HAL
open science

Contrôle multidimensionnel interactif et application à l'édition de contenus cinématographiques virtuels

Charles Périn

To cite this version:

Charles Périn. Contrôle multidimensionnel interactif et application à l'édition de contenus cinématographiques virtuels. Synthèse d'image et réalité virtuelle [cs.GR]. 2011. dumas-00636777

HAL Id: dumas-00636777

https://dumas.ccsd.cnrs.fr/dumas-00636777

Submitted on 28 Oct 2011

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

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

Rapport de stage

Contrôle multidimensionnel interactif et application à l'édition de contenus cinématographiques virtuels

Charles Perin
charles.perin@univ-rennes1.fr
Université de Rennes 1 – ISTIC
Master informatique spécialité Recherche en Informatique

Encadré par :
Frédéric Vernier, Université d'Orsay, CNRS LIMSI et Marc Christie, Bunraku, IRISA Rennes
frederic.vernier@limsi.fr, marc.christie@irisa.fr

Mai 2011

Résumé

De nombreux outils d'aide à la création de contenus 3D sont disponibles mais proposent des modalités d'interaction et des modèles sous-jacents qui restent peu expressifs et fastidieux dans leur utilisation. En parallèle, de nombreux travaux de recherche s'attachent à améliorer l'expressivité des modèles de création de contenus avec des approches largement automatisées mais laissent peu de place à la créativité des utilisateurs. Dans ce contexte, nous nous intéressons au contrôle interactif des éléments cinématographiques d'une scène 3D (principalement le positionnement des caméras). Nous proposons un nouvel outil, nommé Collastar, qui met en place une interaction collaborative sur des paramètres de contrôle haut niveau liés à la cinématographie, en utilisant le support de tablettes tactiles. L'outil repose sur un couplage entre un modèle de représentation en étoile pour le contrôle collaboratif et une représentation de type « linear wall » pour une appréhension des éléments de décision dans une approche « focus+context ». Nos contributions portent sur la visualisation de données multidimensionnelles et l'interaction collaborative avec ces données. Les premières évaluations montrent que l'outil est adapté à des tâches d'édition collaboratives, facilite la visibilité des contrôles et la compréhension des liens entre ceux-ci. Enfin l'outil montre une capacité intéressante à faciliter la création de contenus cinématographiques qu'il reste à explorer et évaluer plus finement.

Mots-clés : Interaction, Collaboration, Visualisation d'informations, Cinématographie virtuelle.

Table des matières

Introduction 3
Chapitre 1. Etat de l'art 4
1.1. Introduction 4
1.2. Le contrôle de caméra en environnement virtuel 5
1.2.1. Les règles cinématographiques 5
1.2.2. Les différents contrôles de caméra 7
1.2.3. Le système Cinesys 9
1.2.4. Indicateurs cinématographiques de haut niveau 10
1.3. Représentations multidimensionnelles et temporelles 12
1.3.1. Processus de visualisation de données 12
1.3.2. Techniques d'interaction avec des données temporelles 16
1.4. Positionnement du travail de stage 18
Chapitre 2. Visualisation de données multidimensionnelles et temporelles 19
2.1. Motivations 19
2.2. Horizon Curves et améliorations 20
2.3. Linear Wall et Focus + Context 22
2.3.1. La représentation de l'espace temporel 22
2.3.2. Le modèle de visualisation proposé 23
2.3.3. Comparaison de données 24
2.4. Tâches d'édition de données 25
Chapitre 3. La représentation en étoile pour une interaction multidimensionnelle collaborative 27
3.1. Motivations 28
3.2. Présentation du modèle 28
3.2. Interaction collaborative 30
3.2.1. Orientation vers l'utilisateur 30
3.2.2. Manipulation des branches 30
3.3. Représentation des contraintes 32
3.3.1. Spécification des contraintes 32
3.3.2. Contraintes bloquantes 33
3.3.3. Contraintes modifiantes 34
3.3.4. Contraintes sur intervalles 35
3.4. Tâches de comparaison 36
3.5. Le modèle CollaStar 38
3.6. Améliorations 40
Chapitre 4. Intégration au système Cinesys 41
Chapitre 5. Expérimentations et évaluations prévues 43
5.1. Pertinence de la représentation en étoile par rapport aux coordonnées parallèles 43
5.2. Evaluation des fonctionnalités spécifiques 44
5.3. Evaluation du modèle Collastar dans un contexte cinématographique 45
Conclusion 46
Références 47

Introduction

Ce manuscrit s'intéresse au contrôle de caméra interactif des éléments cinématographiques dans une scène 3D et principalement au contrôle de caméra. Le contrôle de caméra dans une scène 3D consiste à déterminer une séquence temporelle de points de vue, et des transitions entre ceux-ci. Le cadre cinématographique est quant à lui régi par un ensemble de règles déterminées pragmatiquement. Avec l'avènement récent d'outils permettant un contrôle de haut niveau de l'édition de films de synthèse, se crée un besoin de nouvelles métaphores d'interaction. Ces métaphores doivent supporter des interactions nombreuses (complexité des tâches d'édition) et hétérogènes (diversité des paramètres de haut niveau). Ces métaphores doivent également offrir des manipulations simultanées sur des structures multidimensionnelles et temporelles de données. Face à ce besoin en interaction, des périphériques comme les tablettes tactiles interactives offrent une richesse intéressante.

Les logiciels les plus haut de gamme actuels, tels qu'Autodesk® Maya® ou MotionBuilder® (utilisés pour le film Avatar) offrent de nombreux outils permettant la création et l'édition de films d'images de synthèse. Bien que ces logiciels soient extrêmement performants et très utilisés, le placement des caméras reste fastidieux. En effet, le mouvement des caméras est traité au niveau le plus bas, l'utilisateur modifiant explicitement les sept Degrés De Liberté (DDL) de chaque caméra : 3 DDL pour la translation : *x(dolly),y(boom),z(truck)*, 3 DDL pour l'orientation : *roll*, *tilt* et *pan* (fig. 1) et 1 DDL pour la distance focale (zoom avant ou arrière). Le procédé (commun à tous les logiciels) de manipulation de caméra que nous avons observé est le suivant : les paramètres de bas niveau sont entrés manuellement pour des images-clés (*keyframe*) choisies : position, orientation, distance focale. Les logiciels étudiés permettent ensuite d'éditer ces paramètres sur une échelle temporelle afin d'utiliser des interpolations simples (linéaires) ou plus complexes, comme présenté dans la fig. 6.a. Nous notons que ces logiciels s'interfacent avec des dispositifs de manipulation de plusieurs DDL (*Spacemouse* par exemple) mais que ceux-ci ont pour seule ambition de manipuler les paramètres de caméra à bas niveau. Les modalités d'interaction et de contrôle sont les mêmes avec les sources lumineuses. Afin d'apporter une plus grande facilité d'utilisation en termes de manipulation, nous avons considéré l'utilisation de paramètres alternatifs (de plus haut niveau que les DDL de la caméra) dans le contexte de la création de films, en sélectionnant ces paramètres avec pertinence. Nous nous sommes par exemple intéressés à des paramètres tels que le personnage ou l'objet central de la scène, le dynamisme de la caméra ou bien le type de plan désiré (serré ou large par exemple). L'étude des métaphores d'interaction qui permettent d'accéder à la structure multidimensionnelle des données est un sujet de recherche très actif. En couplant ces métaphores à un périphérique tactile multipoints, les possibilités offertes à l'utilisateur ou aux utilisateurs (*utilisateur* désigne dans cette étude un réalisateur ou monteur de film) sont bien supérieures à celles proposées lors de l'utilisation d'un périphérique d'entrée classique tel que la souris ou la *spacemouse*.

Les trois verrous scientifiques que nous avons identifiés face à cette problématique sont les suivants :

1. La visualisation de données multidimensionnelles et temporelles.
2. L'interaction avec ces données
3. L'édition collaborative de données multidimensionnelles et temporelles

Par conséquent, les quatre axes de travail que nous avons tout particulièrement développés sont :

1. Les différents aspects de la visualisation des données spécifiques à notre cadre applicatif : bien que de nombreuses représentations existent pour l'analyse de données, celles prenant en compte l'aspect temporel sont moins fréquentes. Il nous a fallu déterminer la manière de représenter des données temporelles, de différents types, et en quantité variable. Nous avons aussi dû considérer les opérations de navigation sur l'axe temporel et l'affichage de ces données à différentes échelles, ainsi

que les tâches de manipulation de type *ajout* ou *réorganisation* des données.

2. Le modèle que nous avons conçu se voulant interactif, nous avons proposé plusieurs métaphores dédiées à cette tâche. En premier lieu nous avons intégré des métaphores d'interaction « standard » puis avons déterminé la *représentation en étoile*, satisfaisant nos besoins en interaction.
3. Il nous a semblé essentiel que le modèle proposé soit adapté à des interactions multi-utilisateurs collaboratives. Ces dernières années, les outils collaboratifs connaissent un essor avec la démocratisation des périphériques tels que les tables tactiles. Nous avons donc exploré cette voie afin d'exploiter au mieux ses caractéristiques particulières.
4. Le fait que le système soit multi-utilisateurs et les paramètres interdépendants nous a amené à spécifier des contraintes sur les interactions et les différentes manières de représenter celles-ci.

Un des enjeux du travail a été d'établir les bénéfices du couple tables d'interaction – représentations dans des tâches de manipulation multidimensionnelles haut niveau.
Les contributions que nous avons apportées sont les suivantes :

1. Nous avons proposé une manière efficace de représenter des données continues dans un espace restreint (« Linear wall et Focus+context »), ainsi que des tâches de comparaison de données. Ces différentes propositions sont détaillées dans le chapitre 2.
2. Afin de permettre l'interaction collaborative sur les données, nous avons proposé un modèle d'édition : la représentation en étoile pour une interaction multidimensionnelle collaborative, que nous exposons dans le chapitre 3. Nous développons en particulier les différentes contraintes sur les interactions engendrées par les dépendances entre paramètres et les avertissements visuels associés que nous avons proposés.
3. Nous proposons finalement dans la section 3.5 le modèle « CollaStar », représentation permettant d'une part la visualisation de données multidimensionnelles et temporelles et d'autre part l'interaction collaborative avec des paramètres de types différents.

Contexte du stage

Ce stage s'est effectué en collaboration entre deux laboratoires et avait pour objectif de concilier deux thématiques. L'équipe Bunraku (IRISA), dans ses travaux sur l'animation, a mis au point un outil (Cinesys, cf. 1.2) mettant en œuvre un contrôle de caméra automatique en environnement 3D. Le laboratoire du LIMSI a quant à lui apporté des contributions concernant le contrôle interactif de paramètres et les représentations multidimensionnelles.

Chapitre 1. Etat de l'art

1.1. Introduction

Ce chapitre décrit un état de l'art du contrôle de caméra (interactif et automatique) en environnement 3D (interactif ou non) ainsi que des représentations de données multidimensionnelles et temporelles existantes, afin d'introduire les fondements du travail de recherche effectué, visant à coupler ces deux domaines.

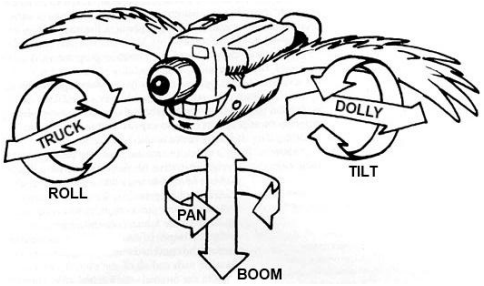


Fig. 1 : Les axes de translation et rotation d'une caméra [Ker00]

La suite de cet état de l'art est organisée comme suit. La section 1.2 traite du contrôle de caméra dans un environnement 3D, en approfondissant le système Cinesys [LCLSO10]. La section 1.3 aborde les différentes représentations multidimensionnelles et temporelles existantes et leur taxonomie. Enfin, nous conclurons dans la section 1.4 en positionnant notre travail de stage par rapport à cet état de l'art.

1.2. Le contrôle de caméra en environnement virtuel

Avec les progrès récents dans le domaine de l'animation et du rendu visuel et l'utilisation croissante de mondes virtuels 3D (par exemple en visualisation de données, dans les jeux vidéo, visites virtuelles ou machinima) le contrôle de caméra – ce par quoi la scène est visualisée – se doit d'évoluer parallèlement. En effet, la position, l'orientation et le mouvement de la caméra en fonction des éléments de la scène et des actions qui s'y déroulent influent grandement sur les impressions du spectateur. De nombreuses techniques manipulant les 7 DDL de la caméra et quelques-unes proposant une interaction de plus haut niveau ont été proposées, que ce soit pour un contrôle interactif ou automatique, faisant émerger de nouveaux problèmes tels que le calcul de trajectoire de la caméra ou la prise en compte des occultations (non visibilité d'un acteur à l'écran). Le contrôle de caméra consiste à choisir les meilleures position, orientation et proximité de la caméra tout en fournissant une vue de l'environnement virtuel sans occultations. Injecter et adapter dans ces applications traitant des mondes virtuels 3D les mêmes règles que celles régissant l'art cinématographique est naturel : ces règles et conventions ont été créées et se sont imposées au cours du XXème siècle. Les personnes les utilisant (réalisateurs) ont pour but d'autoriser des actions compréhensibles. Nous verrons comment l'adaptation de ces règles est possible et quelles sont les principales difficultés liées à cette adaptation.

1.2.1. Les règles cinématographiques

La cinématographie est l'art de construire un film. De nombreuses règles lui sont propres et sont acquises, telles que le placement de la caméra, la composition des prises de vue (*shot*) ou l'arrangement des éclairages. Ces paramètres, qui peuvent définir le style de l'auteur, permettent de capter l'attention du spectateur et de lui présenter un contenu. Nous débuterons en exposant brièvement les principales règles spécifiques à ce domaine avant de considérer la formalisation de celles-ci. Cette section s'inspire de deux ouvrages traitant de l'art cinématographique en général : *Grammar of the edit* [TB09a] et *Grammar of shot* [TB09b].

Composition d'un film sous forme hiérarchique.

Il est suggéré dans la littérature de représenter un film sous une forme hiérarchique : un film est représenté comme étant une séquence de scènes contenant chacune une situation ou une action spécifique. Chaque scène est elle-même composée d'une ou plusieurs prises de vues. Une prise de vue est définie comme un intervalle de temps durant lequel la caméra filme en continu, c'est-à-dire sans coupure ou transition. Une prise de vue ne dure en général que quelques secondes.

Les différents placements de caméra.

Nous nous plaçons dans la situation où la scène contient deux acteurs, par exemple un dialogue (A et B dans la fig. 2). C'est un cas typique, tant celui-ci est fréquent dans la cinématographie. Les metteurs en scène caractérisent les différents placements de caméra relativement à la ligne d'intérêt. La ligne d'intérêt est une ligne imaginaire reliant les deux acteurs de la scène. Les placements de caméra standards sont les vues *external* (vue *over the shoulder*, cf. fig. 3), *internal* et *apex* (fig. 2). La fig. 4 illustre les différents cadrages classiques de caméra lorsqu'un acteur est filmé et la fig. 5 les trois angles standards de la caméra permettant d'effectuer des vues en plongée ou contre-plongée. Ceux-ci permettent là encore au réalisateur d'exprimer un style ou un message à transmettre.

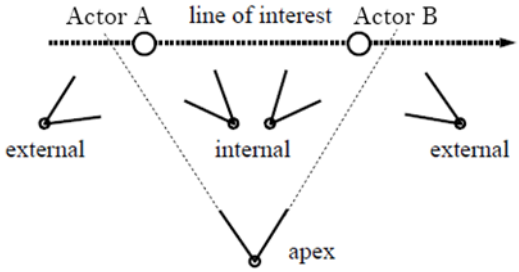


Fig. 2 : Placement de la caméra relativement à la ligne d'intérêt [HCS96]. External : vue d'un acteur par-dessus l'épaule du second. Internal : vue d'un des deux acteurs. Apex : Vue des deux acteurs perpendiculairement à la ligne d'intérêt

Fig. 3 : Une prise de vue par-dessus l'épaule (*over the shoulder*) Source : http://accad.osu.edu/~midori/Materials/camera.html

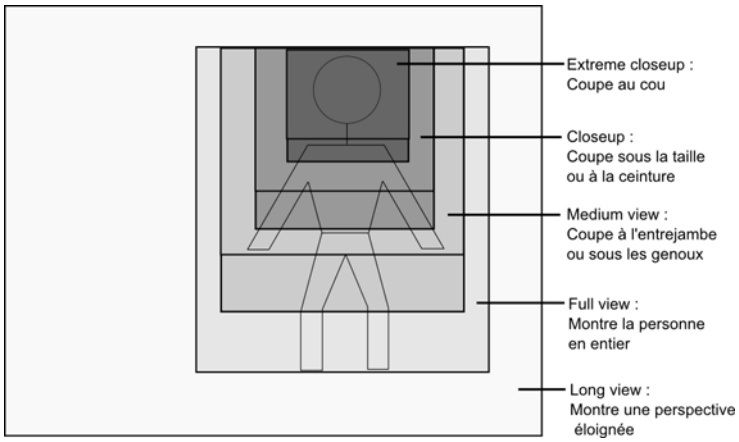


Fig. 4 : Principales hauteurs de coupe

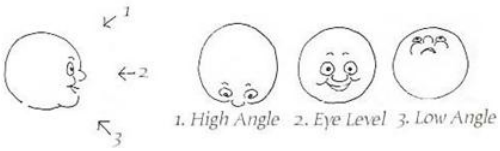


Fig. 5 : Les différents angles verticaux (tilt) de la caméra
Source :http://accad.osu.edu/~midori/Materials/camera.html

Contraintes et heuristiques cinématographiques, séquences de prises de vue.

Dans la littérature, ces placements sont utilisés afin de définir plusieurs règles. Par exemple :

- Dans une même scène il n'est pas souhaitable que la caméra traverse la ligne d'intérêt lors d'une coupure (règle des 180 degrés).
- Deux prises de vue doivent avoir une différence d'orientation d'au moins 30 degrés (règle des 30 degrés)
- Il doit y avoir une différence significative de taille, vue, ou nombre d'acteurs entre deux prises de vue
- Une scène doit être établie à son commencement, c'est-à-dire débuter par une vue d'ensemble.

On trouve dans [TB09a] et [TB09b] une liste exhaustive de ces heuristiques. Un ensemble de stéréotypes de séquences de prises de vue permettant de définir le style du réalisateur (donné dans [Ari76] de manière informelle) est ajouté à ces règles.

L'ensemble de ces règles cinématographiques, si elles sont largement détaillées dans la littérature du domaine, n'ont commencé à être formalisées que récemment. Comme nous le verrons en 1.2.2, l'article de référence [HCS96], datant de 1996, est le premier à proposer une formalisation de ces règles afin de générer automatiquement les paramètres de la caméra selon les événements et le contenu de la scène.

1.2.2. Les différents contrôles de caméra

La Manipulation des 7 DDL de la caméra est fastidieuse et contraignante. Pour pallier ce problème, deux catégories de contrôle de la caméra existent : le contrôle interactif et le contrôle automatisé. Les deux cas sont sujets aux mêmes difficultés : grande complexité due aux 7 DDL, non linéarité (la projection d'un objet 3D sur un plan 2D est fortement non-linéaire), difficultés à inverser (passage d'une contrainte sur l'image à la position de caméra respectant la contrainte) et problème de l'occultation (maintien de la visibilité sur les acteurs). Le problème ne consiste pas à prendre en compte uniquement la géométrie de la scène, mais aussi les relations entre les objets, au sens d'un plan, à la compréhension, au temps, etc.

Dans la suite, nous décrivons deux approches du contrôle de caméra : les contrôles interactifs et les contrôles automatiques.

Le contrôle de caméra interactif.

Le contrôle de caméra interactif a pour but d'apporter la possibilité à l'utilisateur de modifier les paramètres de la caméra en temps-réel. Il peut être organisé selon trois catégories [CON08] :

- **Contrôle direct *(direct control)*** : l'utilisateur modifie directement les paramètres de la caméra, comme c'est le cas dans Maya® (fig. 6.a). Plusieurs métaphores existent afin de manipuler les 7 DDL de la caméra : *eyeball in hand* (l'utilisateur manipule la caméra dans l'espace [PBG92], [Sho92]), *world in hand* (l'utilisateur manipule le monde alors que la caméra reste fixe) ou encore *walking metaphor* (la distance au sol est constante).
- **Contrôle dit « Through the lens » *(TTL)*** : l'utilisateur contrôle la position des objets à l'écran [GW92]. L'objectif de cette technique est de fournir des systèmes interactifs de haut niveau.
- **Contrôle assisté *(Assisted control)*** : La connaissance de l'environnement est utilisée pour assister l'utilisateur. L'assistance peut être basée objets (des boites englobantes contraignent le placement de la caméra) ou basée environnement, en utilisant la planification de mouvement : construction d'un graphe connectant les différentes configurations consécutives puis recherche d'un chemin dans le graphe.

Nous nous sommes intéressés tout particulièrement à la troisième catégorie : les contrôles assistés. En effet, en utilisant des métaphores d'interaction de haut niveau, une manipulation explicite des DDL de la caméra est évitée en laissant le processus placer la caméra en fonction des objectifs de l'utilisateur, en termes de propriétés de prises de vue.

