



Animation 3D en Temps-Réel de Quadrupèdes par Simulation Physique

Grégori Clauzel

► To cite this version:

Grégori Clauzel. Animation 3D en Temps-Réel de Quadrupèdes par Simulation Physique. Synthèse d'image et réalité virtuelle [cs.GR]. 2006. inria-00598383

HAL Id: inria-00598383

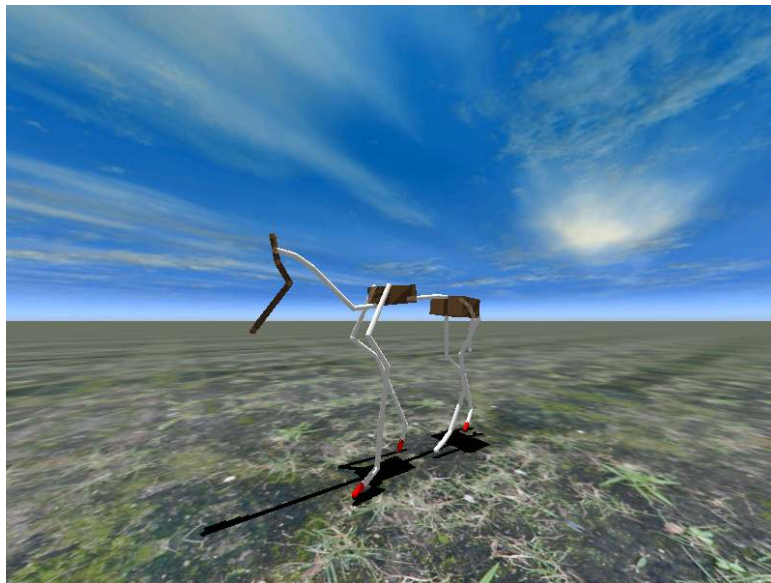
<https://inria.hal.science/inria-00598383>

Submitted on 6 Jun 2011

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.

Animation 3D en Temps-Réel de Quadrupèdes par Simulation Physique



STAGE MASTER 2 IMAGERIE, VISION, ROBOTIQUE

Étudiant : **Grégori Clauzel** — Encadrant : **Lionel Revéret**
elgoret@online.fr — lionel.reveret@inria.fr

Evasion - Laboratoire GRAVIR - INRIA Rhône-Alpes
ZIRST - 655 avenue de l'Europe
Montbonnot - 38334 Saint Ismier Cedex



30 août 2006

Table des matières

1	Introduction	4
2	État de l'Art	5
2.1	Réponse dynamique pour l'animation de capture de mouvement	5
2.2	Synthèse de mouvements complexes à partir d'animations simples	6
2.3	Contrôle interactif en temps-réel	6
3	Modèle physique de quadrupèdes	7
3.1	Le squelette	8
3.2	Les articulations	9
4	Méthodes de contrôle	10
4.1	Commande angulaire des articulations	10
4.2	Les couples de forces, ou segments de contrôle	10
4.3	Vecteur d'équilibre	11
5	Contrôleurs expérimentés	12
5.1	Rigide	12
5.2	Pointe la cible	13
5.3	Les postures	14
5.4	La laisse	15
5.5	Jambe par jambe	15
5.6	Par la colonne	16
6	Résultats	17
7	Conclusion	19
A	Annexes	20
A.1	Logiciel de simulation	20
A.2	Scripts Maya	20
B	Bibliographie	21

Résumé

L'animation 3D en temps-réel de personnages, utilisant uniquement la simulation physique, est un domaine quasi inexploré à l'heure actuelle. Au travers de ce travail, nous instaurons certaines bases méthodologiques, permettant de créer des animations réactives pour un quadrupède au sein d'un environnement entièrement dynamique. Nous avons instauré le concept de couple de forces (ou segment de forces), qui simplifie la simulation complète du système locomoteur pratiquée en biomécanique. Nous nous appuyons aussi, sur le fait que l'élément moteur de la locomotion est la colonne, et non pas les jambes. Par ce procédé, nous obtenons des animations assez réalistes, munie d'un bon contrôle pour l'animateur, et une réactivité sans précédent dans le domaine de l'animation en temps-réel. Les calculs mis en jeu sont minimaux, ce qui offre la possibilité d'avoir de nombreux quadrupèdes simulés en temps-réel, sur les ordinateurs personnels actuels. Le logiciel ainsi développé peut aussi expérimenter des contrôleurs plus complexes se rapprochant de la biomécanique en vue d'une simulation plus aboutie de la locomotion, en profitant des cartes accélératrices physiques.

1 Introduction

Aujourd'hui, dans la production d'animations pour les jeux-vidéos, deux techniques prédominent : la capture de mouvement et l'interpolation de positions clés. Si la capture de mouvement permet d'obtenir des mouvements de grande qualité, elle reste chère à mettre en place, et difficile à appliquer sur des animaux, ou des créatures imaginaires. L'interpolation de positions clés quand à elle, s'applique à n'importe quel modèle, mais demeure très fastidieuse pour l'animateur, qui passe beaucoup de temps à paufiner ses animations.

Aussi, les jeux vidéos actuels et de prochaine génération proposent de plus en plus des univers hautement interactifs simulés par la physique. Les animations des personnages doivent être intégrées à l'environnement dynamique dans lequel ils évoluent, et réagir correctement aux contraintes imposées par cet environnement. Beaucoup de recherches s'articulent autour de cette problématique, en particulier sur comment modifier en temps-réel, les animations issues des techniques traditionnelles, pour qu'elles s'adaptent aux nouvelles conditions extérieures.

Pour répondre à ce problème, nous avons choisi d'animer un modèle physique de squelette de quadrupèdes par programmation de contrôleurs, sans utiliser de positions clés ou de capture de mouvement préalablement définies. Celui-ci est donc parfaitement incorporé dans l'univers physique, puisqu'il est soumis aux mêmes lois. Cela nous permet d'obtenir une grande variété d'animations réactives, répondant aux innombrables configurations possibles de l'environnement dynamique, sans avoir à prévoir à l'avance tous les cas de figures possibles.

Tout la difficulté se trouve donc dans la définition d'un **contrôleur minimal**, utilisant le moins possible de ressources, mais produisant une animation de qualité qui procure l'impression visuelle d'un comportement cohérent de l'animal. De ce fait, une implémentation complète du système locomoteur et de ses muscles, comme on peut le voir en biomécanique, ne peut être réalisée pour fonctionner en **temps-réel**, avec parfois plusieurs animaux simulés en même temps. Le but est donc de simplifier au maximum la modélisation de la locomotion, tout en restant proche d'un certain réalisme.

A terme, un des objectifs est de pouvoir proposer un système de configuration générique de contrôleurs, utilisable par un animateur n'ayant aucune connaissance en programmation, lui permettant de créer un comportement propre à son personnage. Mais aussi de pouvoir développer des contrôleurs largement plus complexes, afin d'obtenir une simulation de la locomotion de quadrupèdes aussi réaliste que possible.

