



**HAL**  
open science

# Simultaneous planning of staging and filming in virtual cinematography

Philippe Rannou

► **To cite this version:**

Philippe Rannou. Simultaneous planning of staging and filming in virtual cinematography. Synthèse d'image et réalité virtuelle [cs.GR]. 2010. dumas-00530764

**HAL Id: dumas-00530764**

**<https://dumas.ccsd.cnrs.fr/dumas-00530764v1>**

Submitted on 29 Oct 2010

**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.

Université de Rennes 1  
ENS Cachan - Antenne de Bretagne

Stage Master2 Recherche M2RI  
Au sein de l'équipe Bunraku  
IRISA/INRIA Rennes

Directeur de stage : Marc Christie

Rapport de stage : Simultaneous planning of staging  
and filming in virtual cinematography

Philippe RANNOU  
<philippe.rannou@gmail.com>

Rennes, le 4 juin 2010

# Table des matières

<b>1</b>	<b>État de l'art</b>	<b>4</b>
1.1	Planification de mouvement . . . . .	4
1.1.1	Représentation du robot . . . . .	4
1.1.2	Espace des configurations . . . . .	4
1.1.3	Recherche de chemin dans l'espace des configurations . . . . .	5
1.2	Positionnement de caméra . . . . .	9
1.2.1	Cinématographie . . . . .	10
1.2.2	Règles cinématographiques . . . . .	10
1.2.3	Planification de caméra . . . . .	11
<b>2</b>	<b>Modèle</b>	<b>16</b>
2.1	Hypothèses . . . . .	16
2.2	Vue générale du problème . . . . .	16
2.3	Modélisation du problème . . . . .	17
2.3.1	Formalisation d'un plan cinématographique de référence pour deux acteurs . . . . .	17
2.3.2	Contraintes sur le modèle . . . . .	18
2.3.3	Abstraction du modèle . . . . .	19
2.4	Représentation de l'espace des configurations . . . . .	21
2.4.1	Décomposition de l'espace . . . . .	21
2.4.2	Exploration des sous-espaces à l'aide de roadmaps probabilistes . . . . .	21
2.4.3	Propriétés du modèle basé plans de référence . . . . .	23
2.5	Calcul et lissage de la trajectoire . . . . .	24
2.5.1	Parcours des PRMs . . . . .	24
2.5.2	Lissage de trajectoire . . . . .	27
2.5.3	Réduction de l'espace de recherche . . . . .	29
<b>3</b>	<b>Résultats et perspectives</b>	<b>33</b>
3.1	Résultats . . . . .	33
3.2	Perspectives . . . . .	34

## Introduction

Dans le domaine de la synthèse d'image et plus particulièrement dans celui de l'animation, la cinématographie virtuelle est très souvent utilisée, dans les domaines tels que les jeux vidéos, l'entraînement virtuel ou le virtual story telling, il est utile voire nécessaire de filmer des acteurs virtuels à l'aide d'une caméra. Que ce soit pour des questions techniques (vision d'une zone de travail pour un ouvrier virtuel), stratégique (vue d'un personnage non-joueur dans les jeux vidéos) ou artistique, ce problème consiste à transmettre par une vidéo des informations intelligibles sur une scène : c'est ce qu'on appelle la cinématographie virtuelle. Dans beaucoup de ces domaines, le problème est résolu en deux phases :

1. planification de la trajectoire des acteurs dans l'environnement,
2. planification de la trajectoire de la caméra à partir de la trajectoire des acteurs.

La caméra est alors asservie aux acteurs dans le sens où ses configurations sont calculées à partir des positions des acteurs. Cette simplification (diviser pour régner) du problème permet de limiter la complexité des calculs (Ce qui est utile pour un rendu en temps réel par exemple). Cependant, quand les configurations de la caméra dépendent de celles des acteurs, il arrive que le rendu ne puisse pas être de bonne qualité (Exemple dans les jeux vidéos : les déplacements des personnages sont contrôlés par le joueur, la caméra doit alors offrir le meilleur rendu possible pour une configuration des personnages donnée : elle est asservie aux acteurs. Voir figure 1).



FIG. 1 – Captures d'écran prise dans un jeu vidéo. Dans la première image, le point de vue de la caméra est de bonne qualité; dans la deuxième, la caméra tente de se placer le mieux possible pour une configuration du personnage donnée, mais le résultat n'est pas de bonne qualité.

Afin de palier à cette perte de qualité, nous proposons une approche qui combine simultanément le placement à la fois des acteurs et de la caméra. Nous nous plaçons dans le cas de deux acteurs humanoïdes et d'une caméra les filmant à travers un environnement « simple » ; notre approche consiste à considérer ces trois entités comme un *triptyque*, et à trouver un chemin pour ce triptyque à travers l'environnement par une méthode de planification de mouvement, tout en maximisant la qualité du film qui en ressortira. Notre objectif principal est l'obtention d'un film de bonne qualité, sans contraintes de temps réel. Nous

avons donc pensé un modèle adapté à cet objectif et nous avons implémenter (en partie) notre méthode.

**Organisation du document** Nous verrons dans une première partie l'état de l'art sur les différentes techniques de planification de mouvement, les contraintes supplémentaires intervenant dans la planification de caméra ainsi que certaines règles cinématographiques. Ensuite, nous décrirons les propriétés que nous avons souhaité exprimer à travers notre modèle.

## 1 État de l'art

Dans cette section, nous faisons un aperçu des techniques existantes pour la planification de mouvement. Nous décrivons également les techniques de réalisation de film virtuel, en quoi ce problème est une spécialisation du problème de planification de mouvement et quelles sont les règles qui s'y applique.

### 1.1 Planification de mouvement

Dans cette partie, on considérera le cas de la planification de mouvement d'un robot. On décrira dans un premier temps la représentation interne d'un robot par un arbre cinématique, puis comment le plonger dans un environnement, enfin on décrira des méthodes d'exploration de cet espace.

#### 1.1.1 Représentation du robot

Une manière de représenter le robot est de l'assimiler à un arbre cinématique (décrit dans [LaV06]) de solides rigides, reliés par des articulations. Ces articulations, que l'on peut toujours considérer comme des rotations, sont représentables formellement par des matrices de rotations. De plus, chaque point de chaque solide est représentable par une translation par rapport au centre articulaire racine.

Étant donné que les solides sont rigides, les seuls paramètres variables sont les rotations des articulations, ainsi la configuration d'un robot est entièrement définie par la position de la racine de l'arbre et de la mesure des angles de rotation des articulations (le squelette a donc autant de degrés de liberté que de paramètres).

Ainsi, dans une configuration donnée, n'importe quel point du robot peut être calculé par une composition de matrices homogènes appliquée à la position de la racine ; ce qui permet de reconstruire entièrement l'image du robot grâce à ces seuls paramètres. On peut également retrouver la valeur de ces paramètres à partir des points du robot ; l'avantage de représenter les états du robot d'une telle manière est que si on trouve un « chemin » entre deux configurations du robot, alors on connaît la suite de transformations qu'il faudra appliquer à ses articulations pour passer d'une configuration à l'autre.

#### 1.1.2 Espace des configurations

On peut donc représenter les configurations possibles dans un espace à autant de dimensions qu'il y a de degrés de liberté dans le robot. Dans cet espace, appelé espace des configurations, il existe cependant des positions « interdites »

qui correspondent par exemple à des configurations dans lequel le robot est en collision avec un élément de l'environnement, en collision avec lui même ou dans une configuration où un de angles de rotation ne respecte pas une contrainte physique.

En théorie, pour résoudre le problème de la planification de mouvement, il suffira donc de définir une position de départ , une position d'arrivée et de chercher un chemin dans l'espace des configurations qui ne passe que par des configurations acceptables.

### 1.1.3 Recherche de chemin dans l'espace des configurations

Il y a deux type de méthode pour rechercher un chemin dans l'espace des configurations, le premier est un type de méthodes exactes, qui calcule la représentation de l'environnement dans l'espace des configuration, et qui ensuite cherche un chemin. Mais en général, ce calcul est très couteux, on va donc parfois préférer d'autres méthodes qui consistent par exemple à s'appuyer sur un détecteur de collision, qui permet de détecter si une position est « interdite » ou non ; on tire alors aléatoirement des positions que l'on relie entre elles de sorte à obtenir un arbre (RRT), ou un graphe (PRM) : c'est ce que l'on appelle des méthodes par échantillonnage. On présente dans la suite les différents types de méthodes.

**Méthode exacte** Il existe des méthodes de planification de mouvement dites *complètes*, à savoir que quand elles rendent un résultat alors il est juste, et que quand il n'y a pas de solution, elles le signalent. Dans cette partie nous allons décrire une de ces méthodes ; pour simplifier on se placera dans un espace des configurations à deux dimensions, on considèrera que les obstacles sont des polygones et que le robot est un point.

Une méthode consiste à trianguler l'espace (voir figure 2). On obtient alors une espace des configurations libres sous forme de cellules liées entre elles par des arêtes qu'elles ont en commun. On peut alors en extraire une carte topologique

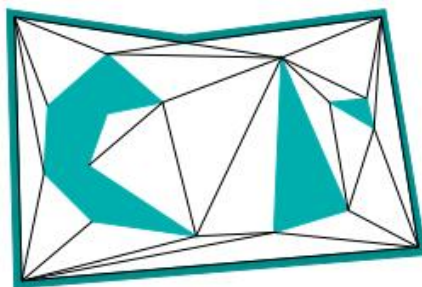


FIG. 2 – Exemple de triangulation d'un espace des configurations en deux dimensions

en reliant les centres des cellules voisines aux centres de leur arête commune (voir figure 3). Il y a alors équivalence entre existence d'un chemin acceptable (c'est à dire ne rentrant en collision avec aucun obstacle) entre deux points et

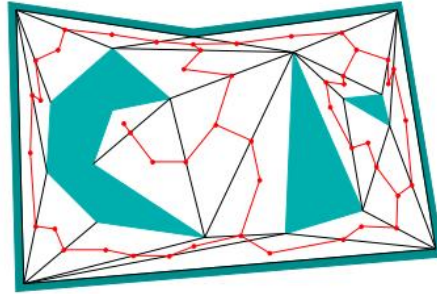


FIG. 3 – Carte topologique obtenue à partir de la figure 2

existence d'un chemin dans la carte topologique, qui est un problème classique de recherche de chemin dans un graphe.

Cette méthode nécessite donc une triangulation, d'un coût au moins linéaire par rapport à la complexité de la scène (en nombre de points), ce qui la rend assez peu efficace si l'environnement est dynamique et qu'il faut recalculer une triangulation trop souvent. On préférera donc des méthodes plus rapides, telles que par exemples les méthodes à échantillonnage probabiliste.

**Méthode approchée** Le problème de planification de mouvement peut également être résolu par une méthode approchée ; ces méthodes consistent à représenter l'espace des configurations de manière approchée par un quadrillage de cellules (Exemple : par des hypercubes, des hypersphères, . . . : voir figure 4). En étiquettant une cellule libre si l'ensemble de la cellule est dans l'espace libre, et non-libre si elle intersecte avec l'espace des configurations interdites : c'est donc une approximation de l'espace des configurations.

Cette méthode ne nécessite pas de déterminer exactement l'ensemble des configurations interdites, elle nécessite juste d'avoir un opérateur permettant de déterminer l'intersection d'une cellule simple avec cet ensemble (par exemple, un détecteur de collision). Elle nécessite cependant une représentation en autant de dimension que l'espace des configurations qui peut être parfois très couteuse (à stocker en mémoire, à calculer, . . .).

**Méthode par RRT** Les méthodes basées sur l'échantillonnage probabiliste nécessitent, pour leur correction, de pouvoir tirer aléatoirement des points dans l'espace des configurations selon une loi de probabilité. Il y a nécessité d'une formalisation mathématique de la notion de distance dans l'espace des configurations (notamment pour ce qui est des angles : utilisation des quaternions) qui est bien développée dans [LaV06].

Les méthodes par exploration d'arbre (*RRT*) sont des méthodes dites *single-query*, à savoir qu'il n'y a pas d'avantages dans un quelconque précalcul, mais que pour chaque donnée d'un point de départ, de sortie et d'un ensemble d'obstacles, il faudra tout recalculer.