De nombreux travaux ont déjà été effectués : StyleCam [BKFBK02] par exemple propose une approche permettant à l'utilisateur de ne manier que 2 DDL et donc d'utiliser une souris. Le principe est d'assister l'utilisateur lors de l'inspection d'objets 3D lorsque celui-ci en est proche (inspection proximale), en le positionnant automatiquement au meilleur point de vue possible selon le détail qu'il cherche à observer. Ceci est possible en construisant des surfaces autour des objets où l'utilisateur sera automatiquement placé ainsi que des transitions automatiques entre celles-ci (fig. 6.b). Toutefois StyleCam requiert la construction au préalable d'un ensemble de surfaces, points de vue et transitions pour chaque objet de la scène 3D. HoverCam [KKSFK05] reprend le même principe, mais la détermination des surfaces est automatisée en gardant constante la distance entre l'utilisateur et l'objet observé et en déterminant la meilleure orientation de caméra lui permettant d'observer l'objet. [HMKBH97] propose une approche encore différente : en assignant à la scène un champ de potentiel des objets, l'utilisateur est poussé à se déplacer là où ce champ est le plus faible, évitant ainsi les collisions.

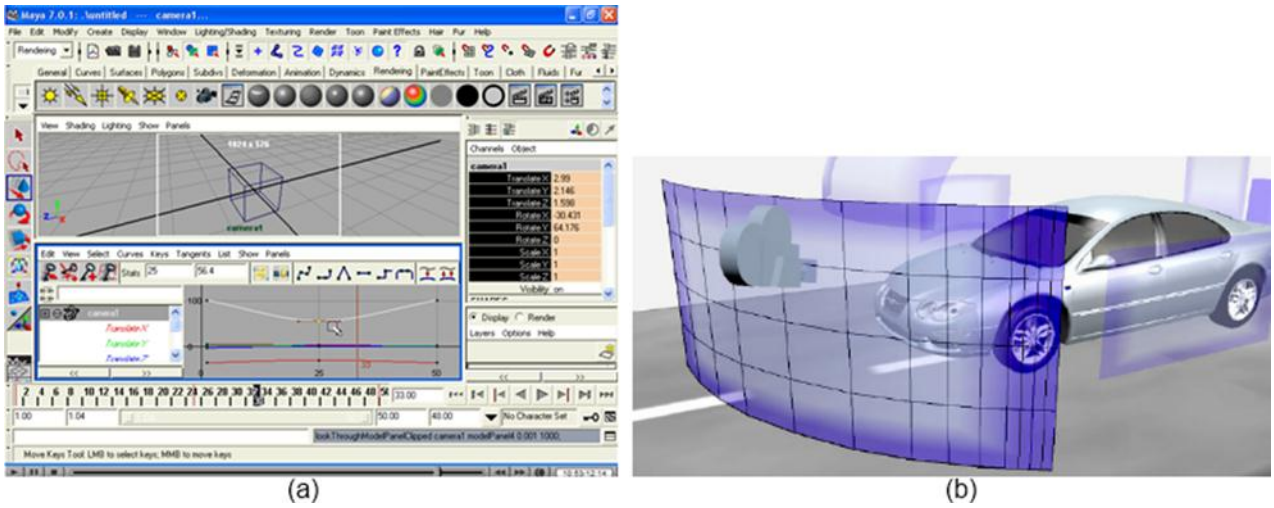


Fig. 6 : (a)Contrôle direct de la caméra sous Maya®, (b)Contrôle assisté dans StyleCam[BKFBK02] : les déplacements de caméra s'effectuent selon 2 dimensions le long de surfaces prédéfinies autour des objets

Le contrôle de caméra automatique.

Dans le cas du contrôle automatique, la caméra est entièrement contrôlée par l'application, l'utilisateur n'ayant aucun contrôle interactif sur ses paramètres. [CON08] différencie quatre catégories d'approches du contrôle de caméra :

l'approche est qualifiée de *réactive* lorsque la modification des paramètres de la caméra est calculée en réponse à un changement dans les propriétés de l'image. Dans ce cas d'asservissement visuel [CM01], une position optimale est recherchée parmi les solutions possibles (utilisation de la Jacobienne image). Cette stratégie est très peu coûteuse en temps de calcul et permet une utilisation en temps-réel. Les approches basées *optimisation* ont quant à elles pour objectif de maximiser des propriétés sur les prises de vue. La principale difficulté de cette approche est de parvenir à exprimer toutes les contraintes (7 DDL) dans une seule fonction objectif. Des algorithmes génétiques avec *crossover* sont en général utilisés [OHPL99], ce qui entraine un grand temps de calcul. Nous parlerons d'approche basée *contraintes* lorsque les contraintes sont exprimées sur les paramètres et résolues par des techniques CSP (*Constraint Satisfaction Problem*) [Sny92]. Le temps de calcul est encore une fois important, mais cette stratégie apporte certaines caractéristiques intéressantes : une approximation de la solution est garantie (les régions évaluées comme vraies contiennent uniquement des solutions alors que celles évaluées comme fausses ne contiennent aucune solution), et les équations linéaires, polynomiales ou non linéaires ainsi que les inéquations peuvent être traitées. Une solution avancée, l'approche par *optimisation contrainte* permet d'obtenir un compromis entre l'approche basée optimisation, qui donne des solutions bonnes à un certain degré et l'approche basée contraintes qui fournit la totalité des solutions, mais a un coût de calcul élevé. Le système modélisé contient donc un ensemble de propriétés obligatoires (contraintes) et un ensemble de propriétés à optimiser (maximisation d'objectifs). Ces quatre catégories d'approche ont pour objectif de calculer le(s) meilleur(s) point(s) de vue de la scène. Néanmoins, celles-ci ne prennent pas en compte l'aspect narratif d'un film. Dans une perspective cinématographique, les points de vue successifs et leurs transitions doivent construire une séquence intelligible.

De nombreux travaux ont été effectués sur la problématique du contrôle de caméra, et autant d'approches différentes en ont résulté. [HCS96] est le premier article à présenter une formalisation des règles cinématographiques et propose un contrôle de caméra automatique.

De manière générale, le contrôle des caméras virtuelles doit faire le lien entre : **1**-une application en temps-réel qui génère des informations sémantiques caractérisant la scène et les actions qui s'y déroulent ; **2**-Le monteur virtuel (*Virtual Cinematographer* : VC) qui détermine les paramètres de la caméra en fonction des indications sémantiques transmises par l'application en temps-réel ; **3**-Le rendu, qui prend en compte les paramètres géométriques de la scène ainsi que les spécifications de la caméra, transmises par le VC. Ces approches permettent de formaliser les règles cinématographiques existantes dans un environnement simple. Le désavantage de ces méthodes est qu'elles ne proposent qu'un contrôle de caméra automatisé. En effet, les automates sont déterministes et l'utilisateur n'a pas de contrôle interactif en temps-réel sur la caméra.

Nous allons dans la partie suivante décrire le système Cinesys [LCLS010], qui bien qu'il propose un contrôle de caméra automatisé, dispose d'opérateurs de haut niveau paramétrant le choix des points de vue. Il prend en compte les règles cinématographiques et les occultations, et permet de calculer les points de vue en temps-réel donc autorise des possibilités d'interaction.

1.2.3. Le système Cinesys

Les approches existantes en termes de contrôle de caméra automatique sont procédurales, se concentrent sur des aspects isolés tels que la visibilité ou les transitions, n'offrant pas de paramétrage de haut niveau qui permettrait de faire varier des données représentatives d'un style cinématographique. Le système Cinesys [LCLSO10] propose un modèle permettant de manipuler des paramètres alternatifs afin de fournir à l'utilisateur une des contrôles de plus haut niveau pour des tâches de création automatique de films. C'est ce système que nous allons détailler ici.

Le système Cinesys prend en entrée un environnement 3D interactif dont les mouvements ne sont pas connus à l'avance, ainsi qu'un flot d'éléments narratifs décrivant les actions effectuées dans cet environnement. Les éléments narratifs peuvent être des actions prototypiques (par exemple « se lever », « marcher », « parler à ») mais peuvent aussi traduire des actions plus subtiles, telles que les relations entre des acteurs (par exemple dominance, conflit ou isolement). L'objectif de l'approche est de produire un film contenant ces éléments narratifs, selon des règles cinématographiques et un style particulier.

Ce processus s'effectue en quatre temps :

1. Sélection des éléments narratifs : lors d'une nouvelle transition, le moteur narratif sélectionne en temps-réel le(s) élément(s) narratif(s) le(s) plus important(s) parmi tous les éléments narratifs (tels que les actions ou les mouvements clés d'acteurs) de la scène.
2. Calcul de volumes directeurs (*director volumes)* : à partir des éléments narratifs, des volumes directeurs sont calculés. Ceux-ci sont les éléments de l'espace depuis lesquels les éléments narratifs peuvent être cadrés (fig. 7).
3. Filtrage des volumes directeurs : cette étape consiste à filtrer les volumes directeurs afin de ne conserver que ceux satisfaisant certains critères. Une succession de filtres est ainsi appliquée : filtres de continuité (ceux qui violent une règle de continuité au sens cinématographique sont éliminés), filtres de style (en pondérant les éléments narratifs selon le style désiré, les volumes directeurs sont triés selon leur pertinence) et enfin des règles de composition sont prises en compte. Parmi les volumes directeurs conservés jusqu'ici, ceux qui permettent d'appliquer une composition (position exacte des acteurs à l'écran) proche de celle souhaitée sont sélectionnés.
4. Calcul des transitions : s'il est nécessaire de changer de volume directeur (par exemple début ou fin d'événement, occultation) une transition est sélectionnée automatiquement, celle-ci pouvant être une coupure (nouvelle prise de vue) ou une transition continue (mouvement de la caméra, sans couper).

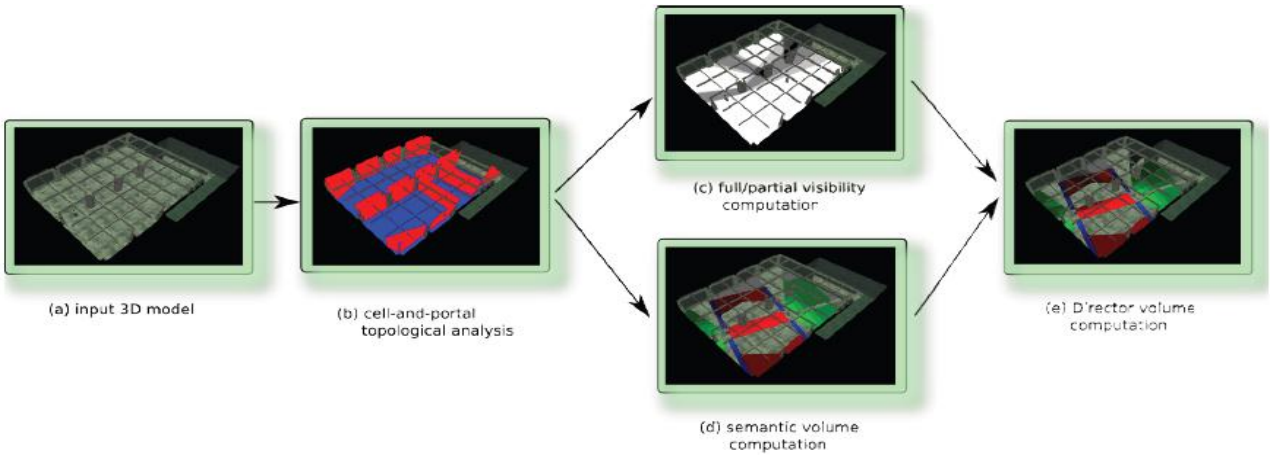


Fig. 7 : Calcul des volumes directeurs. (a) le modèle 3D en entrée avec deux acteurs, (b) le calcul cellules et portes qui partitionne l'espace en cellules convexes connectées par des portes, se basant sur un pré calcul de la topologie de l'environnement [Lam09], (c) les volumes de visibilité pour les deux acteurs, en prenant compte des occultations (d) les volumes sémantiques calculés à partir des éléments narratifs de la scène, (e) les volumes directeurs (intersection des volumes sémantiques et de visibilité) [LCLSO10]

Les règles de continuité et de style déterminent quel volume directeur sera utilisé pour chaque prise de vue, mais aussi les instants auxquels il faut passer d'une prise de vue à une autre et la manière de le faire (coupure ou transition continue). Ces choix sont influencés par des indicateurs sur les acteurs présents dans la scène (table 1) ou sur la caméra (table 2). Notre travail a eu pour objectif de permettre à des utilisateurs d'entrer dynamiquement les paramètres de style de haut niveau afin de contrôler les étapes 3 et 4 de manière interactive.

1.2.4. Indicateurs cinématographiques de haut niveau

La notion de style, qui attribue à des indicateurs des valeurs particulières, peut être vue comme un indicateur de niveau supérieur. Ces indicateurs (liste non exhaustive en tables 1 et 2, indicateurs soit extraits d'articles soit proposés) sont des opérateurs de haut niveau manipulant les 7 DDL de la caméra. Des contradictions peuvent donc apparaître, à plusieurs niveaux de représentation. Ces indicateurs peuvent de plus paramétrer un film entier, une scène ou une ou plusieurs prises de vue, et l'objectif était de parvenir à les manipuler de manière interactive.

Indicateur	Prises de vue	Composition à l'écran	Points de vue
Affinité	Symétriques	Similarité entre les 2 acteurs, même hauteur d'yeux	+ apex - external
Dominance	Asymétriques Dominant vu plus souvent que dominé Dominant vu plus longtemps que dominé	Dominant plus grand que dominé Yeux du dominant plus hauts que ceux du dominé	+ external du dominant + subjective view du dominant
Isolation	La caméra s'attarde sur l'isolé	Isolé entouré d'espace vide en cas de *long shot*	- external - apex +parallel

Table 1 : Caractéristiques de l'affinité, de la dominance et de l'isolation. Ces 3 indicateurs sont implémentés dans Cinesys.

Indicateur	Définition	Type de valeur envisageable
Pacing*	Rythme des coupures : indique la cadence à laquelle les prises de vue seront coupées.	Gaussienne continue centrée sur une valeur dans un intervalle temporel
Dynamicity*	Dynamisme de la caméra : influence le choix des prises de vue favorisées selon le style voulu sur une échelle allant de « caméra statique » à « caméra libre ». Le dynamisme prend une valeur selon : l'autorisation ou non de changer l'orientation de la caméra, l'autorisation ou non de changer la translation de la caméra et la qualité des changements de prises de vue : coupure ou transition	Discret, 4 valeurs possibles : (1 : Orientation 0, Translation 0, Coupure ou transition 0 ; 2 : Orientation 1, Translation 0, Coupure ou transition 0 ; 3 : Orientation 1, Translation 1, Coupure ou transition 0 ; 4 : Orientation 1, Translation 1, Coupure ou transition 1)
Visibilité*	A partir de quelle visibilité un acteur est-il considéré comme visible ? Cet indicateur peut être général ou différent selon les acteurs	Intervalle continu dans [0,1]
Intensité	Plus ce critère est élevé et plus la caméra sera proche des acteurs	Valeur continue dans [0,1]
Règles de filtrage*	Respect ou non des règles de filtrage telles que : -Règle des 30 degrés (1) -Continuité de la ligne d'intérêt(2) -Continuité de la ligne d'action (3) -Continuité de la direction à l'écran (4)	binaire
Idiomes*	Classement arbitraire des différents idiomes. A chaque point de vue est attribué un score selon ce classement, il est alors possible de les trier en fonction de leurs scores	Sous-ensemble d'idiomes
Points de vue*	Classement arbitraire des différents points de vue. A chaque point de vue est attribué un score selon ce classement, il est alors possible de les trier en fonction de ceux-ci	Sous-ensemble de vues
Transitions*	Table de transitions (probabilités)	Matrice de probabilités
Règles de composition*	Utilisation ou non des règles de composition telles que la règle des tiers (5) Chaque plan est sujet à des règles de composition différentes	binaire
Dutch Angle	Indique l'inclinaison du point de vue par rapport à la ligne d'horizon	Valeur continue dans [-π/2,π/2]
Plongée et contre-plongée	Rotation de la caméra (*tilt*) rendant la taille perçue d'un acteur plus grande ou plus petite	Valeur continue dans [-π/4,π/4]
Subjectivité	Prise de vue subjective ou non (6) ?	Binaire ou valeur continue dans [-π/2,π/2]

Table 2 : Définitions et représentations de différents indicateurs de haut niveau extraits à partir de la littérature
*** : indicateur implémenté dans Cinesys.**
(1) Deux prises de vue successives doivent avoir un écart d'orientation d'au moins 30 degrés.
(2) Les acteurs doivent conserver leur position relativement à l'écran : il n'est pas possible que la caméra traverse la ligne d'intérêt.
(3) Un acteur bougeant de gauche à droite à l'écran doit continuer à bouger dans cette direction.
(4) Si un personnage sort de l'écran par la gauche, il doit rentrer par la droite dans la prise de vue suivante et conserver sa direction.
(5) Les éléments clef de l'image sont placés sur les lignes qui séparent les tiers verticaux et horizontaux et leurs intersections.
(6) Prise de vue subjective si l'axe Z de la caméra est confondu avec la direction du regard de l'acteur.

Nous avons vu qu'il existe plusieurs manières de contrôler une caméra. Le contrôle automatique est utilisé dans de nombreux domaines mais de par sa définition, n'offre pas de possibilités d'interaction. Le contrôle interactif quant à lui permet une manipulation soit de paramètres de bas niveau (les 7 DDL de la caméra) soit de paramètres de haut niveau mais uniquement pour des tâches très spécifiques comme l'exploration proximale et non adaptées à l'édition de films (StyleCam).

Nous avons dans ce chapitre exhibé les caractéristiques du contrôle de caméra et ses avancées actuelles. Le système Cinesys présenté en 1.2.3 est un outil prenant en compte à la fois les règles cinématographiques et les problèmes de contrôle de caméra (occultation, transitions) en apportant la notion de style et une possibilité de paramétrage. Une fois les paramètres de haut niveau pertinents sélectionnés, il s'agissait de pouvoir les représenter d'une manière permettant à l'utilisateur de les observer, les analyser et les modifier interactivement (en temps-réel). De par les multiples paramètres d'un tel système, une interface multidimensionnelle offre des propriétés intéressantes permettant d'exploiter cette méthode. Nous développons en 1.3.2 les métaphores d'interaction en plusieurs dimensions existantes, en nous focalisant sur

celles permettant de prendre en compte la dimension temporelle.

Afin de représenter et d'interagir avec des jeux de paramètres, il est nécessaire de représenter ceux-ci. Nous exposons dans la section suivante le processus de visualisation de données ainsi que les tâches d'interaction spécifiques à l'édition de données.

1.3. Représentations multidimensionnelles et temporelles

Afin d'interagir avec les paramètres de haut niveau exposés précédemment, l'utilisation d'une interface permettant la visualisation et la modification de données multidimensionnelles et temporelles est nécessaire. Nous verrons en premier lieu quel est le processus général de visualisation de données et comment ce processus s'applique aux données temporelles et multidimensionnelles. Nous verrons en particulier que dans le cas de l'édition de films comme dans de nombreux autres contextes, la composante temporelle joue un rôle primordial.

1.3.1. Processus de visualisation de données

Afin de situer les visualisations de données temporelles et multidimensionnelles, nous développons le processus de visualisation de données en général qui se décompose en quatre niveaux de données [Chi00] et trois transformations [Ver00] (fig. 8). L'étape *donnée* représente les valeurs de la donnée à visualiser, composée dans le cas qui nous intéresse d'une dimension structurelle et d'une dimension temporelle. L'étape *point de vue sur la donnée* extrait à partir de ces valeurs une abstraction analytique ou métadonnée. Ici est ainsi défini le type d'information à représenter au niveau d'une technique de visualisation. *Espace de visualisation* transforme la représentation analytique en valeurs directement visualisables. Cette transformation fixe les caractéristiques de l'espace de représentation dans lequel les données sont visualisées en fonction du nombre de ses dimensions. Finalement, l'étape *point de vue sur l'espace de visualisation* définit le rendu perceptible par l'utilisateur, en transformant les valeurs directement visualisables en une représentation graphique, ou vue.

Dans la transformation *point de vue sur la donnée*, l'abstraction analytique représente une donnée temporelle comme une séquence de couples <t,v>, où t est l'instant d'observation (valeur temporelle) et v la valeur de la donnée à cet instant (valeur structurelle). La conversion des données brutes en métadonnées, véhiculant une information descriptive à propos de celles-ci afin de rendre leur manipulation plus aisée est essentielle. Deux échelles de temps existent dans la cinématographie : l'échelle temporelle du film lui-même est rarement linéaire et peut effectuer des sauts temporels dans le passé ou le futur, accélérer ou décélérer. La seconde échelle temporelle, adaptée au point de vue du monteur, représente le temps du film observé. Cette échelle va du début à la fin du film et a une progression linéaire.

Processus de visualisation de la dimension temporelle.