2 État de l'Art

Dans le domaine du choix des contrôleurs pour les simulations physiques, on distingue trois approches différentes : à base de capture de mouvement, à l'aide de positions clés, ou entièrement physique. Nous présentons tout d'abord trois travaux précédemment réalisés sur des simulations physiques et leurs contrôleurs associés, basés sur ces trois méthodes, mais possédant leurs propres limitations.

2.1 Réponse dynamique pour l'animation de capture de mouvement

Les données de capture de mouvement pour l'homme sont très répandues en animation à l'heure actuelle. Elles permettent une certaine fidélité à la réalité, puisque ce sont des mesures directes de mouvements exécutés par des acteurs. L'une des difficultés imposée par cette technique, est l'aspect prédéfini de ces animations. Seule une quantité finie d'animations est enregistrée dans le logiciel d'animation, et il est impossible de pré enregistrer toutes les transitions nécessaires pour passer de l'une à l'autre. Aussi, si un événement imprévu dans l'animation intervient, il faut pouvoir modifier celle-ci, afin qu'elle colle à ce qu'il se passe vraiment.

Zordan et al. [3] ont présenté une technique permettant de générer des transitions à tout moment d'une certaine animation perturbée par un événement externe, afin de retomber sur une autre animation venant prendre le relais. Au moment où un choc (par exemple) intervient, l'animation est stoppée, et une phase de simulation physique est effectuée sur les dixièmes de secondes suivants. La simulation s'opère avec un ragdoll remplaçant le modèle dans la position où il était au cours de l'animation, livré à soi-même. Après quelques simulations, permettant de choisir l'animation sur laquelle poursuivre et de déterminer une transition cohérente, l'animation peut reprendre son cours, suivant les résultats obtenus (cf. fig. 1).

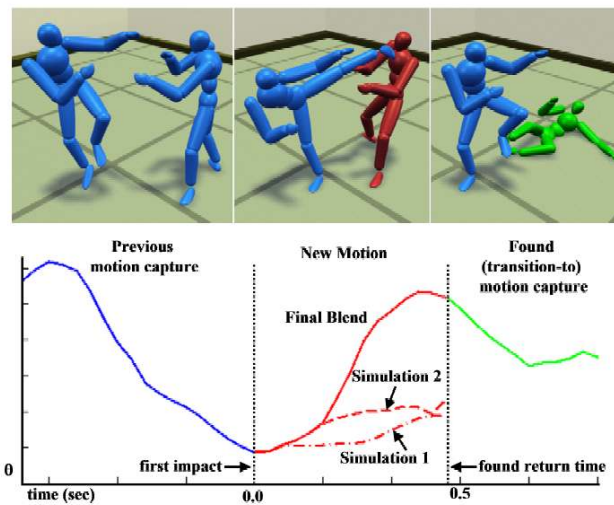
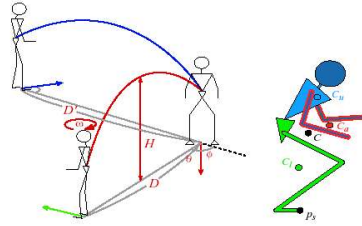


FIG. 1 – Valeur en y de l'articulation de la taille, avant, pendant, et après l'interaction

Un des problèmes de cette méthode est le pic de calculs nécessaire lors d'un choc imprévu. Cette technique est donc bien appropriée pour un calcul off-line de rendu d'animation, mais n'est pas très adéquate à un rendu d'animations en temps-réel. De plus, pour que le rendu soit acceptable, il faut une large base de données de capture de mouvement, afin d'avoir un maximum d'animations répondant à diverses situations possibles. Ceci pose un problème dans le cas des animaux, imaginaires ou non, pour lesquels peu de capture de mouvement sont disponibles, ou inadaptées.

2.2 Synthèse de mouvements complexes à partir d'animations simples

Cette deuxième méthode, est une sorte de large évolution de la technique des positions clés, présentée par Popović [2]. Un modèle physique du personnage à animer est créé. On fournit à ce modèle physique, un ensemble d'animations simples, paramétrables. Par exemple, on intègre une séquence de marche, sur laquelle on fait varier l'amplitude du lever de pied, la hauteur de l'appui, etc. Une fois ce modèle mis en place, on définit plusieurs états dans lesquels le caractère va se retrouver au cours du temps (cf. fig. 2).



3 Modèle physique de quadrupèdes

Suite au travail de Reveret et al. [4], nous disposons de 10 morphologies différentes d'animaux (cf. fig. 5), allant du rat à l'éléphant. Ils sont définis par une hiérarchie d'os, reliés par des articulations. Chaque animal possède la même hiérarchie, seuls la longueur des os et les angles des articulations diffèrent.

L'implémentation de ces squelettes dans le simulateur utilise le moteur physique NovodeX [7]. Ce moteur permet l'utilisation de solides articulés, reliés par des articulations aux degrés de liberté configurables. Il se charge de calculer les collisions, et de résoudre les contraintes induites par les articulations. Nous avons donc traduit les informations rudimentaires sur les squelettes (longueurs d'os, position en rotation), par des solides articulés, définis par une forme, une masse, un centre de gravité, et donc des moments d'inertie (cf. fig. 4).

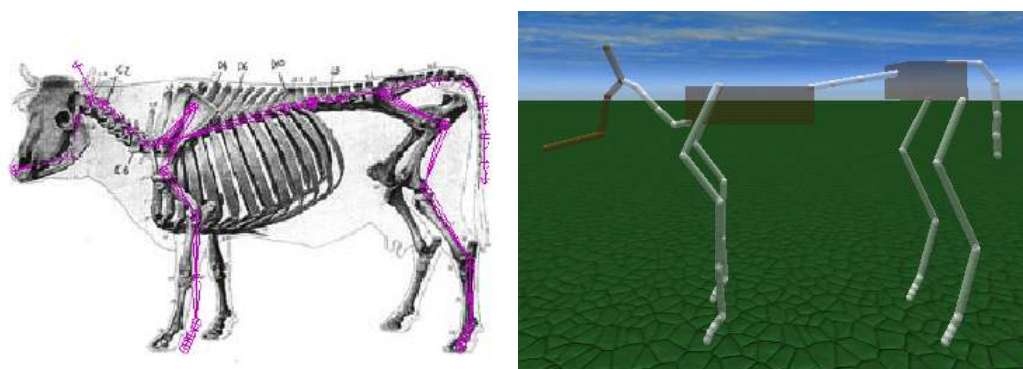


FIG. 4 – A gauche, d'après Reveret et al., et à droite, notre implémentation en solides articulés.

Lors de la création d'un animal, on stocke la hauteur de son pelvis, ce qui caractérise de manière simple les proportions de son squelette. Cette valeur sera souvent utilisée par la suite, pour définir certains paramètres variant selon la corpulence de l'animal. Par la suite, nous appellerons cette valeur HP (Hauteur de Pelvis). A titre d'indication, voici la liste des animaux, avec leur HP associée (cf. fig. 6).