Les méthodes *single-query* fonctionnent à peu près toutes sur le même schéma :

1. **initialisation d'un graphe**  $G = (V, E)$  avec au moins un sommet dans  $V$  ;

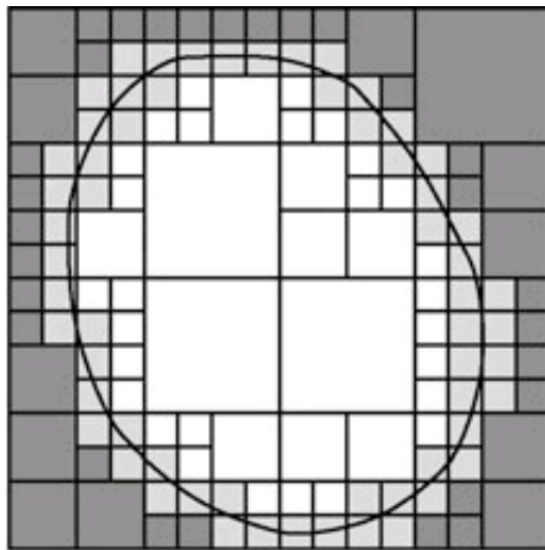


FIG. 4 – Décomposition de l'espace en hypercube

2. **sélection d'un sommet**  $q$  de  $V$  pour l'expansion ;
3. **extension locale du graphe** par tirage d'un nouveau  $q'$  suivant une loi de probabilité prédéfinie dans l'espace des configurations qui soit une configuration acceptable et par liage à l'état courant  $q$  (si c'est possible) ;
4. **ajout d'une arête dans le graphe** si l'étape précédente à fonctionné : ajouter  $(q, q')$  au graphe, et si  $q' \notin V$ , le rajouter ;
5. **test de fin** si  $G$  permet de résoudre le problème ;
6. **retour à l'étape 2** tant qu'une solution n'a pas été trouvée.

Dans le cas de l'exploration par arbre, on ne choisit pas à l'avance quel sommet sera le courant : on choisit le plus proche point du graphe de celui qui vient d'être tiré ; à savoir si c'est un sommet, on tente de rajouter une arête à ce sommet (voir figure 5) et si ce point tombe sur une arête, on crée un nouveau sommet à cet endroit et on rajoute l'arête correspondante (voir figure 6).

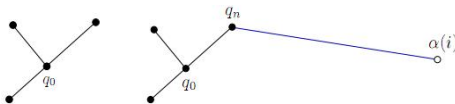


FIG. 5 – Ajout d'un sommet par un autre sommet

Un désavantage de cette méthode est qu'elle ne tient pas compte des simplifications qu'il pourrait y avoir dans un environnement statique : quoiqu'il advienne, l'algorithme recalculera tout l'arbre à chaque appel. Ce problème ne se pose pas dans le cas des *PRM*.



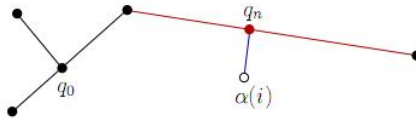


FIG. 6 – Ajout d'un sommet par une arête

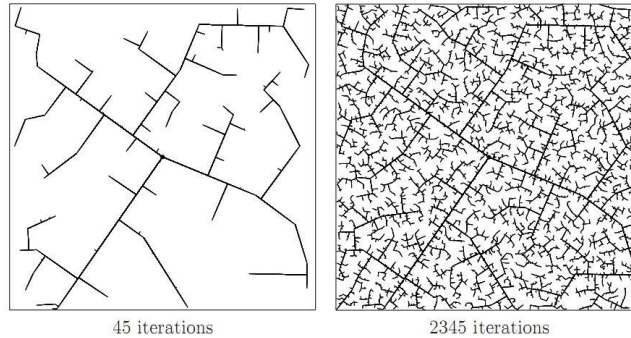


FIG. 7 – Exemple de construction d'un *RRT*

**Méthode par PRM** À la différence des méthodes par *RRT*, les méthodes de planification par *PRM* sont des méthodes dites *multiple-query*. Dans ce type de méthode, les calculs vont pouvoir servir pour plusieurs requêtes, une phase de précalcul peut donc être utile.

La méthode se décompose en deux grandes phases :

- Précalcul du graphe qui couvrira l'espace des configurations comme une carte topologique ;
- Requête d'un chemin entre  $q_i$  et  $q_f$ , deux configurations initiale et finale, qui sera satisfaite par un planificateur local pour relier ces configurations à des configurations de la carte topologique, et d'une recherche de chemin dans la carte topologique.

La phase de précalcul est assez similaire à l'algorithme des *RRT* :

1. **initialisation** du graphe  $G = (V, E)$  ;
2. **tirage aléatoire** suivant une loi de probabilité prédéfinie d'un sommet  $q$  dans une configuration acceptable ;
3. **reliage de  $q$**  à  $N$  plus proches voisins dans le graphe et test des arêtes créées (voir figure 8) ;
4. **mise à jour de  $E$**  en ajoutant les arêtes valides (ne passant pas par des obstacles) ;
5. **mise à jour de  $V$**  en ajoutant  $q$  ;
6. **retour à l'étape 2** tant que le graphe n'est pas assez couvrant.

La phase de requête se déroule comme telle :

1. On soumet  $q$  et  $q'$  deux configurations que l'on considère comme des nouveaux points à l'étape 2 de la phase de précalcul et qu'on tente de relier au graphe ;

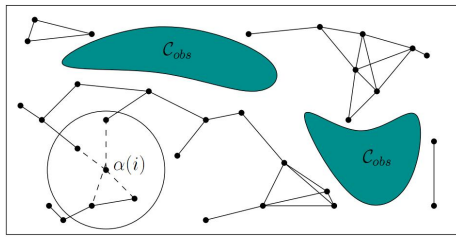


FIG. 8 – Ajout d'un sommet dans une *PRN*

2. On recherche un chemin dans le graphe reliant ces deux points ;
3. Si on en trouve un, alors on le renvoie ;
4. Si on n'en trouve pas, alors on ne peut pas conclure.

Le fait de ne pas toujours pouvoir conclure est parfois handicapant, dans les cas où l'on a besoin d'une réponse sûre, on préférera donc parfois la méthode exacte.

## 1.2 Positionnement de caméra

On appelle cinématographie virtuelle le fait d'utiliser des techniques cinématographiques dans des environnements virtuels. Comme dans le cinéma réel, une caméra possède sept degrés de liberté : sa position (trois degrés de liberté), ses angles de vue (trois degrés) et sa focale. Dans un environnement virtuel, elle peut être assimilée à un point, mais est en interaction avec la scène : on ne peut donc pas s'abstraire des obstacles pour filmer.

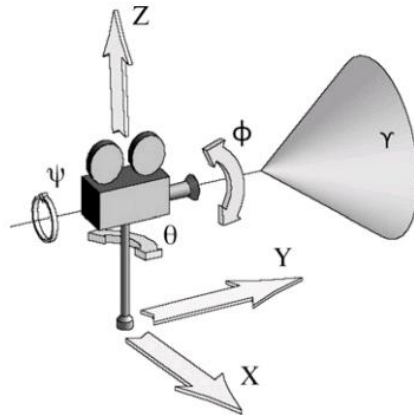


FIG. 9 – Les degrés de liberté d'une caméra

### 1.2.1 Cinématographie

Comme dans le cinéma classique, on considère qu'un *film* est un ensemble de *scènes*, et qu'une scène est un ensemble de *plans* ayant une continuité temporelle.

Un plan possède plusieurs caractéristiques, notamment :

- un angle de vue, lié à la focale de la caméra ;
- un cadre qui correspond aux limites de l'image ;
- des types de plan de la cinématographie classique (voir figure 10), que l'on prendra pour base afin de construire nos séquences.
- ...

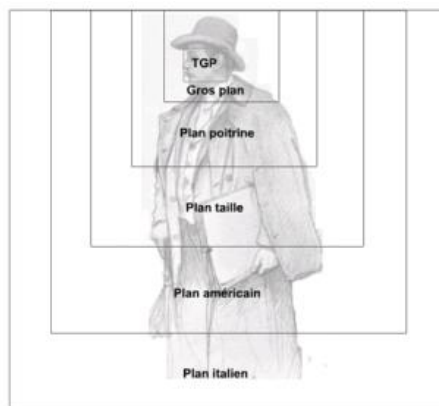


FIG. 10 – Différents types de plans cinématographiques

Toujours de la même manière qu'en cinématographie classique, les plans nécessitent entre eux des transitions. Ces transitions possèdent des contraintes que l'on va devoir prendre en compte (durée d'un plan, cohérence entre deux plans, ...).

Il existe plusieurs styles de tournage suivant les réalisateurs et suivant les actions que l'on a à filmer ; on tentera de formaliser ces types de style pour pouvoir jouer sur la manière de planifier nos séquences.

### 1.2.2 Règles cinématographiques

Dans la cinématographie classique, il existe des règles de composition d'une scène, des règles sur les raccords entre les scènes ainsi que sur les mouvements de caméra. Ces règles sont décrites dans plusieurs ouvrages tels que [War03] ou [Ari93]. La qualité d'un film dépend notamment du respect de ces règles. Aussi, nous utilisons certaines d'entre elles pour définir une fonction de qualité d'un film. Ci-dessous quelques exemples de règles cinématographiques pour les changements de plans :

- règle des  $180^\circ$  : pour permettre au spectateur de se repérer dans la scène, il est important de conserver la position relative des personnages (voir figure 11)
- règle des  $30^\circ$  : afin d'éviter au spectateur une sensation de « jump cut », c'est à dire une sensation de discontinuité dans l'image, les changements de plans de la caméra doivent s'effectuer avec un angle d'au moins  $30^\circ$ .

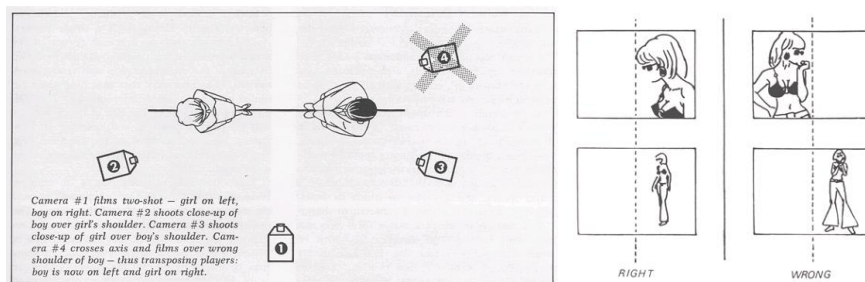


FIG. 11 – Règle des 180°

- règle de continuité dans le mouvement : toujours pour permettre au spectateur de se repérer dans la scène, il est important de conserver le sens du mouvement (voir figure 12).

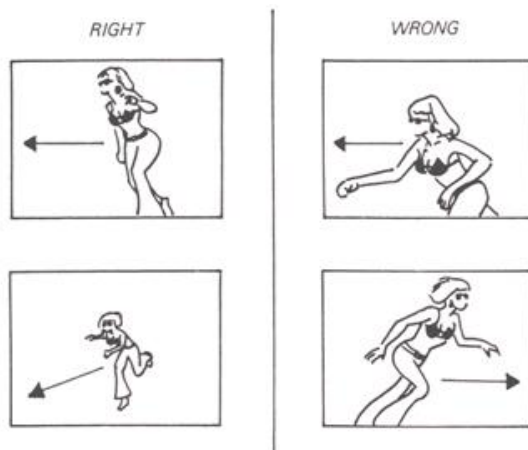


FIG. 12 – Règle de conservation du mouvement

Dans notre modèle, nous prendrons en comptes ces règles de changement de plan.

**Problématique** Dans le cadre de suivi d'objets virtuels, le problème de planification de position de caméra peut être considéré comme une spécialisation du problème de planification de mouvement, dans lequel la visibilité, le cadrage et d'autres paramètres cinématographiques viennent s'ajouter comme contraintes. Notre problématique sera donc de générer un ensemble de plans cinématographiques ainsi que les transitions les articulant tout en respectant les contraintes induites par les plans comme par les transitions.

### 1.2.3 Planification de caméra

Le problème de la planification de caméra étant une spécialisation du problème de planification de mouvement, il existe des méthodes assez similaires à celles-ci

**Exemple d'approche exacte** Dans son article [GAG04], Andújar propose de représenter l'espace à un plus haut niveau, d'y chercher un chemin et de le retranscrire ensuite dans l'espace réel. On se place dans le cas où on l'on connaît entièrement l'environnement dans lequel se trouvent des points d'intérêt que l'on souhaite filmer lors du passage de la caméra. Pour cela : on décompose l'espace



FIG. 13 – Calcul des distances des voxels à la géométrie et décomposition en cellules

en cellules (dans lesquelles figurent des points d'intérêt), en s'appuyant sur les distances par rapport aux obstacles (voir figure 13).

On crée ensuite une carte topologique sur cette décomposition en cellules (voir figure 14). La planification de mouvement de caméra devient alors une recherche de meilleur chemin dans un graphe pondéré.

*Remarque : les sommets et les transitions du graphe sont pondérés en fonction des points d'intérêt que l'on a dans les cellules et les transitions.*

On calcule donc un chemin (presque) optimal permettant de voir le plus de points d'intérêts possible en minimisant le nombre de visites de cellules « vides », la longueur du chemin et le nombre de revisite.