En informatique, le temps est modélisé par un modèle linéaire discret : il est représenté comme une droite partitionnée en une suite d'intervalles disjoints. Chaque intervalle est un point non décomposable numéroté par un entier appelé chronon : l'élément de l'espace-temps. Toute partition de l'échelle temporelle est construite sur les chronons, l'unité choisie étant une quantification de chronons. L'échelle de temps comporte plusieurs granularités (millisecondes, secondes, minutes, etc.) organisées hiérarchiquement. Dans le cas de l'édition de scènes cinématographiques, les unités d'observation sont dites régulières. En effet, le partitionnement se fait en intervalles de même taille (par exemple 1/25ème de seconde). Ces unités sont aussi comparables : une relation d'ordre entre les différentes unités peut être établie (temps hiérarchisé).

La technique la plus couramment utilisée dans la littérature est celle des intervalles d'Allen [All91], définie pour les documents multimédias. Cette méthode permet de représenter spatialement les relations entre événements dans le temps (fig. 9). Dans le cadre de l'édition de films et du point de vue du monteur, le temps est représenté linéairement, un film possédant une durée déterminée et l'échelle de temps ayant un pas constant.

Nous avons vu dans cette section qu'une donnée temporelle était une séquence de couples <temps,valeur> et avons détaillé le processus de visualisation de la structure temporelle. Cependant ce processus est différent pour les valeurs multidimensionnelles des données. Ces valeurs sont les dimensions structurelles de la donnée temporelle.

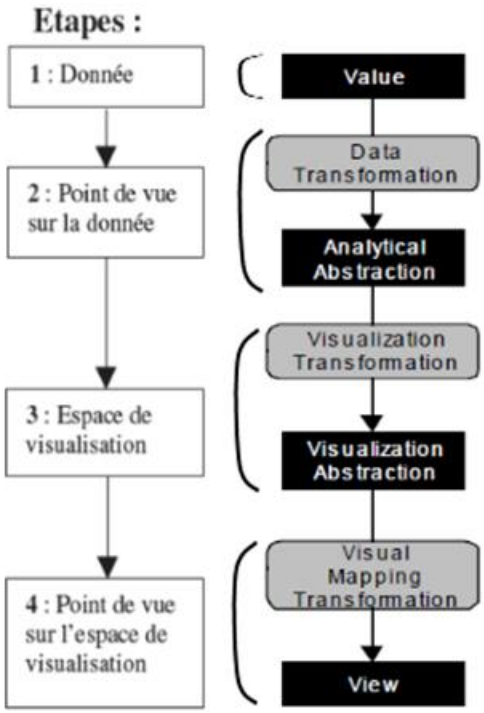


Fig. 8 : processus de visualisation de données [Chi00] structuré en 4 étapes [Ver00]

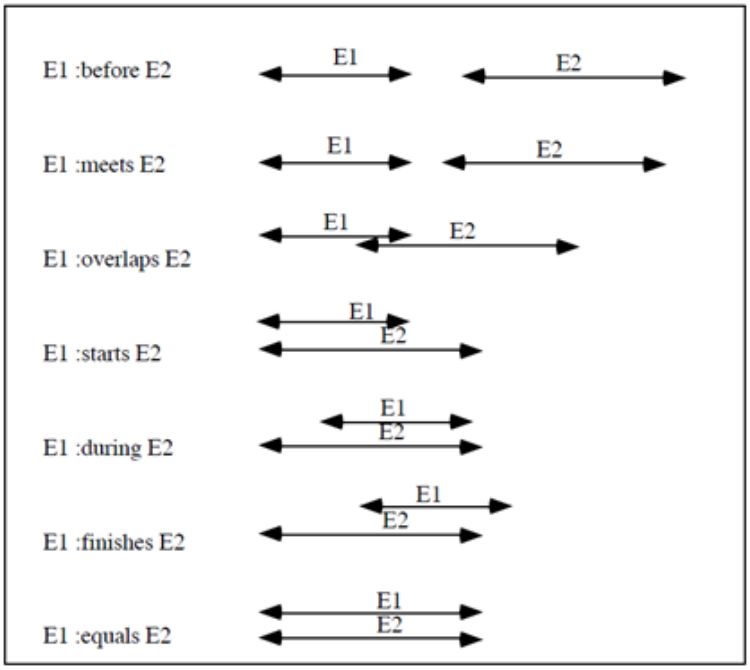


Fig. 9 Représentations spatiales d'événements temporels [All91]

Processus de visualisation des dimensions structurelles.

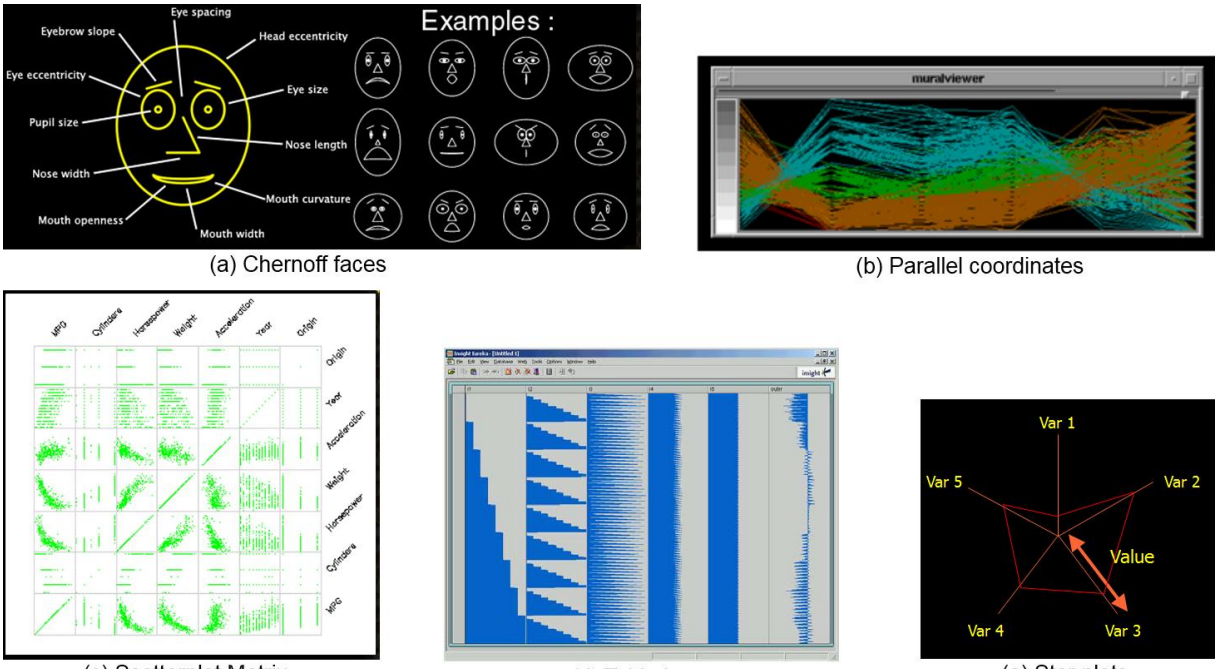
Les valeurs structurelles peuvent être de type quantitatif, qualitatif, voire objet. Pour les dimensions structurelles, l'étape importante dans la fig. 8 est celle de *point de vue sur l'espace de visualisation*. Cette étape est définie par une transformation de l'espace de représentation, par exemple par des fonctions de déformation. Deux stratégies existent à cette fin : celles basées pixels et celles basées éléments graphiques. Les techniques basées pixel représentent chaque entité de l'espace de donnée par un pixel dont la couleur exprime la valeur d'un attribut, alors que les techniques basées éléments graphiques représentent chaque entité par un objet graphique dont varient la taille, la forme, l'orientation, etc.

Les techniques basées pixel étant utiles essentiellement dans le cadre de l'analyse de données, nous nous sommes concentrés sur les techniques basées éléments graphiques, plus orientées manipulation et offrant de meilleures possibilités interactives [Daa00].

Parmi les représentations multidimensionnelles basées éléments graphiques les plus utilisées, nous pouvons citer :

- ***Chernoff faces***, d'Herman Chernoff (fig. 10.a) : encode chaque variable selon une caractéristique du visage humain. En le visualisant un être humain reconnait une expression du visage traduisant l'ensemble des données (par exemple heureux, triste, étonné).
- ***Parallel coordinates (coordonnées parallèles)***, inventées par Maurice d'Ocagne en 1885 (fig. 10.b) : les lignes verticales spécifient les valeurs selon chaque dimension et les variables sont encodées selon une droite horizontale. Le modèle permet l'interpolation de ces droites afin d'en obtenir des courbes. Ce type de représentation permet de se concentrer sur certains objets ainsi qu'une utilisation de la couleur.
- ***Scatterplot Matrix*** [EDF08] (fig. 10.c) : représente dans une matrice carrée 2D chaque paire de variables possible. Si plus de deux variables doivent être comparées, la taille de la matrice devient vite excessive.
- ***Table lens*** [RC94] (fig. 10.d) : affiche chaque variable séparément. Ce type de représentation permet de visualiser toutes les dimensions parallèlement et d'analyser des corrélations.
- ***Star plots*** ou ***star glyphs*** [KHLW09] (fig. 10.e) : représentation en étoile, qui dispose les variables selon des axes à angles égaux circulairement, chaque pointe de l'étoile encodant la valeur d'une des variables. Ce type de représentation permet l'utilisation de données discrètes ou continues, quantitatives ou qualitatives, linéaires ou non.

Cette liste recense les méthodes les plus utilisées ou les plus travaillées. Certaines de ces représentations permettent de comparer facilement des données entre elles mais rendent difficilement possible l'isolation mentale et la signification d'une dimension donnée (*Chernoff faces* par exemple). D'autres au contraire isolent des dimensions mais sont moins adaptées à la comparaison globale des données (*coordonnées parallèles* par exemple). Les tâches de comparaison globale comme celles d'isolation d'une dimension spécifique (pour manipulation ou analyse) étant intéressantes dans le contexte du paramétrage et des modifications à postériori d'un film, ces deux critères ont été pris en compte.



(a) Chernoff faces (b) Parallel coordinates (c) Scatterplot Matrix (d) Table lens (e) Star plots

Fig. 10 : Représentations multidimensionnelles classiques

Techniques de visualisation de données temporelles.

Il existe deux classes de techniques de visualisation de données temporelles : la première est de n'utiliser qu'un seul espace en deux dimensions (une dimension code le temps et l'autre les valeurs en fonction du temps) pour visualiser plusieurs dimensions structurelles séparément (fig. 11.a). La seconde est de considérer pour chaque dimension des données structurelles une dimension temporelle différente (fig. 11.b).

L'utilisation d'un seul espace garantit la mise en correspondance des dimensions structurelles aux mêmes instants. Il est alors possible d'observer et comparer les données à des instants précis. La littérature indique que les données ont pour obligation d'utiliser le même domaine temporel afin que cette méthode soit viable et que les valeurs des données doivent être comparables, c'est-à-dire posséder la même échelle de valeurs. Lors de l'utilisation d'une représentation graphique différente pour chaque donnée structurelle,

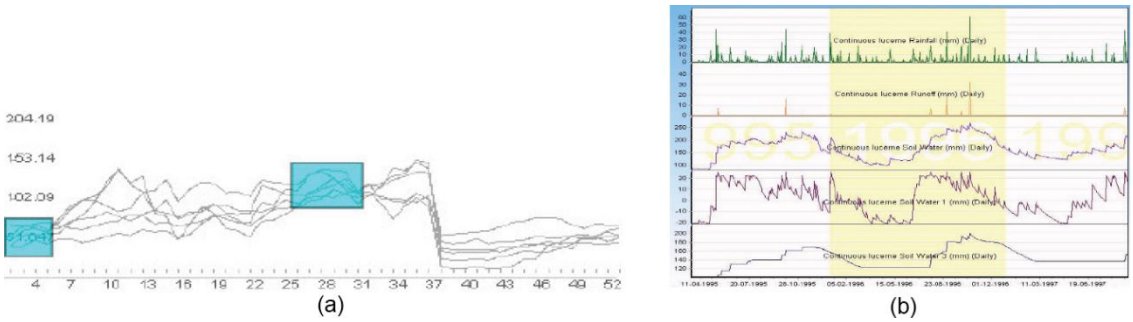


Fig. 11 : (a) Un espace de représentation pour chaque dimension structurelle, (b) Un espace de représentation commun à toutes les dimensions structurelles [Daa00]

l'échelle de temps est en général identique pour chaque représentation et la séparation de chaque dimension permet l'utilisation d'échelles différentes pour les dimensions structurelles.

Toujours dans la catégorie des représentations lorsque le temps est représenté linéairement, nous pouvons citer les techniques suivantes :

- ***Lexis Pencils*** [FP03] (fig. 12) : chaque dimension structurelle est représentée sur une des faces d'un crayon. Cette représentation permet d'afficher simultanément des données continues (illustrées par des dégradés de couleurs) ou discrètes (couleurs unies) et plusieurs crayons peuvent être représentés dans un même espace.

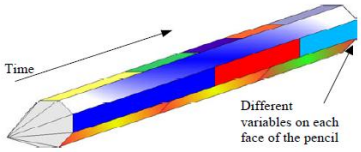


Fig. 12 : Lexis Pencil [FP03]

- ***ThemeRiver*** [HHN00] (fig. 13) : Cette technique prend en compte à chaque pas la valeur de chaque paramètre et l'affichage est rendu continu par interpolation. Le problème majeur de cette représentation est que les paramètres ne sont pas traités équitablement. En effet les paramètres les plus proches horizontalement de l'axe temporel ne sont pas déformés alors que ceux aux extrémités présentent des distorsions.
- ***Multicomb*** [CPDD03] (fig. 14) : Un axe temporel est attribué à chaque variable. Ceux-ci sont disposés circulairement autour d'un cercle. Les axes ne commencent pas au centre, afin de faciliter la lisibilité en ne laissant pas les représentations se chevaucher. Le centre peut alors être utilisé pour afficher des informations supplémentaires.

Dans le cas où le temps est représenté sur une échelle logarithmique, une technique intéressante est celle du ***perspective wall*** [MRC91] (fig. 15) qui offre une vue *focus+context* [Infovis] (les visualisations *focus+context* permettent à l'utilisateur d'observer un objet d'intérêt dans les moindres détails tout en ayant une vue d'ensemble du contexte de cet objet d'intérêt). Ici, sur la face avant (partie *focus*) les données sont

détaillées et non déformées, alors que les faces de gauche et droite, servant à représenter le passé et le futur représentent un avancement plus ou moins rapide dans le temps, de manière non linéaire (partie *context*).

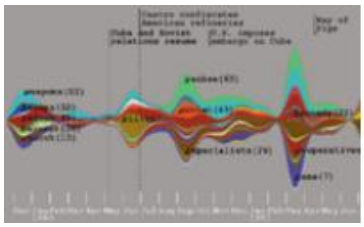


Fig. 13 : ThemeRiver [HHN00]

Fig. 14 : Multicomb [CPDD03]

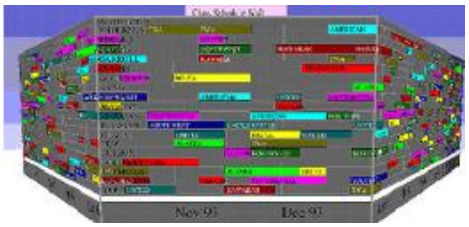


Fig. 15: Perspective wall [MRC91]

Nous avons vu dans cette section qu'une donnée temporelle était caractérisée par la composition d'une dimension particulière qu'est la dimension temporelle et d'une dimension structurelle. Nous nous attardons dans la suite sur les techniques d'interaction avec les données disposant d'une dimension temporelle.

1.3.2. Techniques d'interaction avec des données temporelles

Afin de garantir l'utilisabilité des techniques de visualisations explorées au chapitre précédent, [Daa00] indique qu'il faut définir pour réaliser l'étape du processus *point de vue sur l'espace de visualisation* une approche centrée tâches utilisateur. Cette étape doit traduire chaque tâche utilisateur en un ensemble de propriétés ergonomiques. En combinant le processus de visualisation avec ces tâches utilisateur, des techniques de visualisation sont réalisables, en regard de ces deux critères.

Tâches utilisateur.

Lors de la conception d'un tel système, il est nécessaire d'avoir une approche de conception centrée utilisateur, celui-ci établissant les besoins du dit système. Les tâches utilisateur peuvent être classées selon trois catégories : les tâches d'analyse de données, qui ne sont pas spécifiques aux données temporelles, les tâches d'édition de données concernant l'interaction de l'utilisateur avec les données et leur manipulation dans le but de faciliter leur analyse, et des tâches de navigation nécessaires aux deux tâches précédentes [Daa00].

- **Tâches d'édition de données** : ces tâches consistent à modifier des valeurs de données par le biais de l'interface. Par exemple, dans le cadre du contrôle de caméra, l'utilisateur peut vouloir modifier la largeur temporelle de l'intervalle d'une valeur, ou modifier une valeur à un instant précis.
- **Tâches d'analyse de données** : Les tâches d'analyse sont des tâches que l'utilisateur effectue afin d'extraire des informations pertinentes à partir des données. Elles permettent de visualiser l'évolution globale des données dans le temps ainsi que leur valeur exacte à un instant précis. Par exemple, si l'utilisateur estime que le rythme des coupures est trop irrégulier, il doit pouvoir analyser la répartition des coupures dans le film. Ces tâches permettent aussi la comparaison de données entre elles afin d'étudier leurs corrélations, similarités ou différences.
- **Tâches de navigation** : ces tâches permettent à l'utilisateur, dans un cadre *focus+context* de déplacer le focus, d'anticiper ses actions de navigation (prévisibilité) et de retourner à un endroit connu de cet espace s'il se trouve perdu ; il faut que n'importe quel point de l'espace de données puisse être aisément atteint. Ces tâches sont essentielles car elles permettent d'effectuer les tâches d'édition et d'analyse de données vues précédemment. L'exemple typique de navigation est de devoir retourner au début d'une séquence, ou de vouloir se déplacer à la fin d'un intervalle dans lequel une certaine valeur est constante dans le temps.

Ces tâches de manipulation de données doivent ensuite être traduites en tâches concrètes d'interaction (sélectionner, zoomer, séparer, etc.) pour concevoir des techniques de visualisation. Les tâches utilisateur se rapportant à la dimension temporelle sont aussi à définir : l'utilisateur doit pouvoir sélectionner des valeurs temporelles et naviguer dans le temps. Nous avons dû déterminer quelles représentations s'adaptent le mieux au cadre du contrôle de caméra ainsi que la manière de les instancier.

Deux exemples concrets.

Dans le but d'interagir avec des données temporelles, le modèle présenté dans [DNF04] propose de combiner deux techniques de visualisation décrites précédemment et offre une vue *overview+detail* [Infovis] (utilisation de deux affichages parallèles pour la représentation, le premier affichant une vue d'ensemble des données non détaillées, le second décrivant en détail les valeurs d'une donnée spécifique) : les dimensions structurelles sont représentées par la métaphore de l'étoile, alors que le mur en perspective illustre la progression dans le temps. L'étoile affichée indique la valeur de données selon l'instant choisi (*detail*), alors que le mur permet de se déplacer dans le temps tout en conservant un repère temporel global (*overview*). Les données que représente l'étoile évoluent en fonction de l'instant sélectionné (fig. 16). Cette représentation apporte une bonne compréhension des valeurs des données structurelles et temporelles, mais n'affiche pas le contexte dans lequel évoluent les données.

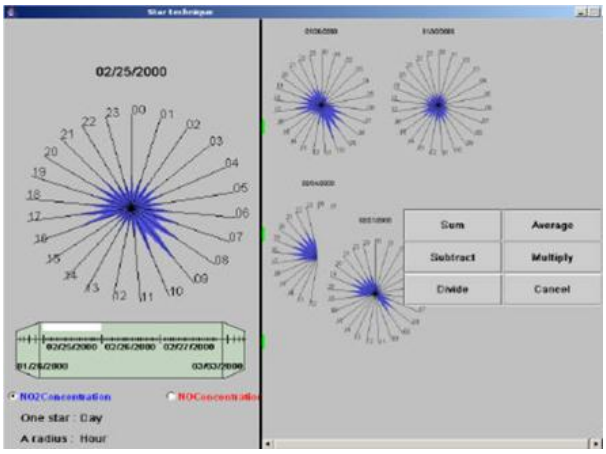


Fig. 16 : Overview+detail grâce à la représentation en étoile et le mur en perspective [DNF04]

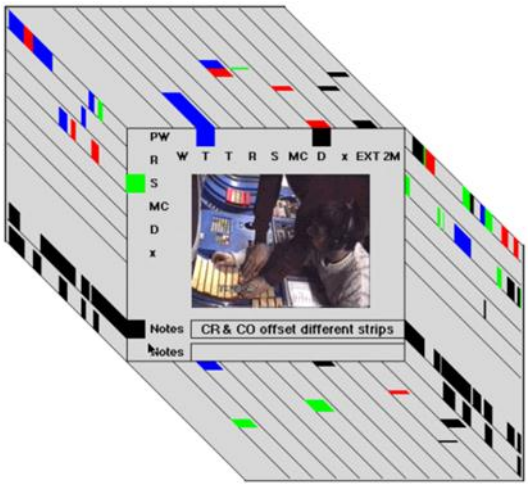


Fig. 17 : Focus+context : Le système DIVA [MB98]

Un second exemple est le système DIVA [MB98] étudié afin d'explorer et analyser des données de flux multimédia. Il permet de visualiser des données qualitatives en fonction du temps (linéaire cette fois-ci). L'affichage est ainsi composé : la vue centrale est la vue spatiale alors que les vues temporelles sont sur les côtés. Lorsque le temps progresse, la vue spatiale est constamment mise à jour et les données temporelles affluent du coin en bas à droite vers le coin en haut à gauche (fig. 17). Cette représentation satisfait le besoin de focus+context évoqué précédemment, mais la représentation des dimensions structurelles des données n'apporte que peu d'informations.

