A virtual cinematography system that learns from examples

Mathieu Chollet

Rapport de stage de fin d'études

A virtual cinematography system that learns from examples

Mathieu Chollet
<mathieu.chollet@telecom-bretagne.eu>

Encadrants de stage :
Marc Christie - IRISA / INRIA Rennes Bretagne Atlantique
Rémi Ronfard - INRIA Grenoble Rhône-Alpes

Conseiller d'études :
Ioannis Kanellos - Telecom Bretagne

Responsable de filière :
Claire Lassudrie - Telecom Bretagne

Rennes, le 13 août 2011

Résumé

La production de films nécessite des connaissances en cinématographie, notamment sur les techniques de cadrage, de placement de caméra et de montage. Dans le cadre d'applications 3D comme le prototypage de films ou les systèmes de narration interactive, il apparaît intéressant d'intégrer ces connaissances pour produire une séquence d'images servant de support à la narration. Les premières approches mettant en oeuvre un contrôle de caméra « cinématographique »utilisent des quantités limitées de caméras. Ceci garantit un montage respectant les conventions cinématographiques, mais ne permet pas de faire varier le style de montage. Nous proposons dans ce rapport un modèle de montage cinématographique évaluant plans et transitions de manière générique. Le montage est évalué indépendamment de la manière dont les caméras sont générées, et permet l'utilisation de tous types de caméras et de transitions. Nous proposons de plus une méthode d'apprentissage pour le paramétrage de cette évaluation à partir d'exemples.

Abstract

The production of movies requires knowledge in cinematography, in particular on techniques such as framing, camera placement and editing. For a range of 3D applications such as film prototyping and interactive narration systems, the automatization of the editing process appears a useful way of providing shot sequences that convey the story and are correct with respect to cinematographic conventions. Existing systems have used limited amounts of carefully chosen cameras. This ensures that the edit respects cinematographic conventions, but lacks variability in directorial style. In this report, we introduce a model of cinematographic editing evaluating shots and cuts in a generic fashion. This casts the problem as independent from camera generation and allows the use of all kinds of shots and cuts. Furthermore, we propose the use of maching learning methods for tuning the parameters of the model with examples.

Remerciements

Travailler dans le milieu de la recherche a ceci d'exceptionnel que l'on y rencontre des individus passionnés et passionnants. Il m'est ici donné l'occasion d'adresser mes remerciements à ceux et celles qui ont permis de faire de ce stage une expérience formidable.

Tout d'abord, un grand merci à mes maîtres de stage, Rémi Ronfard et Marc Christie, pour leur patience, leurs conseils et leurs encouragements, et à Christophe Lino pour sa disponibilité et son aide précieuse tout au long de ce stage.

Pour ses questions avisées et son intérêt rafraichissant pour mon travail, je tiens à remercier mon conseiller d'études, Ioannis Kanellos.

Pour leur ponctualité (à la pause de 16h), leur assiduité (aux séances de jeux du mardi midi), les échanges (balistiques) en salle Batz, toute la bande des stagiaires : Charly, Clément, Carl, Alexandre, Benoit, Quentin, François, Fabien, Aurélien, Charles. Mais aussi les ingénieurs et doctorants de l'équipe Bunraku, mention spéciale pour Orianne et ses talents de chef.

Tous les « POC »pour leur capacité à remplir ma boîte mail à une vitesse impressionnante.

Enfin, une pensée spéciale pour une personne spéciale, Mathilde, pour ton soutien indéfectible, tes encouragements qui font toujours mouche, et pour tout le reste, merci.

Table des matières

Introduction

Le contrôle de caméra est un cas particulier de planification de mouvement. Le but est de déterminer, dans un environnement à 7 degrés de liberté (3 en position, 3 en orientation, et 1 pour le zoom), une configuration de caméra respectant des contraintes particulières : par exemple, assurer une bonne visibilité des objets importants à l'image. C'est un composant essentiel d'un grand nombre d'applications, telles que la visualisation de données, les visites virtuelles (d'un musée par exemple), la narration virtuelle ou les jeux vidéo 3D.

Les travaux sur le contrôle automatique de caméra dans un environnement virtuel s'appuient sur le domaine de la cinématographie, où des règles et conventions ont été identifiées et constituent les pratiques classiques du placement de caméra et du montage. Le montage cinématographique consiste à choisir la succession de plans et des transitions entre ces plans pour filmer une scène : on obtient alors une séquence qui, assemblée avec les séquences des scènes précédentes et suivantes, produit un film. Les difficultés sous-jacentes du problème de contrôle de caméra et du montage résident dans la formalisation de la connaissance cinématographique, souvent empirique, des jeux de règles spécifiques à des styles cinématographiques.

Si certains problèmes du contrôle automatique de caméra ont été résolus, comme la détermination d'images sans occultations ou le respect de règles de continuité sur la position relative des personnages à l'écran entre plans successifs, le choix du montage reste toujours un problème ouvert. Une manière de l'appréhender consiste à sélectionner un ensemble de scènes de films que l'on choisit comme références, et à déduire de l'observation de ces scènes exemples des règles de montage empiriques. Cette approche peut poser problème : les situations rencontrées ne seront jamais parfaitement similaires à celles des films choisis en référence, et l'on risque de disposer d'exemples contradictoires. Il faut donc se demander quels sont les conditions qui amènent à réaliser une transition à un certain moment d'une action, et quels paramètres permettent de choisir le type de plan suivant. Réaliser une telle évaluation manuellement peut être difficile. Une classe de techniques utilisée en informatique, l'apprentissage automatique, permet en revanche d'aborder des problèmes en déterminant des connaissances que l'on veut apprendre, à partir de données exemples. La problématique de ce stage est d'évaluer l'utilisation de telles méthodes pour apprendre un modèle de montage cinématographique.

Dans la première partie de ce rapport, nous présentons le contexte du stage, les objectifs fixés et la méthodologie qui a été choisie pour les atteindre. Nous dressons ensuite un état de l'art sur le contrôle de caméra et sur les techniques d'apprentissage susceptibles d'être transposées au problème. Le troisième chapitre détaille le modèle de cinématographie qui a été mis au point ainsi que son implémentation et son évaluation préliminaire. Nous présentons enfin la méthode d'apprentissage choisie pour paramétrer ce modèle, travail qui est actuellement en cours et dont les résultats ne sont pas encore disponibles.

Chapitre 1

Contexte et objectifs du stage

1.1 Laboratoire d'accueil

Depuis sa création en 1967, dans le cadre du Plan Calcul, L'INRIA (Institut National de Recherche en Informatique et en Automatique), originellement appelé IRIA, est devenu l'un des plus importants centres de recherche en informatique français : de nos jours, l'institut est installé dans 8 centres : Paris, Saclay, Bordeaux, Grenoble, Lille, Rennes et Sophia Antipolis.

Les thèmes de recherche de l'INRIA sont organisés entre cinq domaines : « STIC pour les sciences de la vie et l'environnement », « Mathématiques appliquées, calcul et simulation », « Perception, cognition, interaction », « Réseaux, systèmes et services, calcul distribué », « Algorithmique, programmation, logiciels et architectures ».

Ce stage s'inscrit dans une collaboration entre deux équipes de recherche de l'INRIA travaillant sur le thème « Interaction et visualisation », du domaine « Perception, cognition, interaction »: l'équipe Evasion (INRIA Grenoble) et l'équipe Bunraku (INRIA / IRISA Rennes).

Bunraku

L'équipe Bunraku s'intéresse au domaine de l'interaction entre humains réels et virtuels dans les environnements virtuels. En particulier, l'un des axes de recherche de l'équipe est celui des scénarios interactifs, et de la manière de choisir des séquences de points de vue cohérentes pour transmettre ce scénario à l'utilisateur.

Evasion

L'équipe Evasion développe de nouveaux outils plus intuitifs pour la création numérique, notamment dans les domaines de la création de formes, de mouvements et de films en animation 3D.

1.2 Objectifs du stage

Le choix des plans et le montage d'un film sont faits en fonction de plusieurs aspects :
- Informatif : quelle action décrit-on, que cherche-t-on à apprendre au spectateur ?
- Esthétique : créer une image esthétiquement satisfaisante.
- Émotionnel : placer le spectateur dans un état émotionnel désiré.
- Cognitif : permettre au spectateur de construire une représentation mentale de l'environnement, et de la suite logique de la narration.

Les environnements virtuels de formation, les systèmes de narration interactive, ou les systèmes de prototypage de films sont autant d'applications pour lesquelles la manière de choisir et d'enchaîner les points de vue répond aux mêmes besoins. On peut donc envisager de transposer les techniques de cinématographie afin d'obtenir des choix de séquences correspondant aux besoins de ces applications.

Même s'il existe des conventions cinématographiques appliquées dans la majorité des séquences filmées, il n'y a pas une seule manière de filmer une scène donnée. Des variations de style, par exemple sur le rythme d'enchaînement des coupures ou sur la manière de cadrer les acteurs, peuvent modifier profondément la manière dont la scène est perçue. Dans cette perspective, nous proposons de pouvoir paramétrer le style voulu de montage à partir d'exemples de séquences.

L'objectif du stage est ainsi de proposer un modèle de contrôle de caméra cinématographique pour les environnements virtuels qui puisse être paramétré par apprentissage à partir d'exemples, et d'évaluer les résultats obtenus par une telle approche.

1.3 Méthodologie

La méthodologie utilisée pendant dans le stage est la suivante :

1. Dans une première phase, une étude bibliographique a été menée sur les techniques de cinématographie, le contrôle de caméra en environnement virtuel, ainsi que les grandes classes de méthodes d'apprentissage qui pourraient être transposées au problème.
2. A partir de cet état de l'art, un modèle de cinématographie a été mis au point en s'inspirant des approches existantes et en prenant en compte la perspective de l'apprentissage futur de ses paramètres. Ce modèle a ensuite été implémenté dans le cadre d'un système existant.
3. Afin de valider la pertinence du modèle et l'étendue de la variabilité des séquences produites, nous avons réalisé une évaluation préliminaire du modèle avec un paramétrage manuel.
4. Des méthodes d'apprentissage ont ensuite été proposées pour pouvoir apprendre les paramètres du modèle, et sont actuellement en cours d'implémentation. En parallèle, un ensemble d'exemples de séquences est en train d'être rassemblé.
5. Une comparaison sera alors effectuée entre une séquence de référence réalisée manuellement, et des séquences produites automatiquement (a) par le système existant, (b) le modèle paramétré manuellement, et (c) le modèle paramétré par apprentissage.

Chapitre 2

État de l'art

2.1 Contrôle de caméra

Dans cette partie, nous évoquons d'abord le domaine de la cinématographie et les règles qui le constituent. Nous présentons ensuite les différentes types de contrôles de caméras utilisés dans les applications 3D, en insistant sur les approches automatiques.

2.1.1 Cinématographie

La cinématographie est un domaine complexe : des décennies de pratique ont permis d'identier des règles empiriques qui atteignent un certain consensus dans la communauté étudiant le domaine.

Définitions

Le cadrage consiste à choisir la distance de la caméra par rapport à la scène, et le type de lentille. On peut distinguer un certain nombre de types de cadrages (*cf fig. 2.1*), qui possèdent des intérêts narratifs différents. Par exemple, des plans très larges peuvent être utilisés pour mettre l'accent sur l'environnement de la scène, tandis qu'un gros plan attirera l'attention sur un élément particulier, comme l'expression d'un acteur.

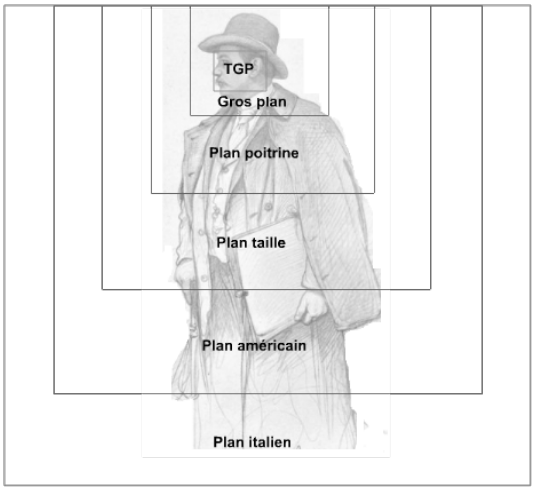


FIGURE 2.1 – Différents types de cadrages

Enfin, les positions et tailles des différents éléments sur le plan final définissent la composition du plan. Celle-ci peut être ajustée en modifiant finement la position de la caméra. Une image est ainsi caractérisée en termes de formes, lignes, mouvements. L'agencement de ces différents éléments a une influence sur la manière dont l'image est perçue [TB09b]. La règle des tiers est un exemple de règle de composition utilisée très fréquemment, notamment en photographie, et recommande de ne pas centrer le sujet de l'action. Des résultats harmonieux seront en revanche obtenus si l'on respecte un ratio 1/3 : 2/3. Ces règles sont basées sur des notions psychologiques et sont ainsi perçues de manière variable selon les individus, ce qui rend difficile l'analyse objective de leur influence sur l'esthétique générale d'un plan.