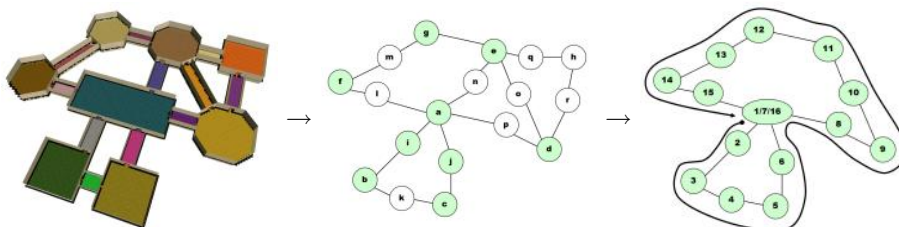


FIG. 14 – Calcul d'une carte topologique et d'un chemin dans cette carte

Enfin, une fois ce chemin calculé, on le retranscrit dans l'environnement réel (voir figure 15) en calculant, pour chaque cellule, le meilleur chemin pour la traverser.

Ainsi, pour chaque cellule, on détermine d'abord les meilleurs points d'entrée et de sortie vers les cellules précédente et suivante dans le chemin (suivant un critère tel que la visibilité qu'ils offrent sur les points d'intérêt), puis un chemin les reliant.

Pour terminer, les trajectoires sont lissées et la vitesse de la caméra est ajustée en fonction des points d'intérêt que l'on souhaite filmer, de sorte à avoir un rendu plus agréable à regarder.

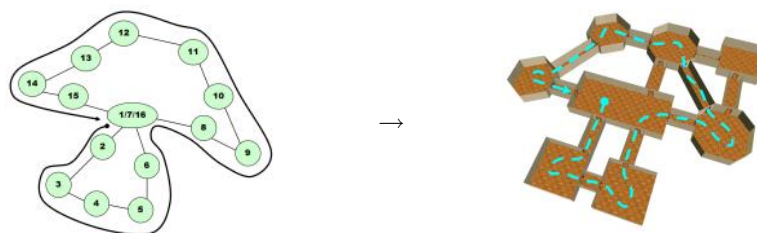


FIG. 15 – Retranscription du chemin topologique dans l'environnement réel

**Exemple d'approche par échantillonnage** Dans son article [NO04], Overmars présente une approche probabiliste pour résoudre le problème de la planification de mouvement de caméra. Il propose de cartographier l'espace grâce à des *PRM*. On se place dans le cas où un utilisateur a sa caméra virtuelle dans une configuration de départ et qu'il souhaite la faire arriver dans une autre en ayant une continuité dans le déplacement.

*Remarque : seule la position de la caméra sera déterminée grâce à ces cartes ; le cadrage, quant à lui, sera calculé postérieurement.*

On considère donc qu'on a précalculé une carte *PRM* de notre environnement (exemple : voir figure 16).

*Remarque : le fait que la carte soit celle d'une caméra a été pris en compte, les positions acceptables sont donc assez loin des obstacles pour pouvoir filmer.*

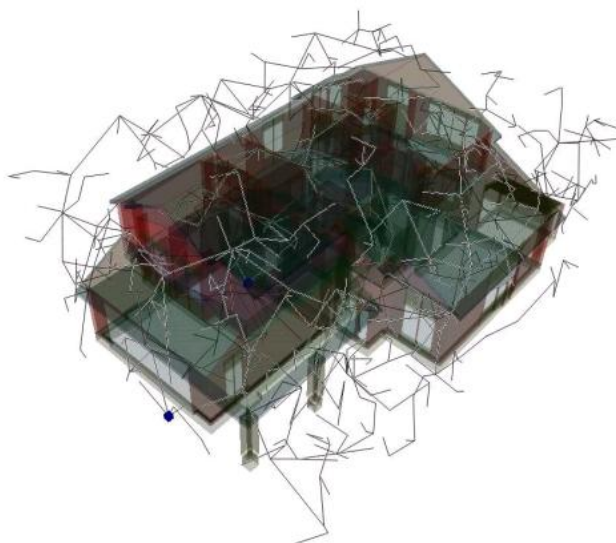


FIG. 16 – Exemple de *PRM*

Quand l'utilisateur décide de la nouvelle position qu'il souhaite pour la caméra, la première étape consiste à trouver un chemin dans la carte permettant de relier sa position initiale à la position désirée. Néanmoins, contrairement à la planification de mouvement, on ne recherche pas nécessairement le plus court chemin ou le plus rapide. On le choisira plus selon des critères tels que : par

exemple, lorsque cela ne le rallonge pas trop, prendre des virages larges plutôt que serrés (cela permet un meilleur rendu cinématographique).

Une fois qu'on a un chemin sans collision, on lisse ses angles (correspondant aux noeuds du graphe engendré par la *PRM*) en les transformant par des arcs de cercle (voir figure 17) ; si le lissage engendre des collisions, on diminue le rayon de l'arc de cercle.

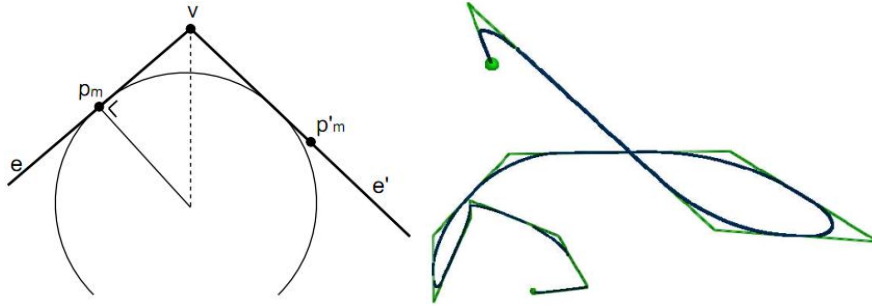


FIG. 17 – Lissage des angles du chemin par arcs de cercle

On définit ensuite une fonction indiquant la vitesse qu'aura la caméra le long du chemin lissé ; cette vitesse dépend de la courbure du chemin, a une borne supérieure, nécessite d'être continue et doit tenir compte que l'accélération est également bornée.

*Remarque : l'article propose une méthode pour optimiser un peu plus le déplacement de la caméra, mais on s'intéresse plus ici à l'orientation de la caméra.*

On peut considérer que l'orientation de la caméra par rapport au temps est une fonction de  $\mathbb{R} \rightarrow [0, 2\pi]^3$  ( l'espace d'arrivée correspond aux mesures des angles de la caméra). Cette orientation, pour permettre un rendu aussi agréable que possible doit vérifier certaines propriétés , comme par exemple la continuité au cours du temps, que l'image de l'horizon doit rester aussi droit que possible (pour éviter le mal de mer) ou que la caméra doit anticiper les virages (c'est à dire filmer non pas suivant le mouvement, mais suivant l'endroit où elle va se trouver juste après).

Le fait de garder l'horizon droit fixe déjà un paramètre, et connaissant déjà le chemin que va parcourir la caméra, on peut également calculer l'angle de vue suivant l'axe vertical (voir figure 18).

Reste alors le dernier paramètre pour l'orientation : l'angle longitudinal, qui est en général tel que la caméra soit parallèle au sol ; dans certaines configurations du chemin, un phénomène appelé « twist » peut arriver, mais l'article explique comment s'en abstraire.

## Conclusion de l'état de l'art

Dans ce rapport, nous avons présenté des méthodes permettant de résoudre les problèmes de planification de mouvement et de positionnement de caméra. Étant donné la mauvaise qualité de certains plans lorsque la caméra est asservie aux acteurs, nous nous sommes intéressés à la planification simultanée des acteurs et de la caméra les filmant. Nous nous intéresserons plus particulièrement à la planification de deux acteurs et d'une caméra, pour l'instant nous

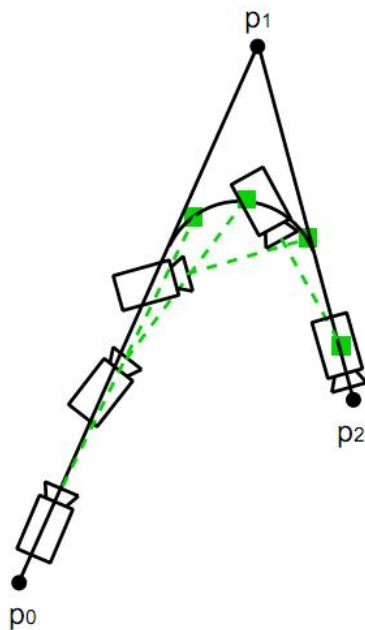


FIG. 18 – Planification de l’angle de rotation vertical

avons retenu l’utilisation d’un triptyque représentant ce trio, et nous tenterons d’appliquer des algorithmes de planification de mouvement sur cette entité.

Plusieurs méthodes résolvent le problème de la planification de mouvement par des méthodes exactes, cependant ces méthodes sont généralement coûteuses et nous préférons donc nous axer sur des méthodes par échantillonnage probabiliste. Le choix d’utiliser des arbres (RRT) ou des graphes (PRM) n’est pas encore défini.

Pour ce qui est du positionnement de caméra, les méthodes existantes asservissant cette tâche aux objets, on ne pourra les utiliser telles quelles.

Nous ferons en sorte que les acteurs aient une trajectoire continue reliant une position de départ à une position d’arrivée et que les plans tournés par la caméra optimisent la fonction de coût que nous aurons définie.

Ainsi, étant donné que l’on va placer les acteurs et la caméra simultanément, on est en droit de penser que l’on obtiendra une meilleure qualité cinématographique de film que lorsque la caméra est asservie aux acteurs.

**Travail à venir** Dans un premier temps, on s’autorisera à ce que la planification du mouvement ne s’effectue pas en temps réel, mais cela reste néanmoins un objectif non négligeable. Un autre objectif secondaire est l’interactivité : on souhaiterait au final que le système puisse gérer des événements non contrôlés, commandés par exemple par un utilisateur externe.



## 2 Modèle

Étant donné des acteurs et un environnement dans lequel ils peuvent circuler, notre objectif est de filmer ces acteurs d'un point de départ à un point d'arrivée dans cet environnement. L'approche classique consiste à planifier premièrement la trajectoire des acteurs puis celle de la caméra. Le problème de cette approche est qu'il est possible, une fois la trajectoire des acteurs calculée, de ne pas trouver de position possible pour la caméra donnant un point de vue de qualité acceptable. Ainsi, afin d'éviter ce genre de situation, nous proposons de planifier simultanément les déplacements des acteurs et de la caméra, en privilégiant la qualité globale du film.

Nous nous sommes donc ramenés à un problème de planification de mouvement dans un espace prenant en compte la configuration de la caméra en plus de celle des acteurs. Comme dans tout problème de planification, notre objectif est donc de trouver un chemin reliant deux configurations du système en restant dans l'espace libre. Étant donné qu'il faut prendre en compte la qualité du film, la navigation dans l'espace libre ne sera pas sans contraintes. Nous différencions donc deux types de contraintes : les contraintes dites *fortes* qui correspondent aux contraintes physiques de l'environnement (collision), et les contraintes dites *faibles* qui correspondent aux contraintes de qualité cinématographique.

Afin de simplifier le problème, nous nous sommes restreints à la planification de deux acteurs filmés par une caméra. Dans un premier temps, nous exposerons quels ont été nos choix pour paramétrer notre système, puis nous décrirons quelle méthode nous avons utilisée pour représenter l'espace dans lequel s'effectuera la planification et nous finirons par exposer notre processus de planification à travers cet espace.

### 2.1 Hypothèses

Le problème général de la planification d'acteurs et de caméra est complexe : combien d'acteurs sont présents dans la scène ? Quels sont les mouvements des acteurs autorisés ? Quels sont les mouvements de caméra autorisés ? L'environnement est-il statique ou dynamique ?...

Nous allons considérer certaines hypothèses simplificatrices qui nous permettront de résoudre ce problème dans des cas particuliers. Nous souhaitons implémenter notre modèle et le valider sur un environnement-test (voir section **Résultats**). Nous ne visons pas de performances calculatoire permettant d'avoir un rendu en temps réel, nous recherchons principalement à avoir un film de bonne qualité.

- Pour simplifier ce problème, nous prendrons donc les hypothèses suivantes :
- l'environnement est représentable par un ensemble fini de triangles,
  - l'environnement est entièrement statique,
  - seul deux acteurs évolueront dans cet environnement.

### 2.2 Vue générale du problème

La résolution d'un problème de planification de mouvement est généralement divisée en plusieurs étapes :

1. définition et identification de l'espace des configurations,
2. choix d'une représentation de cet espace,

3. recherche et parcours de l'espace des configurations.