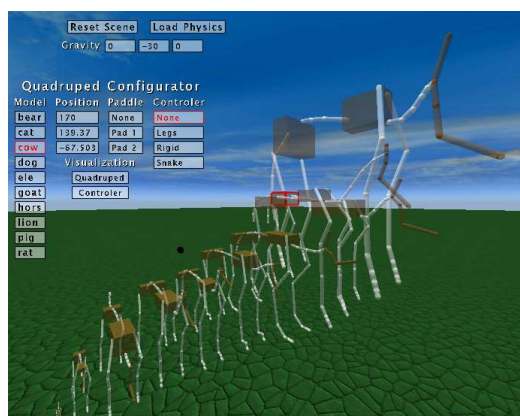


FIG. 5 – Squelettes, du rat à l'éléphant

Animal	Hauteur
Elephant	299,82
Vache	144,17
Cheval	140,69
Ours	93,66
Lion	76,48
Chien	65,56
Chèvre	59,54
Pig	48,41
Chat	27,93
Rat	2,54

FIG. 6 – HP (Hauteur du Pelvis) associé (sans unités)

3.1 Le squelette

Le squelette de référence du quadrupède est composé au total de 39 os. Nous en utilisons 5 pour le cou, 3 pour la colonne, 4 pour la queue, et 4×3 pour les pieds.

Chaque os est modélisé par une capsule, dont nous connaissons la longueur. Pour que le modèle fonctionne, le rayon d'une capsule doit être inférieur à la longueur de cette capsule et de ses voisines directes. Pour simplifier, le rayon de chaque capsule est défini de manière unique pour tous les os des animaux, mais il est différent pour chaque animal, selon sa taille. Le rayon des os est fixé pour chaque animal à $HP / 75 + 0,1$. On obtient alors des os suffisamment petits sur tout le corps pour éviter les capsules dégénérées, mais suffisamment grand pour avoir un contrôle visuel, et une bonne cohérence dans les proportions (cf. fig. 5).

Le rayon n'a que peu d'importance sur le mouvement, entre autre parce que le centre de masse associé est le même. De plus, Renous dans son livre *Locomotion* [5] indique que seuls 3 centres de masse sont importants pour la locomotion : ceux situés au niveau du pelvis, du thorax et de la tête. A ces 3 os majeurs sont donc attribués des masses particulières. N'ayant que très peu d'information sur la répartition des masses de ces animaux, nous avons fixés de manière arbitraire les masses associées à ces 3 os :

- $20 \times HP$ pour le pelvis
- $10 \times HP$ pour le thorax
- $6 \times HP$ pour la tête

Ces os plus complexes (4 articulations pour le pelvis et le thorax), ne sont pas modélisés par une simple capsule. Le pelvis et le thorax sont des pavés, alors que la tête est un encastrement de 5 capsules. Aux autres os du squelette est attribué une densité de 10. Le poids varie donc légèrement selon la taille de l'os (cf. fig. 7). Suite à quelques expérimentations sur les contrôleurs présentés ultérieurement, il s'est avéré que cette valeur a très peu d'influence sur les mouvements.

Partie	Poids
Queue	186
Pelvis	1311
Spine	106
Thorax	656
Neck	106
Head	280
Front Leg	266
Rear Leg	231
Total	3423

FIG. 7 – Poids des différentes parties du chien (sans unités).

3.2 Les articulations

Les os sont reliés entre eux par des articulations, possédant un ou deux degrés de liberté et des limites en rotation.

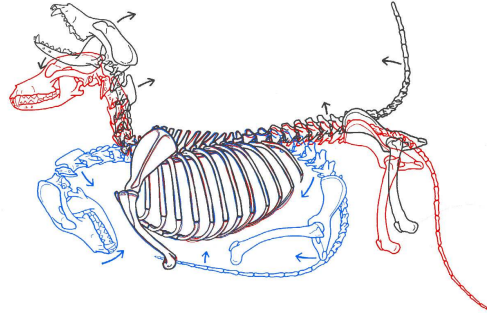


FIG. 8 – Amplitude des articulations de la colonne vertébrale du chien.

Ne disposant que de très peu d'informations sur les limites des articulations des quadrupèdes, nous nous sommes basés sur des dessins d'un artiste (cf. fig. 8), ayant représenté les amplitudes du dos du chien, et des jambes du cheval, pour paramétrer les articulations de nos squelettes. La plupart de ces liaisons sont des liaisons pivot. Les axes de rotation sont extraits du travail de Reveret et al. [4]. Les amplitudes des rotations sont elles tirées des dessins.

Les articulations situées sur la colonne vertébrale possèdent plus de degrés de liberté, mais moins d'amplitude. Les liaisons pour les jambes, situées au niveau du pelvis et du thorax, possèdent un degré de liberté de plus que les autres liaisons de la jambe, dans l'axe de la colonne (cf. fig. 9).

Os	twistmin	twistmax	swing1	swing2
tails	-20	40	0	20
spines	-5	10	10	10
thorax	-10	10	10	10
neck0	-5	60	0	10
necks	-5	10	0	10
head	-30	10	0	10
scapula	-5	15	45	0
shoulder	-10	20	0	0
elbow	-100	0	0	0
wrist	0	130	0	0
fpha0	-30	100	0	0
hip	-20	35	45	0
knee	0	85	0	0
ankle	-110	0	0	0
hpha0	-40	100	0	0
hpha	0	10	0	0

FIG. 9 – Angles limites en degrés sur les 3 axes pour chacune des articulations.

Twistmin et twistmax correspondent aux limites autour de l'axe principal de l'articulation, en général aligné sur l'axe horizontal perpendiculaire à la colonne. Les limites pour les deux autres axes, sont symétriques (swingmin = -swingmax), swing1 étant souvent aligné sur l'axe de la colonne, swing2 sur l'axe vertical.

4 Méthodes de contrôle

Pour contrôler les mouvements des quadrupèdes, nous utilisons deux méthodes d'application de forces. La première s'applique directement sur les articulations. C'est la méthode classique, utilisée généralement en biomécanique. La deuxième est une sorte de simplification de celle-ci, qui peut s'appliquer à un ensemble de solides articulés. Nous utilisons aussi une information extraite des solides, appelé vecteur d'équilibre, qui sert à connaître dans quelle direction l'équilibre peut être trouvé.

4.1 Commande angulaire des articulations

Pour chaque articulation, peut être défini un ressort, qui exerce une force sur l'articulation afin de l'amener dans une position en rotation donnée. Ce mode de commande se rapproche assez du fonctionnement musculaire. Cette méthode est largement utilisée en biomécanique, pour modéliser l'action des muscles, dans l'objectif de créer un modèle de locomotion le plus réaliste possible.

Les objectifs que nous nous sommes fixés, imposent que notre modèle fonctionne en temps-réel. Contrôler le mouvement par commande angulaire, nécessite de lourds calculs d'optimisation, afin d'interpréter correctement la volonté de mouvement de l'animal. En général, nous utiliserons cette force uniquement comme force de rappel, pour ramener la position actuelle d'une articulation vers sa position initiale qui est celle au repos. Afin de contrôler le mouvement, nous avons créé un autre type de commande, agissant de manière plus globale sur le squelette, ne nécessitant pas de gros calculs.