Montage cinématographique

Selon Thompson, le montage cinématographique consiste à définir l'organisation finale d'un film à partir des bandes vidéos et du contenu sonore obtenus après le tournage [TB09a]. Dans notre contexte, nous simplifions le processus en l'assimilant aux choix suivants :

- Le choix des plans de la scène, c'est à dire, pour chaque plan :
  - Le choix de la position de la caméra dans la scène.
  - Le cadrage, c'est à dire le choix de la taille du cadre et de l'angle de la caméra.
  - La composition du plan : déterminée par le position de caméra et le cadrage, la composition d'un plan représente les tailles relatives et les positions des différents éléments par rapport à l'écran.
- Le choix des transitions entre plans, c'est à dire :
  - Le choix de l'instant où la transition est effectuée. Ou encore, de manière équivalente, la durée d'un plan.
  - Le choix du type de transition, c'est à dire le choix de quel plan va succéder au plan courant.

Règles et conventions

Un certain nombre de règles permettent d'assurer dans la plupart des cas un montage qui soit le plus naturel au spectateur. L'une d'elle est d'assurer la continuité du montage, notamment de la position des personnages à l'écran, de leurs directions de regard et de leurs mouvements [TB09a].

Selon les éléments et les actions constituant une scène, des conventions cinématographiques empiriques permettent d'identifier quels plans vont respecter les règles de continuité. Ces règles sont appelées *idiomes*. Par exemple, Arijon décrit le principe du triangle qui, lors d'une conversation entre plusieurs personnages, permet de tracer une *ligne d'intérêt* (*cf fig. 2.2*) : placer l'ensemble des caméras de la scène du même côté de cette ligne permet de s'assurer que le spectateur ne sera pas troublé par des changements de direction de regard, ou de position relative des personnages à l'écran [Ari76].

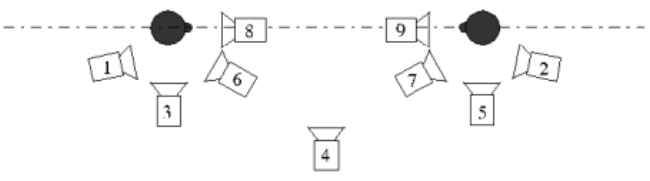


FIGURE 2.2 – (Fig.4 de Christie et al. [CON08]) Positions relatives de caméra pour une discussion entre 2 personnages.

Le rythme du montage est un paramètre important du style d'une séquence filmée. Barry Salt a réalisé dans [Sal03] une analyse des distributions des durées de plans de films, et émet l'hypothèse que ces distributions sont proches de distributions lognormales (*cf fig. 2.3*).

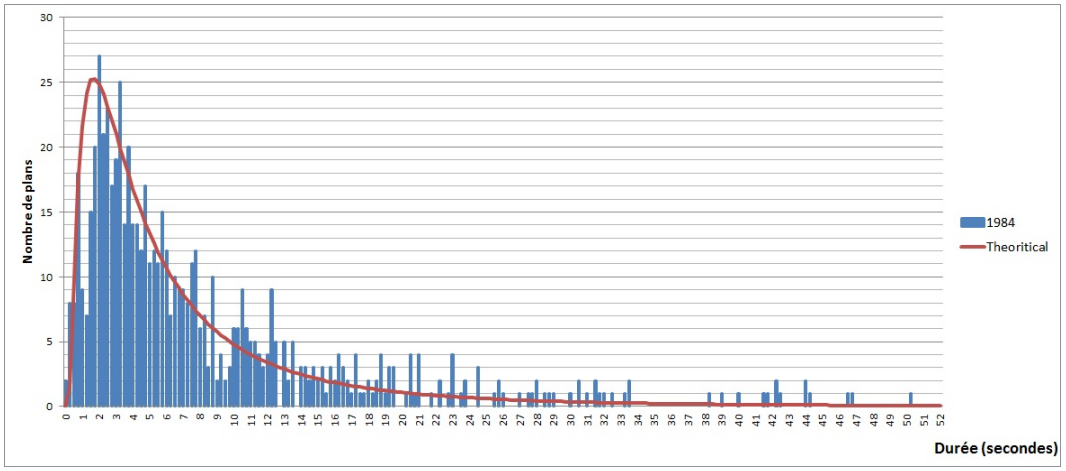


FIGURE 2.3 – Histogramme des durées de plans du film *1984* de Michael radford, et courbe d'une distribution lognormale paramétrée à partir des durées de plan du même film.

Rappelons enfin que s'il existe des règles cadrant les bonnes pratiques en cinématographie, le réalisateur dispose encore d'une grande liberté d'expression dans le choix des prises de vues et dans l'articulation du montage : ce sont les choix qu'il va effectuer qui vont constituer son style cinématographique propre.

2.1.2 Contrôle interactif de caméra

Une caméra virtuelle est un objet disposant de sept degrés de libertés : trois en translation, trois en rotation et un degré de liberté contrôlant la taille du cadre. Une caméra peut être contrôlée de deux manières : soit en laissant son contrôle à l'utilisateur, totalement ou en partie, on parle alors de contrôle interactif. L'autre manière de procéder consiste à automatiser la manipulation de la caméra.

Le contrôle de caméra interactif permet de laisser à l'utilisateur la possibilité d'influer sur le placement de la caméra. On peut classer les techniques de contrôle de caméra interactives en trois catégories [CON08] :

- Le contrôle direct, où l'utilisateur influe directement sur les degrés de liberté de la caméra. Différentes métaphores du déplacement de la caméra ont alors été identifiées : *eyeball in hand*, où l'utilisateur déplace directement la caméra, *world in hand*, où l'utilisateur manipule le monde devant une caméra fixe, *flying vehicule* où la caméra se comporte comme un avion (l'utilisateur modifie les vitesses de rotation et de translation), ou encore *walking metaphor*, où la distance entre la caméra et le sol est constante.
- Le contrôle *Through the lens*, où l'utilisateur modifie la position des différents éléments de l'image à l'écran, laissant ensuite au système la charge de calculer la position de la caméra.
- Le contrôle assisté, où le système utilise sa connaissance de l'environnement pour aider l'utilisateur dans la tâche du contrôle de la caméra, notamment dans des contextes de planification de chemin ou d'évitement de collisions.

2.1.3 Contrôle automatique de caméra

Le premier systême de contrôle automatique de caméra a été introduit par Blinn [Bli88] dans ses travaux à la NASA : ce système permettait de visualiser deux objets, spécifiquement une sonde spatiale et la planète que la sonde approchait, en plaçant la caméra en fonction des positions absolues des deux objets, de leurs positions désirées à l'écran, de la taille du champ de la caméra et d'un vecteur utilisé pour définir l'axe vertical de l'image. Cette première approche était fondamentalement limitée au calcul d'un plan unique pour des images comportant deux objets, et des travaux ultérieurs, comme ceux de Christianson et al. [CAH+96] ou He et al. [HCS96], se sont attachés à décrire des systèmes pouvant s'adapter à d'autres situations et proposant plusieurs points de vue.

On peut faire une disctinction entre les approches réactives, qui contrôlent la caméra en temps réel en choisissant le plan jugé le plus adapté à représenter la situation courante, et en opérant une transition si la situation l'exige (évènement particulier, occultation des éléments à l'écran), et les approches séquentielles, qui calculent le montage entier d'une scène dont on connaît le script à l'avance. Nous présentons à la fin de cette partie le fonctionnement du système qui servira de support au travail du stage.

Approches réactives

He et al. présentent dans [HCS96] un paradigme fondateur pour la conception de systèmes de contrôle automatique de caméra cinématographique, ainsi qu'un système temps réel, le « Virtual Cinematographer ». Ce système permet de créer des « fêtes virtuelles » où des acteurs contrôlés par des humains peuvent se déplacer, discuter ou encore prendre un verre.

Le coeur du système est constitué de deux éléments : les *idiomes* et *modules de caméra*. Les modules de caméra sont chargés de placer la caméra d'une manière spécifique lorsqu'ils sont appelés : par exemple, $external(actor1, actor2)$ prend en attributs deux acteurs et place la caméra au dessus de l'épaule de $actor1$ (plan dit « Over The Shoulder ») de telle façon que $actor1$ soit représenté sur les deux tiers de l'écran, et $actor2$ sur le dernier tiers. Les idiomes contiennent les connaissances cinématographiques du système pour un type de scène en particulier, ainsi l'idiome *2Talk* est utilisé lorsqu'une conversation entre deux personnages est initiée (*cf fig. 2.4*). Un idiome prend la forme d'un automate à états finis où chaque état représente un module de caméra, et possède un certain nombre de conditions qui, lorsque satisfaites, appellent à une transition vers un autre état (et ainsi un autre module de caméra). Il est intéressant de noter qu'un idiome peut en contenir un autre : ainsi l'idiome *3Talk* est constitué en partie de l'idiome *2Talk*, ce qui permet d'utiliser des plans où seulement deux des trois acteurs en conversation apparaissent à l'écran.

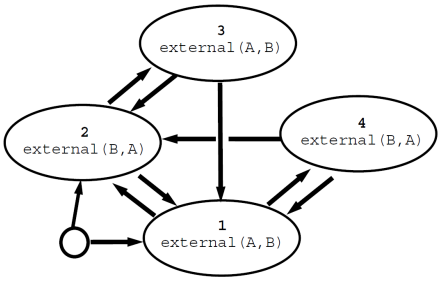


FIGURE 2.4 – (Fig. 7 de He et al. [HCS96]) L'idiome 2Talk.

L'utilisation d'automates à états finis pour représenter les idiomes possède cependant des limitations intrinsèques : les conditions de transitions entre les modules de caméras et la structure des automates restent fixes, et les modules de caméras rendent toujours des images à partir des mêmes points de vue et avec des compositions similaires.

Hawkins a proposé une approche multi-agents pour modéliser les connaissances cinématographiques [Haw04]. Trois types d'agents sont implémentés et communiquent sous forme d'une chaîne : les agents réalisateurs, monteurs et caméramen. L'agent réalisateur propose au monteur un ensemble de plans possibles en fonction de l'action qui se déroule. Le monteur choisit alors l'enchaînement entre ces différents plans, puis l'agent caméraman se charge de manipuler la caméra. Un avantage de cette méthode, qui se base sur l'organisation classique d'un plateau de tournage, est qu'on décompose le contrôle de caméra en trois problèmes séparés : le choix des plans disponibles par rapport à l'action, le choix des transitions entre les plans disponibles, et le placement et contrôle de caméra proprement dit.

Cetta approche a été reprise par Kneafsey et al. [KM05], qui utilisent les personnages non-joueurs (NPCs) de jeux vidéo pour implémenter des agents cinématographes. Lima et al. proposent une méthode de cinématographie automatique pour systèmes narratifs [dLPd+09], et Passos et al. [PMC+10] implémentent un système de cinématographie pour le jeu vidéo. Les agents mis en oeuvre dans ces trois méthodes reprennent les principes de ceux proposés dans [Haw04] : des agents s'occupent de mettre en place les différents éléments de la scène et de référencer l'action en cours, et plusieurs agents personnifient les différents caméramen. L'agent central du système est le monteur, qui choisit les plans (c'est à dire quel agent caméraman va filmer l'action en cours) et les transitions. Les approches de Lima et al. et Passos et al. utilisent l'apprentissage automatique pour modéliser la connaissance de l'agent monteur : un expert réalise manuellement des montages que le système va ensuite utiliser comme exemples pour apprendre son comportement.

Approches séquentielles

Christianson et al. [CAH+96] utilisent le même système d'idiomes que dans [HCS96], en les modélisant cette fois dans un langage spécifique, le *Declarative Camera Control Language* (DCCL). Leur système utilise un script regroupant les informations de positions et d'activités des acteurs au cours du temps. Chaque idiome est spécifié pour être utilisé pour certains types d'activités, mais il n'est pas possible d'évaluer la qualité du rendu d'une scène avant de calculer le résultat : le système va ainsi calculer tous les montages possibles (*cf fig. 2.5*), pour les évaluer dans une phase finale.

Elson et al. ont mis en oeuvre avec le système CamBot [ER07] une méthode similaire : les informations cinématographiques, en revanche, sont ici stockées dans une base de données de configurations de scènes, *stages*, de configurations de placement de personnages, *blockings*, et de caméras, *shots*. Les différentes caméras de cette base sont annotées manuellement de manière à savoir à quelles actions elles sont adaptées (quels *shots* sont adaptés à quels *blockings*), et de manière à pouvoir juger de leur qualité cinématographique (par exemple, l'intensité d'un type de plan). A partir du script fourni en entrée, CamBot énumère dans une première phase les combinaisons de scènes et placements de personnages qui sont adaptées aux actions du script, mettant ainsi en scène le script dans l'application 3D. Le système énumère ensuite les caméras compatibles avec cette mise en scène, qui sont notées en fonction de leur concordance avec des contraintes cinématographiques : respect de la ligne d'intérêt, « intensité » des plans. On obtient ainsi plusieurs montages que l'on peut comparer en fonction de la somme des notes de leurs plans.

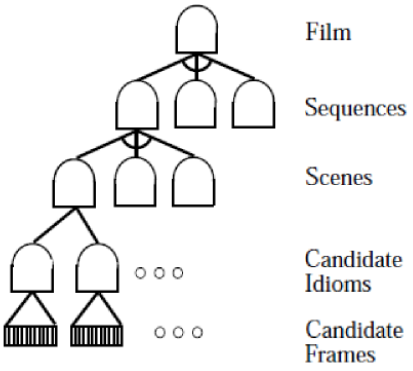


FIGURE 2.5 – (Fig. 7 de Christianson et al. [CAH+96]) Calcul des montages possible à partir des idomes compatibles.