Dans cette troisième section, nous avons pu nous rendre compte de la difficulté de représenter des données d'une part multidimensionnelles, et d'autre part temporelles. Par une approche *focus+context* ces deux caractéristiques peuvent être séparées tout en proposant une interaction permettant à l'utilisateur d'avoir une compréhension aisée. Toutefois nous n'avons pas trouvé de représentation satisfaisant ce besoin de *focus+context* tout en proposant une bonne compréhension des dimensions structurelles et les deux modèles présentés ci-dessus ne permettent pas l'édition des données.

Afin de déterminer quelle(s) représentation(s) utiliser il nous a fallu identifier parmi les dimensions extraites en 1.2.4 celles à afficher, à comparer ou à manipuler. Nous avons aussi dû prendre en compte le fait qu'un film dure en général une centaine de minutes. Les paramètres présents dans l'édition de film ne durent parfois que quelques secondes sur une durée totale d'environ 6000 secondes. Nous nous trouvions donc en présence d'une très grande quantité d'informations plus ou moins courtes dans un intervalle de temps comparativement très vaste. Il nous a fallu en conséquent déterminer une manière astucieuse de représenter l'information.

Les représentations sont spécifiques à chaque cas d'utilisation, et [CPDD03] fournit une taxonomie soulevant les interrogations à prendre en compte lors de l'utilisation de données temporelles. Notre étude a donc considéré les points suivants :

- Temps discret ou intervalles de temps : soit les instants n'ont pas de durée, soit les axes temporels sont des intervalles tels que les jours ou les années.
- Temps linéaire, cyclique ou logarithmique : si le temps est linéaire ou logarithmique, les événements vont d'un point de départ à un point d'arrivée, du passé vers le futur. S'il est cyclique, les instants sont récurrents à la manière des saisons.

De la même manière, pour représenter des données (partie structurelle) plusieurs cas se présentent :

- Représentations en 2d ou en 3d.
- Indépendance ou non des variables et prise en compte de ces dépendances.
- Changements statiques ou dynamiques : statiques si seules les actions de l'utilisateur peuvent provoquer des changements dans la visualisation, ou dynamiques si la représentation peut évoluer dynamiquement en fonction de modifications provoquées par le système.

1.4. Positionnement du travail de stage

Dans cet état de l'art nous avons exposé deux domaines différents : le contrôle de caméra en environnement virtuel et les représentations multidimensionnelles et temporelles. Dans le contexte de l'aide au montage cinématographique et dans le cas particulier d'un contrôle de caméra interactif en environnement dynamique et en temps-réel, il existe peu d'outils proposant des paramètres d'interaction de haut niveau. L'utilisation de tels paramètres requiert l'établissement de métaphores d'interaction pertinentes et de haut niveau. Il faut alors répondre à trois questions primordiales :

- Comment représenter les données et les paramètres sur ces données ?
- Comment contrôler les modifications de ces paramètres et interagir avec en temps-réel ?
- Comment évaluer la mise en œuvre d'un tel système ?

Nous exposons dans la suite du rapport le travail de recherche effectué au cours du stage et détaillons les éléments de réponse apportés à ces questions.

Chapitre 2. Visualisation de données multidimensionnelles et temporelles

Nous n'avons pas trouvé de représentation satisfaisant entièrement les besoins évoqués en 1.4 (représentation, contrôle, interaction) et nous nous sommes proposés dans notre étude d'en déterminer une ou plusieurs. La représentation la plus pertinente parmi celles présentées est celle de l'étoile et du mur en perspective (fig. 16). Même si elle ne satisfait pas notre besoin de « *focus+context* » (cette technique ne propose pas de représentation globale des paramètres manipulés au cours du temps ni de possibilités d'édition), elle propose néanmoins une représentation intéressante des données structurelles.

Le modèle de visualisation que nous présentons dans ce chapitre s'inspire grandement du modèle DIVA. Le choix des paramètres modifiables a aussi eu une grande importance et nous avons exploré les différents niveaux de contrôle possibles pour l'utilisateur afin d'en déterminer les plus pertinents. Ces niveaux de contrôle vont du très haut niveau jusqu'aux paramètres intrinsèques de la caméra, afin de satisfaire tous les besoins potentiels d'une telle application. La prise en compte de la dimension temporelle, primordiale dans l'édition de films de synthèse, a aussi grandement influencé notre choix de représentation des données. La spécificité de cette dimension et les contraintes qu'elle engendre ont été des facteurs essentiels dans la poursuite de cette étude.

2.1. Motivations

Comme nous l'avons vu en 1.3, la visualisation de données à la fois multidimensionnelles et temporelles requiert de séparer la dimension temporelle des autres dimensions. Ceci est possible grâce à la nature des données concernées : à un instant donné, les différentes dimensions prennent chacune une valeur unique. Nous avons vu en 1.3.2 deux exemples concrets issus de la littérature permettant cette séparation, chacune avec ses avantages et ses inconvénients. Ces deux exemples nous ont permis d'introduire les notions de « Overview + Detail » [DNF04] et de « Focus + context » [MB98]. Puisqu'aucun modèle existant ne satisfaisait nos besoins en termes de visualisation ni d'interaction, nous avons dû spécifier et déterminer une interface répondant à ces besoins.

Comme la littérature nous l'a indiqué, il nous était difficile d'envisager, pour des données multidimensionnelles et temporelles, une interface permettant à la fois les tâches de visualisation et d'interaction. Le travail a donc été séparé en deux parties : une partie visualisation, que nous aborderons dans ce chapitre, et une partie interaction, que nous développerons dans le chapitre 3. Enfin, nous expliquerons dans le chapitre 4 comment ces deux modèles sont complémentaires et intégrables à une unique interface.

De l'état de l'art, nous avons déterminé que l'interface de visualisation devait :

- Etre adaptée à un nombre de données représentées variable. Ce point soulève un problème d'optimisation de l'espace de représentation (2.2).
- Disposer d'une représentation de la dimension temporelle permettant d'accomplir les tâches de navigation dans le temps (2.3.1).
- Permettre de visualiser différents types de données (2.3.2).
- Offrir la possibilité d'organiser ces données : opérations de déplacement, ajout, suppression (2.3.2).
- Intégrer au mieux les possibilités qu'apporte une bonne utilisation des couleurs. La couleur ajoute une dimension à l'espace de représentation (2.3.2).
- Proposer un outil de comparaison des données (2.3.3).
- Permettre l'édition des données modifiables par l'utilisateur (2.4).

2.2. Horizon Curves et améliorations

Dans le domaine de la visualisation d'informations, les limites de l'espace de représentation deviennent une contrainte à prendre en compte. En effet, si l'utilisateur a besoin d'afficher simultanément de nombreux éléments, ceux-ci risquent de devenir difficiles à lire et à analyser. Les données binaires ne sont pas sujettes à ce problème, de par leurs deux états possibles (tout ou rien). Le cas des données qualitatives n'a pas été approfondi, mais leur lecture est rendue difficile dans le cas d'un trop grand nombre de paramètres affichés et peut être améliorée. Il en va de même pour les données bidimensionnelles (à deux valeurs pour un instant t : de R dans R^2), mais à un degré moindre. Nous nous sommes concentrés sur les données continues et l'optimisation de leur représentation afin de pallier ce problème d'optimisation de l'espace de représentation.

Les Horizon Curves, introduites dans [HKA09], proposent un modèle permettant d'afficher dans un espace restreint verticalement des données continues. Le principe est le suivant : l'échelle des valeurs (valeurs positives et négatives traitées séparément) est coupée par exemple en sa moitié. Les valeurs supérieures à cette valeur de coupe (respectivement inférieures, pour les valeurs négatives) sont enlevées de la représentation et retranscrites à partir de l'origine de l'ordonnée. Leur nouvelle valeur est : $f'(x) = f(x) - coupe$, avec *coupe* une valeur arbitraire. Un degré de transparence est attribué à la représentation des données suivant leur appartenance à une coupe (plus leur niveau de coupe est élevé, plus leur affichage est opaque ou foncé), ainsi la lecture des données ne se fait plus uniquement en fonction de f(x) mais aussi en fonction de l'opacité de la représentation.

Prenons par exemple l'évolution du Dutch Angle dans une fenêtre temporelle. La fig. 18 illustre ce paramètre en fonction du nombre de coupes qui lui est attribué. L'échelle du paramètre varie en fonction du nombre de coupes, selon la formule :

$$[min', max'] = \left[min, min + \frac{max}{nbCut}\right], avec \begin{cases} min = borne \min des\ valeurs\ possibles \\ max = borne \max des\ valeurs\ possibles \\ min' = nouvelle\ borne \min \\ max' = nouvelle\ borne \max \\ nbCuts = nombre\ de\ coupes \end{cases}$$

Dans le cas du Dutch Angle par exemple, l'intervalle des valeurs possibles est [0,180] degrés (roulis de la caméra). Si l'on applique une, deux ou trois coupes, l'intervalle des valeurs possibles sera respectivement [0,90], [0,60] et [0,45]. De cette première illustration nous pouvons extraire les informations suivantes :

- Pour un nombre de coupes supérieur à 0, il est immédiatement visible que la variable est à son maximum en (2).
- Pour un nombre de coupes supérieur à 0, la similarité entre les maxima locaux en (0), (1), (3) et (4) apparait de manière bien plus prononcée.

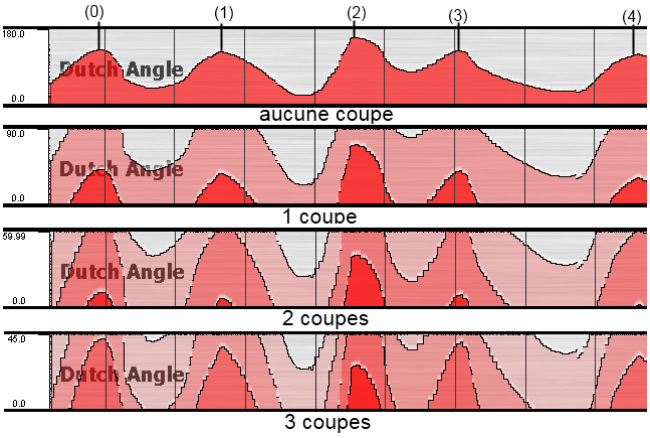


Fig. 18 : Illustration des Horizon Curves avec 0, 1, 2 et 3 coupes. L'échelle de valeurs est différente dans chaque cas et le degré de transparence traduit la valeur de la donnée

Malgré ces deux remarques, il n'est pas évident que l'utilisation des Horizon Curves permette de mieux lire des données dans un espace de représentation restreint. En effet, même si le modèle apporte quelques informations supplémentaires, sa lecture est moins intuitive, et surtout l'analyse du paramètre représenté sans coupes est tout à fait lisible.

Pour illustrer le réel intérêt de ce modèle, il faut prendre un cas moins homogène. Il existe un cas très fréquent, qui est qu'au cours d'un film le Dutch Angle prend une valeur très forte à un instant donné et varie ensuite très faiblement autour d'une valeur référence. La lecture de ces variations est alors rendue très difficile par une visualisation classique sans coupes. Grâce au modèle des Horizon Curves, ces variations apparaissent de manière lisible, tout en illustrant le fait que ce sont des valeurs faibles par leur transparence (fig. 19).

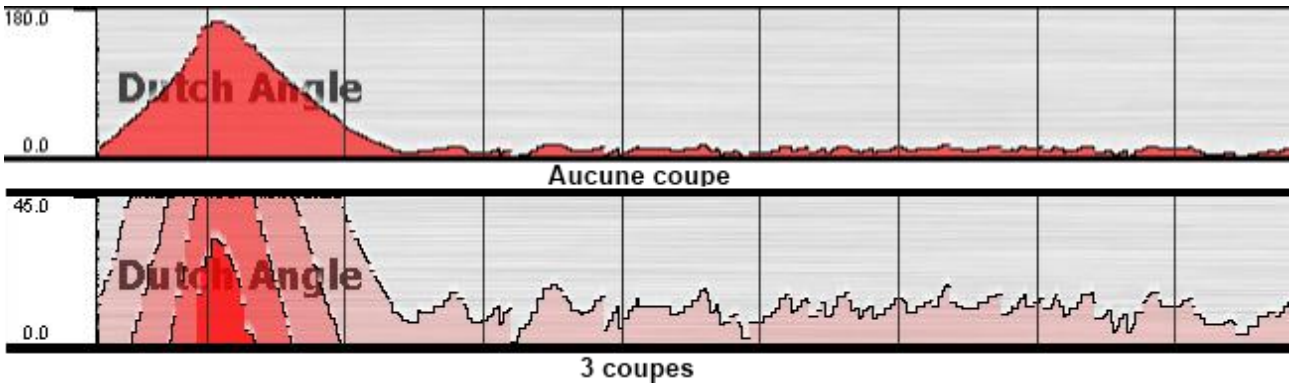


Fig. 19 : Intérêt des Horizon Curves lorsque les valeurs de la donnée sont hétérogènes : facilité de lecture des faibles variations

Nous avons de plus apporté plusieurs contributions par rapport au modèle présenté dans [HKA09] rendant plus aisée la lecture des données représentées sous cette forme (fig. 20) :

- Nous avons dessiné les contours de courbes pour que la démarcation entre deux coupes, c'est-à-dire la limite entre deux zones de couleurs de transparences différentes, soit plus prononcée.
- Afin de mieux délimiter les différents dégradés de couleur et donc les différentes transpositions de la donnée, nous avons ajouté un halo vertical à transparence progressive.
- Nous avons affiché la valeur de la donnée dans le cas où celle-ci stagne assez pour s'assurer qu'il y ait assez d'espace pour l'annoter. Cette amélioration est assez essentielle dans le contexte des Horizon Curves, car l'échelle ne correspond qu'aux valeurs de la première coupe.
- Afin de faciliter la lecture et la comparaison de hauteur (et donc de la valeur) de la donnée à des instants différents, nous avons ajouté un fond (aluminium brossé horizontalement).

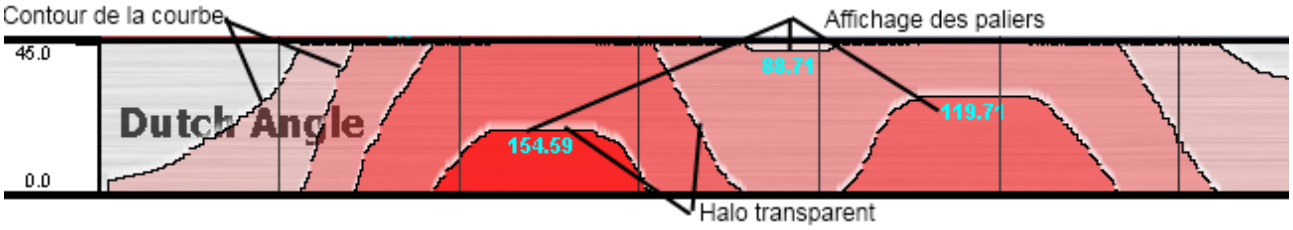


Fig. 20 : Contributions apportées aux Horizon Curves : Halo vertical, contour de la courbe et affichage des paliers

Nous insistons sur l'importance de la couleur dans ce modèle de représentation (il en va de même pour des données discrètes). En effet, l'utilisation de celle-ci permet d'ajouter à l'espace de représentation une dimension supplémentaire, très importante dans le domaine de l'information et de la visualisation [AS99].

2.3. Linear Wall et Focus + Context

Le modèle « Focus + Context » permet la visualisation de nombreuses dimensions tout en proposant l'affichage de la donnée multidimensionnelle à un instant précis. Le système DIVA n'autorise cependant que la manipulation de dimensions binaires et ne prend pas en compte les éventuelles dépendances entre ces dimensions. Il permet en revanche de visualiser les données passées et futures. Le modèle « Overview + Detail » apporte la possibilité de comparer les valeurs d'une donnée à différents instants, mais ne dispose pas de l'affichage de la donnée à l'instant présent. L'approche que nous développons dans ce chapitre permet à la fois de visualiser les données passées et futures, de comparer les valeurs d'une donnée à différents instants, et d'observer la donnée à l'instant présent.

2.3.1. La représentation de l'espace temporel

En gardant à l'esprit notre cadre applicatif (édition de scènes cinématographiques), la représentation temporelle respectant les intervalles de Allen [All91] et étant la mieux adaptée à nos besoins est celle répondant aux critères suivants (issus de la taxonomie [CPDD03]) :

- Temps discret
- Temps linéaire, non cyclique

En 1.3.2 nous avons soulevé le fait que l'intervalle temporel total peut être très élevé comparé à l'espace temporel restreint que l'utilisateur souhaiterait visualiser ou manipuler. Afin de pallier cette éventualité, nous avons mis en œuvre un moyen de naviguer dans le temps, mais aussi de spécifier la fenêtre temporelle désirée. Ces opérations de zoom / dézoom et déplacement sont possibles grâce à l'utilisation d'un BiSlider [BiSlider] (fig. 21).

Bien que le BiSlider permette d'effectuer la totalité des tâches de navigation nécessaires, nous avons choisi d'introduire d'autres métaphores plus classiques, redondantes, liées à ces tâches. Il est par exemple possible de se déplacer dans le temps en effectuant un *drag & drop* sur une courbe à la manière de nombreuses applications, par interaction tactile ou à la souris, ou bien de zoomer et dézoomer grâce à la molette de la souris.

L'espace temporel d'un film étant borné et de taille variable, et le niveau de détail requis pouvant changer selon les besoins de l'utilisateur, il était aussi nécessaire de fournir une échelle temporelle à granularité variable (fig. 22). L'échelle adapte alors son pas à la fenêtre temporelle en cours. Ce modèle est démocratisé et utilisé dans tous les outils de montage vidéo et audio.

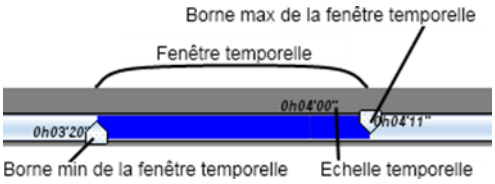


Fig. 21 : Adaptation du BiSlider aux tâches de navigation dans le temps

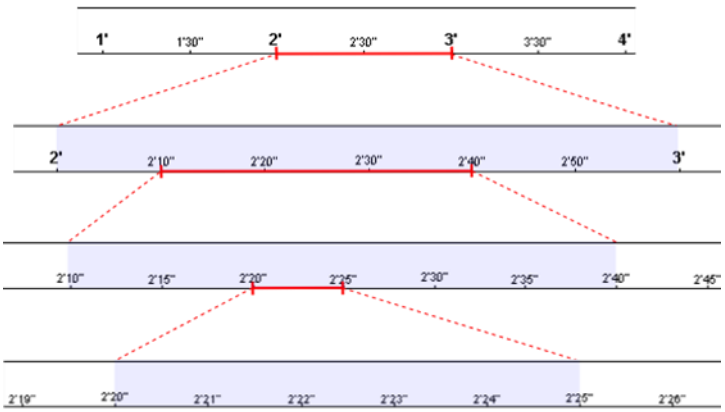


Fig. 22 : Echelle temporelle à granularité dynamique par adaptation du pas de l'échelle à la fenêtre temporelle

2.3.2. Le modèle de visualisation proposé

De la même manière que dans DIVA [MB98], nous avons séparé les valeurs passées et futures par la donnée courante. Les différentes variables ont toutes le même axe temporel et leur propre échelle de valeurs (un espace de représentation pour chaque dimension structurelle). Nous avons ajouté au modèle le Bislider afin de satisfaire les tâches de navigation dans le temps.

Le défaut majeur du système DIVA est qu'il ne permet de manipuler que des données binaires. Nos données pouvant être de nature discrète, continue, binaire, voire bidimensionnelle, il a été nécessaire de déterminer comment représenter chaque type de variable (fig. 23).