4.2 Les couples de forces, ou segments de contrôle

Nous disposons donc de ce que l'on appelle un ragdoll (poupée de chiffon), c'est à dire une sorte de pantin virtuel. Après quelques simples expérimentations avec ce ragdoll, nous avons constaté qu'il était possible de recréer artificiellement des mouvements plausibles pour l'animal, en tirant sur ses membres à l'aide de forces extérieures, comme s'il s'agissait de fils attachés à un pantin, dirigés par un animateur.

Cette méthode est très intéressante, car elle offre une grande liberté de contrôle à la personne chargée d'animer l'animal, le résultat souhaité pouvant être réaliste ou non. Cependant, dans l'idée de créer un comportement plausible, certaines restrictions doivent être apportées. Une règle simple à respecter afin que les mouvements paraissent cohérents, est que les forces appliquées s'annulent (cf. fig. 10).

$$\sum \text{forces exterieures} = 0$$



FIG. 10 – Exemple d'application de forces pour tendre la jambe avant.

Ainsi, pour tendre une jambe, on tire vers le sol à la base de celle-ci, comme par exemple à la cheville, et l'on pousse vers le ciel l'autre extrémité, comme par exemple la hanche. De ce fait, l'animal ne se déplace pas dans l'espace car les forces appliquées s'opposent, mais la jambe se tend.

Afin de formaliser un peu plus cette idée, nous avons mis en place des segments de contrôle. On définit deux points d'attache sur le quadrupède, qui sont les deux extrémités du segment. On peut faire varier la direction de ce segment de contrôle, en appliquant une force à une extrémité (et la force opposée à l'autre extrémité), dans la direction finale souhaitée (cf. fig. 11).

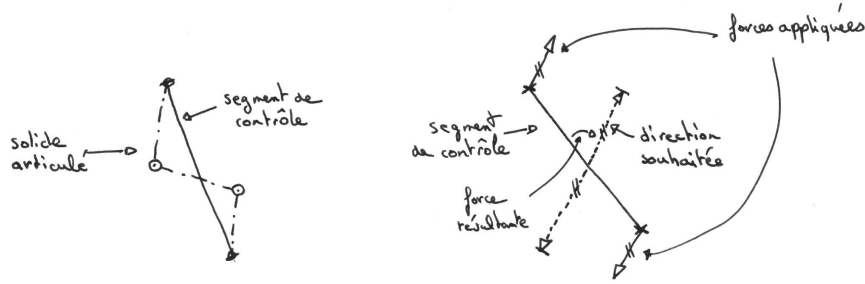


FIG. 11 – points d’applications d’un segment de contrôle, forces mises en jeu pour le changement d’orientation.

Avec ce segment de contrôle, on peut aussi déterminer une longueur de segment à respecter. Dans ce cas, on considère que les deux extrémités sont reliées par un élastique. Si la longueur est trop longue par rapport à celle demandée, deux forces opposées les attirent. Si la longueur est trop courte, deux forces opposées les repoussent.

Cet outil peut donc créer du mouvement au sein du quadrupède, mais ceci ne garantit en aucun cas qu’il va pouvoir se mouvoir au sein de son univers. La locomotion se crée, lorsqu’il rentre en contact avec son univers. Par exemple, la gravité projette le quadrupède vers le sol, et lorsque le contact entre ces deux entités a lieu, les forces résultantes de l’interaction déplacent les deux corps dans la direction correspondant au choc.

4.3 Vecteur d’équilibre

L’un des principes fondamentaux de la locomotion pour un quadrupède, est de devoir garder son équilibre. C’est un élément important, qui est sans cesse remis en question dans un cadre dynamique. Lors d’un mouvement, le quadrupède recherche en permanence son équilibre. Afin de modéliser cette volonté de stabilité, nous avons recherché quel critère le positionnement des jambes devait respecter. Finalement, un calcul ressemblant au ZMP [6] en simplifié nous a fourni de bons résultats.

En robotique, le ZMP (Zero Moment Point) est défini comme le point spécifique sous le mécanisme du pied du robot auquel l’effet de toutes les forces agissant sur la chaîne du mécanisme peut être remplacé par une force unique, et à ce que tous les moments de rotation soient égaux à zéro.

Notre vecteur d’équilibre est défini de la sorte, à partir d’un point quelconque :

$$\overrightarrow{equilibre} = \overrightarrow{gravite} + \overrightarrow{vitesse\ linéaire}$$

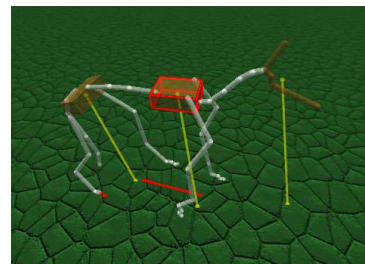


FIG. 12 – Vecteur d’équilibre tracé en jaune pour les 3 masses principales.

Ce vecteur donne une information sur la direction que tend à prendre le point du corps concerné (cf. fig. 12). C’est très utile, comme par exemple pour placer la jambe de manière à maintenir le corps de l’animal au dessus du sol, là où il a tendance à venir s’écraser.

5 Contrôleurs expérimentés

Disposant de ces divers outils, plusieurs méthodes ou concepts de contrôle peuvent être opérés. Voici une description des diverses approches intéressantes mises en place.

5.1 Rigide

Ce contrôleur illustre très bien l'utilisation des segments, élément clé de tous les contrôleurs expérimentés. C'est la seule technique employée ici, pour contrôler le quadrupède. Le corps est divisé en 7 segments :

- la queue
- la colonne
- le cou
- les 4 jambes

Chaque segment reçoit une direction cible, qu'il va donc s'efforcer de calquer, d'après le principe expliqué dans la section précédente (cf. fig. 11) . Une force est attribuée à chaque segment, égale à la distance entre les 2 extrémités du segment élevée au carré. A chaque étape de calcul, on ajoute à une extrémité une force égale à la direction recherchée multipliée par la force du segment, et à l'autre extrémité, sa force opposée.

Voici le tableau récapitulant pour chaque segment, les valeurs mises en jeu :

Segment	Direction	force
Queue	(0,-1,-1)	1490
Colonne	(0,0,1)	2132
Cou	(0,1,1)	335
Jambe Arrière	(0,-1,0)	2922
Jambe Avant	(0,-1,0)	3804

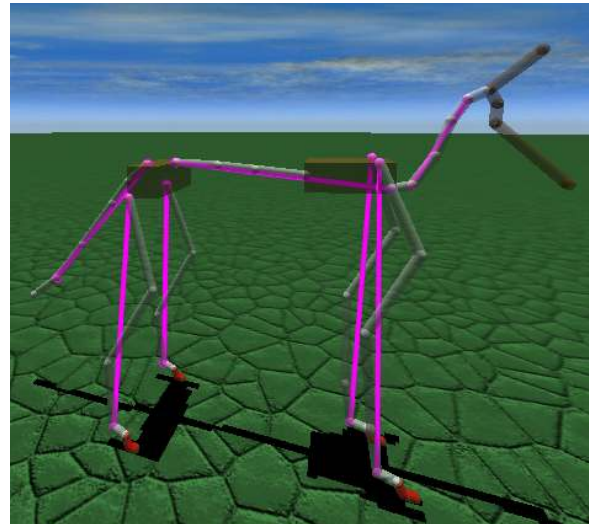


FIG. 13 – Direction et force imposées aux segments de contrôle (tracés en violet).