Une autre approche permettant un montage complet est définie dans le brevet déposé par Xtranormal [Ron09]. Ici, les différentes caméras sont définies manuellement par l'utilisateur. Un ensemble de règles cinématographiques sur les plans et les transitions est implémenté par des fonctions de coût : par exemple, la fonction de coût *jumpleftright* pénalise les transitions qui vont inverser les positions relatives de couples de personnages à l'écran. Les fonctions de coût sont additives entre elles, ce qui permet de calculer le score d'un plan en faisant la somme des coûts de ces fonctions. On calcule ensuite le montage optimal d'une scène par récurrence, c'est à dire que le score d'une scène passant à un plan S_{i+1} d'une durée Δt se calcule à partir du score de la scène filmée par les plans $(S_1, ..., S_i)$ jusqu'au moment t_i, auquel on ajoute le coût de la transition du plan S_i au plan S_{i+1} et le coût du plan S_{i+1} pendant une durée Δt.

Le système existant

Dans le cadre de ce stage, nous avons étendu le système développé par Lino et al. à l'INRIA Rennes [LCL+10] : ce système permet un contrôle automatique de caméra dans un cadre temps réel. Il utilise les idiomes d'Arijon pour modéliser les connaissances cinématographiques, en catégorisant l'espace autour des éléments importants de la scène en zones en fonction du type de plan que l'on obtient en y plaçant la caméra (*cf fig. 2.6*). Son fonctionnement est réactif, et est constitué de quatre étapes successives :

1. Sélection des éléments narratifs les plus importants de la scène (comme les protagonistes principaux de l'action), par l'utilisation d'un moteur narratif séparé.
2. Calcul des *volumes directeurs* : cette étape consiste à construire autour des éléments narratifs principaux un ensemble de zones. Des *volumes sémantiques*, zones qui correspondent au type de plan et de cadrage que l'on obtiendrait en y plaçant la caméra, et des *volumes de visibilité* qui définissent les zones où les éléments principaux de la scène sont visibles. Ces deux types de volumes sont combinés pour obtenir les volumes directeurs.
3. Filtrage des volumes directeurs en fonction de critères cinématographiques : ces critères peuvent filtrer selon des règles de cinématographie classiques (respect de la ligne d'intérêt), selon un style cinématographique voulu (en favorisant certains volumes directeurs), ou filtrer sur la composition de l'image rendue.
4. Enfin, le système calcule s'il est nécessaire d'effectuer une transition, par exemple si une occultation d'un des éléments principaux est détectée, si le plan a duré suffisamment longtemps ou encore si le moteur narratif indique un changement d'élément narratif principal. Le système calcule alors le prochain plan en réitérant les étapes précédentes. La transition peut être continue (mouvement de caméra) ou nette (*cut*).

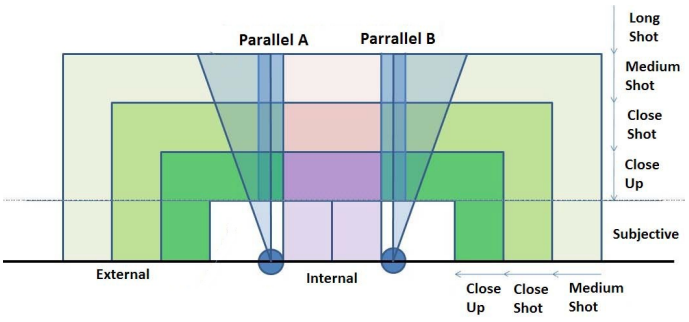


FIGURE 2.6 – (Fig. 3 de Christie et al. [CON08]) Volumes sémantiques pour un dialogue entre deux personnages (les personnages sont représentés par les cercles).

Ce système est intéressant dans le cadre du stage car il nous permet de disposer d'un système pouvant d'ores et déjà résoudre des problèmes comme la détermination d'images sans occultations ou du placement précis de caméra en fonction d'un type de plan désiré.

2.2 Apprentissage

La discipline de l'apprentissage automatique se consacre à l'étude des méthodes qui peuvent permettre à un ordinateur d'apprendre, c'est à dire, selon Mitchell, de s'améliorer avec l'expérience. L'apprentissage automatique est utilisé pour approcher de nombreux problèmes : la reconnaissance de la parole, le contrôle de véhicule automatique ou encore la fouille de données sont des exemples d'applications où les méthodes d'apprentissage automatique ont démontré leur efficacité [Mit97].

Nous présentons dans la partie suivante la méthode générale de définition d'un problème d'apprentissage et son application à notre problématique. Nous introduisons ensuite les principales classes d'algorithmes d'aprentissage automatique et analysons des méthodes utilisées dans des domaines proches qui seraient transposables à notre problématique.

2.2.1 Définition du problème

La conception d'un algorithme d'apprentissage consiste à faire différents choix :
- Choisir les connaissances que l'on veut apprendre et leur représentation.
- Choisir les données que l'on va utiliser comme exemples.
- Choisir les observations qui vont être faites sur les exemples pour extraire les connaissances voulues.
- Une fois les trois choix précédents réalisés, choisir une méthode d'apprentissage pour résoudre le problème.

Ces choix ne sauraient bien sûrs être faits indépendamment. Cette formalisation du problème est centrale à toute implémentation d'algorithme d'apprentissage. Le cas de l'utilisation de films réels pour un système de contrôle automatique de caméra représente un problème non traité dans la littérature actuelle, l'étude de cette formalisation est donc d'autant plus important. Celle-ci continuera donc pendant le stage : nous donnons ici quelques pistes sur la manière d'aborder le problème.

Connaissances recherchées

Les connaissances que l'on souhaite tirer des exemples peuvent être de natures et complexités différentes. Pour Passos et al. [PMC+10] et Lima et al. [dLPd+09], l'objectif est d'apprendre quelle caméra représente au mieux l'action courante. Dans le contexte du stage, l'objectif est d'apprendre a réaliser le montage d'un script de film dans son ensemble : on peut ainsi utiliser, en plus de l'action courante, les informations des actions précédentes et des plans que l'on a utilisé précédemment.

On différencie deux types de connaissances différentes qui seront utilisées par notre système, et que nous allons apprendre à partir d'exemples de films réels :

- Quels sont les plans qui sont préférables en fonction de la prochaine action du script, et des actions précédentes.
- Connaissant les plans précédents qui ont été utilisés, quel type de transition est préférable, et à quel instant la transition doit elle être faite.

Le problème peut se formaliser mathématiquement de plusieurs manières. Si l'on décompose la scène en intervalles de temps Δt, et en posant x_t l'action se passant pendant l'intervalle de temps t, et y_t le plan choisi pour représenter l'action pendant l'intervalle de temps t :

- Calculer les probabilités conditionnelles $p(y_t|x_0, ..., x_t, y_0, ..., y_{t-1})$
- De manière équivalente, déterminer la fonction f qui calcule un score z correspondant au montage $x_0, ..., x_t$ et $y_0, ..., y_t$, c'est à dire $f(x_0, ..., x_t, y_0, ..., y_t) = z$, puis calculer le maximum de cette fonction pour trouver le montage optimal.

Enfin, on peut choisir de considérer des scores booléens, un plan sera alors convenable ou non, ou des scores compris entre 0 et 1. Dans le premier cas, on se place dans un problème de classification. Dans le second cas, dans un problème de régression.

Choix des exemples

Les approches d'apprentissage de montage de Passos et al. et Lima et al. utilisent des exemples générés manuellement. En revanche, les exemples que nous allons utiliser dans ce stage vont provenir de séquences de films réels. Ces séquences devront être choisies afin que toutes les actions de la scène que nous voulons monter y soient présentes, c'est à dire qu'on trouve un exemple de scène de film réel qui contienne cette action.

Une particularité inhérente à notre formulation du problème d'apprentissage, est que les films exemples vont nous permettre de disposer de scènes réputées adéquates par rapport au style que l'on veut représenter. En revanche, il va être impossible d'en tirer automatiquement des scènes que l'on pourrait qualifer d'inadéquates. Pour résoudre ce problème, il faudra soit utiliser une méthode d'apprentissage qui puisse apprendre efficacement en se basant uniquement sur un ensembles de bons exemples, soit générer artificiellement une base de contre-exemples, c'est à dire de plans et de transitions considérés comme non-optimaux.

Représentation des exemples

L'utilisation des données vidéo brutes nécessiterait l'implémentation de techniques d'analyses d'images qui sortiraient du contexte du stage. On peut toutefois utiliser un symbolisme plus ou moins complexe pour représenter les scènes exemples, en s'inspirant de langages de description d'images existants.

Pickering décrit un langage de description d'image qui utilise des prédicats atomiques qui peuvent être combinés pour représenter des informations de haut niveau [PD03]. Les prédicats représentent des informations factuelles sur l'image, comme les notions d'occultation partielle ou totale d'un objet, de taille d'un objet par rapport au cadre, de positions absolues et relatives (au-dessus, devant...) des objets, ou encore d'angle de caméra.

Le système de cinématographie automatique de Xtranormal [Ron09] utilise un langage de description symbolique de plan (SDL) qui comprend les positions et orientations des acteurs et de leurs yeux, leur vitesse, la surface de leur visage visible à l'écran, et l'angle de caméra par rapport à leurs yeux. Pour des scènes comprenant plusieurs personnages, les mêmes attributs cités précedemment sont indiqués pour chaque acteur, même si un acteur n'est pas à l'écran : cela permet de conserver des informations importantes comme la ligne d'intérêt. Les actions sont aussi extraites du script en stockant l'intervalle temporel où se passe l'action, la classe de l'action (dialogue, animation...) et les différents éléments et acteurs impliqués dans l'action et leur rôle (agent, patient, objet...).

Une fois choisie la manière de décrire un plan, on peut décrire la totalité d'un montage sous forme d'une séquence de symboles, par exemple sous forme matricielle (*cf fig. 2.7*). vPour

Cut no.	1	2	3	4	5
Plan	Long shot	Medium shot	Close-up	Medium shot	Long shot
Durée	1.5s	2s	3s	4s	2s
Objet	Acteur A	Acteur B	Acteur A	Objet	Acteur A
Action	Regard	Regard	Déplacement	-	Regard
Direction	Droite	Gauche	Droite	-	Droite
Son	Dialogue	Dialogue	Dialogue	Bruitage	Musique

FIGURE 2.7 – (Adapté de Fig. 2 de Matsuo *et al.* [MSU03]) Description de montage sous forme matricielle.

décrire plus précisément les actions du script, nous pouvons utiliser l'ontologie décrite par Nevatia et al. dans [NHB04]. Celle-ci a été conçue dans le cadre de recherches sur la détection d'évènements sur des vidéos et fournit un cadre dans lequel exprimer différents types de propriétés des objets (objet mobile, objet de type contenant), différents types de relations entre objets (distance entre deux objets), ou différents évènements. Propriétés, relations et évènements peuvent être primitifs ou complexes, auxquels cas ils peuvent être décomposés en plusieurs éléments primitifs (par exemple, avancer une jambe peut se décomposer en : lever le pied, faire une rotation de la jambe vers l'avant, baisser le pied).

Observations

Il faut ensuite définir les observations qui seront tirées des exemples afin d'évaluer les conditions dans lesquelles différents plans et différentes transitions sont choisis. Selon la complexité de la représentation des données, on peut observer différents types de caractéristiques des scènes exemples :

- Des caractéristiques liées aux actions : par exemple le choix d'un plan « Apex » pour représenter deux personnes en train de parler, comme dans le premier tableau de la Fig.2.8.
- Des caractéristiques temporelles, comme le temps d'attente avant de couper sur un plan « Over The Shoulder ».
- Des caractéristiques de séquences d'édition, comme le choix d'un plan large si un gros plan l'a précédé, comme dans le second tableau de la Fig.2.8.

Les caractéristiques selon alors représentées par des fonctions du type suivant :

$$\phi_i(x_t, y_t) = \begin{cases} 1 & si\ x_t = Dialogue\ et\ y_t = OTS \\ 0 & sinon \end{cases}$$

ϕ_i étant alors la caractéristique associant les choix de plans « Over the shoulder » aux actions de type « Dialogue » .

2.2.2 Choix de la méthode

Les choix réalisées précédemment vont orienter le choix de la méthode d'apprentissage que l'on va utiliser. Nous classons d'abord les techniques d'apprentissage en catégories générales, puis nous analysons des méthodes d'apprentissage empruntées à des domaines connexes qui pourraient s'adapter à notre problème particulier.

Catégories de techniques

On peut classer la méthode d'apprentissage selon le type de données fournies au système :

- L'apprentissage supervisé est utilisé lorsque que le comportement que l'on veut apprendre est connu, c'est à dire qu'à chaque donnée fournie en entrée est associée une réponse cible. Un exemple de méthode d'apprentissage supervisé est l'apprentissage d'arbres de décision [Mit97]. Cette méthode est notamment utilisée dans le domaine du diagnostic médical et dans l'estimation de risques bancaires.
- L'apprentissage non-supervisé s'applique aux problèmes où il n'y a pas de connaissance à priori sur les données d'apprentissage. Ces techniques permettent de classer des données exemples en sous-groupes regroupant des données possédant des caractéristiques similaires. Ces méthodes sont très utilisées en fouille de données.
- L'apprentissage semi-supervisé est utilisé lorsque l'on dispose de connaissances sur une partie des données.