Dans notre cas nous planifions simultanément le déplacement de deux acteurs et d'une caméra les filmant. Aussi, dans la mesure où notre procédure est similaire à la résolution d'un problème de planification de mouvement, elle se déroulera en plusieurs étapes :

1. Identification de « plans cinématographiques de référence » et définition de la qualité d'un plan,
2. Génération d'un modèle adapté au calcul de la qualité d'un film et basé « plan de référence »,
3. Représentation de l'espace des configurations associé au modèle,
4. Parcours de l'espace des configurations et génération d'une trajectoire de « bonne qualité ».

Dans la suite, nous détaillerons ces étapes une à une, les choix que nous avons fait dans chacune d'entre elles ainsi que les raisons qui nous ont poussés à faire ces choix.

## 2.3 Modélisation du problème

La qualité d'un film repose sur plusieurs paramètres, notamment la qualité des plans cinématographiques (voir figure 20) qui le composent ; cette qualité peut par exemple se mesurer à l'aide d'une distance à ces plans.

Aussi, étant donné que la qualité d'un plan est difficilement exprimable à partir de la configuration des acteurs et de la caméra, nous proposons de n'autoriser qu'un sous-ensemble de plans cinématographiques. Dans la mesure où un style cinématographique est notamment décrit par l'ensemble des plans utilisés lors de la réalisation du film, nous estimons qu'ils seront fournis par l'utilisateur comme entrée du problème, en plus d'un environnement et d'un point de départ et d'arrivée dans l'environnement considéré.

Notre problème se résume donc, étant donné un environnement, un point de départ et d'arrivée et une liste de plans de référence cinématographiques, à déterminer les trajectoires de deux acteurs et d'une caméra partant du point de départ et joignant l'arrivée sans discontinuité de position pour les acteurs et en ne filmant que par les plans de référence donnés en entrée.

### 2.3.1 Formalisation d'un plan cinématographique de référence pour deux acteurs

Dans notre modèle, nous définissons un plan cinématographique de référence  $\mathcal{P}$  (voir figure 19) comme un tuple  $(r, \theta_c, h, f, \vec{o}_c, \theta_d, \theta_1, \theta_2)$  où, pour une position des acteurs  $P_1$  et  $P_2$  fixée,

- $r$  est le rapport des distances (caméra-milieu des acteurs) et distance entre acteur :  $d_c/d_a$ ,
- $\theta_c$  est l'angle entre la ligne d'acteur et l'axe (milieu des acteurs  $\rightarrow$  caméra),
- $h$  est la hauteur de la caméra,
- $f$  est la focale de la caméra,
- $\vec{o}_c$  est l'orientation de la caméra,
- $\theta_d$  est l'angle (préférentiel) de déplacement du couple d'acteurs,
- $\theta_1$  et  $\theta_2$  sont les orientations des deux acteurs.

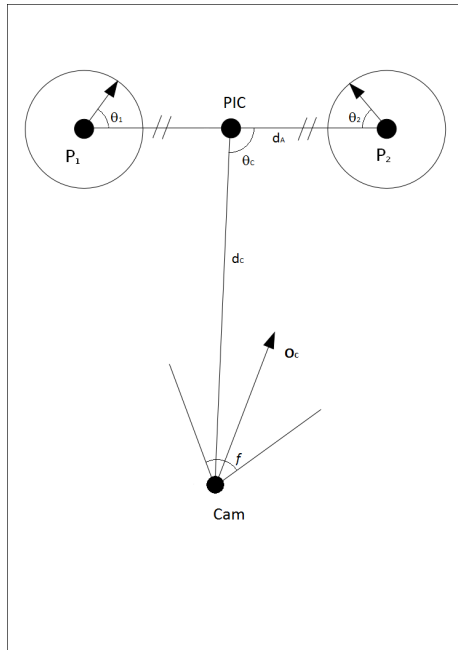


FIG. 19 – Paramètres d'un plan de référence pour deux acteurs : une fois les positions des acteurs  $P_1$  et  $P_2$  fixées, les paramètres du plan de référence fixent les autres paramètres du système, à savoir la configuration complète de la caméra (position, orientation, focale) et les orientations des acteurs.

Ainsi, étant donné une position des acteurs, un plan de référence détermine entièrement la configuration de la caméra, de manière à assurer une certaine disposition de ceux-ci à l'écran (voir figure 20). Ces plans peuvent être issus de la littérature, ou directement extraits de films existants. Dans notre modèle, tous les plans dont nous nous servirons seront donnés en entrée du problème.

### 2.3.2 Contraintes sur le modèle

À présent que nous avons décrit les degrés de liberté de notre système, nous allons le paramétrer de manière à facilement prendre en compte certaines contraintes. Les contraintes 'fortes' sont :

- la non-collision des acteurs et du plan de projection de la caméra avec l'environnement,
- la continuité du déplacement des acteurs du point de départ à l'arrivée,
- la réalisation du film uniquement par des plans de référence donnés en entrée.

Ces contraintes sont dites 'fortes' car elles déterminent des configurations interdites. Des contraintes dites 'faibles' sont également prises en compte :

- la qualité des plans cinématographiques : distance par rapport à un plan de référence,
- la qualité des transitions : réalisme des trajectoires des acteurs et qualité de la trajectoire de la caméra,

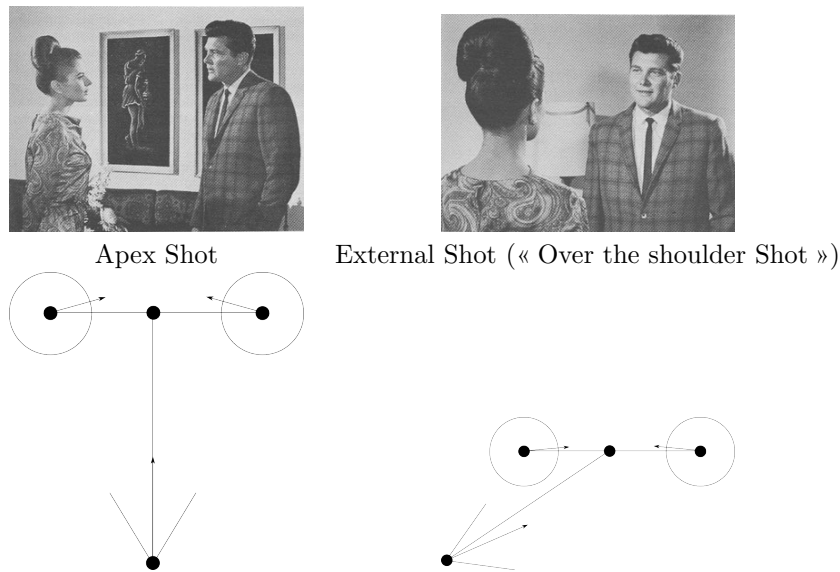


FIG. 20 – Exemple de deux plans cinématographiques classiques dans la cinématographie classique. Les schémas correspondent à la modélisation de ces plan de référence.

- la qualité des changements de plan : différentes règles cinématographiques issues de la littérature ([DH95] : règle des  $180^\circ$ , règle des  $30^\circ$ ).

Contrairement aux contraintes fortes, ces contraintes ne déterminent pas de configurations interdites, mais elles vont orienter les choix de réalisation du film.

### 2.3.3 Abstraction du modèle

Notre système est constitué de trois entités (dans la suite, nous l'appellerons *triptyque*) : deux acteurs et une caméra ( $Act1, Act2, Cam$ ). Ce triptyque possède sept degrés de liberté pour ce qui est de la caméra et plusieurs dizaines de degrés de liberté pour les acteurs (si elles sont toutes considérées). Donc si toutes les dimensions sont prises en compte, l'espace des configurations correspondant possèdera plusieurs dizaines de dimensions, et il sera calculatoirement difficile à représenter et à parcourir. Aussi, certaines simplifications s'imposent.

**Représentation des acteurs** Plutôt que de considérer les acteurs comme des humains virtuels, nous allons en considérer des cylindres englobants et orientés. Le fait de choisir des volumes englobants permet d'assurer que leur non-collision avec l'environnement induise la non collision des acteurs avec cet environnement et le fait de choisir des cylindres permet de réduire le coût des calculs de collisions avec l'environnement. Il restera alors, en considérant que ces cylindres touchent le sol, deux degrés de liberté pour la position dans l'environnement et un degré pour l'orientation  $Act1 = (P_1, \theta_{A_1})$  et  $Act2 = (P_2, \theta_{A_2})$  où  $P_i$  est la position de l'acteur  $i$  et  $\theta_{A_i}$  son orientation).

**Représentation de la caméra** La caméra quant à elle possède sept degrés de liberté  $Cam = (P_{Cam}, h_{Cam}, \vec{o}_{Cam}, f_{Cam})$  où :

- $P_{Cam}$  correspond à la position en abscisse et en ordonnée de la caméra,
- $h_{Cam}$  à sa hauteur,
- $\vec{o}_{Cam}$  à son orientation,
- $f_{Cam}$  à sa focale.

Étant donné que nous sommes dans un environnement virtuel, nous ne considérerons comme entité physique que son plan focal (qui ne doit pas être en collision avec l'environnement). Notre triptyque simplifié possède donc six degrés de liberté en ce qui concerne les acteurs et sept degrés de liberté pour la caméra.

**Notion de PIC** Les mouvements des deux acteurs ne sont pas indépendants, en effet pendant le déroulement du film, les acteurs sont 'proches' et vont (globalement) dans la même direction. Aussi, plutôt que de planifier indépendamment les mouvements des acteurs, nous allons planifier la trajectoire d'une base locale dans laquelle nous définirons les positions relatives des acteurs. Nous définissons un *point d'intérêt central* de la scène (PIC), lié à la position des acteurs (dans notre modèle, nous choisissons le milieu des positions  $P_1$  et  $P_2$ ), une distance  $d$  correspondant à l'écartement des acteurs et une orientation  $\gamma$  de la ligne d'acteur ( $P_2 \rightarrow P_1$ ) par rapport à l'axe des abscisses. Les positions des acteurs sont alors réexprimables en fonction de ces trois paramètres :

$$\begin{cases} P_1 = PIC + e_\gamma \cdot d/2 \text{ où } e_\gamma \text{ correspond à } (\cos\gamma, \sin\gamma) \\ P_2 = PIC - e_\gamma \cdot d/2 \end{cases},$$

et une fois les positions des acteurs et un plan de référence  $\mathcal{P} = (r_{\mathcal{P}}, (\theta_c)_{\mathcal{P}}, h_{\mathcal{P}}, f_{\mathcal{P}}, (\vec{o}_c)_{\mathcal{P}}, (\theta_d)_{\mathcal{P}}, (\theta_1)_{\mathcal{P}}, (\theta_2)_{\mathcal{P}})$  définis, la configuration entière du triptyque peut être calculé, en effet :

$$\begin{cases} \theta_{A_1} = (\theta_1)_{\mathcal{P}} \\ \theta_{A_2} = (\theta_2)_{\mathcal{P}} \\ P_{Cam} = PIC + r_{\mathcal{P}} \cdot M_{(\theta_c)_{\mathcal{P}}} \cdot d \cdot \vec{e}_\gamma \\ \text{où } M_{(\theta_c)_{\mathcal{P}}} \text{ correspond à une matrice de rotation d'angle } (\theta_c)_{\mathcal{P}} \\ h_{Cam} = h_{\mathcal{P}} \\ \vec{o}_{Cam} = (\vec{o}_c)_{\mathcal{P}} \\ f_{Cam} = f_{\mathcal{P}} \end{cases}$$

**Propriétés du modèle** Il y a donc équivalence de représentation entre  $(Act1, Act2, Cam)$  et  $(\mathcal{P}, PIC, \gamma, d)$ . Cette représentation permet de réguler séparément des contraintes fortes "filmage par plan de référence  $\mathcal{P}$ " et "non-collision des acteurs avec l'environnement". Elle permet également de facilement contrôler les changements de plan. En effet lors d'un changement de plan, les acteurs ne changent pas de position (même triplet  $(PIC, \gamma, d)$ ), seul le plan de référence change ( $\mathcal{P}_1 \rightarrow \mathcal{P}_2$ ); ainsi, pour tester si un changement de plan à partir d'une configuration  $(\mathcal{P}_1, PIC, \gamma, d)$  vers un plan  $\mathcal{P}_2$  est réalisable, il suffit de tester si la nouvelle position de la caméra n'est pas en collision avec l'environnement et si le changement de plan ( $\mathcal{P}_1 \rightarrow \mathcal{P}_2$ ) respecte les contraintes cinématographique [War03].