Avec ce contrôleur, le quadrupède reste en place. Il est assez rigide, et ne tombe pas, pour peu que la gravité soit dirigée dans l'axe (0,-1,0). Mais il ne possède aucun mouvement volontaire, même s'il est sujet à une interaction, comme par exemple si on le pousse. Il va subir passivement l'action, ses pieds vont tout d'abord rester collés à l'endroit où il se trouve. Puis lorsque la différence entre la direction du segment de la jambe devient trop grande avec celle imposée, par rapport à la friction exercée par le contact avec le sol, le quadrupède se déplace, les jambes frottant contre le sol.

5.2 Pointe la cible

Pour ce contrôleur, nous avons repris la même configuration que précédemment. A la différence près, que les directions imposées aux segments, peuvent évoluer dans le temps. Une cible a été placée à un endroit de l'univers. A chaque étape, on demande aux segments du cou, de la colonne, de la queue, et de 2 jambes : l'avant droite et l'arrière gauche, de se placer en direction de cette cible.

La direction appliquée à leur segment devient donc :

$$direction = \frac{\overrightarrow{position\ du\ segment - position\ de\ la\ cible}}{||\overrightarrow{position\ du\ segment - position\ de\ la\ cible}||}$$

Aux deux autres jambes restantes, la direction choisie à chaque étape est celle du vecteur d'équilibre, calculée à l'extrémité du segment reliant la jambe à la colonne (cf. section 4.3). Ceci nous donne une bonne approximation de la direction dans laquelle la jambe doit se trouver, afin que l'animal se tienne en équilibre. Voici le type de résultat obtenu, quelques secondes après le déclenchement de ce contrôleur.

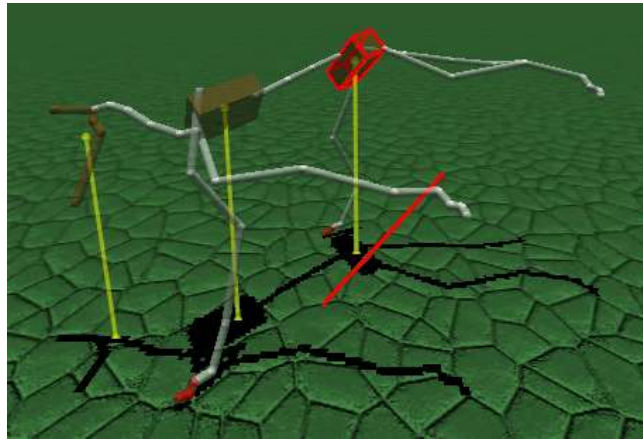


FIG. 14 – Le quadrupède se positionne en équilibre sur 2 jambes, et pointe sa cible

Le cou, la queue, et les 2 jambes pointent correctement dans la direction souhaitée (cf. fig. 14). La colonne quand à elle, n'est pas inclinée comme on le lui a demandé. Ceci est dû aux contraintes imposées par le squelette et ses bloquages en rotation. Malgré cela, le squelette trouve un point d'équilibre, qui satisfait au mieux les directions voulues. Les deux autres jambes se positionnent afin que le corps soit bel et bien au dessus du sol, les pieds an l'air contrebalançant correctement le poids de la tête.

5.3 Les postures

Pour ce contrôleur, les jambes ont cette fois 3 paramètres : la direction, la longueur au repos, et le ratio de cette longueur à respecter. Grâce à ce paramètre de longueur de jambe, expliqué auparavant (cf. section 4.2), nous avons ainsi pu définir 3 attitudes pour le quadrupède : assis, debout, couché (cf. fig. 16). Pour les positions hautes, une valeur supérieure à 1 est nécessaire (cf. fig. 15). Ceci est dû à l'influence de la gravité, qu'il faut prendre en compte. Ainsi, le changement sans transition de ces valeurs, amène le quadrupède à passer d'une position à une autre de manière fluide.

Attitude	Jambe Arrière	Jambe Avant
Debout	1,3	1,3
Assis	0,5	1,3
Couché	0,5	0,5

FIG. 15 – Valeurs des longueurs de jambes selon l'attitude.

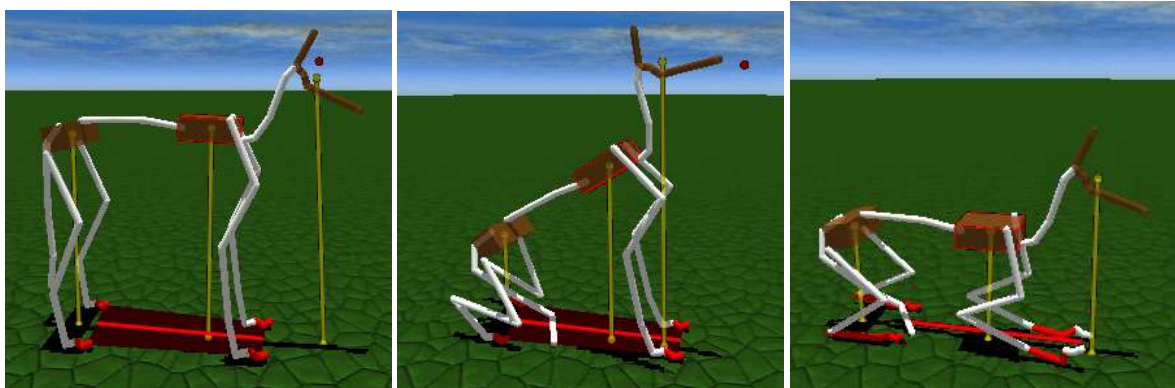


FIG. 16 – Debout, assis et couché.

5.4 La laisse

Une fois que le travail de recherche d'équilibre fut satisfaisant, l'objectif suivant a été de simuler la locomotion. Une première idée pour y parvenir, est de créer un déséquilibre chez notre animal en lui tirant dessus, un peu comme avec une laisse, le quadrupède étant censé se positionner automatiquement afin d'accompagner le mouvement, pour retrouver son équilibre perdu.

Pour cela, seul le contrôle des jambes varie au cours du temps. En effet, toutes les articulations partant du pelvis jusqu'à la tête ont pour contrôleur, un élastique qui tend à les repositionner dans leur position d'origine (cf. section 4.1) . Ceci afin d'éviter que la colonne ne se courbe trop de manière irréaliste, et entraîne un désalignement important du thorax et du pelvis. Cela sert aussi à maintenir la tête droite, pour ne pas qu'elle traîne par terre. C'est aussi le cas des phalanges, qui pour éviter que le pied ne se retourne, reviennent dans leur position d'origine. Ce qui permet aussi de recréer une légère poussée automatique, quand le pied force contre le sol. La queue quand à elle, n'a aucun contrôle.

Afin d'accompagner un mouvement créé par déséquilibre, les jambes ont deux états possibles : en recherche d'équilibre, ou pied porteur. Pour le premier, la jambe va se déplacer en direction du vecteur d'équilibre. Pour le second, elle va tout simplement rester en place, tout en respectant la longueur de jambe indiquée. Les jambes passent d'un état à un autre, lorsque la direction actuelle de la jambe dépasse un certain seuil de tolérance par rapport à la direction d'équilibre.