Notre problème semble ainsi s'intégrer dans un contexte d'apprentissage semi-supervisé : les plans et transitions utilisés dans les scènes de films exemples sont considérés comme corrects, et l'on ne dispose pas de connaissance à priori sur les plans et transitions qui n'y sont pas utilisés. Ce problème est particulier, car les données exemples utilisées dans les algorithmes d'apprentissage sont en général constituées d'exemples positifs et négatifs. Il peut toutefois être abordé en adaptant les méthodes d'apprentissage classique : ainsi, dans [EN08], Elkan et Noto démontrent que l'on peut résoudre un problème de classification semi-supervisé où l'on dispose de données réputées correctes mais pas de données incorrectes.

Apprentissage du score d'un plan et d'une transition

Avant de pouvoir considérer le problème du montage complet, il faut pouvoir attribuer un score (booléen ou numérique) à l'utilisation d'un plan y_t par rapport à une action x_t, et à l'utilisation d'un plan y_t par rapport au plan y_{t-1} qui le précède.

Dans les méthodes d'apprentissage s'appliquant à ces types de problème, on peut citer les classificateurs Bayésiens Naïfs : dans notre contexte, une telle méthode permettrait de calculer le plan p par rapport aux caractéristiques observées ϕ_i sur l'action x_t ou le plan précédent y_{t-1} : $p = \text{argmax}_p\, P(p|\phi_1, ..., \phi_n)$. Les caractéristiques sont ici considérées indépendantes (hypothèse « naïve »), on obtient alors en utilisant la loi de Bayes :

$$P(p|\phi_1, ..., \phi_n) = \frac{P(p) \times P(\phi_1|p) \times ... \times P(\phi_n|p)}{P(\phi_1, ..., \phi_n)}$$

Les classificateurs Bayésiens Naïfs ont l'avantage d'être très simples à implémenter et permettent néammoins d'obtenir de bons résultats dans de nombreuses applications [Mit97].

On peut aussi utiliser les machines à vecteur de support (SVM). Utilisées dans [dLPd+09], le principe de ces méthodes est d'essayer de séparer les différentes classes de données (i.e. les différents types de plans) par la plus grande marge possible, en se plaçant dans un hyperplan où les caractéristiques $\phi_1 \times ... \times \phi_n$ sont les dimensions.

L'apprentissage des plans et transitions peuvent tous les deux être réalisées avec les méthodes ci-dessus. Ces deux problèmes diffèrent principalement dans le choix des caractéristiques ϕi qui vont être observées par l'algorithme d'apprentissage. Les connaissances apprises peuvent être modélisées sous forme de tableaux du type suivant :

x_t ; y_t	OTS	Gros plan	Apex	y_{t-1} ; y_t	OTS	Gros plan	Apex
A parle à B	0.6	0.3	0.1	OTS	X	0.3	0.7
A se rapproche de B	0.4	0.1	0.5	Gros plan	0.8	X	0.2
A s'éloigne de B	0.4	0.1	0.6	Apex	0.8	0.2	X

FIGURE 2.8 – Représentation d'une scène de film exemple sous forme des tableaux des probabilités $p(y_t|x_t)$ et $p(y_t|y_{t-1})$.

Apprentissage de la séquence de plans complète

Le choix de la méthode d'apprentissage doit être fait en gardant à l'esprit des considérations d'efficacité : les méthodes choisies doivent converger vers une solution dans un temps raisonnable. Cette question a son importance dans le cadre du calcul d'un montage complet : s'il faut considérer l'ensemble des successions de plans possibles pour chaque intervalle de temps, on fait face à un nombre exponentiel de possibilités, et il faut faire des hypothèses simplificatrices.

Une possibilité consiste à utiliser des méthodes de programmation dynamique pour limiter l'apprentissage aux probabilités $p(y_t|x_t)$ et $p(y_t|y_{t-1})$, apprises par des méthodes décrites au paragraphe précédent. Le montage optimal à l'instant t est alors calculé à partir du montage optimal à l'instant $t-1$ et des probabilités $p(y_t|x_t)$ et $p(y_t|y_{t-1})$. Cette méthode est notamment utilisée dans [Ron09].

Deux grands types de méthodes d'apprentissage permettent de résoudre les problèmes combinatoires de ce type : les méthodes à base d'apprentissage par renforcement, et les réseaux bayésiens (comme les modèles de Markov cachés). Nous présentons ici des domaines où de telles méthodes ont été utilisées pour résoudre des problèmes transposables à celui du montage cinématographique.

Réseaux bayésiens Dans le traitement de la langue naturelle, des méthodes d'apprentissage sont utilisées pour étiqueter chaque mot d'une phrase à sa fonction grammaticale. On peut, dans ce cadre, décrire un ensemble de caractéristiques qui observent le choix d'étiquette pour un mot donné, et des caractéristiques pour observer les séquences d'étiquettes. Collins utilise une méthode basée sur les modèles de Markov cachés [Col02]. Un modèle de Markov caché est un processus Markovien particulier : la séquence d'états du processus Markovien est inconnue mais on dispose d'une séquence de sortie produite par le processus. Inversement, connaissant les paramètres de l'automate, on peut calculer quelle est la séquence de sortie la plus probable en fonction de la séquence d'états du système. Dans le domaine de l'étiquetage grammatical, la

séquence d'états de l'automate s'apparente aux mots de la phrase à étiqueter et la séquence de sortie décrit la séquence d'étiquettes la plus probable. L'apprentissage d'un modèle de Markov caché se réalise en affectant à chaque caractéristique ϕ_i un poids α_{ϕ_i}. Pour chaque phrase donnée en exemple, l'algorithme fait correspondre toutes les séquences possibles d'étiquettes, et calcule leur score en faisant la somme des caractéristiques pondérées par leurs poids. La séquence au score le plus élevé est comparée à l'exemple, et selon les erreurs d'étiquetage, on ajuste les poids des caractéristiques. Par exemple, si le mot « Nous » a été étiqueté en « Verbe », alors le poids $\alpha_{\phi_{Nous \to Verbe}}$ va être diminué. La résolution du problème complet est réalisée en utilisant l'algorithme de Viterbi, qui est efficace pour les problèmes fortement combinatoires.

Ce problème est proche du choix de placement de caméra : on peut comparer le problème d'attribuer les étiquettes correctes aux mots d'une phrase à celui de choisir les placements de caméra pour une séquence d'action données. Ici, la séquence d'états du modèle de Markov caché sera constituée des actions de la scène, et la séquence de sortie sera la séquence des plans utilisés.

Apprentissage par renforcement Les méthodes d'apprentissage renforcé peuvent résoudre des problèmes fortement combinatoires modélisés par des processus de décision Markovien (MDP). La modélisation du montage cinématographique par un MDP consiste, par rapport à l'état actuel et passé du système, à choisir l'action à effectuer, c'est à dire faire ou non une transition vers un autre plan. Les actions sont choisies de manière à obtenir un montage optimal, dans le sens où ce montage va maximiser une fonction dite de récompense. Les méthodes d'apprentissage renforcé cherchent à établir une politique optimale par rapport à la fonction de récompense, une politique étant la fonction déterminant quelle action doit être effectuée par rapport à l'état du système. La politique optimale maximisera alors la récompense au cours du temps.

L'application d'une telle méthode consiste à modéliser les connaissances cinématographiques dans la fonction de récompense. Les méthodes d'apprentissage renforcé inverse peuvent résoudre ce problème en permettant d'apprendre les fonctions de récompense à partir d'exemples. Lee et Popovic proposent une méthode d'apprentissage renforcé inverse permettant de reproduire des comportements à partir d'un nombre restreint d'exemples réalisés par un humain, dans le cadre d'applications de contrôle des déplacements d'un personnage [LP10]. La fonction de récompense est alors considérée comme une combinaison linéaire d'un ensemble de caractéristiques, comme le temps mis pour déplacer le personnage à sa destination, où encore une fonction spécifiant si le personnage a évité un obstacle. On comparera alors les politiques en fonction de l'espérance de leurs différentes caractéristiques par rapport à un état initial donné, et le but de l'apprentissage va être de trouver les pondérations de caractéristiques qui font que la fonction de récompense calculée va avoir une espérance de ses caractéristiques la plus proche possible de celle des trajectoires exemples.

La méthode décrite par Lee et Popovic est générique et est directement transposable au problème du montage cinématographique : il faut simplement choisir les caractéristiques adaptées à la description de notre système..

Chapitre 3

Modèle de montage cinématographique

Les approches existantes de contrôle de caméra cinématographique mettent en œuvre des idiomes extraits du cinéma. La variabilité des résultats produits par cette catégorie de méthode est intrinsèquement limitée et il est difficile de les rendre génériques, i.e. introduire de nouvelles actions, d'avoir des actions se déroulant en parallèle, ou encore d'avoir des actions avec plus de deux acteurs.

Afin de pouvoir disposer d'une certaine variabilité de montages possibles, nous avons développé un modèle de génération de montage basé sur une évaluation générique des plans et des transitions disponibles selon un ensemble de critères. Ces critères se matérialisent par des fonctions de coût et sont basés sur des principes classiques de cinématographie comme le principe d'Hitchcock, qui postule qu'il faut montrer à l'image les actions de manière proportionnelle à leur importance [ST85], ou des règles de continuité comme le non-franchissement de la ligne d'intérêt (*cf fig. 2.2*).

Dans ce chapitre, nous introduisons d'abord un formalisme sur les éléments constituant un film. Nous détaillons ensuite le modèle de montage cinématographique que nous avons conçu. Enfin, nous présentons la manière dont la modèle a été mis en œuvre puis évalué.

3.1 Définitions

Un film est décomposé en séquences, une séquence correspondant à une partie de film se déroulant dans un lieu unique pendant un intervalle de temps continu. Une séquence est associée à :

- Un instant de départ t_{debut} et de fin t_{fin}
- Un lieu, i.e. un environnement 3D
- Un ensemble d'actions
- Une succession de plans

Actions

Une action a représente un évènement unique impliquant un objet (i.e. un acteur) $o_j(a)$ et ayant lieu entre $t_{debut}(a)$ et $t_{fin}(a)$. Une action du film impliquant deux acteurs ou plus (A parle à B) se matérialise par 2 actions distinctes dans le modèle. Par exemple : A parle à B et B écoute A. De plus, un acteur est toujours considéré par défaut comme impliqué dans l'action *Inactif*, c'est à dire l'action de « ne rien faire », de priorité $imp(Inactif) = 0$. Les

actions sont classés dans une des catégories $c(a)$ suivantes afin de réduire la complexité dûe au nombre d'actions différentes :

- Dialogue (*« Speaking »*) : l'action de parler à un interlocuteur proche.
- Réaction (*« Reacting »*) : les actions entrainant des mouvements de tête liés au fait de réagir à quelque chose (e.g. acquiescer, écouter, tourner la tête vers, regarder).
- Manipulation (*« Manipulating »*) : les actions entrainant des mouvements de mains et de bras liés à la manipulation d'objets proches (e.g. verser, manger, soulever).
- Geste (*« Gesturing »*) : les actions engendrant des mouvements de mains sans manipulation d'objets (e.g. faire des gestes en parlant).
- Déplacement (*« Moving »*) : les actions de déplacement (e.g. marcher, s'asseoir, se lever).
- Inaction (*« Idle »*) : action par défaut pour les acteurs n'ayant pas d'action particulière à un moment donné, ou une action n'ayant pas d'importance par rapport à l'histoire (e.g. action donnée aux figurants).

On associe enfin à chaque action une priorité $imp(a)$, qui permet de classer plusieurs actions en fonction de leur importance narrative.

Plans

Un plan p est une succession d'images filmées par une même caméra de manière continue entre l'instant $t_{debut}(p)$ et l'instant $t_{fin}(p)$. A chaque instant compris dans cet intervalle, le plan est décrit par un vecteur contenant les acteurs (objets d'une action a) à l'écran, ordonnés de gauche à droite. Le plan est décrit relativement à chacun de ces acteurs par :

- La position des yeux et du nez sur l'image. La fonction $centre(p, a, t)$ permet de récupérer la position du centre des yeux de l'acteur projetée à l'écran.
- La taille de plan $taille(p, a)$: *« close-up »*, *« close shot »*, *« medium shot »*, *« long shot »*, *« extreme long »*, obtenue à partir de l'écart entre le centre des yeux et nez.
- Le profil $profil(p, a)$: *face*, *trois-quarts*, *profil*, *trois-quarts de dos*, *dos*, ou obtenu par calcul à partir la différence de position entre les yeux, et de la taille de plan.

> Dans le cinéma classique, il est courant qu'une caméra soit mobile, à la fois en position, en orientation, mais aussi en zoom. Les plans dynamiques sont soumis à des règles sensiblement différentes des plans statiques. Dans une volonté de réduire la complexité du problème, nous n'utilisons pas de plans dynamiques dans notre modèle.

3.2 Processus de montage

Notre modèle procède au montage d'une séquence selon le processus suivant (*cf fig. 3.1*) :

1. Création d'un graphe de montage
2. Evaluation de chaque noeud du graphe
3. Calcul du montage par recherche de chemin

3.2.1 Graphe de montage

On se place dans le cadre du montage d'une scène connue à l'avance : on dispose ainsi d'un fichier rassemblant toutes les informations sur les actions de la scène.