L'écartement des acteurs ainsi que l'orientation de la ligne d'acteurs, deux paramètres intervenant dans la qualité de plan pour le premier, et la qualité des transitions (faibles variations de l'orientation au cours d'un mouvement réaliste)

pour le deuxième, peuvent également être contrôlés séparément : cela permet par exemple de favoriser une composante de la qualité. Enfin, la continuité du mouvement des acteurs correspond à la continuité des trois paramètres  $(PIC, \gamma, d)$ . Ainsi en planifiant un chemin continu pour ces paramètres, la continuité du mouvement des acteurs est assurée.

Le fait d'utiliser cette représentation permet de délimiter l'espace des configurations, et ainsi d'éviter une exploration inutile de certaines zones (par exemple, les zones dans lesquelles  $d$  est trop grande).

Une fois les paramètres globaux (largeur et hauteur des cylindres,...) du triptyque fixés, une *configuration* du triptyque est alors un quadruplet  $(\mathcal{P}, PIC, d, \theta)$ , avec  $\mathcal{P}$  un plan de référence,  $PIC$  la position (en abscisse et en ordonnée) du point d'intérêt central de la scène,  $d$  la distance entre les acteurs et  $\theta$  l'orientation de la ligne des acteurs par rapport à l'axe des abscisses.

## 2.4 Représentation de l'espace des configurations

Une configuration du triptyque est un quadruplet  $(\mathcal{P}, PIC, d, \theta)$ . Afin de relier la configuration de départ à celle d'arrivée, il nous faut calculer des configurations intermédiaires de manière à obtenir un chemin continu entre le départ et l'arrivée.

### 2.4.1 Décomposition de l'espace

Certains degrés de liberté de notre triptyque correspondent à ceux calculés à partir des plans de référence (une fois les deux positions des acteurs  $P_1$  et  $P_2$  fixées, la configuration de la caméra et l'orientation des acteurs sont déterminés par le plan de référence), mais dans la mesure où ces plans sont fournis par l'utilisateur et qu'ils sont les seuls autorisés, les degrés de liberté correspondant ne font pas partie de l'espace de recherche. Aussi nous décomposons l'espace des configurations en autant de sous-espaces  $\mathcal{S}_1, \mathcal{S}_2, \dots$  qu'il y a de plans de référence  $\mathcal{P}_1, \mathcal{P}_2, \dots$  en entrée.

Chaque sous-espace  $\mathcal{S}_i$  est de dimension quatre, et correspond à l'ensemble des configurations dans lesquelles le plan de référence  $\mathcal{P}_i$  est fixé. Cette décomposition de l'espace préserve la facilité de changement de plan : une fois les sous-espaces calculés, pour savoir si un changement de plan  $\mathcal{P}_i \rightarrow \mathcal{P}_j$  peut s'effectuer à partir d'une configuration des acteurs  $C = (PIC, \gamma, d)$ , il suffit de tester l'appartenance de ce point aux deux espaces  $\mathcal{S}_i$  et  $\mathcal{S}_j$ . Reste maintenant à représenter ces sous-espaces de manière à pouvoir planifier le déplacement d'une configuration à une autre dans un espace à quatre dimensions où le plan de référence est fixé.

### 2.4.2 Exploration des sous-espaces à l'aide de roadmaps probabilistes

Plusieurs méthodes existent pour représenter ces sous-espaces. Nous avons choisi de les représenter à l'aide de PRMs. Leurs constructions s'effectuent en deux étapes :

1. tirage aléatoire de nouvelles configurations correspondant au sommet du graphe des PRMs,
2. liaison des configurations entre elles.

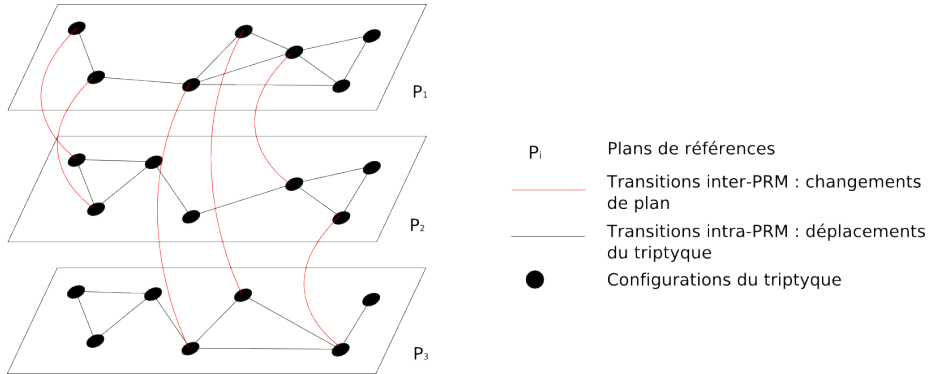


FIG. 21 – Espace des configurations : superposition des sous-espaces (représentés grâce à des PRM) associés à chaque plan de référence. Les transitions inter-PRM représentent les changements de entre plans de référence. Les transitions intra-PRM représentent les changements de configurations du triptyque à plan de référence fixé.

**Tirage** Dans notre approche, les configurations sont tirées de la manière suivante :

1. un point de l'environnement correspondant au PIC est choisi aléatoirement de manière non-biaisée (Exemple : dans un environnement rectangulaire, tirage uniforme en  $x$  (resp.  $y$ ) entre  $[x_{min}, x_{max}]$  (resp.  $[y_{min}, y_{max}]$ )),
2. une distance entre acteurs correspondant à  $d$  est tirée selon une loi de probabilité dont la distribution est représentée sur la figure 22. La distance entre acteurs joue un rôle dans la qualité d'un plan cinématographique : plus cette distance est proche d'une distance préférentielle  $d_{opt}$ , meilleure est la qualité.  $d$  est donc tirée entre une distance minimale  $d_{min}$  et une distance maximale  $d_{max}$  et de manière à privilégier la distance préférentielle  $d_{opt}$ .
3. une fois ces deux paramètres fixés, nous calculons l'ensemble  $\Gamma$  des orientations pour lesquelles le triptyque n'est pas en collision avec l'environnement (voir figure 23).

Une configuration est valide dans un plan de référence si le triptyque n'est pas en collision avec l'environnement pour au moins une orientation (les contraintes 'fortes' de collisions sont alors respectées).

**Liaison** Pour relier une nouvelle configuration ( $PIC_{new}, d_{new}, \Gamma_{new}$ ) au reste du graphe, les potentiels voisins ( $PIC_v, d_v, \Gamma_v$ ) doivent avoir une orientation  $\gamma \in \Gamma_{new} \cap \Gamma_v$  commune avec la nouvelle configuration pour laquelle avec une distance entre acteurs  $d \in \{d_{new}, d_v\}$ , le triptyque se translate de  $PIC_{new}$  vers  $PIC_v$  sans être en collision avec l'environnement. Ce qui donne formellement ( $PIC_{new}, d_{new}, \Gamma_{new}$ ) est liée à ( $PIC_v, d_v, \Gamma_v$ ) si et seulement si :  $\exists \gamma \in \Gamma_{new} \cap \Gamma_v, \exists d \in \{d_{new}, d_v\}$  tels que la translation du triptyque entre  $PIC_v$  et  $PIC_{new}$  avec les paramètres  $d$  et  $\gamma$  soit libre de collision.

Une liaison entre deux configurations correspond donc à un chemin continu (au sens des paramètres  $d, PIC, \theta$ ) les reliant.

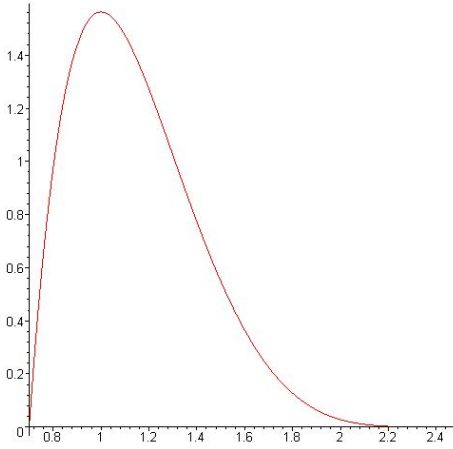


FIG. 22 – Distribution de la loi de probabilité selon laquelle la distance inter-acteur est choisie. Dans ce cas, la distance minimale  $d_{min}$  vaut 0.7 mètres, la distance maximale  $d_{max}$  : 2.5 mètres et la distance préférentielle  $d_{opt}$  : 1.0 mètre.

Une configuration de la PRM correspond alors à un plan de référence  $\mathcal{P}$ , une position pour le point d'intérêt central PIC, une distance entre acteurs  $d$  et l'ensemble des orientations possibles une fois les autres paramètres fixés. L'ajout de configurations s'arrête quand il y en a assez par PRM (nous nous fixons un seuil arbitraire. Exemple : 10 configurations par mètre carré).

### 2.4.3 Propriétés du modèle basé plans de référence

Le fait de calculer exhaustivement l'ensemble des orientations admissibles pour le triptyque permet de réduire d'une dimension le problème ; ce caractère particulier a été appliqué à l'orientation plutôt qu'à la position du PIC ou à la distance inter-acteurs car elle est dans un intervalle borné ( $[0, 2\pi]$ ) indépendamment de l'environnement (contrairement à la position) et qu'elle intervient grandement dans la planification (contrairement à la distance inter-acteurs).

Cette décomposition des dimensions peut être vue comme une superposition de PRMs : une par plan de référence (voir figure 21), dans lesquelles l'espace des configurations est à quatre dimensions. Étant donné qu'un changement de plan correspond à un saut d'un sous-espace à l'autre en fixant les paramètres (PIC,  $d$ ,  $\gamma$ ), nous tirons les mêmes points dans chaque sous-espace, à savoir le PIC et la distance  $d$ . Ainsi, les changements de plan possibles sont facilement identifiables : il suffit de vérifier que deux configurations ayant même PIC et même  $d$  possèdent une orientation en commun dans deux PRMs. Tirer les mêmes paramètres pour chaque PRM de chaque plan de référence permet donc de définir les points pour lesquels des changements de plan sont possibles.

Notre représentation permet ainsi de facilement contrôler la qualité des plans (par la pondération de sommets dans une PRM) et la qualité des changements de plan (par pondération des arêtes inter-PRM), et de les contrôler séparément : ce qui permet éventuellement de favoriser plus ou moins une qualité par rapport à une autre.



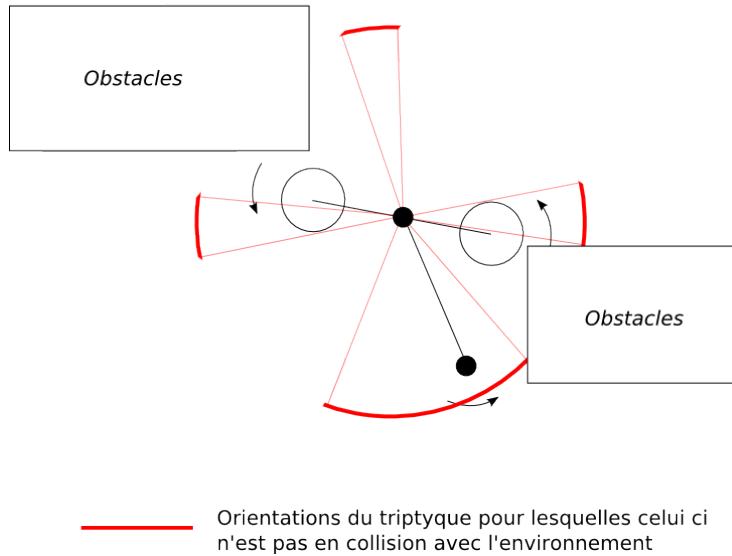


FIG. 23 – Calcul d’une configuration : une fois le point d’intérêt central PIC et la distance inter-acteur  $d$  fixés, les paramètres du plan de référence permettent de construire le tryptique à une rotation près. Toutes les orientations pour lesquelles le triptyque n’est pas en collision avec l’environnement (c’est à dire, ni les acteurs, ni le plan focal de la caméra) sont alors calculées.

Dans la suite, le terme ‘PRM’ sera utilisé pour parler de PRM associée à un plan de référence.

## 2.5 Calcul et lissage de la trajectoire

Dans cette section, nous allons exposer comment nous parcourons notre ensemble de PRMs afin de trouver une suite continue de configurations valides reliant le point de départ à celui d’arrivée, tout en maximisant la qualité globale du film obtenu.

### 2.5.1 Parcours des PRMs

Afin d’estimer la qualité d’un chemin, nous évaluons la qualité des plans cinématographique, la qualité des transitions ainsi que la qualité des changements de plan. Lors du parcours de l’ensemble des PRMs, nous sommes capables d’évaluer en chaque nœud de chaque PRM la qualité des plans, qui dépend :

- de la préférence du plan de référence,
- de la distance entre acteurs,
- de la qualité de l’arrière plan,
- de la visibilité des acteurs entre eux et depuis la caméra,
- ...