Les résultats furent plutôt encourageant, mais loin d'être concluant. A faible vitesse, l'animal arrive à faire quelques pas, lorsqu'on lui tire la tête en avant doucement, mais a tendance à laisser trainer une patte avant. Si on lui tire fortement sur la laisse, les jambes s'alignent au niveau de son corps, sans opérer de mouvement semblable au galop. En fait, le seuil de changement d'état est difficile à déterminer, et les jambes restent souvent bloquées dans un des deux états. Des valeurs numériques simples comme par exemple un angle limite, n'ont pas été satisfaisantes, car un tel critère se doit d'être dynamique, selon la situation.

Ceci a donc constitué un premier pas dans l'étude de la locomotion, et nous a permis de mettre en évidence l'efficacité du vecteur d'équilibre, et la difficulté pour déterminer le changement d'état pour accompagner un mouvement.

5.5 Jambe par jambe

Ce contrôleur poursuit l'idée émise auparavant, en remplaçant l'état de pied porteur par un état de poussée. Cette poussée s'effectue en appliquant sur l'extrémité de la jambe rattachée à la colonne, une force dirigée dans la direction que le quadrupède souhaite prendre, la force opposée étant appliquée sur les chevilles.

Selon l'intensité de cette force, la poussée est plus ou moins forte. Avec une faible poussée, le quadrupède avance légèrement de lui-même, mais n'enchaîne pas les pas correctement. Avec une poussée plus forte, il patine sur place, et n'avance pas bien plus vite qu'auparavant, il recule même parfois. Nous avons à nouveau rencontré le même problème, qui est de définir correctement le critère pour changer l'état d'une jambe.

On aurait pu penser que le fait de contrôler volontairement la marche, simplifierait le choix du statut de fonctionnement de la jambe, mais ce n'est pas le cas. Bien choisir ce critère de changement d'état est primordial pour simuler la locomotion. Devant la difficulté rencontrée pour le définir correctement, nous avons décidé d'explorer un autre concept, permettant de contourner ce problème : le contrôle par la colonne vertébrale.

5.6 Par la colonne

Dans le domaine de l'étude de la locomotion, deux stratégies s'opposent : soit la locomotion est dirigée par les jambes, soit c'est à travers la colonne vertébrale que tout est orchestré. Après avoir exploré en partie la première théorie, nous nous sommes donc intéressés à la deuxième.

En analysant des vidéos mettant en scène des quadrupèdes, au travers de diverses allures, on peut apercevoir un mouvement régulier au niveau des hanches. Ce mouvement peut être assimilé à un pédalier, comme si la hanche était traversée par une pagaie, et que l'animal s'en servait pour ramer sur les cotés.

Nous avons donc modélisé ce phénomène en appliquant deux forces opposées de chaque côté des hanches qui tournent autour d'un axe perpendiculaire à la colonne (un peu comme une roue) dans un cycle de 38/30ème de secondes, ce qui correspond à un cycle normal de marche. On applique aussi le même cycle de force sur les épaules, en décalage de phase.

Le contrôle des pattes est déduit de la colonne. On demande aux pattes de venir calquer leur direction sur le vecteur d'équilibre pris à l'extrémité de la patte rattachée à la colonne, au même endroit où l'on applique le cycle de force de pédalier. Les pattes sont contrôlées indépendamment. Les forces extérieures opposées sont appliquées pour l'arrière à la cheville et à la hanche, et pour l'avant sur l'épaule et le genou. En effet, après divers essais, il s'avère qu'un contrôle par le genou plutôt que par la cheville pour les jambes avant est bien plus efficace, et donne un rendu plus proche de la réalité (cf. fig. 17). Les pieds avant avaient tendance à gratter le sol à chaque mouvement, incitant le pied à se retourner. Ceci est évité en agissant sur le genou.

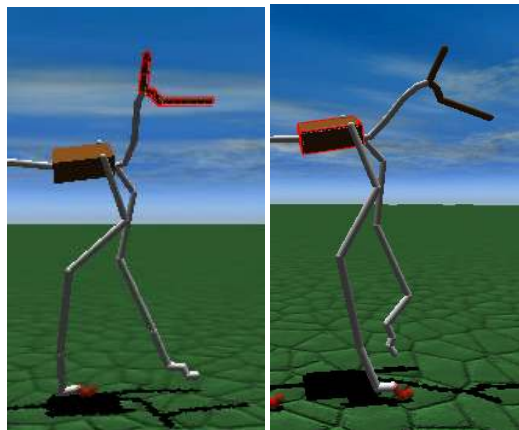


FIG. 17 – Contrôle sur la cheville, contrôle sur le genou.

Ce modèle fonctionne très bien, l'animal marche, à une allure respectable, sans frotter par terre. Seules, les hanches et les épaules, censées être en décalage de phase, fonctionnent en phase dans la simulation. Cette allure correspond donc à la marche à l'amble, alors que nous voulions modéliser la marche normale. Nous n'avons pas eu le temps d'approfondir la raison de cette perte de décalage de phase, mais c'est probablement dû à la réaction de la colonne, qui s'oppose au déphasage, par sa résistance en torsion.

Car comme précédemment, nous avons placé des élastiques angulaires sur toute la colonne, et sur les pattes. De plus, une force entre le thorax et le pelvis impose à la colonne d'être tendue en permanence. Une force impose aussi à toute la colonne de s'orienter vers l'objectif à atteindre (cela joue le rôle de gouvernail). Cette force tire la tête vers l'objectif, tandis que la moitié de l'opposé de cette force est appliqué sur le thorax, l'autre moitié sur le pelvis.

6 Résultats

Grâce à ce dernier contrôleur, les résultats ont été très intéressants, le quadrupède pouvant se déplacer de lui-même. Le principal atout de cette méthode de simulation physique, est sa réactivité face à un environnement dynamique. Nous avons voulu pousser cet algorithme à ses limites, et avons commencé par créer une petite colline dans l'univers, que notre quadrupède se doit de grimper. Il a alors buté dessus, réussi à monter un peu ses jambes avant, mais sans arriver à poser ses jambes arrières sur la colline, il glissait puis se retrouvait à son point de départ. Si on le posait sur le sommet de la colline pour le faire descendre, il réussissait à descendre, mais avait tendance à pédaler dans le vide avec ses jambes avant, ce mouvement étant loin de paraître réaliste.

Pour y remédier, nous avons ajouté un couple de forces, afin d'orienter verticalement la colonne, en fonction du relief rencontré sous les pieds. On extrait l'information donnée par la normale du contact entre les pieds du quadrupède et le sol, afin de déterminer l'orientation du terrain. Cette normale est fournie quelque soit la géométrie de l'objet rencontré, par le moteur physique. Ainsi, l'on peut savoir si le quadrupède se trouve dans une descente ou une montée. Si il monte, on lui applique alors une force qui a tendance à plaquer son pelvis au sol, et à relever son thorax. A l'inverse, si il descend, on plaque son thorax vers le sol, et relève son pelvis.