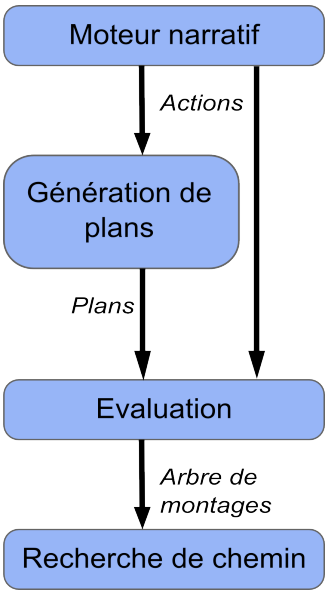


FIGURE 3.1 – Processus de montage de séquence

La première étape est d'obtenir une collection de plans filmant la scène. Notre système se voulant générique, la manière dont sont obtenus ces plans n'a pas d'incidence sur le calcul de montage, seule leur évaluation va compter. Les plans peuvent donc être obtenus par exemple par un placement manuel de caméras dans la scène (en plaçant une caméra et en indiquant pendant quel intervalle de temps celle-ci est disponible), ou encore par des procédés de génération automatique de plans par rapport aux actions de la scène. Dans notre implémentation, nous utilisons le système existant [LCL+10] (décrit dans la section 2.1.3) pour obtenir une suggestion de plan pour chaque volume sémantique associé à chaque action contenue dans le script. Ces plans ont alors une durée égale à celle de l'action, et sont étendus pendant un intervalle de temps Δ_{avance} avant le début et après la fin de l'action.

L'intervalle de temps pendant lequel dure la séquence que l'on veut obtenir est ensuite découpé en sections de durée $\Delta_{section}$. On obtient ainsi le graphe de montage en créant un nœud pour chaque plan disponible pendant une section, et en créant les arcs en reliant chaque nœud à l'ensemble des nœuds de la section suivante.

3.2.2 Évaluation de séquence

Pour pouvoir évaluer et comparer des plans, on définit des fonctions de coûts évaluant chacune le degré de violation d'un critère de composition d'image, et le coût d'un plan est alors la somme pondérée des évaluations de ces fonctions. En posant w_k le poids et ϕ_k la fonction de coût correspondant au k_{ieme} critère de qualité de plan, le coût C^P du plan i à l'instant t est alors :

$$C^P(i,t) = \sum_k w_k \phi_k(i,t)$$

De la même manière une transition est évaluée comme la somme pondérée de fonctions de coût correspondant à des évaluation de violations de règles de montage. En posant w_l le poids et ϕ_l la fonction de coût correspondant au l_{ieme} critère de qualité de transition, le coût C^T de

la transition du plan i au plan j à l'instant t est alors :

$$C^T(i,j,t) = \begin{cases} \sum_l w_l \phi_l(i,j,t) \text{ si } i \neq j \\ 0 \text{ sinon} \end{cases}$$

Le coût d'une séquence complète $seq : c \in [1,M], t \in [1,N]$ composée de N sections t de durée $\Delta_{section}$ et utilisant M caméras c est alors calculé par la somme suivante :

$$C(seq) = \sum_t \sum_k w_k C_k^S(c(t),t) + \sum_l w_l C_l^T(c(t),c(t+1),t)$$

Critères d'évaluations de plan

Cette section détaille les critères utilisés pour évaluer la qualité d'un plan. Rappelons que selon notre formalisme, une action a lieu entre $t_{debut}(a)$ et $t_{fin}(a)$, implique un objet unique $o(a)$ et est associée à une priorité $imp(a)$, et que dans le plan p à l'instant t, on connaît pour l'action a la taille de plan $taille(p,a,t)$, le profil $profil(p,a,t)$, et la position de l'objet de l'action $centre(p,a,t)$. On définit de plus la fonction δ qui détermine si l'objet i est à l'écran dans le plan p à l'instant t :

$$\delta(p,i,t) = \begin{cases} 1 \text{ si } i \text{ est visible dans } p \text{ à l'instant } t \\ 0 \text{ sinon} \end{cases}$$

Cadrage de l'action Ce critère ϕ^{ACTION} mesure si le cadrage utilisé respecte le principe d'Hitchcock, c'est à dire si la quantité d'information visible est proportionnelle à son importance. On discrétise le problème en considérant qu'à chaque type de cadrage de l'action a (i.e. la taille s et le profil pr de l'objet $o(a)$) correspond un coût $A[a, s \in tailles, pr \in profils]$. On ajoute un coût particulier correspondant au cas où l'objet de l'action n'apparaît pas à l'écran : $A[a, hors-champ]$. On dispose alors du terme

$$A(a,p,t) = \begin{cases} A[a, taille(p,a,t), profil(p,a,t)] & \text{si } \delta(p,o(a),t) = 1 \\ A[a, hors-champ] & \text{sinon} \end{cases}$$

Le critère de cadrage $\phi^{ACTION}(p,t)$ pour un plan p à l'instant t est alors calculé à partir de la somme des termes $A[a,p]$ pour chaque action dont l'intervalle comprend t, et des importances respectives de chaque action $imp(a)$.

$$\phi^{ACTION}(p,t) = \sum_{\substack{a \in actions,\ t_{debut}(a)<t, \\ t_{fin}(a)>t}} \frac{imp(a) \times A(a,p,t)}{Z_{ACTION}(t)}$$

Avec $Z_{ACTION}(t)$ terme de normalisation sur les importances des actions :$Z_{ACTION}(t) = \sum_{\substack{a \in actions,\ t_{debut}(a)<t, \\ t_{fin}(a)>t}} imp(a)$

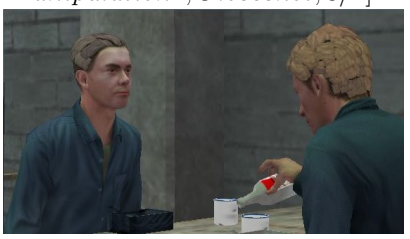
La quantité de valeurs $A[a,p]$ à paramétrer permet de définir précisément pour chaque action une hiérarchie représentant la préférence entre les différents cadrages.

Visibilité Il est parfaitement possible que certains plans reçus par le système présentent des occultations : un critère de qualité $\phi^{VISIBILITE}$ est donc défini pour pénaliser les plans qui présentent ce défaut. On parle d'occultation statique lorsqu'un élément du décor cache un objet important (e.g. un pilier entre la caméra et un acteur), et d'occultation dynamique lorsque c'est un objet non statique qui cache un objet important (e.g. un figurant entre

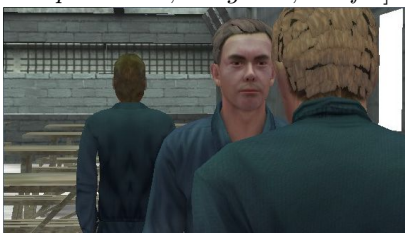
$A[\text{"}Manipulation\text{"}, Mediumshot, 3/4] = 0$

$A[\text{"}Manipulation\text{"}, Closeshot, 3/4] = 0.2$

$A[\text{"}Manipulation\text{"}, Longshot, Profil] = 0.5$



$A[\text{"}Manipulation\text{"}, Mediumshot, 3/4\ dos] = 0.7$



$A[\text{"}Manipulation\text{"}, Close-up, Dos] = 1$

$A[\text{"}Manipulation\text{"}, Close-up, 3/4] = 1$

FIGURE 3.2 – Plusieurs cadrages pour l'action « Smith verse du gin »de type « Manipulation »et des valeurs possibles pour les termes $A[a, p]$ correspondants.

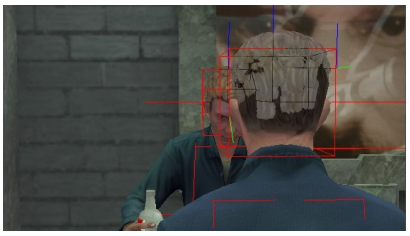
l'acteur principal et la caméra). Dans notre implémentation, il existe déjà un mécanisme dans le système existant gérant les occultations statiques. Les caméras présentant un tel défaut sont donc d'ores et déjà mises de côté, et on peut faire l'hypothèse que le seul cas à prévenir est celui des occultations dynamiques.

Le critère $\phi^{VISIBILITE}$ se base sur le pourcentage de surface occultée pour chaque objet important de la scène (i.e. impliqué dans une action). Pour évaluer ce terme tout en limitant la complexité, on utilise les projections des extrémités des boîtes englobantes des objets à l'écran. On vérifie que les boîtes englobantes des objets importants ne soient pas en superposition avec la projection d'une autre boîte englobante plus proche de la caméra. S'il y a superposition, le critère rend alors un coût proportionnel au pourcentage de surface occultée. Avec $boite(i)(p,t)$ la fonction retournant la surface de la boîte englobante d'un acteur i dans le plan p à l'instant t, le critère s'exprime de la manière suivante :

$$\phi^{VISIBILITE}(p,t) = \sum_{\substack{a \in actions,\ \delta(a,p,t)=1 \\ b \in objets,\ \delta(b,p,t)=1}} \frac{area(boite_{o(a)}(p,t)\ \cap\ boite_b(p,t))}{aire(boite_{o(a)}(p,t) \times Z_{VISIBILITE}(p,t)}$$

Avec $Z_{VISIBILITE}(p,t)$ terme de normalisation sur le nombre d'actions en cours à l'écran :$Z_{VISIBILITE}(p,t) = \sum_{\substack{a \in actions,\ t_{debut}(a)<t, \\ t_{fin}(a)>t}} \delta(p, o(a), t)$

Composition Le critère $\phi^{COMPOSITION}$ mesure l'équilibre de l'image. En photographie, les objets d'importance sont habituellement placés le long des lignes verticales des tiers ou du centre de l'image, pas sur les extrémités. De plus, si les objets d'importance sont placés dans



$\phi^{VISIBILITE} = 0.8723$

$\phi^{VISIBILITE} = 0$

FIGURE 3.3 – Gauche : mauvaise visibilité sur l'acteur du fond de l'image. Droite : cadrage sans problème de visibilité.

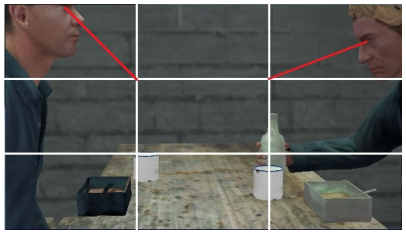
un plan sur une des ligne des tiers horizontale, alors le reste de la séquence va placer les objets sur la même ligne. C'est ce qu'on appelle la règle des tiers (*cf fig. 3.4*)

FIGURE 3.4 – Exemple de photographie respectant la règle des tiers.

Ce critère mesure donc la distance entre la position des objets d'importances (i.e. le centre des yeux pour un acteur) et les points d'intersection des lignes ci-dessus. De cette façon, un plan avec des personnages placés à une extrémité de l'écran sera pénalisé, ainsi qu'un plan changeant brusquement de ligne horizontale alors que le reste de la séquence est sur une autre ligne.

$$\phi^{COMPOSITION}(p,t) = \sum_{\substack{i \in objets, \\ j \in lignesH \cap lignesV}} \frac{\delta(p,i,t) \times |(centre(i),j)|}{Z_{COMPOSITION}(p,t)}$$

Avec $Z_{COMPOSITION}(p,t)$ terme de normalisation sur le nombre d'objets à l'écran :$Z_{COMPOSITION}(p,t) = \sum_{i \in objets} \delta(p,i,t)$



$\phi^{COMPOSITION} = 1$

$\phi^{COMPOSITION} = 0$

FIGURE 3.5 – Gauche : très mauvaise composition. Droite : bonne composition.

Lookroom, headroom Une des bonnes pratiques en cinématographie consiste à laisser suffisamment d'espace autour d'un acteur, en particulier dans la direction de son regard (lookroom), et au-dessus de ses yeux (headroom), en fonction de la taille de plan (cf partie

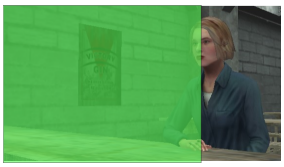
cinématographie). Le critère $\phi^{LOOKOOM}$ est calculé en projetant un point dans la direction du regard de l'acteur à une distance proportionnelle à la taille de plan. Si le point projeté sort de l'écran, alors on considère qu'il n'y a pas assez de lookroom. De la même manière, $\phi^{HEADROOM}$ est calculé en projetant un point dans la direction du haut de la tête de l'acteur.

$$\phi^{LOOKROOM}(p,t) = \sum_{i \in acteurs} \delta(p, centre(i,p,t) + \lambda_{LOOKROOM}(size(p,i,t)) \times regard_i(p,t), t)$$

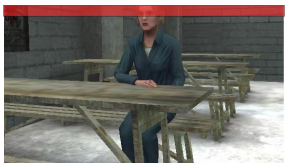
$$\phi^{HEADROOM}(p,t) = \sum_{i \in acteurs} \delta(p, centre(i,p,t) + \lambda_{HEADROOM}(size(p,i,t)) \times up_i(p,t), t)$$

Avec $regard_i$ vecteur unitaire de la direction du regard de l'acteur, up_i vecteur unitaire de la direction du haut de la tête de l'acteur, $\lambda_{LOOKROOM}(size)$ et $\lambda_{HEADROOM}(size)$ fonctions scalaires de la taille de plan, et $Z_{LOOKROOM}(p,t), Z_{HEADROOM}(p,t)$ termes de normalisation sur le nombre d'objets à l'écran :$Z_{LOOKROOM}(p,t) = Z_{HEADROOM}(p,t) = \sum_{i \in objets} \delta(p,i,t)$

$\phi^{LOOKROOM} = 1$



$\phi^{LOOKROOM} = 0$



$\phi^{HEADROOM} = 1$

$\phi^{HEADROOM} = 0$

FIGURE 3.6 – Gauche : rouge, trop peu de lookroom. Vert, suffisamment de lookroom. Droite : rouge, trop peu de headroom. Vert, suffisamment de headroom.