- Données continues (fig. 23.a) : de manière classique, ce type de donnée peut être représenté par sa courbe. Nous avons détaillé ce cas en 2.2 dans le cadre de l'optimisation de l'espace de représentation en introduisant les Horizon Curves et leurs améliorations.
- Données binaires (fig. 23.b) : ces données sont les plus simples à représenter, un intervalle de couleur indiquant un intervalle où la variable concernée vaut vrai.
- Données discrètes (fig. 23.c) : les données discrètes sont plus compliquées à gérer, et des choix doivent être faits concernant la représentation associée. Dans un premier temps celles-ci ont été représentées, à la manière des données binaires, par une succession d'intervalles de couleurs dégradées, en précisant la valeur de la donnée dans chaque rectangle intervalle. Par la suite, nous avons décidé de modifier cette représentation en associant à chaque valeur discrète une valeur différente sur l'échelle de la donnée, tout en conservant le dégradé et l'affichage de la valeur. Il nous a semblé que l'analyse des données discrètes était rendue plus facile par cette représentation. Il faut préciser que cette visualisation est possible uniquement parce que les données discrètes que nous avions à traiter comportaient peu de valeurs différentes. Dans le cas contraire, la lecture de celles-ci serait rendue compliquée.
- Données bidimensionnelles (fig. 23.d) : ce type de donnée a nécessité un traitement particulier pour que notre méthode de visualisation puisse être appliquée. En effet, si une donnée prend deux valeurs distinctes à chaque instant et que cette donnée possède une borne Min et une borne Max partageant la même échelle (avec $\forall\, t, Min(t) \leq Max(t)$) alors il est possible de la représenter comme une suite d'intervalles $f(t) = [Min(t), Max(t)]$.

De manière plus générale, nous avons permis à l'utilisateur de personnaliser certains paramètres liés à la visualisation selon ses propres préférences ou besoins :

- Choix de la couleur de représentation d'un paramètre.
- Ecriture pour les données binaires et discrètes de la valeur du paramètre en *inverse video* (calcul de la couleur opposée à la couleur de fond afin de l'attribuer au texte). Pour les données continues, la valeur est affichée selon le même procédé dans le cas où celle-ci est constante sur un intervalle de taille assez grande pour y inscrire cette valeur.
- Inscription du nom de chaque variable en arrière-plan de la courbe, afin de gagner de l'espace.
- Organisation des variables affichées totalement personnalisable par l'utilisateur : il est possible d'ajouter et supprimer des variables, de modifier la position de variables sélectionnées vers le haut ou le bas, ou encore d'échanger les positions de deux variables.

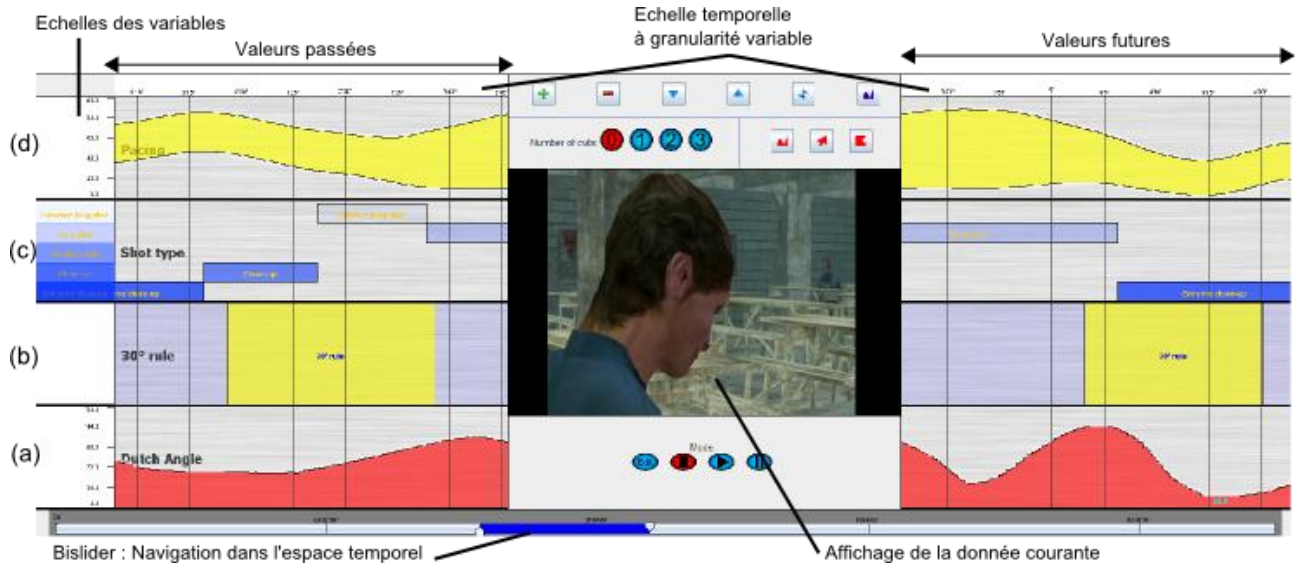


Fig. 23 : Le modèle de visualisation proposé. Permet l'affichage de données (a) continues, (b) binaires, (c) discrètes quantitatives, (d) bidimensionnelles. La dimension temporelle est séparée en deux parties : « valeurs passées » et « valeurs futures », et la donnée courante (image du film) est affichée entre ces deux parties. Les valeurs et la granularité de la dimension temporelle évoluent selon la fenêtre temporelle spécifiée.

Nous avons aussi envisagé, puisque le modèle permet de visualiser des informations futures, d'ajouter des données informatives à la liste des visualisations possibles. Ces données seraient de type qualitatif, et en langage cinématographique. Par exemple, des informations sur le suspens, le côté dramatique ou comique désiré par le réalisateur et le scénariste, ou encore sur les apparitions et actions futures des personnages. Cela pourrait se traduire par des messages informels, ou par des courbes telles que « comique de la scène ».

2.3.3. Comparaison de données

Dans le domaine de la visualisation de données, l'utilisateur peut vouloir comparer des données entre elles, si celles-ci sont comparables. La légitimité d'une comparaison dépendant du type de donnée. Pour qu'une comparaison soit possible, il faut que deux données comparées soient définies dans un même intervalle temporel et sur une même échelle de valeurs.

- Le cas des données binaires est limité : on peut s'intéresser à A ou B, A et B, A et non B et leurs inverses. Nous avons choisi d'exprimer toutes ces opérations logiques par la superposition des visualisations, comme illustré en fig. 24. La comparaison de données discrètes suit le même principe de superposition.

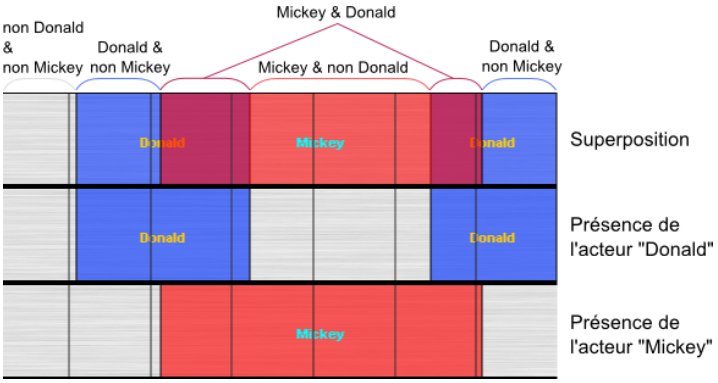


Fig. 24 : Comparaison de données binaires par superposition

- Le cas des données continues autorise de nombreuses comparaisons, et a donc été plus approfondi. La fig. 25 illustre différentes opérations de comparaison sur deux données continues comparables A et B, avec ou sans coupes. La superposition, de la même manière que pour les données binaires ou discrètes, permet d'analyser les deux courbes en simultané. Afin d'apporter au modèle des opérations de comparaison, nous avons par exemple proposé de pouvoir afficher Min(A,B) et Max(A,B). Min(A,B) détermine le minimum entre A et B à chaque instant. La couleur de la courbe correspondant au minimum en ce point est appliquée à la comparaison. Max(A,B) suit le même

procédé. De nombreuses autres comparaisons peuvent être utilisées, selon les besoins de l'utilisateur : différences, dérivées nièmes, valeurs absolues, etc.

Plusieurs détails méritent d'être relevés : on s'aperçoit que la superposition d'Horizon Curves donne un résultat difficile à lire et à comprendre, alors que le max et le min restent lisibles. Il faut donc, lorsqu'on ajoute une comparaison, vérifier sa compatibilité avec les Horizon Curves et désactiver ce mode de visualisation en fonction de cette compatibilité.

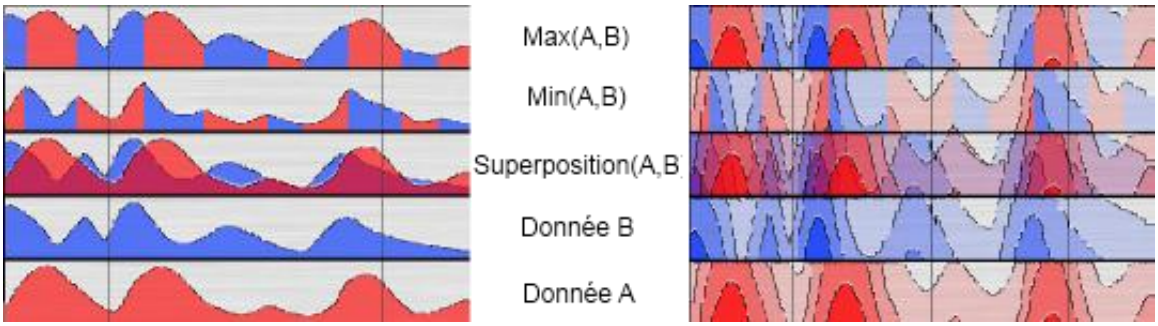


Fig. 25 : Tâches de comparaison entre deux données continues A et B couplées avec les Horizon Curves

Dans le modèle que nous avons proposé, il est possible d'ajouter des comparaisons de données comme on ajouterait un paramètre. L'interface permet donc une composition des données et de leurs comparaisons.

Nous avons exploré en détail le processus de visualisation de données de plusieurs types et les tâches de navigation et d'analyse spécifiques aux données évoluant au cours du temps. En 1.3.2 nous avons aussi évoqué les tâches d'édition de données. La section suivante aborde une première approche de ces tâches, en s'inspirant des modèles déjà existants et couramment utilisés et en les adaptant à nos spécifications.

2.4. Tâches d'édition de données

Afin de permettre l'édition des données, l'utilisateur doit être capable de sélectionner un intervalle temporel et d'y appliquer des modifications. Ce processus est illustré en fig. 26. Nous avons rendu la sélection (qui s'effectue soit à la souris soit par un périphérique tactile) plus aisée par la magnétisation des bornes de l'intervalle de sélection d'une part à la grille temporelle et d'autre part aux changements de valeur brusques de la donnée à modifier. Une fois cette sélection effectuée, l'utilisateur peut soit attribuer une nouvelle valeur à tout l'intervalle en manipulant un curseur (Slider) (fig. 26.a), soit effectuer une interpolation personnalisée entre les deux bornes de l'intervalle par l'intermédiaire d'un outil de transitions (fig. 26.b, fig. 26.c). Nous avons choisi de contrôler cette interpolation en paramétrant les valeurs aux bornes min et max de l'intervalle sélectionné, ainsi que deux points de contrôle d'une courbe de Bézier de degré 2.

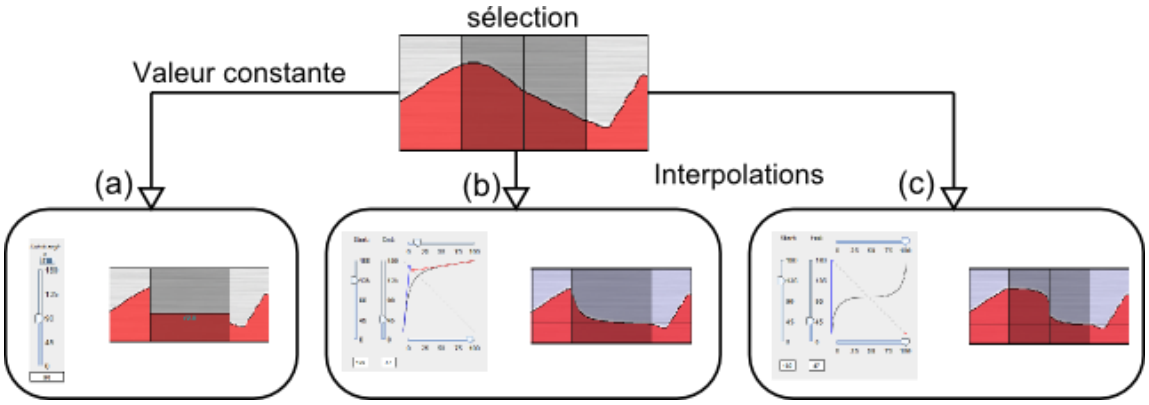


Fig. 26 : Notre processus d'édition d'une variable sur un intervalle temporel sélectionné, (a) par slider pour attribution d'une valeur constante, ou (b), (c) par spécification des valeurs de départ et d'arrivée ainsi que de l'interpolation à effectuer entre ces deux points

Ce modèle est un standard de l'édition de données. Nous avons déterminé et mis en œuvre un second modèle, plus intuitif pour l'utilisateur. En effet, la spécification des points de contrôle peut s'avérer fastidieuse. Ce modèle, de concept simple, consiste à dessiner (à la souris, ou au doigt) l'évolution de la donnée au cours du temps. Cela nécessite néanmoins une grande zone d'interaction afin d'obtenir une précision satisfaisante. Dans le cas où un grand nombre de paramètres est affiché simultanément, l'espace d'interaction est bien trop réduit pour envisager un dessin précis. Pour permettre à l'utilisateur de disposer d'une surface de taille suffisante, nous avons donc ajouté une fenêtre contextuelle déroulante s'affichant à la demande de celui-ci et présentant la donnée à modifier par-dessus l'interface (fig. 27). Cette zone contextuelle permet ainsi de modifier les données dans l'intervalle temporel en cours d'affichage.

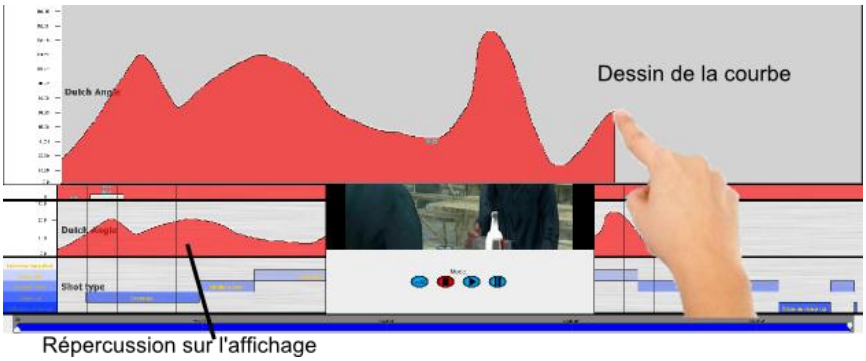


Fig. 27 : Notre métaphore de dessin de courbe au doigt ou à la souris pour des données continues

Cette métaphore de dessin est similaire pour les données discrètes (A chaque valeur discrète est associée un intervalle de valeurs continu) et les données binaires (d'intérêt limité, celles-ci ne possédant que deux intervalles : celui pour lequel la donnée vaut faux et celui pour lequel la donnée vaut vrai). L'utilisation d'un périphérique tactile permet de dessiner des données bidimensionnelles (fig. 28), ce qui n'est pas possible avec une souris. La métaphore associée est la suivante :

$$[min', max'](t) = [val_{down}, val_{up}](t_{up})$$

$$avec \begin{cases} val_{down} \text{ la valeur de la donnée associée à la position du doigt supérieur} \\ val_{up} \text{ la valeur de la donnée associée à la position du doigt inférieur} \\ t_{up} \text{ l'instant temporel associé à la position du doigt supérieur} \end{cases}$$

L'instant courant est guidé par l'abscisse du doigt supérieur, et les valeurs de la donnée sont associées aux ordonnées des doigts.

Fig. 28 : Notre métaphore de dessin de courbe à deux doigts pour des données bidimensionnelles

Le modèle présenté dans ce chapitre satisfait les critères que nous avions spécifiés en 1.3. En effet, celui-ci :

- Comporte un processus de visualisation de la dimension temporelle,
- Comporte un processus de visualisation des dimensions structurelles,
- Permet d'accomplir les tâches de navigation, analyse et édition de données,
- Offre une vue Focus+Context.

Nous avons évoqué l'utilisation d'un périphérique tactile dans le but de manipuler cette interface. Il n'est pourtant pas évident que ce critère soit essentiel. En effet, à part pour l'édition d'un paramètre bidimensionnel (qui pourrait d'ailleurs être scindé en deux paramètres uni dimensionnels) toutes les tâches sont réalisables à l'aide d'un périphérique standard tel que la souris. De plus, les interfaces telles que les tables tactiles ne sont efficaces, si ce n'est essentielles, que dans le cadre d'applications interactives et collaboratives. Même si notre modèle est orienté mono-utilisateur, l'adaptation au tactile s'explique par le fait qu'il n'est qu'une brique du modèle plus global, orienté multi-utilisateur, que nous définirons par la suite. La lecture de données représentées par des courbes nécessite une orientation spécifique à une personne en particulier. Nous avons par conséquent décidé que chaque utilisateur devait posséder sa propre visualisation, orientée selon sa position par rapport au périphérique d'interaction.

Puisque l'un des axes de recherche était de permettre à plusieurs utilisateurs d'interagir de manière simultanée et sur les mêmes données, nous avons dû déterminer une métaphore de visualisation et d'interaction spécifique. En effet, le modèle de visualisation proposé précédemment ne permet ni l'interaction simultanée de plusieurs utilisateurs de manière confortable, ni la spécification des contraintes d'édition dues aux dépendances entre les variables. La signalisation de ces dépendances est pourtant un élément majeur de l'édition collaborative de paramètres interdépendants. Dans le chapitre 3, nous proposons un modèle d'édition de données multidimensionnelles et temporelles, permettant une interaction collaborative multi utilisateurs sur table tactile.

Chapitre 3. La représentation en étoile pour une interaction multidimensionnelle collaborative

Le contrôle des paramètres et leur modification ont soulevé deux problèmes : nous avons dû déterminer le moyen d'interaction et le moyen de contrôler les paramètres. Etant donné que les interfaces standard (souris, clavier) n'apportent pas satisfaction lorsqu'il est nécessaire de travailler simultanément sur plusieurs dimensions, l'utilisation d'une tablette tactile autorisant la manipulation synchrone de différents paramètres (tablette multipoints) a été explorée. Ce périphérique offre entre autres une portabilité intéressante et est adapté aux représentations graphiques et symboliques. L'atout majeur des tables tactiles multipoints est qu'elles permettent d'envisager des interactions collaboratives, qui sont au cœur même de notre problématique. L'emploi d'un tel périphérique ayant ses spécificités (ambiguïté lors de la sélection et données masquées par la main par exemple), nous avons dû prendre en compte celles-ci lors des choix de représentation.
Le contrôle des paramètres pour sa part doit prendre en considération différentes sortes de données (discrètes, continues, linéaires ou non, etc.). Nous présentons dans ce chapitre la métaphore proposée permettant à l'utilisateur ou aux utilisateurs d'interagir avec les paramètres.

Comme nous avons pu le voir en 1.3, même s'il existe de nombreuses façons de représenter des données multidimensionnelles, celles prenant en compte la composante temporelle sont plus rares. Cette dimension particulière nécessite de permettre l'observation de changements au cours du temps, tout en proposant une vue d'ensemble des autres dimensions à un instant précis. De plus, nous avons envisagé des tâches de comparaison ; ce critère a donc été pris en compte dans l'élaboration d'une métaphore d'interaction.

On trouve dans la littérature quelques modèles permettant de visualiser des données à la fois temporelles et multidimensionnelles. [FCI05] propose par exemple d'intégrer la dimension temporelle à la représentation en étoile (fig. 29). Ce genre de modèle permet d'observer les données selon l'axe temporel de manière plus ou moins efficace. Cependant, il est très difficilement envisageable d'interagir avec les données. C'est pour cette raison que nous avons choisi de séparer l'aspect visualisation de l'aspect interaction. Le modèle présenté en 2.2 remplit les tâches de navigation, analyse et manipulation de données. La seconde étape est de concevoir le module permettant l'interaction avec ces données.

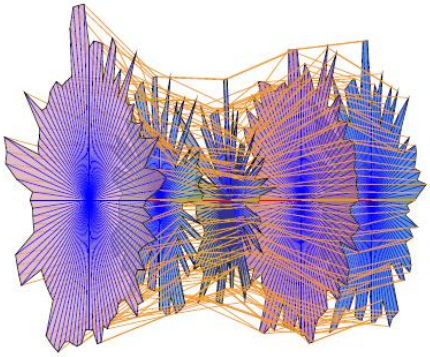


Fig. 29 : Exemple de représentation intégrant la dimension temporelle à une représentation multidimensionnelle

3.1. Motivations

Bien que les métaphores d'interaction présentées dans le chapitre 2 permettent la modification des données, elles sont mal adaptées à notre contexte. En effet, les tâches d'édition que nous avons identifiées requièrent que le modèle possède plusieurs caractéristiques. Nous avons par conséquent proposé une représentation en étoile collaborative, qui satisfait aux besoins suivants :

1. **Types variés** : le modèle doit permettre de représenter et éditer des paramètres de différents types.
2. **Collaboration et manipulation** : le modèle doit proposer une interaction collaborative multi-utilisateurs et autoriser des tâches de manipulation des paramètres dans l'espace de représentation.
3. **Indication des dépendances** : le modèle doit permettre la spécification de contraintes induites par les dépendances entre variables. Ces spécifications doivent non seulement contraindre les tâches d'édition mais aussi fournir aux utilisateurs des avertissements leur permettant de comprendre ces contraintes et leurs conséquences.
4. **Tâches de comparaison** : le modèle doit offrir la possibilité de comparer des jeux de paramètres à des instants différents.
5. **Dimension temporelle** : le modèle doit intégrer la dimension temporelle.