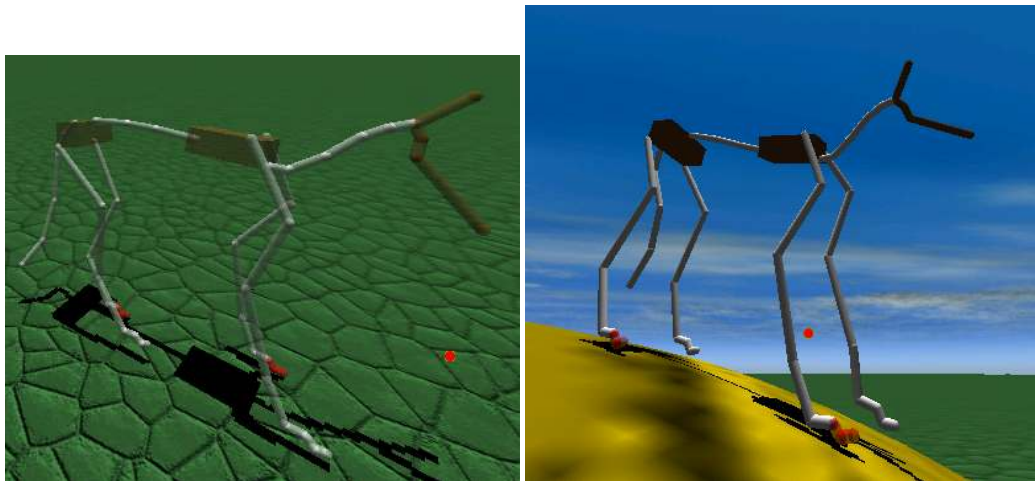


FIG. 18 – Marche à l'amble, et descente contrôlée du quadrupède.

Cette force dynamique extraite du sol, a été très convaincante. Quand le quadrupède rencontre la colline, et que la pente est forte, il lance alors ses pattes avant en l'air, ce qui lui permet d'avancer un peu sur la colline, d'éviter de taper bêtement dans la pente. Le pelvis étant plaqué quelque peu vers le bas, il joue un rôle plus porteur. Puis une fois que tout son corps est placé sur la pente, il monte sans soucis (pour peu que la pente ne soit pas trop forte). Lorsque le quadrupède descend, il ne pédale plus en l'air, et nous donne l'impression au contraire de contrôler sa descente. Quand la pente devient trop raide, les pattes arrières glissent, et le quadrupède contrôle cette glisse, avant de retomber sur le sol plat, de se remettre en place pour avancer, et de reprendre son rythme normal, imposé par les hanches.

Ce modèle est donc plutôt réussi, mais n'apporte pas tous les éléments nécessaires. Nous avons essayé de faire descendre un escalier assez raide au quadrupède, et celui-ci se retrouve passablement bloqué. Autre point noir que cet escalier a mis en évidence, non seulement le quadrupède reste bloqué, mais il a les pattes avant qui flottent en l'air (cf. fig. 19). Comme s'il y avait un sol invisible. En réalité, les forces mises en jeu sur l'arrière train et la colonne, sont bien trop importantes, par rapport à la gravité, et à la structure du quadrupède définie. Cependant, si l'escalier n'est pas trop raide, cet effet ne se fait pas ressentir.

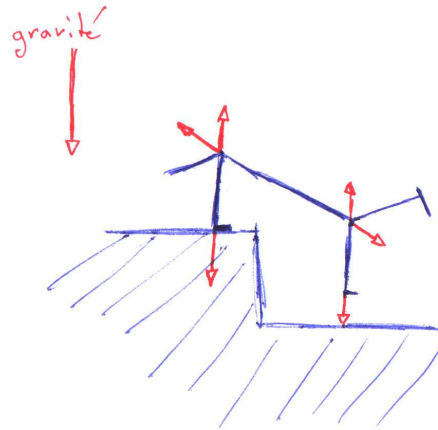


FIG. 19 – Forces sur les jambes arrières et la colonne trop élevées.

Malgré certains cas particuliers, le rendu est bon. On peut jeter des objets sur le quadrupède, celui-ci encaissera le choc, et se repositionnera pour contrecarrer cet événement, avant de reprendre sa route normale. Il peut aussi pousser des objets, s'ils se trouvent sur son itinéraire (cf. fig. 20).

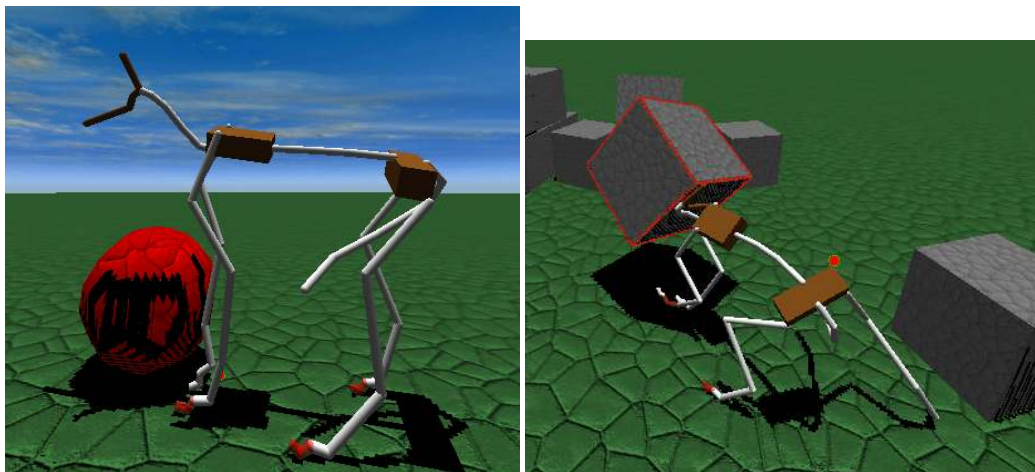


FIG. 20 – Poussée d'un ballon, et résistance face à un bloc projeté sur le quadrupède.

7 Conclusion

Permettre d'animer grâce à la physique uniquement, des animaux ou un quelconque personnage, serait une grande avancée dans le domaine du jeu vidéo, ou des acteurs virtuels dans le cinéma. Cette méthode d'animation peut faire gagner beaucoup de temps, puisque le personnage est directement en interaction avec son environnement. Cependant, ce n'est pas une chose aisée, surtout si l'on veut que ce soit exploitable en temps-réel pour le jeu-vidéo. Les algorithmes nécessitent d'être très rapides et efficaces, on ne peut donc pas se permettre de simuler l'entière réalité, comme dans le domaine de la biomécanique.

Dans ce travail, nous avons montré qu'il était possible de rendre des animations assez réalistes, en se contentant de manipuler la structure physique d'un quadrupède, comme un pantin. Les premières phases d'expérimentation ont commencé à montrer, que considérer la colonne vertébrale comme appareil moteur procure des mouvements réalistes de manière plus évidente qu'en considérant chacune des pattes comme un élément moteur.