Dans notre modèle, nous considérons pour le moment seulement la visibilité des acteurs depuis la caméra ; cette visibilité est calculée de la manière suivante :

- $N$  rayons sont lancés aléatoirement depuis la caméra vers les têtes des acteurs,
- $n$  rayons intersectent l'environnement avant d'atteindre les cibles : nous appelons *visibilité* le rapport  $\frac{N-n}{N}$  et *non-visibilité* le rapport  $\frac{n}{N}$

Nous sommes aussi capables d'évaluer la qualité des changements de plans sur les arêtes entre PRMs, également selon des critères cinématographiques (règle des 30 degrés, règle des 180 degrés, règle de préservation du sens de déplacement, ...).

Ainsi, en associant à chaque sommet et à chaque arête inter-PRM une valeur (dans la suite nous appellerons respectivement ces fonctions de coût *QualityNode* et *QualityArc*), l'ensemble des PRMs peut être considéré comme un graphe pondéré. Par conséquent, en appliquant un algorithme de meilleur chemin (maximisant la qualité des plans et des changements), nous sommes capables de trouver quel chemin aura la meilleure valeur a priori, selon une fonction de coût basée uniquement sur la qualité des plans et des changements de plan.

---

**Algorithme 1** Dijkstra( $G, QualityArc, QualityNode, s_{init}$ )

---

**Initialisation :**

$S :=$  ensemble des sommets de  $G$   
*Valeur* := Tableau[Card( $S$ )]  
*Parent* := Tableau[Card( $S$ )]  
**pour** chaque sommet  $s \in G$  **faire**  
    *Valeur*[ $s$ ] :=  $\infty$

**fin pour**

*Valeur*[ $s_{init}$ ] := *Quality*( $s_{init}$ )

**Procédure principale :**

**tantque**  $S \neq \emptyset$  **faire**

$s_1 := \text{argmin}_{s \in S} (\textit{Valeur}[s])$

$S := S \setminus \{s_1\}$

**pour tout**  $s_2$  voisin de  $s_1$  dans  $G$  **faire**

**Mise à jour**

**si** *Valeur*[ $s_1$ ] + *QualityArc*( $s_1 \rightarrow s_2$ ) + *QualityNode*( $s_2$ ) < *Valeur*( $s_2$ )

**alors**

*Valeur*[ $s_2$ ] := *Valeur*[ $s_1$ ] + *QualityArc*( $s_1 \rightarrow s_2$ ) + *QualityNode*( $s_2$ )

*Parent*[ $s_2$ ] :=  $s_1$

**finsi**

**fin pour**

**fin tantque**

---

Nous avons choisi de parcourir le graphe par un algorithme de Dijkstra (voir Algorithme 1). Aussi, nos fonctions de qualité *QualityNode* appliquée aux sommets et *QualityArc* appliquée aux arêtes de la PRM doivent vérifier certaines propriétés afin de garantir la correction de l'algorithme :

- plus la qualité augmente plus les valeurs des fonctions diminuent car l'algorithme est un calcul de chemin de poids minimal,
- être positive de manière à vérifier que lorsque qu'un sommet est marqué, le chemin de poids minimal a été trouvé jusqu'à ce point.

Dans notre modèle, nous avons pris comme fonction de coût pour les sommets *QualityNode* la non-visibilité des acteurs depuis la caméra.

Il est important de remarquer qu'étant donné la construction de nos PRMs, nous sommes assurés d'une chose : il existe un chemin continu au sens de la position, de l'orientation et de la distance entre acteurs entre chaque configurations reliées deux à deux ; cependant, il n'existe pas nécessairement de chemin continu entre deux configurations ayant une de leur configurations voisines en commun (voir figure 24). Aussi, lors du parcours de graphe, avant de déterminer qu'un sommet est le parent d'un autre, nous testons si l'ensemble du chemin depuis  $s_{init}$  est valide (ce qui augmente la complexité de l'algorithme de dijkstra).

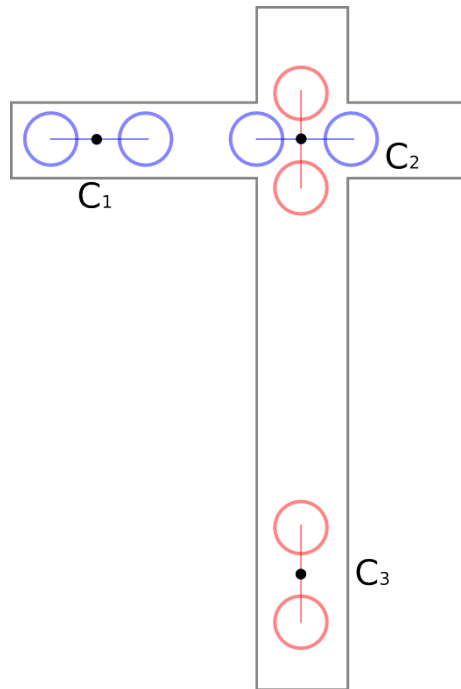


FIG. 24 – Exemple de cas où deux configurations sont reliées deux à deux mais pas globalement : il existe une orientation pour laquelle le triptyque peut se translater de  $C_1$  à  $C_2$ , et il existe une orientation pour laquelle le triptyque peut se translater de  $C_2$  à  $C_3$ , cependant il n'existe pas de chemin libre de collision permettant de rejoindre  $C_1$  à  $C_3$ .

À la sortie de l'algorithme, pour savoir s'il existe un chemin entre  $s_{init}$  et un sommet quelconque  $s$ , il suffit de tester que  $Valeur[s] \neq \infty$  ; il reste alors à parcourir le tableau *Parent* pour retrouver quel est le meilleur chemin valide a priori.

Cependant, même si le chemin trouvé aura la meilleure qualité du point de vue des changements de plan et des plans cinématographiques, il aura une très faible qualité du point de vue des transitions. En effet les chemins trouvés lors du parcours de PRMs sont souvent chaotiques (voir figure 25). Aussi, pour maximiser la qualité totale du film, il est important de « lisser » ces chemins.

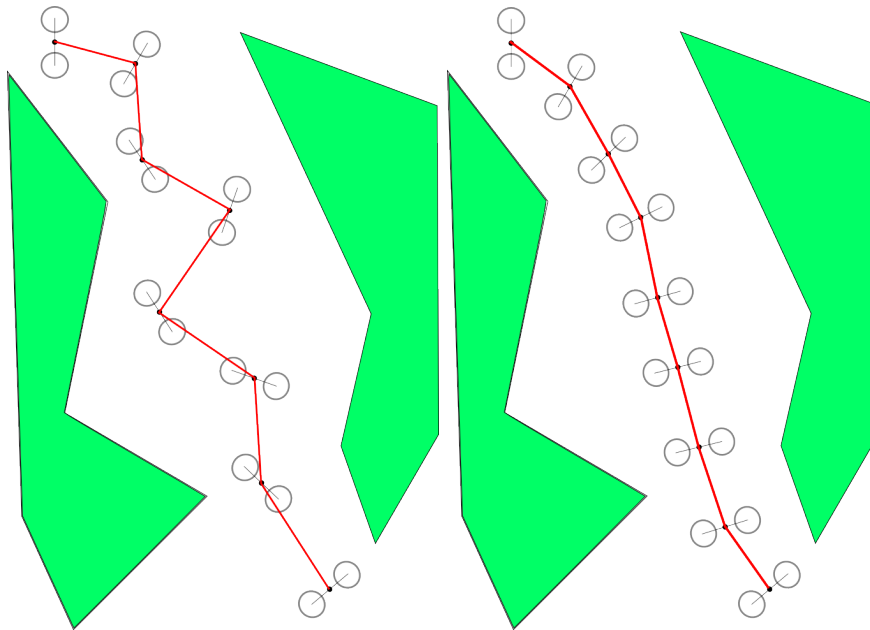


FIG. 25 – Exemple de chemin trouvé dans une PRM avant et après lissage

### 2.5.2 Lissage de trajectoire

Une suite continue de configurations reliant un point de départ à un point d'arrivée n'étant pas toujours cinématographiquement qualitative, nous procédons à un *lissage* de cette trajectoire afin d'en augmenter la qualité (par rapport à une fonction donnée). Ce procédé consiste par exemple à limiter les variations de vitesse et d'accélération induites par la trajectoire de sorte à la rendre physiquement réaliste.

Dans notre problème, cette étape contribue grandement à la maximisation de la qualité des transitions (déplacement du triptyque avec un plan de référence fixe) car la qualité d'une transition dépend du déplacement de la caméra ainsi que du déplacement des acteurs. Cette étape contribue également à la maximisation de la qualité des plans, en effet le lissage modifiant la trajectoire, il est possible de favoriser le passage par des endroits où la visibilité est bonne et de pénaliser ceux où la visibilité est réduite ; il est également possible de favoriser le passage par des endroits où l'arrière plan est intéressant (notion de points d'intérêt de l'environnement).

Dans notre modèle, nous n'avons pas encore explicité de fonction de coût pour les trajectoires, nous choisissons donc comme fonction de lissage "le plus court chemin à vue" (voir figure 26) qui correspond à un chemin « naturel » pour un déplacement. Nous avons en entrée de la fonction de lissage une trajectoire valide issue du parcours de graphe de la PRM (qui est une succession de sommets entre lequel le PIC se déplace en ligne droite), valide dans le sens où nous avons trouvé un chemin continu du point de départ à celui d'arrivée libre de collision avec l'environnement. Nous déterminons alors les configurations successives du triptyque de la manière suivante :

- la trajectoire est parcourue sommet par sommet, de la destination vers le

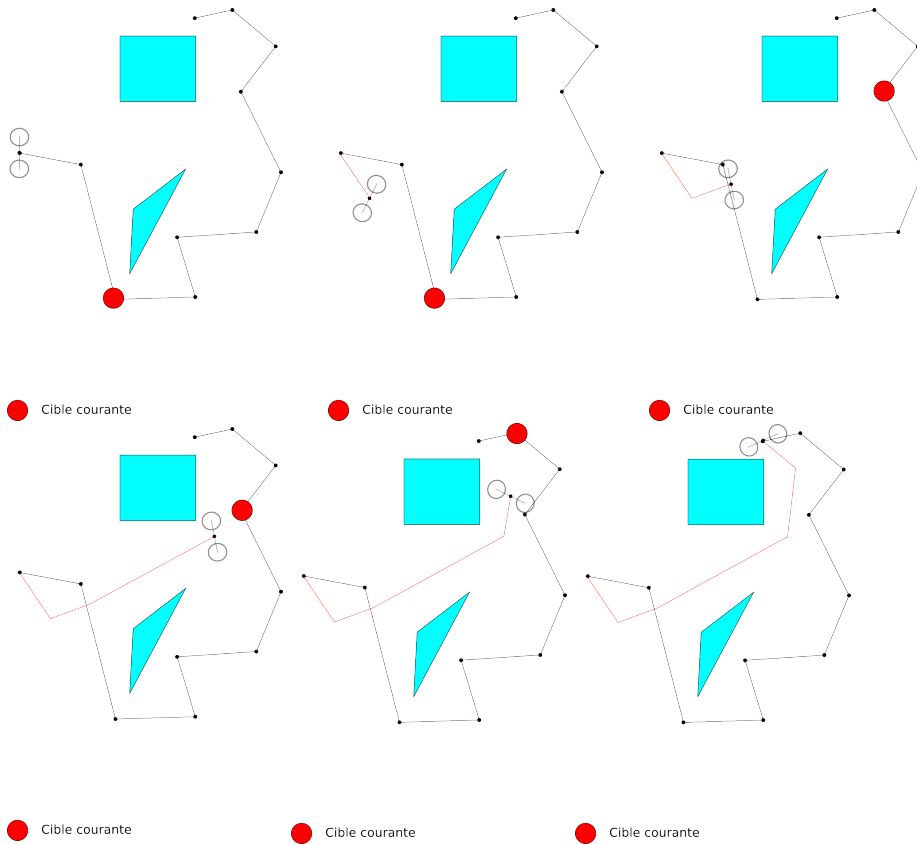


FIG. 26 – Procédure de lissage d’une trajectoire : la cible courante du triptyque est le sommet le plus loin sur la trajectoire initiale qui est accessible en ligne droite (en adaptant l’orientation et la distance inter-acteur) ; cette cible est recalculée à chaque pas du triptyque jusqu’à l’arrivée. La configuration du triptyque est alors calculée de manière à rejoindre sa cible en ligne droite en adaptant son orientation  $\gamma$  et sa distance inter-acteur  $d$ .

départ,

- si le triptyque "voit" un sommet, c’est à dire qu’il peut s’y déplacer en ligne droite en ajustant son orientation  $\gamma$  et sa distance entre acteurs  $d$ , ce sommet devient alors sa prochaine cible.
- une fois déterminée la prochaine cible  $(x_c, y_c)$  du triptyque, sa nouvelle configuration peut être calculée :
  - si la prochaine cible est à la même position et orientation que la configuration actuelle  $(\mathcal{P}, PIC_{ini}, \gamma_{ini}, d_{ini})$  mais avec un plan de référence différent  $\mathcal{P}'$ , la nouvelle configuration est alors  $(\mathcal{P}', PIC_{ini}, \gamma_{ini}, d_{ini})$ .
  - sinon, le plan de référence reste le même  $\mathcal{P}$ , alors :
    - le PIC se calcule en faisant le plus grand pas possible (majoré par une constante définie par l’utilisateur) jusqu’à la prochaine cible en suivant une ligne droite,
    - $\gamma$  se calcule en se rapprochant le plus possible de l’angle préférentiel de déplacement donné par le plan de référence.