Critères d'évaluations de transition

Discontinuité de position Un changement brusque de position à l'écran d'un acteur ou d'un objet important peut perturber le spectateur. Le critère $\phi^{POSITION}$ est utilisé pour mesurer de telles variations de position. Une transition du plan p_1 au plan p_2 à l'instant t_{cut} se calcule de la manière suivante, avec ψ une fonction de la distance entre deux positions à l'écran (e.g. une fonction sigmoïde ou de Heaviside) :

$$\phi^{POSITION}(p_1,p_2,t_{cut}) = \sum_{\substack{i \in objets, \\ \delta(p_1,i,t_{cut})=\delta(p_2,i,t_{cut})=1}} \frac{\psi(|centre(p_1,i,t_{cut}) - centre(p_2,i,t_{cut}))}{Z_{POSITION}(p_1,p_2,t_{cut})}$$

Avec $Z_{POSITION}(p_1,p_2,t_{cut})$ terme de normalisation sur le nombre d'objets à l'écran à la fois sur p_1 et p_2 :$Z_{POSITION}(p_1,p_2,t_{cut}) = \sum_{i \in objets} \delta(p_2,i,t_{cut}) \times \delta(p_1,i,t_{cut})$

$\phi^{POSITION} = 0.3271$

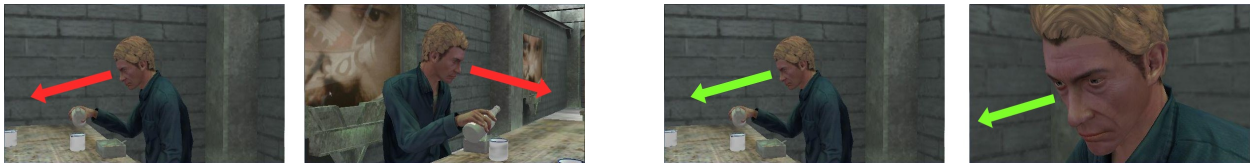
$\phi^{POSITION} = 0$

FIGURE 3.7 – Gauche : transition avec une forte variation de position d'un acteur. Droite : transition ne faisant pas trop varier la position de l'acteur à l'écran.

Discontinuité de regard De la même manière qu'un changement de position, une inversion de direction de regard peut pénaliser la compréhension d'une scène. Les transitions présentant des discontinuités de regard sont pénalisées par le critère ϕ^{REGARD} calculé de la manière suivante :

$$\phi^{REGARD}(p_1, p_2, t_{cut}) = \sum_{\substack{i \in acteurs, \\ \delta(p_1,i,t_{cut})=\delta(p_2,i,t_{cut})=1}} \frac{sign(regard_i(p_1, t_{cut})) - sign(regard_i(p_2, t_{cut}))}{Z_{REGARD}(p_1, p_2, t)}$$

Avec $Z_{REGARD}(p_1, p_2, t_{cut})$ terme de normalisation sur le nombre d'objets à l'écran à la fois sur p_1 et p_2 : $Z_{REGARD}(p_1, p_2, t_{cut}) = \sum_{i \in objets} \delta(p_2, i, t_{cut}) \times \delta(p_1, i, t_{cut})$



$\phi^{REGARD} = 1$ $\phi^{REGARD} = 0$

FIGURE 3.8 – Gauche : transition avec une inversion de direction de regard d'un acteur. Droite : transition conservant la direction de regard.

Discontinuité de mouvement Une autre discontinuité pénalisante est celle d'une inversion d'une direction de mouvement. Le calcul de ce terme $\phi^{MOUVEMENT}$ s'effectue de la même façon que ϕ^{REGARD}, mais en considérant le vecteur vitesse plutôt que le vecteur du regard :

$$\phi^{MOUVEMENT}(p_1, p_2, t_{cut}) = \sum_{\substack{i \in acteurs, \\ \delta(p_1,i,t_{cut})=\delta(p_2,i,t_{cut})=1}} \frac{sign(\frac{dcentre(p_1,i,t_{cut})}{dt}) - sign(\frac{dcentre(p_2,i,t_{cut})}{dt}))}{Z_{MOUVEMENT}(p_1, p_2, t_{cut})}$$

Avec $Z_{MOUVEMENT}(p_1, p_2, t_{cut})$ terme de normalisation sur le nombre d'objets à l'écran à la fois sur p_1 et p_2 : $Z_{MOUVEMENT}(p_1, p_2, t_{cut}) = \sum_{i \in objets} \delta(p_2, i, t_{cut}) * \delta(p_1, i, t_{cut})$

$\phi^{MOUVEMENT} = 1$ $\phi^{MOUVEMENT} = 0$

FIGURE 3.9 – transition avec une inversion de direction de mouvement. Droite : transition conservant la direction de mouvement

Jump cut Une transition entre deux plans très similaires (i.e. deux caméras avec un angle et une position très similaires) perturbe donne une impression d'ellipse temporelle : c'est un *jump cut*. Une règle classique en cinématographie établit qu'un changement de taille de plan ou un écart d'au moins trente degrés entre deux plans successifs de la même taille permet d'assurer qu'il n'y a pas de *jump cut*.

On définit donc un critère $\phi^{JUMPCUT}$ suivant cette règle :

$$\phi^{JUMPCUT}(p_1, p_2, t_{cut}) = \begin{cases} 0 & \text{si } 30deg(p_1, p_2, t_{cut}, i) = 0 \text{ ou } size(p_1, i, t_{cut}) \neq size(p_2, i, t_{cut}), \\ & \forall i \in acteurs,\ \delta(p_1, i, t_{cut}) = \delta(p_2, i, t_{cut}) = 1 \\ 1 & \text{sinon} \end{cases}$$

Avec $30deg(p_1, p_2, t_{cut}, i)$ une fonction détectant si l'écart entre les deux plans p_1 et p_2 par rapport à l'acteur i à l'instant t_{cut} est d'au moins 30 degrés.

$\phi^{JUMPCUT} = 1$ $\phi^{JUMPCUT} = 0$

FIGURE 3.10 – Gauche : la transition ne change pas de taille significative (haut) et la différence d'angle est inférieure à 30 degrés (bas). Droite : deux transitions sans jumpcut.

Changement trop brusque de taille de plan

Inversement, une transition entre deux plans de taille très différentes (e.g. un « long shot »vers un « close-up ») peut déstabiliser le spectateur.

$$\phi^{MAXCHG}(p_1, p_2, t_{cut}) = \begin{cases} 0 & \text{si } \exists i \in acteurs,\ \delta(p_1, i, t_{cut}) = \delta(p_2, i, t_{cut}) = 1 \\ & size(p_1, i, t_{cut}) - size(p_2, i, t_{cut}) < 2 \\ 1 & \text{sinon} \end{cases}$$

$\phi^{MAXCHG} = 1$ $\phi^{MAXCHG} = 0$

FIGURE 3.11 – Gauche : transition brutale entre un « extreme long shot »et un « close shot ». Droite : transition douce entre un « extreme long shot »et un « medium shot ».

Franchissement de la ligne d'intérêt La ligne d'intérêt est une ligne virtuelle que l'on peut tracer dans la scène entre deux objets importants : cela peut-être un acteur et son interlocuteur, ou un acteur et la personne qu'il regarde. Une fois la caméra placée d'un côté de cette ligne, franchir la ligne d'intérêt va introduire des discontinuités. On introduit donc un critère ϕ^{LIGNE} qui détecte ce franchissement :

$$\phi^{LIGNE}(p_1, p_2, t_{cut}) = \sum_{\substack{i \in acteurs, \\ \delta(p_1, i, t_{cut}) = \delta(p_2, i, t_{cut}) = 1}} \frac{sign(centre(p_1, i, t) - ligne(p_1, i, t_{cut})) - sign(centre(p_2, i, t_{cut}) - ligne(p_2, i, t_{cut})}{Z_{LIGNE}(p_1, p_2, t_{cut})}$$

Avec $ligne(p,i,t)$ fonction rendant la position de l'objet d'intérêt de l'acteur i à t dans le plan p (e.g. si l'acteur i regarde un acteur j, alors $ligne(p,i,t) = centre(p,j,t)$), et $Z_{LIGNE}(p_1,p_2,t_{cut})$ terme de normalisation sur le nombre d'objets à l'écran à la fois sur p_1 et p_2 : $Z_{LIGNE}(p_1,p_2,t_{cut}) = \sum_{i \in objets} \delta(p_2,i,t_{cut}) \times \delta(p_1,i,t_{cut})$

$\phi^{LIGNE} = 1$

$\phi^{LIGNE} = 0$

FIGURE 3.12 – Gauche : la transition franchit la ligne d'intérêt, on passe du côté A au côté B. Droite : on reste du bon côté de la ligne d'intérêt.

Critère d'évaluation de rythme

Le rythme entre coupures est un paramètre important du style de montage. Nous introduisons ainsi un critère ϕ^{RYTHME} basé sur l'hypothèse de Barry Salt [Sal03] que les distributions de durées de plans se rapprochent d'une loi lognormale.

Le critère ϕ^{RYTHME} mesure alors l'écart entre une durée de plan Δt et une distribution lognormale paramétrée par μ et σ, respectivement la moyenne et l'écart type du logarithme des durées de plans.

$$\phi^{RYTHME}(\Delta t) = \frac{(log(\Delta t) - \mu)^2}{2\sigma^2}$$

3.2.3 Recherche de chemin

Dans un premier temps, l'utilisation d'un algorithme de recherche de solution optimale a été envisagée. Pour que des solutions soient calculables dans un temps raisonnable, l'algorithme choisi fut une variante de l'algorithme A* proposée par Korf dans [Kor85]. Les résultats de l'implémentation de cet algorithme se sont révélés décevants, car le temps de calcul pour calculer un montage optimal étaient trop longs. Nous avons alors décidé de mettre au point un algorithme permettant un calcul plus rapide de solution sous-optimale.

La recherche de chemin dans l'arbre de montage s'effectue en calculant le chemin minimisant une estimation de la somme des coûts des plans et transitions sur une fenêtre de longueur $N_{window} \times \Delta_{section}$, les coûts étant pondérées par leur proximité à l'instant courant :

$$C(p,t_1) = \sum_{i \in [0..N_{window}]} (\sum_k w_k C_k^S(s(t_1 + i\Delta_{section}), t_1 + i\Delta_{section})$$

$$+ \sum_l w_l C_l^T(s(t_1 + i\Delta_{section}), s(t_1 + (i+1)\Delta_{section}), t_1 + i * \Delta_{section}))/Z(N_{section}, i)$$

Avec $Z(N_{section}, i) = \frac{N_{section} - i}{\sum_{i \in [0..N_{section}]} N_{section} - i}$ terme de pondération.

Le premier plan choisi est ainsi celui minimisant le coût lors des premiers instants de la séquence. A chaque nouvelle section, on considère si la meilleure solution locale à la fenêtre de calcul est de rester dans le plan courant ou de faire une transition vers un autre plan, et si oui, on calcule quel est le meilleur moment pour faire cette transition. Si la transition a

lieu dans la section courante, alors elle est effectivement choisie. Sinon, on décale la fenêtre de calcul d'une section en avant.

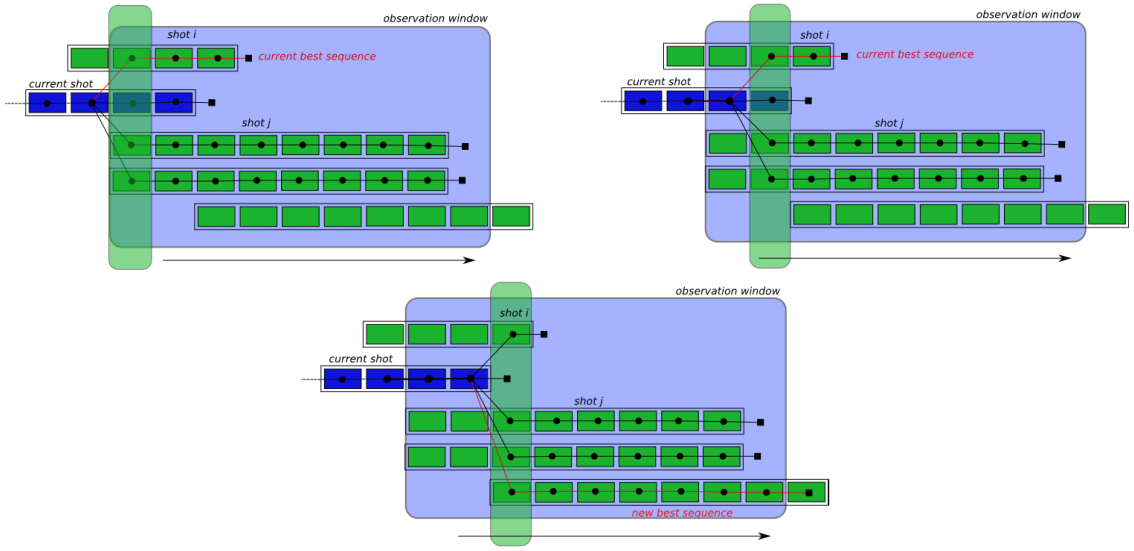


FIGURE 3.13 – Illustration du fonctionnement de l'algorithme de recherche de chemin.

3.3 Mise en oeuvre du modèle

3.3.1 Prototype de montage

Le système que nous étendons reproduit dans un environnement virtuel une scène du film *1984* de Michael Radford. Nous disposons ainsi d'une scène 3D et d'animations complètement modélisées, ainsi que d'un fichier cataloguant les actions correspondant à ces animations ayant lieu dans la scène. Le catalogue des actions et les animations couvrent une durée totale de trois minutes.