3.2. Présentation du modèle

Une spécification forte de ce travail était que notre proposition permette une interaction collaborative sur des paramètres de haut niveau afin de contrôler une caméra. L'utilisation d'une interface tactile multi points (donc orientée multi-utilisateurs) se justifie alors par ses propriétés intrinsèques. La fig. 30 illustre cette représentation, sans prendre en compte l'aspect collaboratif. Cette représentation permet l'utilisation de données continues (fig. 30.a), discrètes et qualitatives (fig. 30.b) et bidimensionnelles (fig. 30.c).

Le modèle que nous avons proposé comporte les caractéristiques suivantes et satisfait notre première motivation (**Types variés**) :

- Chaque branche représente un paramètre dont la valeur peut être modifiée par un bouton associé, le long de la branche, à la manière d'un slider classique.
- La manipulation des données discrètes est aimantée aux marqueurs de l'échelle.
- Les données bidimensionnelles sont manipulées par deux boutons de modification : un pour la borne min de l'intervalle, et un pour la borne max, en respectant (min<=max). Les utilisateurs ont aussi la possibilité de spécifier pour ces données un intervalle de taille minimum, et l'intervalle des valeurs est représenté par un segment de couleur et épaisseur personnalisables (fig. 30.c).

- L'utilisation d'une échelle logarithmique, préconisée pour certains paramètres, est possible (fig. 30.e).
- Le nom du paramètre, ainsi que la valeur courante (qui se met à jour en fonction des modifications effectuées) sont spécifiés à l'extrémité de chaque branche (fig. 30.d).
- Afin de ne pas rendre illisibles ni difficiles à manipuler les valeurs proches du centre de l'étoile, celui-ci est délimité par un polygone reliant l'origine des branches. Les branches commencent alors à une distance r_{centre} du centre de l'étoile (fig. 30.f).
- Les utilisateurs peuvent personnaliser l'apparence des branches afin de différencier celles-ci et les reconnaitre visuellement : couleur de la branche, de l'échelle, des labels de l'échelle des boutons modificateurs par exemple.
- La forme de l'étoile est affichée en transparence (fig. 30.g). Ceci nous a permis, comme nous le verrons en 3.4, de tirer profit de l'avantage de cette représentation en étoile : l'analyse d'un ensemble de paramètres est immédiate et intuitive par simple aperçu de la forme dessinée et nous a permis de proposer une métaphore de comparaison de données.

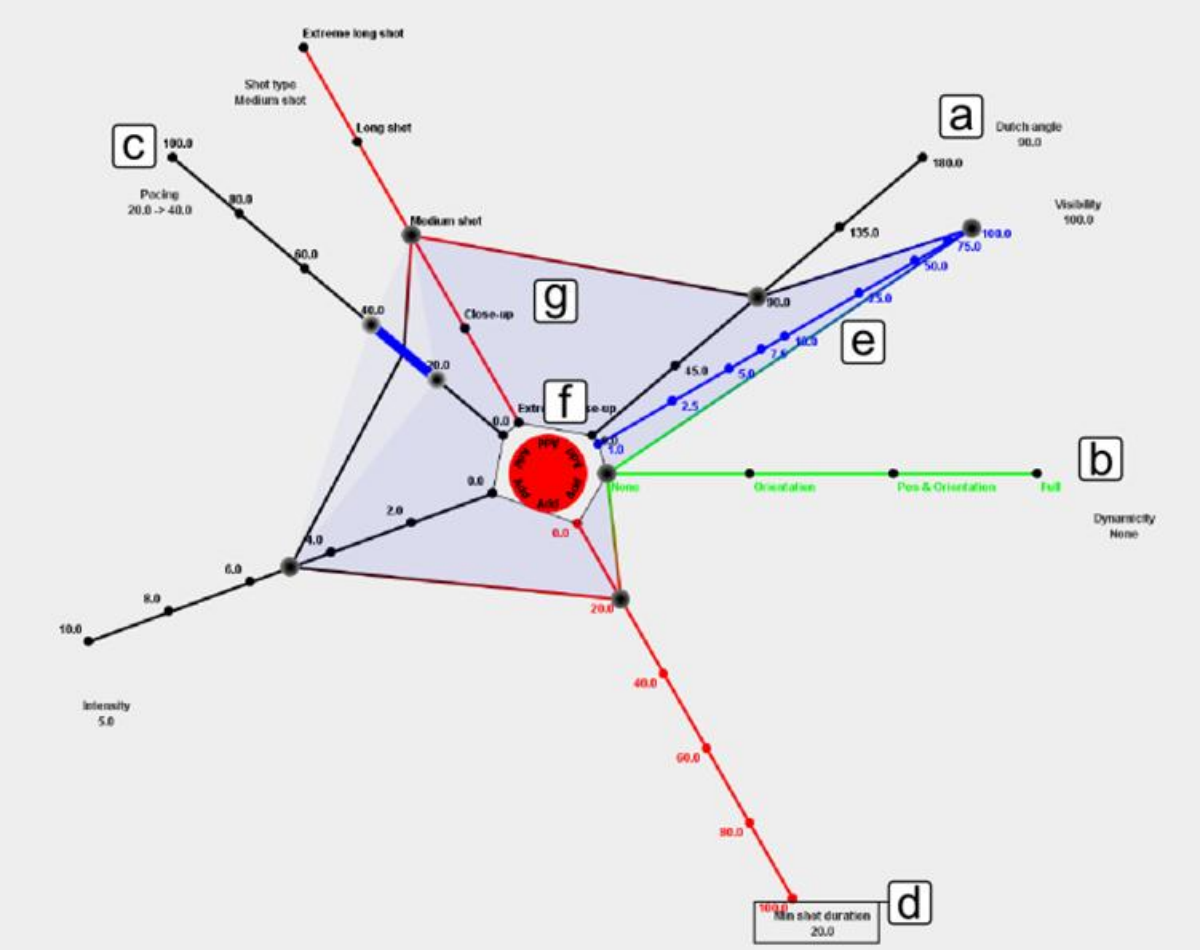


Fig. 30 : Notre modèle de la représentation en étoile non collaborative. Il permet :

-La manipulation de données : (a) continues, (b) discrètes et qualitatives, (c) bidimensionnelles.
-(d) l'affichage des valeurs courantes
-(e) L'utilisation d'échelles logarithmiques

et propose :

-(f) D'établir une zone centrale de l'étoile critique où l'édition des données serait compliquée
-(g) De représenter le glyphe de la configuration des paramètres sous la forme d'une étoile en transparence

3.2. Interaction collaborative

Comme évoqué précédemment, un avantage majeur de la représentation en étoile est qu'elle est facilement adaptable à un contrôle collaboratif. Nous allons dans cette section voir comment le modèle peut permettre à plusieurs utilisateurs d'interagir simultanément et donc de résoudre notre deuxième motivation (**Collaboration et manipulation**).

3.2.1. Orientation vers l'utilisateur

Nous avons orienté les différents labels (échelles, noms et valeurs des paramètres, fenêtres contextuelles) de l'angle de la branche thêta + π/2 (fig. 31). Ceux-ci sont alors disposés idéalement pour une personne spécifique autour de la table de travail (à la manière de [SVFR04]). Les différents utilisateurs peuvent ainsi travailler chacun sur les paramètres les concernant, tout en ayant une vue globale des opérations des autres utilisateurs, les éléments de la représentation illustrant ces paramètres n'étant pas cachés par leurs mains. Il est important de noter que, sur le même scénario, l'utilisation des coordonnées parallèles soulèverait des problèmes d'occultation des données, ce que permet d'éviter la représentation en étoile.

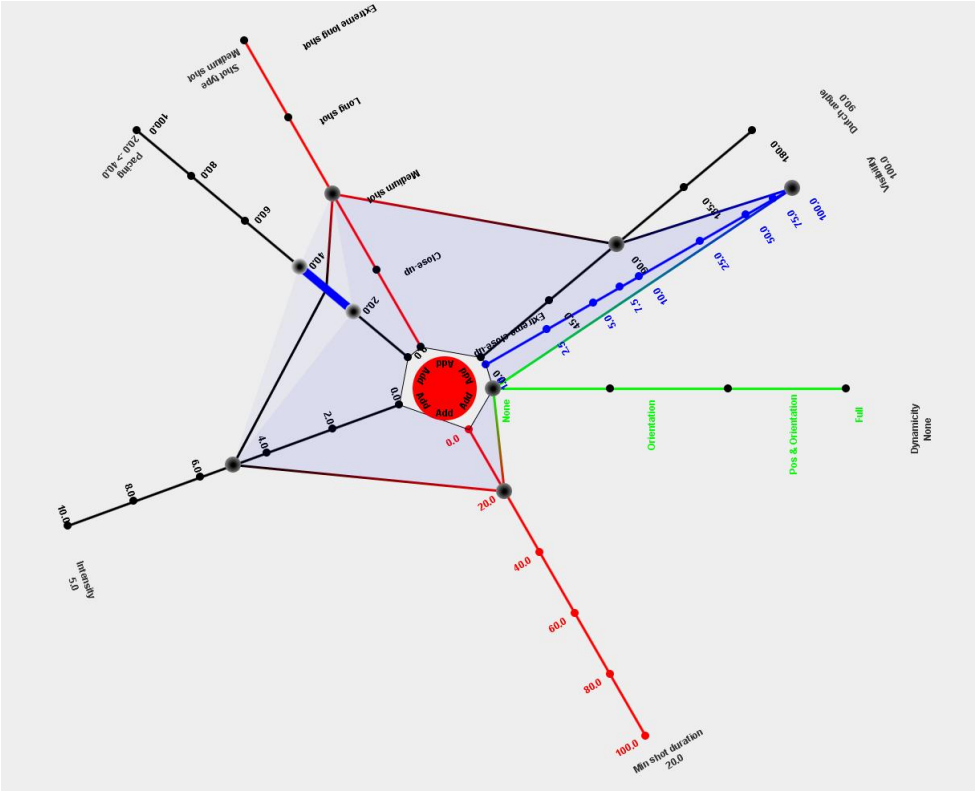


Fig. 31 : Notre modèle de représentation en étoile collaborative : Les éléments sont orientés selon la position des utilisateurs

3.2.2. Manipulation des branches

Dans le contexte d'une interaction collaborative, les opérations d'ajout, déplacement et suppression de branches sont des manipulations courantes :

- **Ajout de branches** :
  Si un paramètre modifiable n'est pas affiché, un utilisateur peut vouloir ajouter celui-ci à la visualisation. La métaphore que nous avons mise en œuvre est « tirer une branche » : si un utilisateur effectue un drag & drop depuis le centre de l'étoile (disque rouge) vers lui, la future branche s'affiche en segment discontinu jusqu'à faire la longueur d'une branche de l'étoile et propose la liste des branches qu'il est possible d'ajouter lorsque l'utilisateur relâche la branche en cours de création

(fig. 32). Cette liste des branches possibles est mise à jour selon les branches ajoutées ou supprimées précédemment.

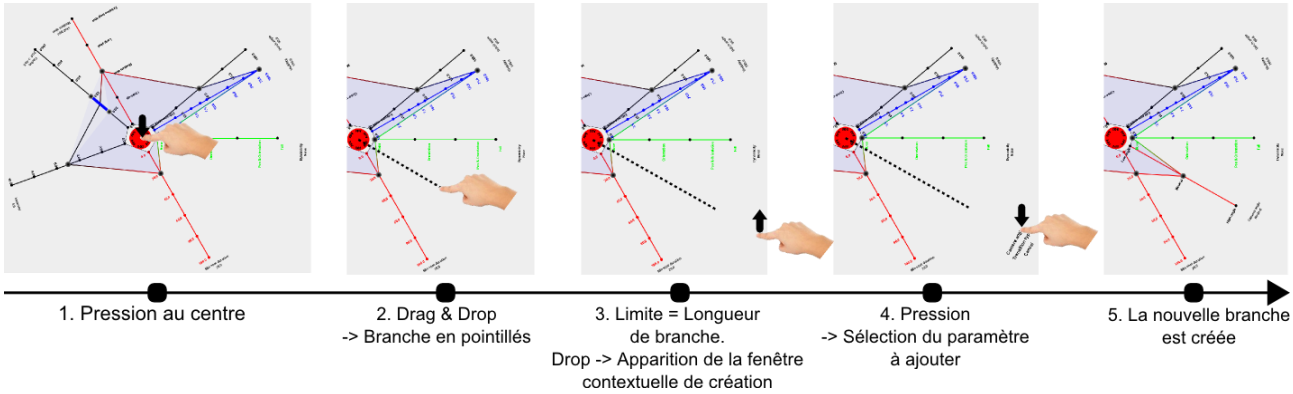


Fig. 32 : Notre métaphore d'ajout de branche

- **Déplacement de branches :**
  Si un utilisateur U1 utilisait une branche à un instant donné mais désire « donner » cette branche à un utilisateur U2, ou si U2 veut « prendre » la branche à U1, nous proposons qu'il appuie longuement sur le nom du paramètre afin d'activer le déplacement de la branche. Celle-ci est alors mise en surbrillance et suit l'orientation du doigt la manipulant (fig. 33). Au cours du déplacement, la liste des branches est mise à jour selon l'angle de chacune. Ainsi la forme de l'étoile et le lien entre les branches voisines sont à recalculer lorsque cet ordre est modifié.

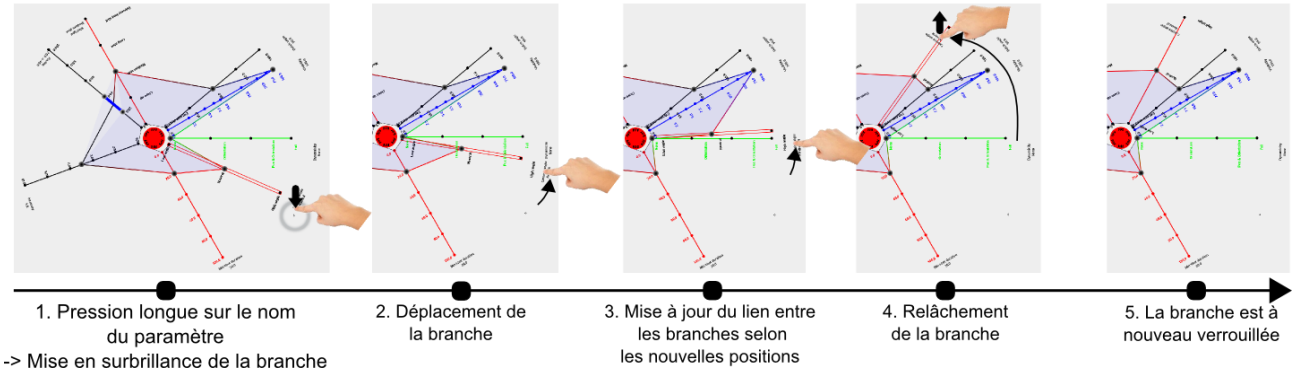


Fig. 33 : Notre métaphore de déplacement de branche

- **Suppression de branches :**
  Si un paramètre est temporairement inutile pour la totalité des utilisateurs, nous avons offert la possibilité de le supprimer de l'affichage afin de libérer de l'espace de visualisation : cela est effectué en double-cliquant sur le nom de la branche à supprimer afin d'ouvrir la fenêtre contextuelle correspondante (fig. 34). De la même manière que lors des deux opérations précédentes, la liste des branches est mise à jour en fonction de la suppression.

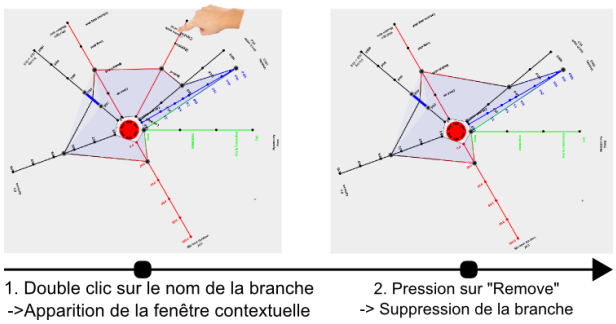


Fig. 34 : Notre métaphore de suppression de branche

Nous avons détaillé dans cette section les contributions que nous avons apportées afin de permettre une interaction collaborative sur des données multidimensionnelles à un instant donné. Néanmoins, les variables peuvent être interdépendantes (ce qui est le cas dans le cadre du contrôle de caméra). Il nous a donc été nécessaire d'une part de spécifier ces dépendances sous la forme de contraintes, et d'autre part de déterminer les avertisseurs visuels permettant de signaler aux utilisateurs l'effet de cette contrainte sur une manipulation.

3.3. Représentation des contraintes

Nous avons abordé en 2.4 les dépendances entre variables et la nécessité de représenter ces dépendances. Nous appelons par la suite une *règle* l'expression de la dépendance entre deux variables modifiables. Nous développons dans cette section les méthodes que nous avons déterminées et ajoutées au modèle afin de satisfaire la troisième motivation (**Indication des dépendances**). Nous spécifierons tout d'abord différentes contraintes sur les variables dues aux interdépendances, puis développerons nos contributions quant aux avertisseurs que nous avons déterminés afin de communiquer ces contraintes aux utilisateurs.

3.3.1. Spécification des contraintes

Les règles que nous avons mises en œuvre opposent deux variables v1 et v2. Pour l'application de ces règles, nous proposons de définir une relation variable dominante / variable dominée. La variable dominante peut être spécifiée librement, modifiant les valeurs de la variable dominée si besoin, alors que la modification de la variable dominée est contrainte par la valeur de la variable dominante.

La table 3 référence quelques exemples de règles sélectionnées et mises en œuvre (cf. 1.2.4) :

Variable dominante			Variable dominée			Contraintes	
Nom	**Définition**	**Valeur**	**Nom**	**Définition**	**Valeur**		
Min Shot Duration	Durée minimale des plans	Continue [0,100]	**Pacing**	Intervalle de durée des plans	Bidimensionnel-le continue [0,100]	Min(Pacing)>=Min Shot Duration	
Intensity	Proximité de la caméra aux acteurs	Continue [0,10]	**Shot Type**	Type de point de vue	Discrète {Ex. Close-up, Close-up, Medium Shot, Long Shot, Ex. Long Shot}	Intensity	Shot Type
						[0,1[	Ex. Long-shot
						[1,3[	Long Shot
						[3,7[	Medium Shot
						[7,9[	Close-up
						[9,10[	Ex. Close-up
Dynamicity	Dynamisme de la caméra	Discrète {Aucun, Orientation, Position & Orientation, Total}	**Transition Type**	Type de transition	Discrète {Aucune, Coupe, Continue, Dissolution, Apparition en fondu, Disparition en fondu}	Dynamicity	Transition Type
						Total	Aucune
						{Aucun, orientation, Position & Orientation}	{Coupe, Continue, Dissolution, Apparition en fondu, Disparition en fondu}
Dynamicity	Dynamisme de la caméra	Discrète {Aucun, Orientation, Position & Orientation, Total}	**Dutch Angle**	Inclinaison de la caméra par rapport à la ligne d'horizon	Continue [0,180]	Dynamicity	Dutch Angle
						Aucun	Statique durant le plan
						{Orientation, Position & Orientation, Total}	Modifiable

Table 3 : exemples de règles et de leurs spécifications mettant en relation une variable dominante et une variable dominée

Cette liste n'est pas exhaustive. En effet, de nombreuses autres règles ont été identifiées : L'angle (de «tilt ») de la caméra est lié aux relations entre personnages : expression de la dominance ou de l'affinité. L'isolation

ou non d'un acteur dans la scène n'est spécifiable que si l'acteur en question est présent dans cette même scène. Ou encore, les relations entre acteurs ne sont spécifiables que si les acteurs concernés sont tous présents dans la scène. Nous nous focalisons ici sur les variables dites **persistantes**, c'est-à-dire qui sont observables et modifiables à tout instant de la scène, quels que soient les acteurs présents. Nous discuterons en 3.6 des représentations liées aux variables sur les acteurs, celles-ci étant des cas particuliers : ces variables sont dites **contextuelles**.

Nous avons dû déterminer et associer à cet ensemble de règles une visualisation pertinente. Nous avons identifié deux sortes de contraintes, provoquées par ces règles. Nous appelons *contrainte bloquante* une règle rendue active par la tentative de modification de la variable dominée mais interdite par la variable dominante, et *contrainte modifiante* une règle rendue active par la modification de la variable dominante influant la valeur de la contrainte dominée. Nous avons estimé nécessaire de différencier ces deux cas, et surtout de fournir à l'utilisateur (respectivement aux utilisateurs) des avertissements visuels différents.