- $d$  se calcule en faisant une interpolation linéaire entre la distance actuelle  $d_{ini}$  et la distance inter-acteurs de la cible.

Pour s'assurer que la trajectoire modifiée reste valide, le chemin de la nouvelle configuration à la nouvelle cible et celui de la nouvelle cible à l'arrivée sont également testés avant de valider la nouvelle cible.

Cette procédure étant la dernière étape dans la réalisation du film, il est important que la fonction de coût appliquée pour le lissage détermine, à partir d'une suite de configuration donnée, les meilleures trajectoires pour les acteurs et la caméra d'un point de vue qualité des plans et qualité des transitions.

**Problème** Dans l'idéal, afin de trouver le film de meilleur qualité possible, il faudrait considérer tous les chemins possibles dans le graphe de planification, les lisser et comparer leur qualité. Néanmoins, l'opération de lissage étant coûteuse, il n'est pas calculatoirement envisageable de lisser tous les chemins du graphe pondéré en un temps raisonnable. Il faut donc, en faisant certaines approximations, réduire le nombre de chemins à lisser.

### 2.5.3 Réduction de l'espace de recherche

Nous allons réduire l'ensemble des chemins à considérer grâce à des propriétés topologiques de l'environnement. Nous considérons des zones de passage *obligatoire* à savoir des zones de l'environnement par lesquelles il sera nécessaire de passer pour relier le départ à l'arrivée – et nous divisons considérer le problème global en deux plus petits : nous faisons donc une approximation qui sera que le meilleur chemin global est une composition d'un des meilleurs chemins du départ à un de ces points obligatoires et d'un des meilleurs chemins de ce point à l'arrivée.

Les zones de passage obligatoire peuvent être trouvées à partir d'une carte topologique de l'environnement, en appliquant par exemple un algorithme de coupe minimale qui rend un ensemble de zones parmi lesquelles au moins une sera visitée quelque soit le chemin reliant le départ à l'arrivée.

**Carte topologique de l'environnement** À l'aide d'un logiciel existant (TOPOPLAN [Lam09]), nous calculons une triangulation de l'environnement (voir section **Résultats**). Cette triangulation constitue déjà une carte topologique (où les sommets représentent les triangles et les arêtes, l'accessibilité entre ces triangles). Cependant, étant donné que cette triangulation est fine, les ensembles de zones de passage obligatoire sont nombreux. Aussi, afin de réduire le nombre d'éléments dans ces ensembles, nous fusionnons certains triangles entre eux de manière à avoir une représentation de l'environnement avec moins de zones (voir figure 27).

Les zones (initialement les triangles) sont donc fusionnées de manière à obtenir des zones plus grande et une carte topologique avec moins de sommets. Deux zones sont fusionnées si elle partagent une arête commune et si la convexité de la zone est préservée par la fusion. Les fusions s'effectuent d'abord entre les zones partageant les plus arêtes de plus grande dimension. Le processus s'arrête quand plus aucune fusion n'est possible sans perdre la propriété de convexité. Reste alors un ensemble de zones convexes et connectées entre elles, assimilable à un graphe appelé graphe topologique.

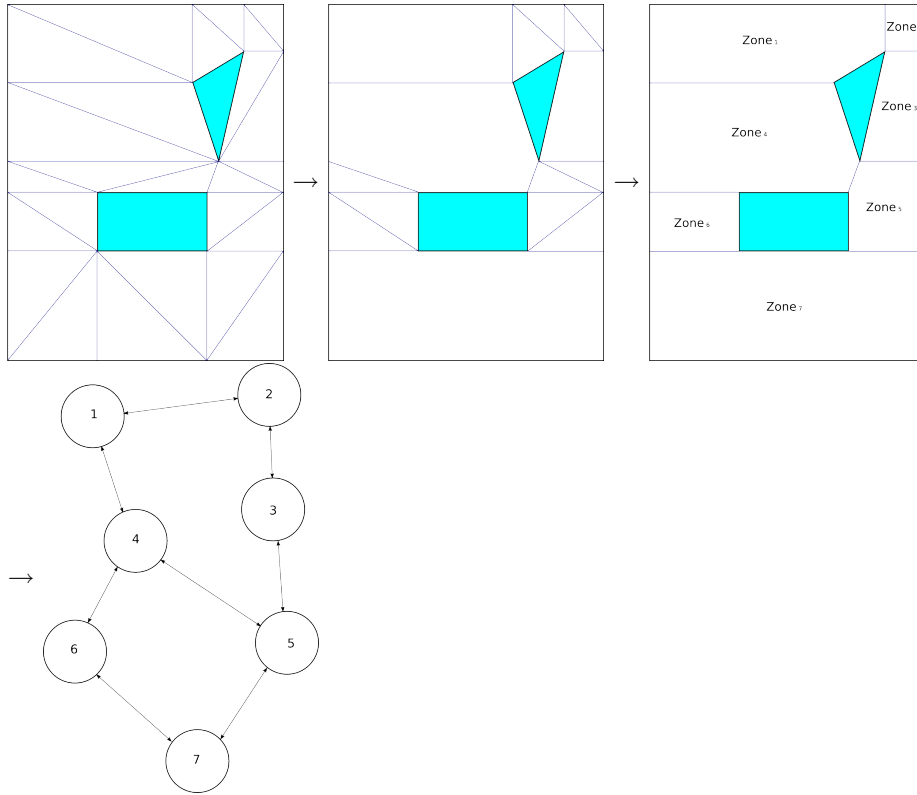


FIG. 27 – Construction du graphe topologique. Partant d’une triangulation de l’espace navigable, les triangles sont fusionnés en zones convexes (fusion préférentielle par les arêtes les plus longues), elles-mêmes fusionnées en zones de plus grandes dimensions. Une fois toutes les fusions possible effectuées, l’environnement est assimilable à un graphe où les sommets représentent les zones convexes et les arêtes, les accessibilités entre chaque zone.

**Points de passage obligatoires** Une fois le graphe topologique de l’environnement calculé, nous plaçons les point de départ et d’arrivée du tritpyque dans la carte topologique (voir figure 28), ce qui induit un état initial et un état final dans le graphe topologique.

Ensuite, en appliquant un algorithme de coupe minimale ([PS98]), nous exhibons un ensemble de sommets (dans notre exemple, l’ensemble  $\{4, 5\}$  convient) qui, s’ils sont supprimés, rendent le graphe non connexe. Cela induit donc la propriété suivante : quel que soit le chemin choisi du départ pour rejoindre l’arrivée, au moins une de ces zones de la coupe minimale sera visitée. Nous avons donc, en notant  $\mathcal{C}(x, y)$  l’ensemble des chemins possibles du point  $x$  au point  $y$ , et  $E_{MinCut}$  un ensemble de zone correspondant à une coupe minimale (nous simplifions les notation départ ( $\mathcal{D}$ ) et arrivée ( $\mathcal{A}$ )), :

$$\mathcal{C}(\mathcal{D}, \mathcal{A}) = \cup_{c \in E_{MinCut}} \{t \cdot t' | t \in \mathcal{C}(\mathcal{D}, c) \text{ et } t' \in \mathcal{C}(c, \mathcal{A})\}.$$

Ainsi, pour toute fonction  $Q$ , nous avons la propriété :

$$\max\{Q(x) | x \in \mathcal{C}(\mathcal{D}, \mathcal{A})\} = \max_{c \in E_{MinCut}} \max\{Q(x \cdot y) | x \in \mathcal{C}(\mathcal{D}, c) \text{ et } y \in \mathcal{C}(c, \mathcal{A})\}.$$

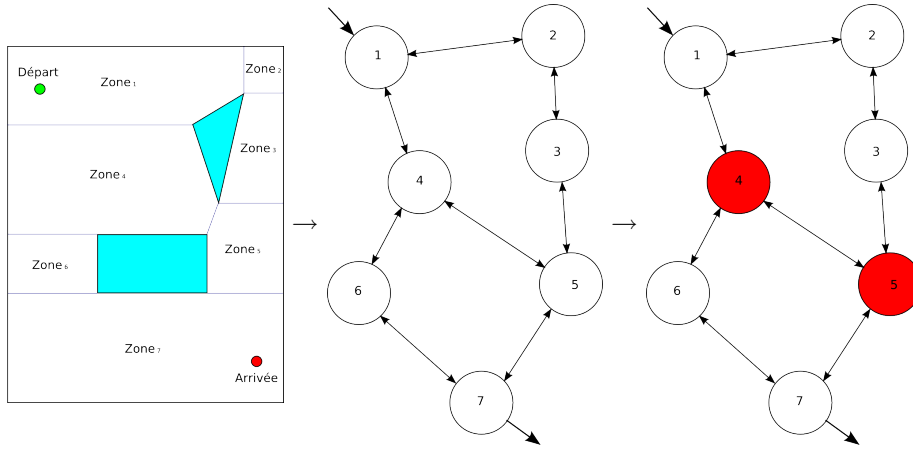


FIG. 28 – Une fois le point de départ (resp. d’arrivée) fixé, le sommet du graphe topologique correspondant à la zone considérée est défini comme sommet initial (resp. terminal). Grâce à un algorithme de coupe minimale, nous obtenons un ensemble de sommets (correspondant à des zones) qui, s’ils sont supprimés du graphe, rendent celui-ci non connexe : l’arrivée n’est plus accessible depuis le départ.

Pour restreindre l’ensemble des chemins à tester, nous faisons donc une approximation : étant donné une zone  $c$ , nous estimons que le meilleur chemin partant du départ  $\mathcal{D}$ , rejoignant l’arrivée  $\mathcal{A}$  et passant par  $c$  est une composition d’un des  $k$  meilleurs chemins de  $\mathcal{D}$  à  $c$  et d’un des  $k$  meilleurs chemins de  $c$  à  $\mathcal{A}$ , ce qui revient à dire formellement, en notant  $MAXC_k(x, y)$  l’ensemble des  $k$  chemins de meilleure qualité de  $x$  à  $y$  :

$$\max\{Q(x \cdot y) \mid x \in C(\mathcal{D}, c) \text{ et } y \in C(c, \mathcal{A})\} = \max\{Q(x \cdot y) \mid x \in MAXC_k(\mathcal{D}, c) \text{ et } y \in MAXC_k(c, \mathcal{A})\}.$$

Ce procédé nous permet ainsi de décomposer le problème initial en  $2 \cdot Card(E_{MinCut})$  sous-problèmes de planification où  $Card(E_{MinCut})$  est le nombre d’éléments dans la coupe minimale du graphe topologique. De plus chacun des sous-problèmes étant lui même un problème de planification, nous pouvons donc appliquer récursivement cette procédure de décomposition.

Une fois deux sous-problèmes entre  $d$  et  $a$  passant par une zone  $c$  résolus (i.e.  $MAXC_k(d, c)$  et  $MAXC_k(c, a)$  calculés), nous lisons les différentes combinaisons de chemins  $x \cdot y$  avec  $x \in MAXC_k(d, c)$  et  $y \in MAXC_k(c, a)$  et nous en élisons les  $k$  meilleurs chemins lissés : cela nous permet alors d’approximer  $MAX_k(d, a)$ . De cette manière, nous remontons la chaîne des sous-problèmes jusqu’à résoudre le problème principal  $MAX_1(\mathcal{D}, \mathcal{A})$ . Nous restons cependant bien conscient que, à chaque décompositions, il y a une possibilité de perte du chemin globalement optimal : nous obtiendrons un chemin localement de qualité optimale mais pas nécessairement globalement.