FIGURE 3.14 – Scène originale du film *1984*, et l'environnement virtuel correspondant.

Pour interagir avec le système de cinématographie, nous avons fait évoluer l'interface existante. Ainsi, à la vue de la caméra active dans la scène, nous avons ajouté un module préexistant permettant de connaître les actions ayant cours et leurs priorités. Nous avons développé un tableau de bord permettant de suivre en temps réel l'évolution des coûts des différents critères d'évaluation de plans et de transitions.

Différents modes d'interaction sont disponibles :

- Mode manuel : le choix des plans et des transitions est ici intégralement laissé à l'utilisateur. Il peut à tout moment interrompre le plan courant pour faire une transition,

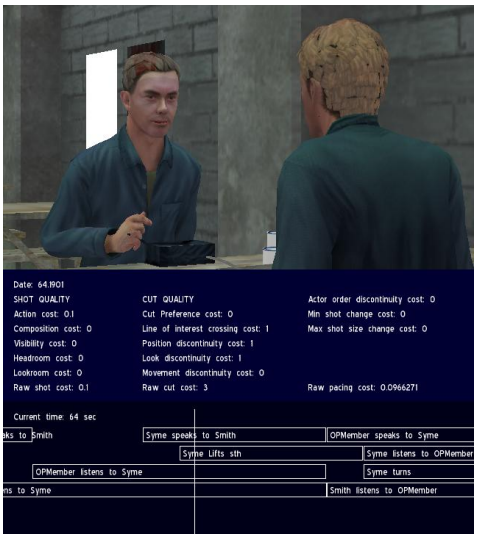


FIGURE 3.15 –

ou revenir en arrière pour modifier ses choix. Le montage est finalement sauvegardé en conservant les données de chaque plan : intervalle de temps, position et orientation de la caméra.

- Mode automatique : le choix des plans et des transitions est effectué par l'algorithme de recherche de chemin. Le montage est sauvegardé de la même manière.
- Mode correction : une séquence sauvegardée ou calculée par l'algorithme de recherche de chemin est visualisée par un utilisateur, qui peut indiquer quand des plans ou des transitions sont incorrects. L'utilisateur choisit alors le plan de remplacement et la fin de la séquence est recalculée à partir de cet instant. On sauvegarde alors la séquence finale ainsi que les informations sur les plans et les transitions qui ont été remplacées.
- Mode lecture : ce mode permet de rejouer une séquence sauvegardée.

3.3.2 Résultats intermédiaires

Une première évaluation des résultats a été réalisée pour valider que le modèle était suffisant pour produire des montages de style variés. Pour cela, nous avons paramétré le système manuellement de différentes façons pour produire des montages de styles différentes :

- Variations de rythme en modifiant les paramètres de la distribution lognormale.
- Variations des termes du critère de cadrage $A[a, p]$ pour privilégier des plans plutôt courts, moyen ou longs.

Variabilité

Nous avons d'abord vérifié que le système était en mesure de produire des montages aux rythmes variés. Pour cela nous avons produit 5 séquences sur l'intégralité des trois minutes de la scène, en faisant varier le ratio entre le poids associé au critère de rythme par rapport à la somme des poids, et en faisant varier les paramètres de la loi lognormale du critère de rythme.

Une première validation expérimentale a consisté à réaliser un montage en ne prenant pas en compte les critères autre que le rythme. Tous les plans durent alors le nombre de sections minimisant le critère de rythme (i.e. le multiple de $\Delta_{section}$ le plus proche de $\exp(\mu)$).

Nous avons ensuite fait varier les paramètres de la loi lognormale que l'on veut approcher, en utilisant un poids du critère de rythme égal au cinquième de la somme des poids. En augmentant le mode $\exp(\mu)$ de cette loi, on obtient alors effectivement une augmentation de la durée moyenne des plans calculés. L'écart-type reste fort, car la durée d'un plan dépend aussi des actions en cours : si l'action en cours change, il y a de grandes chances que le meilleur cadrage pour filmer la nouvelle action ne soit pas celui en cours. Le coût d'effectuer une transition même si le critère de rythme n'est pas minimal peut alors devenir inférieur au coût de garder le plan actuel.

Rythme	$exp(\mu)$	σ	$\frac{w_{RYTHME}}{\sum w}$	# plans	durée moyenne	écart-type
Rapide	3s	1.02	0.2	63	2.85s	2.11
Moyen	4.3s	1.02	0.2	49	3.87s	1.45
Moyen	4.3s	1.5	0.2	43	4.18s	2.77
Lent	5s	1.02	0.2	42	4.24s	2.48
Moyen	4.3s	1.02	1	40	4.5s	0

TABLE 3.1 – Génération de séquences d'une durée totale de 180 secondes avec variations de paramètres de rythme. La dernière ligne correspond à un calcul de chemin en prenant uniquement en compte le coût du rythme.

Nous avons ensuite fait varier les paramètres $A[type, taille, profil]$ du critère de cadrage pour essayer d'obtenir des montages de style différents. Les poids du système ont été réglés pour que le critère de cadrage soit important mais ne domine pas les autres (ici, $w_{CADRAGE}$ est égal à un cinquième de la somme des poids). Nous avons d'abord essayé d'obtenir des montages avec une forte préférence pour des plans d'une certaine taille, sans se préoccuper du profil :

- Plans larges : $A[type, long\ shot, *] = A[type, extreme\ long\ shot, *] = 0$, $A[type, medium\ shot, *] = 0.5$, $A[type, taille, *] = 1$ sinon.
- Plans moyens : $A[type, medium\ shot, *] = 0$, $A[type, close\ shot, *] = A[type, long\ shot, *] = 0.5$, $A[type, taille, *] = 1$ sinon.
- Plans rapprochés : $A[type, close\ shot, *] = A[type, closeup, *] = 0$, $A[type, medium\ shot, *] = 0.5$, $A[type, taille, *] = 1$ sinon.

Nous avons ensuite paramétré manuellement les termes $A[type, taille, profil]$ en essayant de raisonner sur quels cadrages sont les plus adaptés pour un type d'action. Par exemple, pour une action de type « Manipulation », il est important de voir les mains de l'acteur. Nous avons donc fortement pénalisé les plans de dos et les plans trop rapprochés (e.g. $A[Manipulation, *, Dos] = A[Manipulation, *, 34\ Dos] = 1$ et $A[Manipulation, closeup, *] = 1$). La vidéo produite a servi de support pour la présentation d'un poster sur le modèle à la conférence *Symposium on Computer Animation*, et est disponible à l'adresse `http://vimeo.com/27597860`.

Les résultats obtenus prouvent qu'on peut effectivement faire varier les choix de cadrages, et donc le style de montage, par le paramétrage du modèle.

Evaluation des critères du modèle

L'analyse des séquences produites par le système a montré que des erreurs pouvaient être commises par le système sans que des critères ne soient définis pour les détecter, ce qui nous a amené à modifier certains critères et à enrichir le modèle :

- Des transitions pouvaient donner une impression de relation n'ayant pas lieu d'être entre des personnages, par des jeux de regards accidentels d'un plan à l'autre.

Montage	w_{ACTION}	$\sum w_{SHOT}$	$\sum w_{CUT}$	# close shots	# medium shots	# long shots	# acteurs moyen onscreen	# plans
Plans larges « long shots »	1	2	3	0,5	1,5	53,5	1,66	56
Plans moyens « medium shots »	1	2	3	6.5	33	9.5	1.367	49
Plans rapprochés « close shots »	1	2	3	30	1.5	1.5	1.09	33
Plans équilibrés	1	2	3	13.83	10.43	23.73	1.35	48

TABLE 3.2 – Génération de séquences d'une durée totale de 180 secondes avec variations de paramètres de cadrage. Note : lorsqu'un plan contient n acteurs pour lesquels le plan est de taille différente, celui-ci compte comme $1/n$ plan pour chaque taille différente

FIGURE 3.16 – Transition pouvant faire croire que les deux acteurs du premier plan regardent l'actrice du deuxième plan, alors que ce n'est pas le cas.

– Le modèle était insuffisant sur la gestion des transitions sur les actions de mouvement : des transitions pouvaient donner l'impression qu'un personnage revenait en arrière.

FIGURE 3.17 – Les deux premières sont images issues du même plan. La transition donne l'impression avec la troisième image que l'acteur en déplacement revient en arrière.

3.3.3 Perspectives

Compte-tenu des limites en temps d'un stage de Master, nous avons choisi de seulement modéliser les caractéristiques qui nous semblaient les plus essentielles à l'évaluation des plans et des transitions, afin de conserver du temps pour la mise en place de méthodes d'apprentissage des paramètres du modèle. Une première évaluation des résultats a confirmé que le modèle remplissait les objectifs que nous nous étions fixés, c'est à dire permettre des variations dans les styles de montage, notamment sur les types de cadrages et de transitions.

Cependant, un des avantages majeurs de notre modèle est que l'évaluation des plans et des transitions repose sur des critères totalement indépendants. Par construction, il est donc très facile d'enrichir le modèle avec de nouveaux critères permettant de varier encore plus les styles de montages. Par exemple, des extensions naturelles pourraient être les suivantes :

– Ajouter un critère indiquant si un plan est horizontal, ou en « plongée »ou « contre-plongée ». Ces plans particuliers constituent un élément stylistique important permet-

tant d'introduire des effets de dominance entre acteurs.

- Introduire des plans dynamiques et des critères propres à ces plans : certains styles utiliseraient alors plutôt des caméras statiques, d'autres plutôt dynamiques, ce qui ferait augmenterait encore les possibilités de variations stylistiques. De plus, les caméras dynamiques peuvent se révéler utiles pour filmer des actions de déplacement pendant une longue durée, à l'inverse des plans statiques où les acteurs ont forcément tendance à sortir de l'écran.
- Faire varier la distribution du critère du rythme en fonction du type de plan : par exemple, un plan large peut contenir beaucoup d'informations (e.g. beaucoup d'acteurs à l'écran) et on peut vouloir le garder à l'écran plus longtemps qu'un plan rapproché sur un acteur, qui est plus « rapide à lire ».

Chapitre 4

Apprentissage des paramètres

Les résultats intermédiaires produits montrent que le système permet d'ores et déjà d'obtenir des séquences de styles variés respectant les règles classiques de cinématographie. Toutefois, le grand nombre de paramètres fait que leur ajustement manuel est une tâche difficile. Pour procéder à cet ajustement, nous comptons étudier la pertinence de l'utilisation d'une technique d'apprentissage automatique. Nous définissons ici le point de vue adopté pour ce problème d'apprentissage, puis nous présentons les méthodes que nous avons retenues.

4.1 Cadre du problème

4.1.1 Formalisation

Nous formalisons le problème de la manière suivante. Comme noté précédemment, une séquence exemple est composée de N sections de durée $\Delta_{section}$. Chacune de ces sections correspond à un exemple de plan, sous la forme (x_i, y_i), avec x_i les données sur les actions (*cf* section 3.1, § *Actions*) pendant la section i, et y_i les données sur le plan choisi pendant la section i (*cf* section 3.1, § *Plans*). Un exemple de transition a alors la forme $(x_i, x_{i+1}, y_i, y_{i+1})$.

L'objectif est de trouver, à partir d'une séquence exemple (i.e. N paires (x_i, y_i) et M tuples $(x_i, x_{i+1}, y_i, y_{i+1})$) un jeu de poids w_k associés aux fonctions de coût ϕ_k tel que la séquence calculée automatiquement par recherche de chemin choisisse les mêmes plans y_i que ceux de l'exemple.

4.1.2 Approche proposée

L'apprentissage du jeu de paramètres s'opère de la manière suivante :

Phase 1 : Collecte d'exemples Un utilisateur réalise N séquences en utilisant le prototype en mode manuel, séquences qui sont sauvegardées comme exemples. Ces séquences n'ont pas à être toutes du même style, l'utilisateur est libre dans ses choix. Pour chacune de ses séquences, on extrait les paramètres de la fonction de rythme (μ et σ) et les valeurs de la fonction de cadrage en construisant un histogramme des occurrences de chaque type de cadrage par rapport à chaque type d'action dans la scène. Plus un type de plan aura été utilisé, plus le terme $A[a, p]$ correspondant sera faible.

Phase 2 : Séparation des ensembles d'exemples On sépare l'ensemble d'apprentissage en deux ensembles, un ensemble d'apprentissage et un ensemble de validation.

Phase 3 : Apprentissage des paramètres Les poids du système sont ajustés par une des méthodes d'apprentissage présentées dans la section suivante.

Phase 4 : Évaluation des résultats En utilisant le jeu de poids obtenu à la fin de la phase 3, on calcule les montage optimaux correspondant aux exemples des séquences de l'ensemble de validation. On évalue alors les résultats en calculant :

- Le nombre d'images de la séquence calculée identiques à celles de la séquence originale
- Le nombre d'images de la séquence calculée étiquetées correctes
- Le nombre d'images de la séquence calculée étiquetées non correctes

4.2 Méthodes d'apprentissage retenues

La phase 3 de la démarche d'apprentissage correspond au calcul des poids des différentes fonctions de coût. Pour effectuer ce paramétrage, nous allons comparer deux méthodes : une méthode dérivée de l'algorithme classique du perceptron (réseau de neurones), et une analyse linéaire discriminante (méthode statistique).