Tout ce travail a globalement souffert du manque de données anatomiques sur les quadrupèdes. Nous avons dû faire énormément de suppositions sur entre autres les masses de chacune des parties du squelette, et les amplitudes des articulations. La comparaison avec un animal existant aurait été largement mieux exploitée. Mais il faut tout de même avoir à l'esprit qu'il doit être possible d'animer avec cette technique, des animaux imaginaires, dont l'anatomie est complètement arbitraire. Donc l'incohérence des proportions corporelles ne doit pas être un problème, peut-être juste un frein pour mettre au point certaines règles génériques de comportement devant ressembler à la réalité.

Créer un contrôleur pouvant s'adapter automatiquement à l'animal, suivant certains critères spécifiés, serait nécessaire à mettre en place pour une éventuelle utilisation professionnelle. Aussi, il serait intéressant de poursuivre sur l'idée que la locomotion est régie par la colonne vertébrale, et voir tout ce que cela peut apporter. Il faudrait améliorer le système de répartition des forces, le fait de simplement appliquer une force opposée à un endroit quelconque du corps, ne garantit pas forcément que les efforts s'annulent. Enfin et surtout, un seul type d'allure a pu être mis en place, et ce n'est pas celui qui a été voulu. Il est impératif de comprendre cet effet, afin de mieux contrôler les différents types de marche.

Ce sujet contribut au développement de méthodes de création en temps-réel d'animations dynamiques, pour les applications réactives comme les jeux-vidéos. Aujourd'hui, aucun jeu-vidéo commercialisé ne possède ce type d'animations émergentes, mais on trouvera certainement plus d'animations entièrement générées par la physique dans ce type de produits à l'avenir. Aussi, le logiciel ainsi développé au cours de ce stage, sera utilisé en collaboration avec le Museum National d'Histoire Naturelle de Paris, afin d'étudier à partir de données réelles le comportement adopté par l'animal pour se déplacer, dans le cadre du projet ANR KAMELEON [8].

A Annexes

Avant de pouvoir débiter les expérimentations, un long travail de génie logiciel à du être effectué. En voici brièvement les grandes lignes.

A.1 Logiciel de simulation

Le logiciel de simulation fût écrit en C++. Plusieurs librairies externes ont été utilisées afin d'en simplifier la programmation :

- glfw pour la gestion des entrées-sorties
- opengl pour la représentation des objets 3D
- fnt (de plib) pour l'affichage de texte
- ul (de plib) pour un accès aisé aux fichiers et répertoires sur le disque dur
- xmlparser pour lire et ranger des données xml dans un arbre
- novodex pour la résolution des calculs dynamiques liés à la physique des objets [7]

Nous avons choisi d'utiliser NovodeX comme simulateur physique, de part son nombre de fonctionnalités, sa stabilité et sa bonne documentation. Mais aussi car ce moteur permet d'exploiter les cartes physiques, pouvant accélérer en grande partie la résolution des calculs de la dynamique. Le moteur est configuré pour un pas de temps fixe. Chaque pas dure 1/120ème de seconde, et on calcule 4 pas pour une frame d'affichage. L'affichage est donc bloqué a 30fps, afin de visualiser et calculer la scène en temps-réel.

Les liaisons entre les objets rigides sont en réalité modélisés par des élastiques de raideur infinie. Le simulateur itère donc la résolution des contraintes, jusqu'à obtenir une erreur inférieure à un certain seuil. Par défaut, le simulateur n'opère pas plus de trois itérations, avant de passer à l'étape de temps suivante. Pour obtenir une réaction cohérente pour notre squelette de quadrupèdes, disposant d'une quarantaine d'objets rigides inter-connectés, nous devons augmenter ce nombre d'itérations jusqu'à au moins environ 32 itérations. A titre d'exemple, en dessous de ce nombre, la colonne vertébrale peut avoir tendance a se vriller, les jambes de devant pouvant alors pivoter de 360 degrés, tandis que les jambes arrières restent en place. Cet effet disparaît en augmentant ce nombre d'itérations.

Aussi, la technique des shadow maps a été implémentée, pour une meilleure appréhension de la profondeur, en particulier pour visualiser si l'animal se situe loin ou non du sol.

Afin d'importer les quadrupèdes dans le logiciel, un format de fichier xml représentant un quadrupède a été défini. Il est donc possible de charger a tout moment un ou plusieurs quadrupèdes à partir de fichiers, mais aussi de charger des objets rigides simples, comme une boîte, une sphere, ou un capsule, pour créer un univers complexe et dynamique dans lequel évolue nos quadrupèdes.

A.2 Scripts Maya

Les squelettes de quadrupèdes présentés par Reveret et Al. [4] sont stockés dans des fichiers au format du logiciel Maya. Pour pouvoir exploiter ces données au mieux, nous avons développé un script Maya d'export de squelettes, dans un format exploitable par un logiciel tiers.

Les données utiles sont les positions des articulations dans le repère global, ainsi que la direction de l'axe de rotation principal pour chacune de ces articulations. A ces données, doivent être ajoutées les limites angulaire en rotation de ces articulations. Pour cela, un autre script a été développé, venant rajouter aux fichier Maya ces informations. Le script est facilement éditable, afin de modifier ses informations, pour le moins approximative, à l'heure où ce stage fût effectué.

Le format d'exportation choisi, est un dérivé simplifié du format pml, qui utilise une syntaxe inspiré du xml, mais adapté à la description des scènes physiques. Ceci permet une relecture facile, et éventuellement une édition à la main pour modifier certaines caractéristiques précises.

B Bibliographie

Références

- [1] Laszlo, J. F., van de Panne, M., Fiume, E., “Interactive Control For Physically-Based Animation,” *ACM SIGGRAPH*, 2000. <http://www.cs.ubc.ca/~van/papers/>
- [2] Popović, Z., Karen Liu, C., “Synthesis of Complex Dynamic Character Motion from Simple Animations,” *ACM SIGGRAPH*, 2002. <http://www.cs.washington.edu/homes/zoran/>
- [3] Zordan, V. B., Majkowska, A., Chiu, B., Fast, M., “Dynamic Response for Motion Capture Animation,” *ACM SIGGRAPH*, 2005. <http://graphics.cs.ucr.edu/rglRes.html>
- [4] Reveret, L., Favreau, L., Depraz, C., Cani, M.P., “Morphable model of quadrupeds skeletons for animating 3D animals,” *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005. <http://www-evasion.imag.fr/people/Lionel.Reveret/>
- [5] Sabine Renous, *Locomotion*, Editions Dunod, 1994.
- [6] Vukobratović, M. K., Rodić, A. D., “Contribution to the Integrated Control of Artificial Human Gait,” *2nd Serbian-Hungarian Joint Symposium on Intelligent Systems (SISY)*, Subotica, Serbia and Montenegro, October 1-2, 2004. <http://www.bmf.hu/conferences/sisy2004/vuk.pdf>
- [7] AGEIA, *PhysX API*, <http://www.ageia.com>
- [8] Museum National d’Histoire Naturelle de Paris, Paris 5ème : Jardin des Plantes, 2 rue Buffon. <http://www.mnhn.fr/museum/foffice/transverse/transverse/accueil.xsp>