_P \cdot \vec{dos}| \approx 0$  et  $|\vec{N}_A \cdot \vec{dos}| \approx 0$ .



**Figure 9:**  $\vec{N}_P$  et  $\vec{N}_A$  sont parallèles au dos pour un quadrupède et lui sont perpendiculaires pour un bipède.

Cela est vrai pour toutes les positions du modèles, car les nœuds  $A_1, A_2, P_1$  et  $P_2$  correspondent aux débuts des membres qui sont des parties rarement animées (rigidité des épaules et des hanches). On peut ainsi détecter si le modèle est un quadrupède ou un bipède, en calculant les deux produits scalaires  $\vec{N}_P \cdot \vec{dos}$  et  $\vec{N}_A \cdot \vec{dos}$ .

#### 4.2. Plongement spécifique à un quadrupède

Nous ne décrivons ici que le plongement adapté à un quadrupède, mais le même raisonnement peut s'appliquer pour un bipède.

En comparant un squelette fait par un animateur professionnel et un squelette généré automatiquement par notre méthode (respectivement quatrième et deuxième images de la figure 1), on note plusieurs points à corriger sur le squelette calculé :

- la position du bassin est trop basse, il devrait être plus près du dos ;
- la colonne vertébrale possède trois nœuds intermédiaires, situés plus en arrière car cette zone est plus articulable ;
- il manque deux nœuds intermédiaires entre les épaules et le début de chaque patte avant (omoplates) ;
- il manque un nœud intermédiaire entre le bassin et le début de chaque patte arrière (hanches).

La correction du plongement est purement technique : il suffit de déplacer le nœud correspondant au bassin, et d'insérer de nouveaux nœuds, plongés à des endroits adéquats. Tout ceci est possible automatiquement car, en ayant choisi le sommet de départ sur la tête du modèle, on est capable d'associer une zone anatomique à chaque nœud du squelette. La troisième image de la figure 1 montre le résultat sur le modèle du chat et la figure 11 montre le résultat sur deux autres modèles. Dans le cas du chat, par souci de lisibilité, le squelette n'a pas été raffiné : seuls sont présents les plongements des nœuds du graphe des contours. Pour les deux autres modèles, un nombre total de nœuds supplémentaires a été choisi au départ de l'algorithme, puis ceux-ci ont été placés automatiquement.

#### 5. Résultats et validation

Les figures 1, 10 et 11 montrent quelques résultats de squelettes construits pour divers modèles d'animaux. Le temps de calcul pour chaque modèle présenté, sur un PC de bureau sous Pentium IV à 2.4 GHz et avec une mémoire RAM de 1 Go, est indiqué en table 1. La différence de temps de calcul entre le plongement "standard" et le plongement adapté aux bipèdes et aux quadrupèdes est négligeable.

Modèle	Nombre de faces	Temps de calcul
Chat	2542	0.06s
Cheval	96966	4.42s
Chien	158204	8.48s

**Table 1:** Temps de calcul sur différents modèles.

La validation des résultats demande plus qu'un simple constat visuel, comme le souligne [LKA06]. Néanmoins, à la connaissance des auteurs, il n'existe à ce jour aucune méthode objective permettant de comparer différents squelettes d'un même modèle. Nous avons choisi de valider nos



squelettes de quadrupèdes en comparant deux paramètres définis dans [RFDC05] pour nos squelettes et pour des squelettes faits à la main par une professionnelle chevronnée. Les résultats sont détaillés dans le tableau 2. La différence entre deux valeurs d'un même paramètre et pour un même modèle est au plus de 7%. Une validation quantitative plus poussée de notre approche est nécessaire et prévue ; néanmoins à la lumière de ces premiers résultats nous pouvons dire que les squelettes construits avec notre méthode sont proches de squelettes faits à la main par des infographistes.

Modèle	Spine tilt		Legs height	
	auto	manuel	auto	manuel
Chat	1.13	1.17	1.30	1.23
Chien	1.09	1.15	1.38	1.45
Lionne	0.83	0.89	1.04	1.01

Table 2: Paramètres des squelettes.

## 6. Conclusion

Nous avons présenté dans ce papier une nouvelle méthode de construction automatique d'un squelette hiérarchique pour l'animation. L'utilisation d'une fonction harmonique nous assure de manière théorique l'absence de bruit sur les squelettes calculés, mais nous proposons néanmoins deux méthodes de filtrage, afin notamment de rendre le graphe symétrique. L'utilisateur a le choix entre laisser l'algorithme fonctionner seul (il devra simplement choisir un sommet de départ), ou fixer quelques paramètres (extrémités du squelette, seuils de filtrage, ...). A partir de la position du sommet de départ, nous pouvons calculer un axe de symétrie sur le graphe dont le squelette est le plongement, puis obtenir des informations sémantiques sur le squelette, et ainsi améliorer son positionnement, notamment dans le cas des bipèdes et des quadrupèdes, afin qu'il approche d'un squelette créé manuellement par un infographiste. Nous avons validé notre démarche par une comparaison quantitative avec de tels squelettes.

Les limitations actuelles de notre méthode concernent les données de départ : les modèles doivent être représentés sous forme de 2-variétés triangulées, de genre nul (cette dernière condition est nécessaire pour le calcul de l'axe de symétrie du graphe). Par ailleurs, la validation de notre approche reste à approfondir, et le plongement spécifique aux quadrupèdes reste améliorable.

## References

[Blu67] BLUM H.: A transformation for extracting new descriptions of shape. *Models for the Perception of Speech and Visual Form* (1967), 362–380.

[BMMP03] BIASOTTI S., MARINI S., MORTARA M., PATANÈ G.: An overview on properties and efficacy of topological skeletons in shape modelling. In *Shape Modelling International* (2003), pp. 245–256.