4.2.1 Perceptron

Principe

Le perceptron est le réseau de neurones le plus simple possible, c'est à dire ne comportant qu'un neurone. Dans [Col02], Collins s'inspire du principe de l'algorithme classique du perceptron pour proposer une méthode de classification dans le cadre de séquences. Nous proposons d'adapter cette méthode pour l'apprentissage des paramètres de notre système. Le principe est le suivant : on réalise M passes sur l'ensemble d'apprentissage. A chaque passe et pour chaque exemple $(x_i, y_i), i \in N$, on calcule la solution optimale $(x_i, z_i), i \in N$, c'est à dire dans notre cas la séquence calculée automatiquement par le système. Si la séquence calculée est différente de l'original, alors on ajuste chaque poids w_k du système proportionnellement à l'écart entre l'évaluation du critère de plan ϕ_k sur tous les plans de l'exemple calculé et de l'exemple original :

$$w_k = w_k + \sum_{i \in sections} \phi_k(x_i, y_i) - \phi_k(x_i, z_i)$$

On procède de la même manière pour les transitions :

$$w_l = w_l + \sum_{i \in sections, y_i \neq y_{i+1}} \phi_l(x_i, x_{i+1}, y_i, y_{i+1}) - \phi_l(x_i, x_{i+1}, z_i, z_{i+1})$$

A la fin de cette étape, on choisit le jeu de paramètres qui a minimisé le nombre d'erreurs.

Application au problème

Pour adapter cette méthode à l'apprentissage des paramètres de notre système, nous avons adapté le formalisme de la manière suivante :

- Dans le cas de l'étiquetage grammatical d'une phrase comprenant n mots, un exemple (x_i, y_i) est constitué d'un mot x_i et de l'étiquette y_i. Les critères utilisés pour déterminer quelle est l'étiquette la plus adaptée à un mot sont soit des critères basés sur le mot en court, ou sur les deux mots précédents.
- Dans notre cas, on considère que les x_i sont les descripteurs des actions de la scène, et les y_i sont les descripteurs des plans choisis. Les critères de plans n'utilisent que des données du plan courant, et la plupart des critères de transitions utilisent des données du plan courant et des plans de la section précédente. En revanche, le critère de rythme ϕ^{RYTHME} évalue le nombre de sections pendant lesquelles on est resté dans le même plan.

Lors de la phase d'adaptation des poids, la méthode de Collins implique que l'on calcule la solution optimale, c'est à dire la séquence de plans ayant le meilleur score. Dans notre implémentation, l'algorithme de recherche de chemin implémenté dans le système rend une solution optimale localement mais pas forcément globalement. Le montage est avant tout un problème séquentiel plutôt que global, dépendant en particulier des conditions initiales, c'est à dire du premier plan choisi. Nous faisons ainsi l'hypothèse qu'une optimisation locale est suffisante pour calculer la solution optimale, si les conditions initiales sont les mêmes dans l'exemple et dans la séquence calculée. Lors du calcul de meilleur montage pour l'apprentissage, nous indiquons alors au système de quel côté de la scène filmer la première action, le reste des décisions se faisant automatiquement.

4.2.2 Analyse linéaire discriminante

Principe

On classe chaque exemple de plan (x_i, y_i) dans une classe c_i, par exemple « bon »ou « mauvais »plan (idem pour les transitions). En se plaçant dans l'espace des critères (i.e. chaque dimension de cet espace représente un ϕ_k), un exemple de plan (x_i, y_i) est alors représenté par le point $(\phi_0(x_i, y_i), ..., \phi_n(x_i, y_i))$. L'analyse linéaire discriminante est une méthode de réduction de dimensions dans l'espace des critères : elle permet de trouver un axe (i.e. une combinaison linéaire des critères ϕ_k, c'est à dire un jeu de poids w_k) séparant de manière optimale les deux classes dans cet espace.

Une fois cet axe trouvé, on peut alors construire un classifieur qui permet, à partir d'un nouvel exemple (x_i, y_i) et de ses évaluations ϕ, de calculer quelle est la classe à laquelle appartient le plus probablement l'exemple (x_i, y_i).

Application au problème

On sépare les séquences de montages obtenues à la fin de la phase 2 en regroupant d'un côté les bons (calculés et conservés, ou choisis par l'utilisateur pour corriger un mauvais plan) et d'un autre coté les mauvais plans (remplacés par l'utilisateur). La même étape est faite pour les transitions qu'on sépare en bonnes et mauvaises transitions. On peut alors réaliser une analyse linéaire discriminante, et on obtient un jeu de poids constituant une séparation linéaire optimale entre les classes des bons et mauvais exemples de plans, et une séparation linéaire entre les classes des bons et mauvais exemples de transitions.

4.3 Validation des résultats

4.3.1 Evaluation des méthodes

Pour valider les résultats des méthodes d'apprentissage, on sépare les exemples en un ensemble d'apprentissage et un ensemble de validation. L'apprentissage des poids est alors réalisé sur le premier ensemble, et on peut déjà évaluer les résultats en analysant, pour la méthode du perceptron, l'évolution du nombre d'erreurs au fur et à mesure des passes faites sur l'ensemble d'exemples. Pour l'analyse linéaire discriminante, on peut analyser les matrices de confusions (c'est à dire les exemples pour lesquels le jeu de poids se trompe de classe) sur l'ensemble d'apprentissage.

On utilise alors le modèle paramétré par les méthodes précédentes pour calculer les séquences optimales sur l'ensemble de validation. L'utilisateur procède alors à une annotation manuelle sur les plans et les transitions du montage calculé différents du montage original, en

annotant si un plan ou une transition sont corrects ou incorrects. On évalue les résultats en analysant :

- Le nombre de sections où le plan calculé est le même que le plan de l'exemple.
- Le nombre de sections où le plan calculé est différent de l'exemple mais est noté comme correct par l'utilisateur.
- Le nombre de sections où le plan calculé est différent et est jugé mauvais.

4.3.2 Evaluation des séquences produites

Une fois les méthodes d'apprentissage mises en place, on dispose de plusieurs séquences :

- La séquence originale du film *1984*
- La reproduction manuelle de la séquence originale du film *1984* dans l'environnement 3D
- La séquence calculée par le système existant par un système d'idiomes
- La séquence calculée par le système paramétré manuellement avant validation
- La séquence calculée par le système paramétré manuellement après correction du modèle
- La séquence calculée par le système paramétré par la méthode du perceptron
- La séquence calculée par le système paramétré par analyse linéaire discriminante

On peut alors mener une étude perceptive des séquences produites en séparant la séquence en sections courtes (e.g. 30s). On alors en premier la séquence originale du film *1984*, puis les autres séquences. On pose ensuite les questions suivantes pour chaque séquence à un ensemble de spectateurs :

- La séquence de plans permet-elle de comprendre la situation (les actions) ?
- La qualité des plans est-elle bonne ?
- La qualité des transitions est-elle bonne ?
- La séquence vous a-t-elle semblée proche de la séquence originale ?

4.4 Travaux en cours

A la date de rédaction de ce rapport, il reste un mois et demi avant la fin du stage. Sur cette période, nous avons prévu de terminer les travaux entamés sur l'apprentissage des paramètres. Plus précisément, des modules ont déjà été implémentés permettant de réaliser et de sauvegarder des séquences exemples manuellement, et d'annoter les plans et transitions d'une séquence en tant qu'exemples « corrects »ou « incorrects ». L'analyse linéaire discriminante est une méthode disponible dans des outils de référence comme Matlab. Il ne reste ainsi qu'à mettre en place l'algorithme d'adaptation des poids par la méthode de Collins, et tous les outils auront été rassemblés pour procéder à l'apprentissage.

Conclusion

L'objectif de ce travail était de mettre au point un modèle de montage cinématographique permettant des variabilités dans les styles de montages produits et d'évaluer l'apport d'une méthode d'apprentissage pour le paramétrage du modèle.

Dans ce but, nous avons représenté le processus du montage comme celui d'une recherche de chemin dans un graphe de montage. Le système existant fournit à chaque instant de la scène une collection de plans devenant les nœuds du graphe, et les arcs entre les nœuds sont les transitions. Aux plans et aux transitions sont associés des coûts, calculés à partir de critères d'évaluations modélisant des conventions et règles cinématographiques. On peut alors calculer le coût d'une séquence comme la somme pondérée de ces critères. Une première évaluation des résultats produits par le modèle paramétré manuellement à été réalisée et a validé la possibilité d'obtenir des résultats corrects et variés, tout en nous poussant à enrichir le modèle sur certains aspects. Enfin, un cadre d'étude pour l'apprentissage du modèle a été proposé et des méthodes ont été sélectionnées. Les travaux en cours portent sur la mise en place des outils nécessaires à l'utilisation de ces méthodes dans le système existant.

Le sujet que nous avons abordé est très vaste, et nous pouvons d'ores et déjà dresser quelques perspectives sur les orientations possibles de travaux futurs.

Premièrement, le modèle que nous avons conçu ne se veut pas exhaustif et pourrait être enrichi pour décrire et évaluer plus précisément les plans et les transitions. On peut par exemple citer l'utilisation des plans en plongée ou en contre-plongée, ou l'utilisation de caméras dynamiques.

Ensuite, les exemples que nous utilisons sont réalisés manuellement dans le cadre de notre système, mais on pourrait envisager un apprentissage à partir de films réels. On procéderait alors à une annotation sur chaque plan d'un film exemple des positions des yeux et du nez des acteurs, pour en déduire la taille du plan et le profil, et des actions. Cela poserait le problème de la généralisation à une scène de connaissances apprises sur des scènes aux actions et géométries différentes.

On pourrait alors essayer d'apprendre le style de montage de genres particuliers, comme le montage d'une sitcom, ou le montage d'un jeu télévisé. Si les résultats se révélaient probants, cela pourrait donner des indications sur les caractéristiques propres au montage de ces genres.

Bibliographie

[Ari76] Daniel Arijon. *Grammaire Du Langage Filmé.* Editions Dujarric, deuxième edition, 1976.

[Bli88] Jim Blinn. Where Am I? What Am I Looking At? *Computer Graphics and Applications, IEEE*, 8(4):76–81, 1988.

[CAH+96] D.B. Christianson, S.E. Anderson, L.W. He, D.H. Salesin, D.S. Weld, and M.F. Cohen. Declarative camera control for automatic cinematography. In *Proceedings of the National Conference on Artificial Intelligence*, pages 148–155. Citeseer, 1996.

[Col02] Michael Collins. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 1–8, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[CON08] Marc Christie, Patrick Olivier, and Jean-Marie Normand. Camera control in computer graphics. *Comput. Graph. Forum*, 27(8):2197–2218, 2008.

[dLPd+09] Edirlei E. S. de Lima, Cesar T. Pozzer, Marcos C. d'Ornellas, Angelo E. M. Ciarlini, Bruno Feijó, and Antonio L. Furtado. Virtual cinematography director for interactive storytelling. In *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*, ACE '09, pages 263–270, New York, NY, USA, 2009. ACM.

[EN08] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *KDD*, pages 213–220, 2008.

[ER07] David K. Elson and Mark O. Riedl. A lightweight intelligent virtual cinematography system for machinima production. In *AIIDE*, pages 8–13, 2007.

[Haw04] Brian Hawkins. *Real-Time Cinematography for Games (Game Development Series).* Charles River Media, Inc., Rockland, MA, USA, 2004.

[HCS96] Li-wei He, Michael F. Cohen, and David H. Salesin. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 217–224, New York, NY, USA, 1996. ACM.

[KM05] James Kneafsey and Hugh Mccabe. Camerabots: Cinematography for games with non-player characters as camera operators. In *DIGRA Conf.*, 2005.

[Kor85] Richard E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 27:97–109, 1985.

[LCL+10] Christophe Lino, Marc Christie, Fabrice Lamarche, Guy Schofield, and Patrick Olivier. A Real-time Cinematography System for Interactive 3D Environments. In *2010 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2010.

[LP10] Seong Jae Lee and Zoran Popovic. Learning behavior styles with inverse reinforcement learning. *ACM Trans. Graph.*, 29(4), 2010.

[Mit97] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[MSU03] Yuya Matsuo, Kimiaki Shirahama, and Kuniaki Uehara. Video data mining : Extracting cinematic rules from movie. In *SIGKDD*, 2003.

[NHB04] Ram Nevatia, Jerry Hobbs, and Bob Bolles. An ontology for video event representation. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 7 - Volume 07*, pages 119–, Washington, DC, USA, 2004. IEEE Computer Society.

[PD03] Jonathan H Pickering and Yo Dd. Intelligent Camera Planning for Computer Graphics, 2003.

[PMC+10] Erick B. Passos, Anselmo Montenegro, Esteban W. G. Clua, Cezar Pozzer, and Vinicius Azevedo. Neuronal editor agent for scene cutting in game cinematography. *Comput. Entertain.*, 7:57:1–57:17, January 2010.

[Ron09] Rémi Ronfard. Automated cinematographic editing tool, 2009. International Patent Application, Xtranormal Technologies.

[Sal03] B. Salt. *Film Style and Technology: History and Analysis (2nd edition)*. Starword, 2003.

[ST85] Helen G. Scott and Francois Truffaut. *Hitchcock-Truffaut (Revised Edition)*. Simon and Schuster, 1985.

[TB09a] Roy Thompson and Christopher Bowen. *Grammar of the Edit*. Focal Press, second edition, 2009.

[TB09b] Roy Thompson and Christopher Bowen. *Grammar of the Shot*. Focal Press, second edition, 2009.