Afin d'attirer l'attention d'un utilisateur sur des composants de l'interface en particulier, une technique courante en visualisation est de griser les autres composants. Dans notre cas d'une règle faisant intervenir deux variables, et donc deux branches de la représentation en étoile, il serait naturel de griser les autres branches afin de ne faire ressortir que les deux branches concernées par la règle active. Cependant, notre contribution étant à caractère collaboratif, il nous a paru important qu'une contrainte devant alerter un ou plusieurs utilisateurs ne gêne pas l'interaction des autres utilisateurs. Cette stratégie d'avertissement visuel standard n'est par conséquent pas applicable à une interaction collaborative multi utilisateurs. Nous avons donc déterminé deux avertisseurs visuels, selon que la règle est activée par une contrainte bloquante ou par une contrainte modifiante.

3.3.2. Contraintes bloquantes

Une contrainte bloquante traduit le fait qu'un paramètre, de par l'existence d'une règle, ne peut pas être modifié. Ce cas se produit lorsque l'on essaie de modifier la valeur d'une variable dominée et que la nouvelle valeur est interdite par la variable dominante associée. Prenons l'exemple de la règle liant « Pacing » à « Min Shot Duration ». Cette règle indique, avec la priorité donnée à Min Shot Duration, que la borne Min du Pacing doit être inférieure ou égale à Min Shot Duration. Comme le montre la fig. 35, la modification de la borne Min dans l'intervalle des valeurs autorisées (dans l'exemple, tant que Min(Pacing)>=26.4) ne provoque pas d'avertissement visuel. Mais :

$$min_{voulu}(Pacing) < val(MinShotDuration) \Rightarrow \min(pacing) = val(MinShotDuration)$$

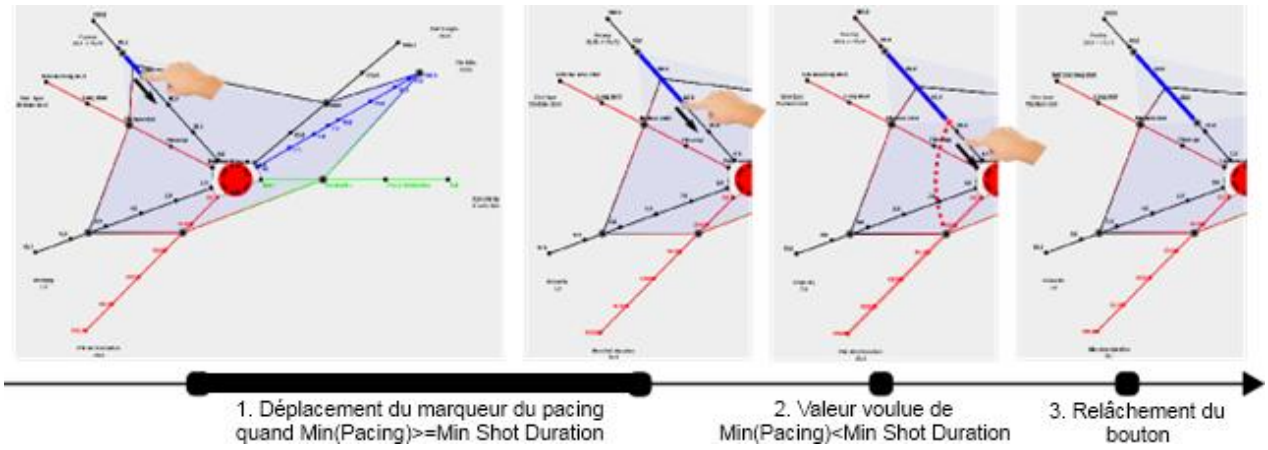


Fig. 35 : Notre contribution permettant d'interdire la modification d'une variable et l'avertisseur visuel associé : (1) L'attribution d'une valeur légale à la variable dominée n'entraine pas d'avertissement visuel, (2) Le bouton de modification de la variable dominée reste à sa dernière valeur légale et l'avertisseur visuel de blocage s'active en cas de tentative d'attribution d'une valeur illégale à la variable, (3) Disparition de l'avertisseur visuel de blocage au relâchement du marqueur

En effet, le marqueur de la borne Min du Pacing ne peut pas prendre de valeur inférieure à celle de Min Shot Duration et se retrouve donc verrouillé à cette valeur. Néanmoins, l'augmentation de cette borne Min du Pacing étant une opération légale, cette action est toujours possible. Le verrouillage du marqueur est accompagné d'une mise en surbrillance des deux branches concernées et d'un arc rouge joignant la valeur bloquante et la valeur bloquée. Afin de ne pas traverser l'étoile en son centre dans le cas où l'angle séparant les deux branches serait proche de π, nous proposons de déterminer cet arc (plutôt qu'un segment) comme détaillé en fig. 36 selon un point de contrôle et les points de départ et d'arrivée :

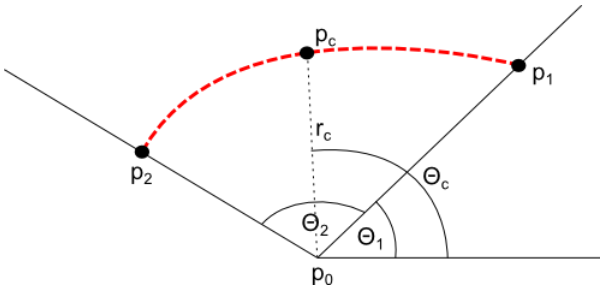


Fig. 36 : Calcul de l'arc reliant les deux valeurs des paramètres concernés par la violation d'une règle et passant par un point de contrôle intermédiaire. L'angle de cet arc vaut au maximum π.
L'arc relie les points p_1 et p_2 et passe par le point de contrôle p_c

On a : $\begin{cases} r_c = \max\big(d(p_0,p_1), d(p_0,p_2)\big), avec\ d(.,.) la\ distance\ euclidienne\ entre\ deux\ points \\ \theta_c = \left|\left(\frac{\theta_1-\theta_2}{2}\right)\right| + \min(\theta_1,\theta_2) \end{cases}$

Avec : $r_c = Rayon\ du\ point\ de\ contrôle\ par\ rapport\ au\ centre\ de\ l'étoile\ p_0$
$\theta_c = Angle\ du\ point\ de\ contrôle$
$p_1, p_2\ les\ points\ des\ valeurs\ des\ deux\ branches\ concernées$
$\theta_1, \theta_2\ les\ angles\ des\ deux\ branches\ concernées$

De plus, si $|\theta_c - \theta_1| > \frac{\pi}{2}$ alors $\theta'_c = \theta_c - \frac{\pi}{2}$ est le nouvel angle du point de contrôle. Ceci permet d'être certain que la courbe dessinée sera la plus courte possible. On déduit de $r_c\ et\ \theta'_c$ les coordonnées du point de contrôle p_c.

3.3.3. Contraintes modifiantes

Une contrainte modifiante traduit le fait que la modification d'un paramètre dominant entraîne la modification du paramètre dominé associé. En gardant l'exemple de la règle liant « Pacing » à « Min Shot Duration », et comme illustré en fig. 37, la modification de Min Shot Duration entraine la modification de la valeur min du Pacing si la nouvelle valeur du premier est supérieure à la valeur du second.
Nous pouvons de plus observer que lorsque Max(Pacing)-Min(Pacing)<MinRange (avec MinRange la taille minimale de l'intervalle de la donnée bidimensionnelle), étant donné que la borne Min ne peut plus être rapprochée de la borne Max, les deux bornes sont modifiées afin de respecter la contrainte sur MinRange (fig. 37.3).

Le calcul de l'arc reliant les deux points de chaque branche est le même qu'en 3.3.2, mais sa couleur a été changée en bleu afin de mieux différencier les deux sortes de contraintes. Il faut aussi noter le fait que l'utilisateur manipulant la variable dominée en perdra le contrôle si plusieurs utilisateurs modifient simultanément les deux variables concernées par une règle et l'enfreignent

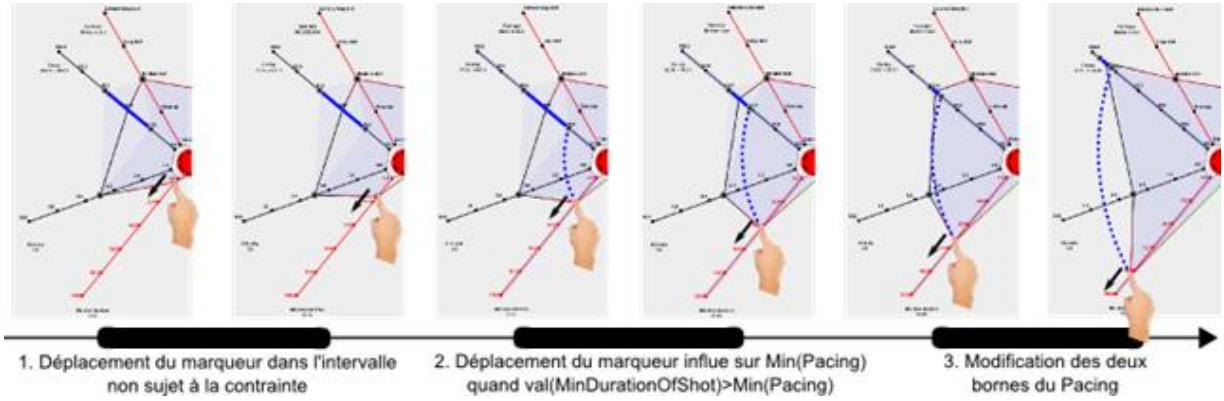


Fig. 37 : Notre contribution permettant d'avertir que la modification d'une variable modifie la valeur d'une variable dépendante : (1) Un changement de valeur de la variable dominante n'entraine pas d'avertissement visuel si celui-ci n'affecte pas la variable dominée associée, (2) La variable dominée est automatiquement modifiée lorsque la modification de la variable dominante l'impose, (3) La modification de la variable dominante peut entrainer la modification des deux bornes de la variable dominées, si celle-ci est de type bidimensionnel

3.3.4. Contraintes sur intervalles

Dans le cas où, pour chaque valeur de la variable dominante, la règle propose un intervalle de valeurs possibles pour la variable dominée, nous avons proposé d'illustrer cet intervalle lors de la tentative de modification de la valeur dominée (fig. 38).

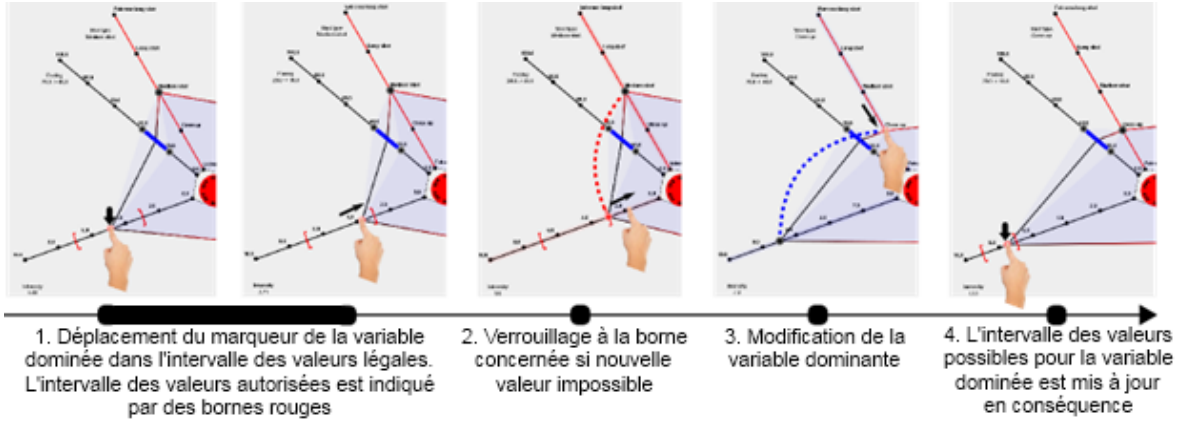


Fig. 38 : Notre contribution permettant d'afficher l'intervalle des valeurs possibles d'une variable dominée lors de la manipulation de celle-ci : (1) Lors de la modification d'une variable dominée restreinte à un intervalle de valeurs, cet intervalle est affiché, (2) le déplacement du marqueur est limité à cet intervalle (avertisseur visuel de blocage), (3), (4) l'intervalle des valeurs possibles de la variable dominée est modifié lorsque la modification de la variable dominante associée spécifie un nouvel intervalle de valeurs légales

Si l'utilisateur souhaite allouer à la variable dominée une valeur impossible, l'alerte visuelle des contraintes bloquantes est activée (fig. 38.2). Lors de la modification de la valeur dominante, l'alerte visuelle des contraintes modifiantes est activée. La variable dominée prend alors pour nouvelle valeur la borne du nouvel intervalle des valeurs possibles qui est la plus proche de sa valeur précédente (fig. 38.3). Lors des modifications futures de la variable dominée, l'intervalle de ses valeurs possibles est mis à jour (fig. 38.4).

Nous avons présenté notre spécification de règles liant deux variables dépendantes, en détaillant par l'exemple le processus de visualisation par alertes visuelles utilisé pour représenter ces règles. Dans le cas particulier d'un contrôle multi-utilisateurs, nous avons proposé des méthodes innovantes afin de signaler des événements bloquants ou modifiants aux utilisateurs concernés, sans perturber l'interaction des autres participants. Cependant, nous soulevions en introduction de ce chapitre que cette représentation ne propose pas de visualisation de la dimension temporelle. Même si l'analyse des paramètres au cours du temps est rendue possible par le Linear Wall présenté en 2.2, il nous semble intéressant de tirer profit d'un des atouts majeurs de la représentation en étoile : l'analyse immédiate et intuitive de la valeur globale de l'ensemble

des variables grâce à la forme de l'étoile. C'est cette piste que nous allons explorer en 3.4.

3.4. Tâches de comparaison

Les tâches de comparaison possibles sont multiples. Les utilisateurs peuvent par exemple, après avoir observé une modification de la configuration d'une scène, vouloir connaitre les paramètres qui ont été modifiés et qui sont responsables de ces changements ; ou encore vouloir reproduire un ensemble de paramètres d'une scène pour les appliquer à une nouvelle scène à un instant différent.

Nous utilisons l'hypothèse forte que le glyphe de la représentation en étoile fournit une vue intuitive de l'ensemble des paramètres afin de répondre à notre quatrième motivation : offrir la possibilité de comparer des jeux de paramètres à des instants différents (**Tâches de comparaison**). Les utilisateurs peuvent, grâce à notre modèle, observer deux configurations de paramètres Star0 et Star1 respectivement à des instants t0 et t1. Notre proposition est illustrée en fig. 39.

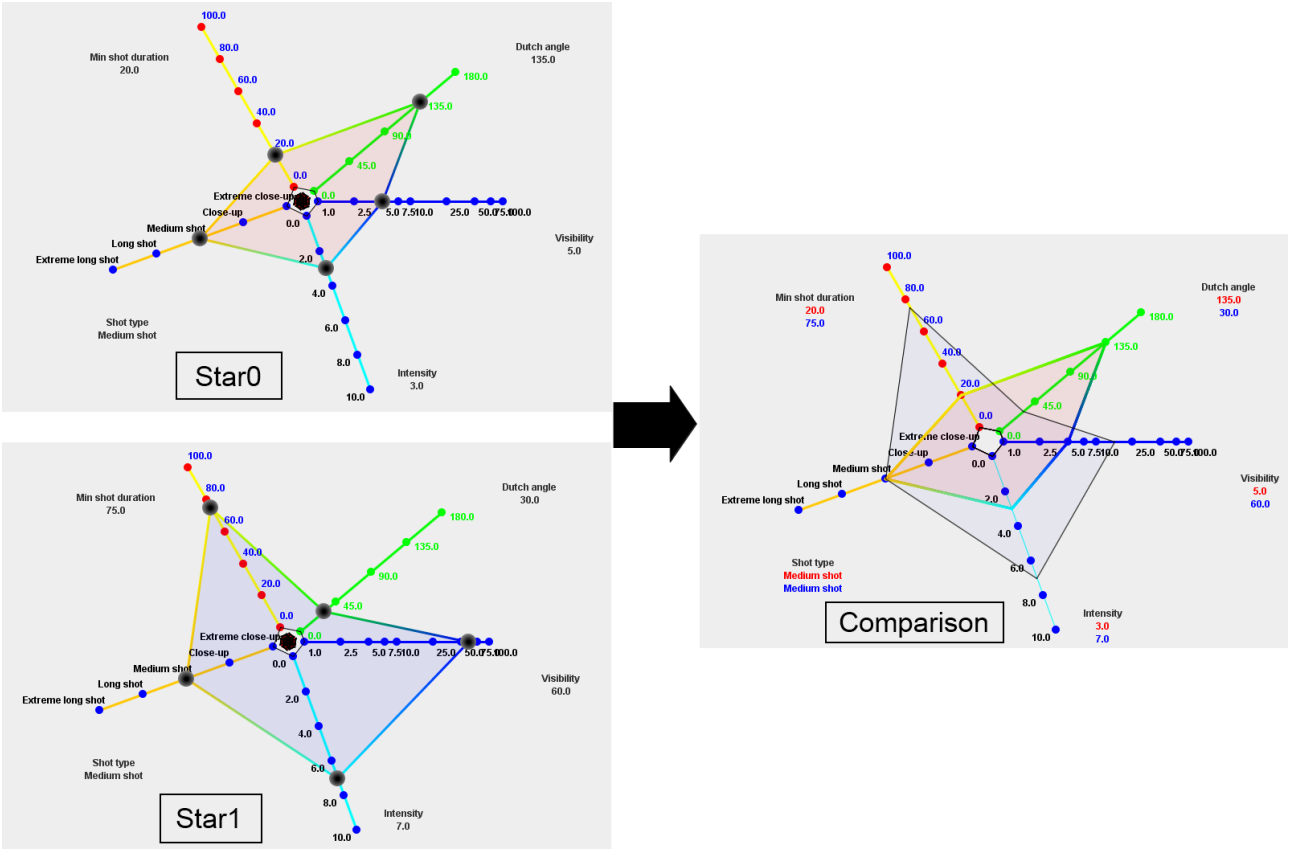


Fig. 39 : Comparaison de données à des instants différents par superposition des glyphes

Chacune des étoiles est indépendante et peut être modifiée selon notre modèle présenté précédemment. La comparaison des deux configurations affiche les deux étoiles superposées. Les différentes valeurs sont indiquées sous le nom de la variable en respectant la couleur du glyphe de chaque configuration. Afin que les configurations soient comparables, nous avons proposé d'organiser les branches de la comparaison selon l'organisation de Star0. Les valeurs en t1 sont alors projetées sur les axes de Star0 : le glyphe de Star1 est par conséquent adapté à la configuration de Star0. Ainsi, la modification des positions des branches de Star1 n'a pas d'impact sur la comparaison (fig. 40). A l'inverse, la réorganisation des branches de Star0 impacte le glyphe de la comparaison (fig. 41).