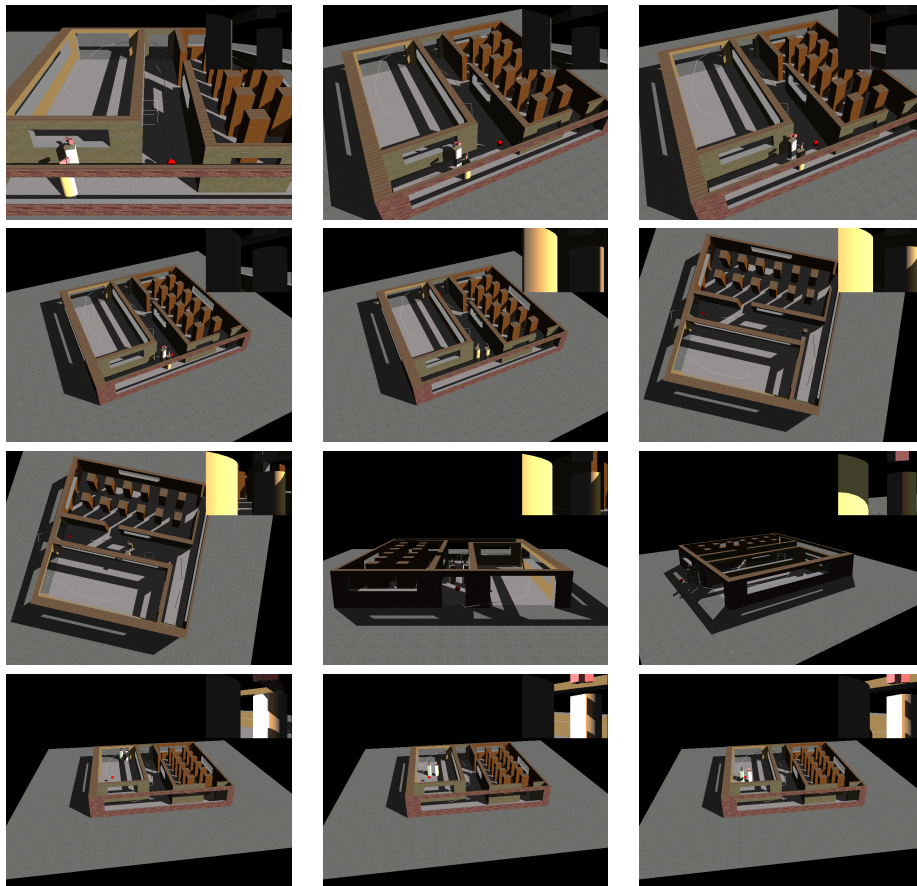


FIG. 29 – Exemple de calcul de trajectoire dans notre environnement de test pour un plan de référence donné (c'est à dire dans une seule PRM). La petite fenêtre en haut à droite correspond au point de vue de la caméra, la fenêtre principale montre le déplacement des acteurs dans l'environnement : les acteurs sont représentés par des cylindre orientés (orientation représentée par une flèche au dessus des cylindres), le chemin calculé dans le graphe de la PRM est affiché comme une succession de segments colorés en blanc (peu visibles sur les images) et la cible courante du triptyque, qui intervient dans la fonction de lissage, est représentée par une sphère rouge.

## 3 Résultats et perspectives

### 3.1 Résultats

Dans le but de pouvoir valider la pertinence de notre modèle, nous avons en vons fait une implémentation. Cette implémentation nous permettra également de tester différentes fonctions de qualité dans des situations-types (Exemples : zones où la visibilité est réduite, zones où les déplacements sont très contraints, ...). Nous avons donc construit un environnement type (voir figure 30) dans lequel nous pouvons tester notre modèle sur ces différentes situations.

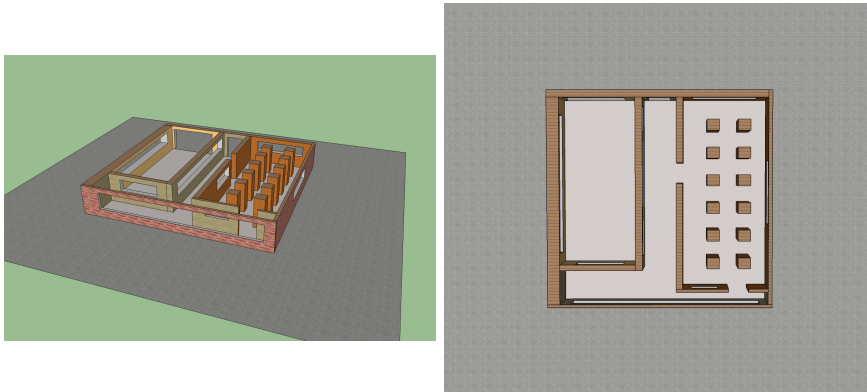


FIG. 30 – Environnement de test permettant d’exhiber des situations-types : zone de visibilité réduite (salle avec piliers), zone de déplacement contraint (couloir étroit); l’environnement permet aussi de pouvoir tester le choix de notre algorithme quand plusieurs types de chemin sont possibles (Préfère-t-on un chemin court dans un environnement contraint ou un chemin long dans un environnement libre?)

À l’aide de TOPOPLAN([Lam09]), nous déterminons une triangulation de l’espace navigable (voir figure 31). Ce logiciel nous fourni également un détecteur de collision que nous utilisons dans la construction de nos PRMs.

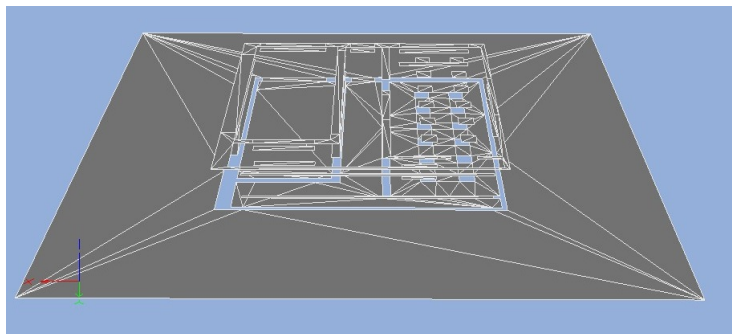


FIG. 31 – Triangulation de l’environnement à l’aide du logiciel TOPOPLAN [Lam09]

Dans un premier temps, nous avons construit un ensemble de PRMs pour différents types de plans ("Two shot" et "OverTheShoulder" pour les deux acteurs). Nous avons ensuite implémenté l'algorithme de Dijkstra ainsi que la fonction de lissage explicités dans la section précédente.

Pour le moment, nos PRMs sont indépendantes, dans le sens où nous ne tirons pas les mêmes points pour chacune d'entre elles, nous ne pouvons donc pas encore effectuer de changement de plan. Nous n'avons pas non plus terminé l'implémentation de la construction de la carte topologique et de fait, la division du problème en sous-problèmes. Nous arrivons cependant à parcourir le graphe d'une PRM, à plan de référence fixe donc, à en extraire une trajectoire valide (pour l'instant notre algorithme de Dijkstra nous fournit une seule trajectoire) ainsi qu'à lisser cette trajectoire grâce à notre fonction de lissage (voir figure 29).

Pour le moment, notre code n'étant pas optimisé, nous calculons une PRM en approximativement une heure sur un processeur *Intel(R) Xeon(R) CPU 2.93 GHz*; *RAM : 6 Go*, et une fois la PRM calculée, un chemin tel que celui fourni en exemple (voir figure 29 : environ 25 mètres) se calcule (à savoir : application de l'algorithme de Dijkstra et lissage) en approximativement 5 minutes.

## 3.2 Perspectives

Dans notre implémentation, il nous faut définir plus de plans de référence (extrait de films ou de la littérature) et construire simultanément (en proposant les mêmes configurations) les PRMs qui leur sont associées. De cette manière les changements de plans sont possibles et nous pouvons parcourir l'ensemble des PRMs pour planifier les trajectoire des acteurs.

Il nous faut également implémenter la division du problème général en plusieurs sous-problèmes ; c'est à dire achever l'implémentation de la construction de la carte topologique, permettre à notre algorithme de Dijkstra de rendre les  $k$  meilleurs chemin à priori et comparer la qualité des chemins lissés.

Dans un troisième temps, il faudra aussi décrire complètement une fonction de qualité pour le film, à savoir expliciter une fonction de qualité pour les changements de plans et une pour les transitions de manière à avoir une fonction de qualité globale  $Q_{globale} = f(Q_{plans}, Q_{transitions}, Q_{changements})$ . Une fois ces fonctions définies, nous pourrions optimiser les paramètres de  $f$  (par apprentissage par exemple), de manière à favoriser plus ou moins un critère de qualité (le style de l'utilisateur pourra également être pris en compte).

Nous avons conçu notre modèle en définissant un « point d'intérêt central » pour une scène. Cette notion est généralisable à des scènes plus complexes (plus d'acteurs par exemple). Il suffit donc, pour pouvoir planifier les déplacements des acteurs de la caméra, d'étendre notre modèle en définissant des plans de référence adaptés pour chaque scène et d'adapter les fonctions de qualité pour le lissage et le parcours de graphe.

## Conclusion

Dans le cas de la réalisation de films mettant en scène des acteurs virtuels, plusieurs sous-problèmes de planification de mouvement complexes se posent. Pour limiter la complexité du problème général, ces sous-problèmes sont généralement résolus les uns après les autres plutôt que simultanément ; cette opération

est souvent inévitable, par exemple si la résolution du problème général nécessite de se faire en temps réel. Cependant, le fait d'avoir plusieurs étapes de résolution, avec par exemple un asservissement de la caméra aux acteurs, engendre parfois des films dont la qualité n'est pas optimale.

Lors de ce stage, nous nous sommes intéressés à ces questions de perte de qualité. Notre objectif était de générer automatiquement un film sur deux acteurs en privilégiant sa qualité, sans contraintes de temps réel. Nous avons tenté de résoudre ce problème par planification de mouvement, dont nous connaissons des techniques de résolution.

Nous avons donc défini la notion de qualité d'un film à partir de règles dans le domaine de la cinématographie classique. Nous avons ensuite proposé un modèle permettant de prendre en compte cette notion de qualité. Nous avons implémenté ce modèle en utilisant des PRMs, une méthode permettant de les parcourir ainsi qu'une fonction de lissage; tout cela en gardant comme objectif principal la qualité du film. Cette implémentation, bien qu'encore incomplète, nous a permis de valider la pertinence de notre modèle.

De plus, nous avons construit notre modèle de manière à ce que la qualité, qui est une notion subjective, soit paramétrable par l'utilisateur de manière à ce qu'il puisse définir un style cinématographique. Ce modèle a également été conçu de manière à pouvoir l'étendre à des scènes plus complexes (plus que deux acteurs), mais en ayant toujours en ligne de mire la qualité, au détriment parfois de la rapidité de calcul.

## Références

- [Ari93] Daniel Arijon. *Grammaire du langage filmé*. Éditions Dujarric, 1993.
- [CON08] Marc Christie, Patrick Olivier, and Jean-Marie Normand. Camera control in computer graphics. In *COMPUTER GRAPHICS Forum*, July 2008.
- [DH95] John Douglass and Glenn Harnden. *The Art of Technique : An Aesthetic Approach to Film and Video Production*. Allyn & Bacon, 1995.
- [Dur99] Frédo Durand. *3D Visibility : analytical study and applications*. PhD thesis, Université Joseph Fourier, Grenoble I, July 1999.
- [GAG04] Andújar C. G., Vásquez P. P. A., and Fairén M. G. Guided tours through complex walkthrough models. In *Eurographics 2004 Conference*, 2004.
- [Lam09] Fabrice Lamarche. Topoplan : a topological path planner for real time human navigation under floor and ceiling constraints. In *COMPUTER GRAPHICS Forum*, 2009.
- [Lat91] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [LaV06] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [LC08] Tsai-Yen Li and Chung-Chiang Cheng. Real-time camera planning for navigation in virtual environments. In *International Symposium on Smart Graphics*, 2008.
- [LKJ00] Steven M. Lavalle, James J. Kuffner, and Jr. Rapidly-exploring random trees : Progress and prospects. In *Algorithmic and Computational Robotics : New Directions*, pages 293–308, 2000.
- [NO04] Dennis Nieuwenhuisen and Mark H. Overmars. Motion planning for camera movements in virtual environments. In *IEEE International Conference on Robotics and Automaton*, 2004.
- [OSTG09] Thomas Oskam, Robert W. Sumner, Nils Thuerey, and Markus Gross. Visibility transition planning for dynamic camera control. In *Eurographics Symposium on Computer Animation*, 2009.
- [PGT07] J. Pettre, H. Grillon, and D. Thalmann. Crowds of moving objects : Navigation planning and simulation. *Robotics and Automation, 2007 IEEE International Conference on*, pages 3062–3067, 10-14 April 2007.
- [PS98] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization : Algorithms and Complexity*. Prentice-Hall, Inc, 1998.
- [War03] Peter Ward. *Picture Composition for Film and Television*. Focal Press (Oxford,Boston), 2003.