[Cha04] CHASKALOVIC J.: *Méthode des éléments finis pour les sciences de l'ingénieur : abrégé de cours et recueil de problèmes corrigés*. Lavoisier, 2004.

[CMEH\*04] COLE-MCLAUGHLIN K., EDELSBRUNNER H., HARER J., NATARAJAN V., PASCUCCI V.: Loops in reeb graphs of 2-manifolds. *Discrete Computational Geometry* 32 (2004), 231–244.

[DKG05] DONG S., KIRCHER S., GARLAND M.: Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer Aided Geometry Design* 22 (2005), 392–423.

[DS06] DEY T., SUN J.: Defining and computing curve-skeletons with medial geodesic function. In *Eurographics Symposium on Geometry Processing* (2006).

[HSKK01] HILAGA M., SHINAGAWA Y., KOMURA T., KUNII T.: Topology matching for fully automatic similarity estimation of 3d shapes. In *ACM Transactions on Graphics* (2001), pp. 203–212. SIGGRAPH.

[LKA06] LIEN J., KEYSER J., AMATO N.: Simultaneous shape decomposition and skeletonization. In *ACM Symposium on Solid and Physical Modeling* (2006), pp. 219–228.

[LV99] LAZARUS F., VERROUST A.: Level set diagrams of polyhedra objects. In *ACM Solid Modeling* (1999), pp. 130–140.

[LWM\*03] LIU P., WU F., MA W., LIANG R., OUHYOUNG M.: Automatic animation skeleton construction using repulsive force field. In *Pacific Graphics* (2003), pp. 409–413.

[NGH04] NI X., GARLAND M., HART J.: Fair morse functions for extracting the topological structure of a surface mesh. *ACM Transactions on Graphics* 23, 3 (2004), 613–622. Proceedings of SIGGRAPH.

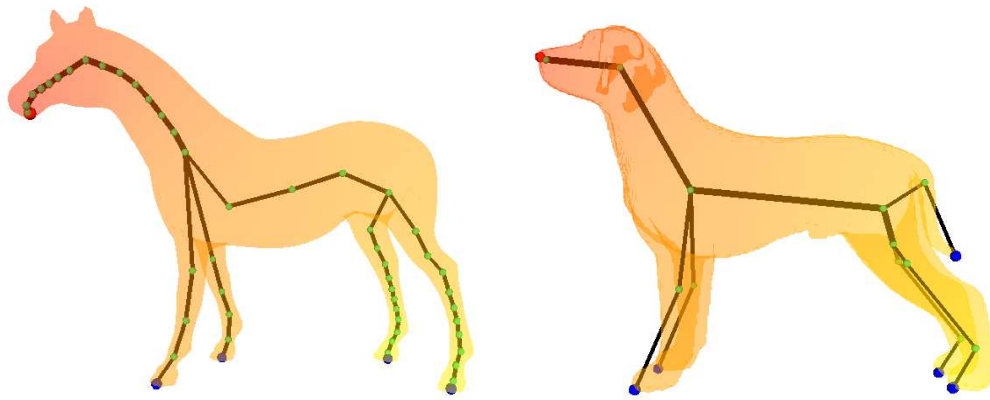
[Ree46] REEB G.: Sur les points singuliers d'une forme de pfaff complètement intégrable ou d'une fonction numérique. *Comptes-rendus de l'Académie des Sciences* 222 (1946), 847–849.

[RFDC05] REVÉRET L., FAVREAU L., DEPRAZ C., CANI M.: Morphable model of quadrupeds skeletons for animating 3d animals. In *ACM/Eurographics Symposium on Computer Animation* (2005), pp. 135–142.

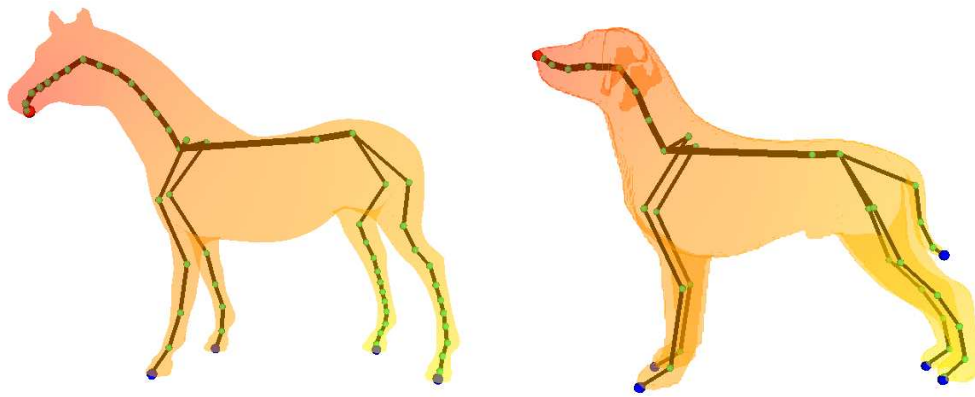
[SKS06] SIMARI P., KALOGERAKIS E., SINGH K.: Folding meshes: Hierarchical mesh segmentation based on planar symmetry. In *Eurographics Symposium on Geometry Processing* (2006).

[TVD06] TIERNY J., VANDEBORRE J., DAOUDI M.: 3d mesh skeleton extraction using topological and geometrical analyses. In *Pacific Graphics* (2006).

[WP02] WADE L., PARENT R.: Automated generation of control skeletons for use in animation. *The Visual Computer* 18, 2 (2002), 97–110.



**Figure 10:** *Squelettes obtenus (plongement "standard").*



**Figure 11:** *Squelettes obtenus (plongement adapté aux quadrupèdes).*