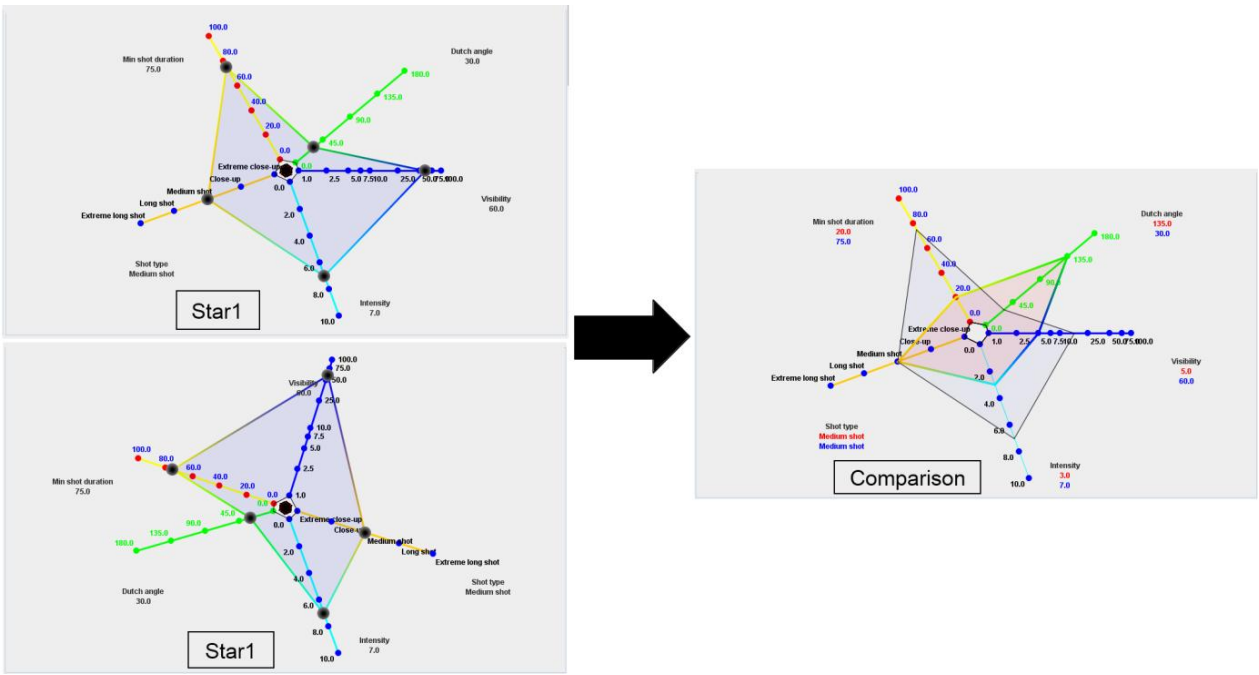


Fig. 40 : Star0 guidant l'organisation des branches, la comparaison donne un résultat identique pour deux organisations de Star1 différentes

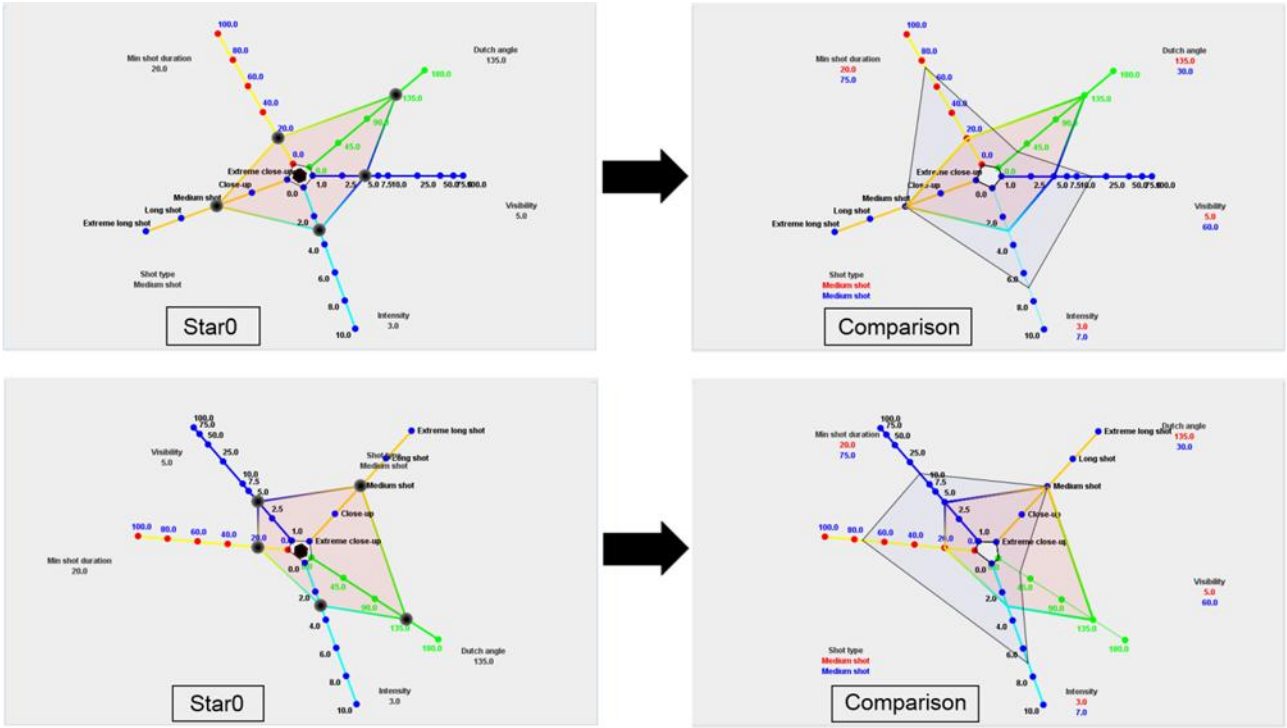


Fig. 41 : Deux organisations des branches de Star0 différentes donnent deux configurations de comparaison différentes

Pour que les configurations restent comparables si des modifications sur les branches sont effectuées, nous avons estimé que la comparaison ne pouvait conserver que les variables affichées à la fois dans Star0 et dans Star1.

De nombreuses améliorations, dérivées de ce modèle, sont envisageables. Par exemple, nous envisageons de permettre un système de sauvegarde et chargement de configuration, afin que les utilisateurs puissent fabriquer leur propre base de données de configurations types. Grâce au glyphe de l'étoile, ces configurations sont facilement reconnaissables. Ou encore, étant donné que ce modèle ne permet pas de visualiser un glyphe à chaque instant, nous prévoyons d'incorporer à l'interface de visualisation présentée dans le chapitre 2 (*Linear Wall & Focus+Context*) une donnée particulière : la succession des glyphes, à pas dynamique selon la granularité temporelle de la vue en cours, comme illustré en fig. 42. Cette amélioration permettrait d'intégrer la dimension temporelle à notre modèle et de satisfaire notre cinquième motivation (**Dimension temporelle**).

Fig. 42 : Proposition d'ajout de la succession des glyphes en fonction du temps à l'interface de visualisation

3.5. Le modèle CollaStar

Notre modèle de la représentation en étoile pour une interaction collaborative satisfait les tâches d'édition et de comparaison que nous avions spécifiées. Toutefois ses deux points faibles sont : (1) la représentation de l'axe temporel et des évolutions des valeurs des différents paramètres selon cette dimension particulière et (2) l'intégration des paramètres non modifiables par les utilisateurs. En effet, les variables peuvent être classées en deux catégories : les variables *informatives* et les variables *modifiables*.

Nous définissons une variable comme *informative* si la visualisation de son évolution au cours du temps est possible, mais que les utilisateurs ne peuvent pas en modifier la valeur. Par exemple, la distance entre deux acteurs dans une scène peut être intéressante à visualiser, mais ne peut plus être modifiée une fois que la scène a été tournée. Nous définissons de même une variable comme *modifiable* si elle peut être à la fois visualisée et modifiée par les utilisateurs. Les variables informatives sont dans notre cadre applicatif celles que l'on peut extraire d'une scène existante et les variables modifiables sont celles qui permettent le contrôle de caméra. Seules les variables modifiables ont un intérêt à être affichées dans l'interface d'interaction en étoile, mais les variables informatives doivent néanmoins être visualisables par les utilisateurs.

Afin de palier à ces deux lacunes, nous nous sommes proposé de coupler notre représentation en étoile avec notre interface de visualisation décrite dans le chapitre 2. Ce couplage permet aux différents utilisateurs d'interagir sur la même représentation en étoile tout en ayant leur propre vue de l'ensemble des paramètres les intéressant (fig. 43). Dans ce cas, chaque utilisateur affiche des courbes différentes et personnalise son affichage. Les valeurs des données sont évidemment les mêmes sur chaque affichage, et une contrainte doit être respectée afin que ce système soit valide : seul un utilisateur peut contrôler la dimension temporelle. En effet, pour que la représentation en étoile ait un sens, les utilisateurs doivent tous observer le même instant.

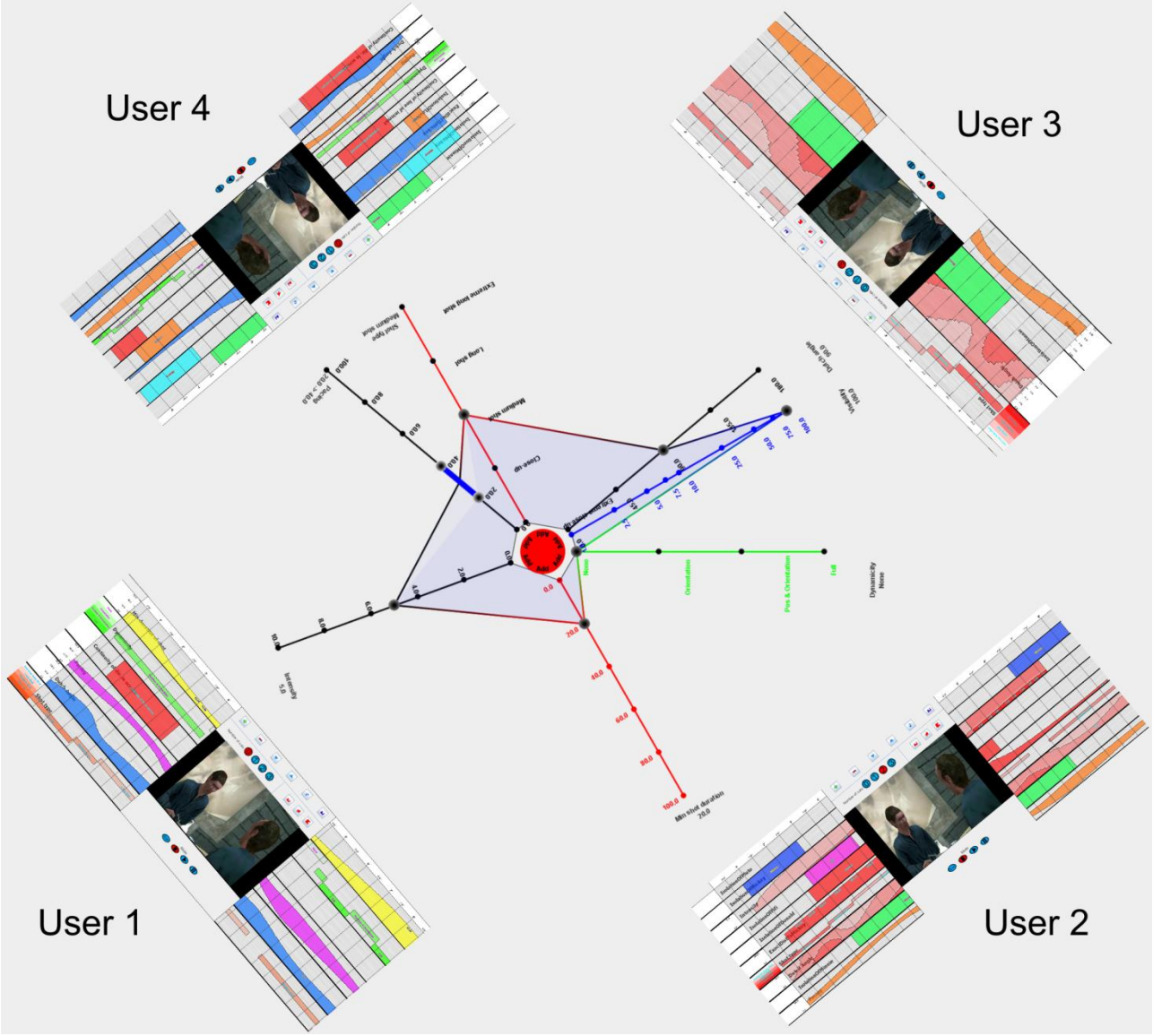


Fig. 43 : Notre modèle Collastar : chaque utilisateur dispose de sa propre interface de visualisation et tous interagissent en collaboration sur la représentation en étoile située au centre de la table interactive

Notre modèle permet à un nombre variable d'utilisateurs d'interagir sur les branches de l'étoile. Les interfaces de visualisation des données sont créées selon les désirs des utilisateurs et peuvent être déplacées autour de l'étoile, en adaptant leur orientation à leur position. En plus d'intégrer la dimension temporelle à la représentation en étoile, CollaStar permet aux utilisateurs de visualiser les données informatives les intéressant sans encombrer l'espace de représentation des autres utilisateurs. Notre modèle est aussi extensible : nous avons estimé qu'il ne fallait pas nous restreindre au contrôle de caméra, mais que l'on pouvait aussi contrôler d'autres paramètres tels que les éclairages ou les prises de son. Nous permettons ainsi qu' un caméraman, un technicien lumière et un technicien du son interagissent sur la même représentation en étoile et contrôlent chacun les paramètres qui sont propres à leur domaine. La disposition du modèle permet alors à tous les intervenants d'avoir une vue d'ensemble des modifications effectuées par chacun, d'observer ces modifications et de disposer en temps-réel du résultat sur leur interface de visualisation *Linear Wall & Focus+Context* personnelle.

Une autre piste que nous avons explorée et mise en œuvre à propos de l'affichage de la dimension temporelle dans le modèle (motivation **Dimension temporelle**) consiste à conserver une trace des valeurs passées proches. En spécifiant une durée d'enregistrement des valeurs t_{rec} il est possible d'afficher la progression des valeurs des paramètres sur l'intervalle de temps $[t_{cur} - t_{rec}\ ,\ t_{cur}]$, avec t_{cur} l'instant courant (fig. 44). Le polygone translucide indique les valeurs récentes qui ont été successivement attribuées à la valeur du paramètre modifié durant la période d'enregistrement. Si le paramètre n'est plus modifié pendant l'intervalle de temps spécifié, la valeur en t_{cur} sera égale à la valeur en $t_{cur} - t_{rec}$ et le polygone translucide sera d'aire nulle. Nous avons envisagé d'utiliser un dégradé de transparence afin de signifier l'âge des valeurs enregistrées (plus la valeur est âgée, moins celle-ci sera visible). Si cette méthode apporte une première intégration de la dimension temporelle au sein du modèle, elle est limitée par l'intervalle de temps alloué à l'enregistrement des valeurs passées. Enfin, il faudrait étudier l'utilité de cette méthode en considérant tout particulièrement la surcharge ou confusion visuelle qu'elle peut entrainer.

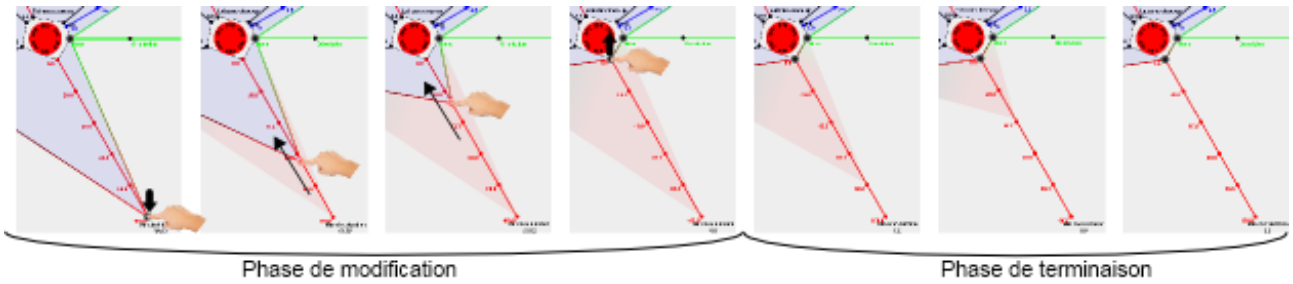


Fig. 44 : décomposition de notre proposition intégrant partiellement la dimension temporelle à notre représentation en étoile : durant la phase de modification les valeurs passées sont conservées et affichées en un polygone translucide. Durant la phase de terminaison (après que l'utilisateur ait terminé de modifier la variable, les valeurs passées se rapprochent de la valeur courante jusqu'à l'atteindre

Notre modèle satisfait la totalité des motivations que nous avions identifiées. Néanmoins, il comporte un défaut non négligeable : afin que les utilisateurs disposent d'un certain confort visuel, il est nécessaire que la définition des éléments constituant l'interface soit très haute. Notre interface de visualisation comportant énormément de détails, nous en avons conclu que la surface de travail devait être de taille conséquente et l'affichage de haute qualité. Ce genre de table interactive a aujourd'hui un coût très élevé, et la contrainte sur les dimensions de la table réduit considérablement la mobilité du système. Puisque les *Linear Wall & Focus+Context* sont spécifiques à un utilisateur en particulier et ne nécessitent pas de satisfaire les contraintes liées au collaboratif, il peut être envisagé que chacun possède son propre écran de visualisation afin que la table interactive propose uniquement la représentation en étoile.

3.6. Améliorations

Le modèle que nous proposons dans ce chapitre, même s'il nous semble tout à fait convenir à notre cadre applicatif, est incomplet. Il faudrait en effet encore approfondir les points suivants :

- Si les données persistantes (définies à tout instant) sont parfaitement gérées par la représentation en étoile, la représentation des données contextuelles (définies partiellement sur des intervalles temporels) est encore à étudier. Par exemple, tout au long d'un film, les acteurs sont visibles dans une scène par intermittence. Etant donné que le contrôle de caméra prend en compte des opérateurs de haut niveau sur les acteurs, tels que la dominance, l'affinité ou l'isolation, il nous faudra déterminer une manière efficace de gérer ces informations.
- Le modèle n'est actuellement pas adapté aux données binaires. Attribuer à chaque variable binaire une branche serait une perte d'espace de visualisation. Il nous faut par conséquent déterminer comment représenter ce type de données, en tirant profit de leur nature. Il est par exemple envisagé

de disposer les variables binaires en cercles concentriques au centre de l'étoile (fig. 45), afin que celles-ci puissent être visualisées et manipulées par tous les utilisateurs, tout en occupant un minimum d'espace.

- Toujours à propos du gain d'espace de visualisation, et afin de ne pas surcharger visuellement les utilisateurs, il nous a semblé judicieux de permettre d'afficher ou non les différents labels de l'étoile comme les noms des branches ou les valeurs intermédiaires des échelles de valeurs. Nous envisageons ainsi de n'afficher que ceux concernant les variables en cours de modification.
- Une autre piste que nous voulions explorer consistait à définir et proposer une métaphore permettant une manipulation des données avancée : nous avons envisagé de considérer l'accélération instantanée du doigt de l'utilisateur pour déterminer si le bouton de réglage du paramètre devait être aimanté par une valeur de l'échelle ou non. En effet, cette métaphore nous a semblée à la fois naturelle et très utile : si l'utilisateur désire régler avec précision une valeur, il déplacera son doigt lentement. Si une valeur approximée lui convient, alors en déplaçant rapidement son doigt le curseur s'aimantera aux paliers de l'échelle et entrainera une discrétisation dynamique de la variable.

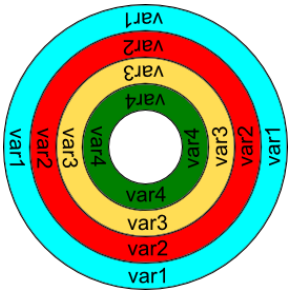


Fig. 45 : Proposition de disposition des variables binaires en cercles concentriques au centre de la représentation en étoile

Dans ce chapitre nous avons présenté le modèle de visualisation que nous avons déterminé. Celui-ci permet la manipulation de données multidimensionnelles par plusieurs utilisateurs, grâce à la possibilité d'orienter les composants vers chaque utilisateur suivant leurs positions, et par les opérations de manipulation des branches de l'étoile. De plus, la comparaison de configurations à des instants distincts est rendue intuitive grâce au glyphe représentant une configuration de paramètres. La limitation de la représentation en étoile, mais qui n'est pas propre à ce modèle, est que la dimension temporelle est difficile à traiter. Nous avons cependant proposé plusieurs façons de gérer cette dimension particulière et problématique et nous comptons évaluer expérimentalement ces techniques afin d'estimer leur utilité et leur pertinence. Cette phase expérimentale sera développée dans le chapitre 5.

Chapitre 4. Intégration au système Cinesys

Afin d'évaluer nos propositions, nous les avons adaptées à notre cadre applicatif qu'est le contrôle de caméra en environnement virtuel en les intégrant, ou plutôt en les faisant communiquer, avec le système Cinesys. Ce système peut recevoir en entrée des valeurs de paramètres de haut niveau et rendre en sortie la scène 3D ogre calculée selon ces paramètres d'entrée. La scène étant générée en temps-réel, Cinesys nous a semblé tout à fait approprié à l'intégration de notre contribution.

Nous différencions deux sortes d'édition des paramètres à transmettre au système :

- Modifications en temps-réel : dans le cadre d'une interaction collaborative sur des données temporelles, il est essentiel que la valeur temporelle courante soit la même pour tous les utilisateurs afin que l'interaction reste cohérente. Nous avons ainsi décidé qu'un seul utilisateur pouvait contrôler la dimension temporelle (modes *Lecture* ou *Stop*) : un unique utilisateur qui possède le contrôle sur la dimension temporelle, peut activer le mode *Lecture*. Le temps défile alors, ainsi que l'affichage des paramètres dans les *Linear Wall & Focus+Context* de chaque utilisateur. Les modifications effectuées sur l'étoile impactent directement les valeurs de chaque paramètre à chaque instant (un instant = 1/30ème de seconde). Notre système capture à chaque pas de temps les valeurs des paramètres et détermine lesquelles ont été modifiées. Ces paramètres modifiés sont envoyés au

moteur Cinesys qui prend en compte les nouvelles valeurs pour calculer les points de vue sur la scène. Cinesys passe alors d'un système contrôlant les caméras automatiquement à un système interactif.

- Modifications à posteriori : Cinesys rend possible la modification de paramètres dans un intervalle de temps. Nous appelons modification à postériori les tâches d'édition de paramètres sur un film déjà réalisé. Nous avons vu que le *Linear Wall & Focus+Context* proposé permettait de sélectionner un intervalle temporel afin de modifier les valeurs des données dans cet intervalle. Il suffit dans ce cas d'envoyer au moteur l'intervalle sélectionné et les valeurs des paramètres dans celui-ci. Cinesys détermine alors si les points de vue dans l'intervalle temporel sélectionné sont modifiés et transmet aux *Linear Wall & Focus+Context* la nouvelle scène.

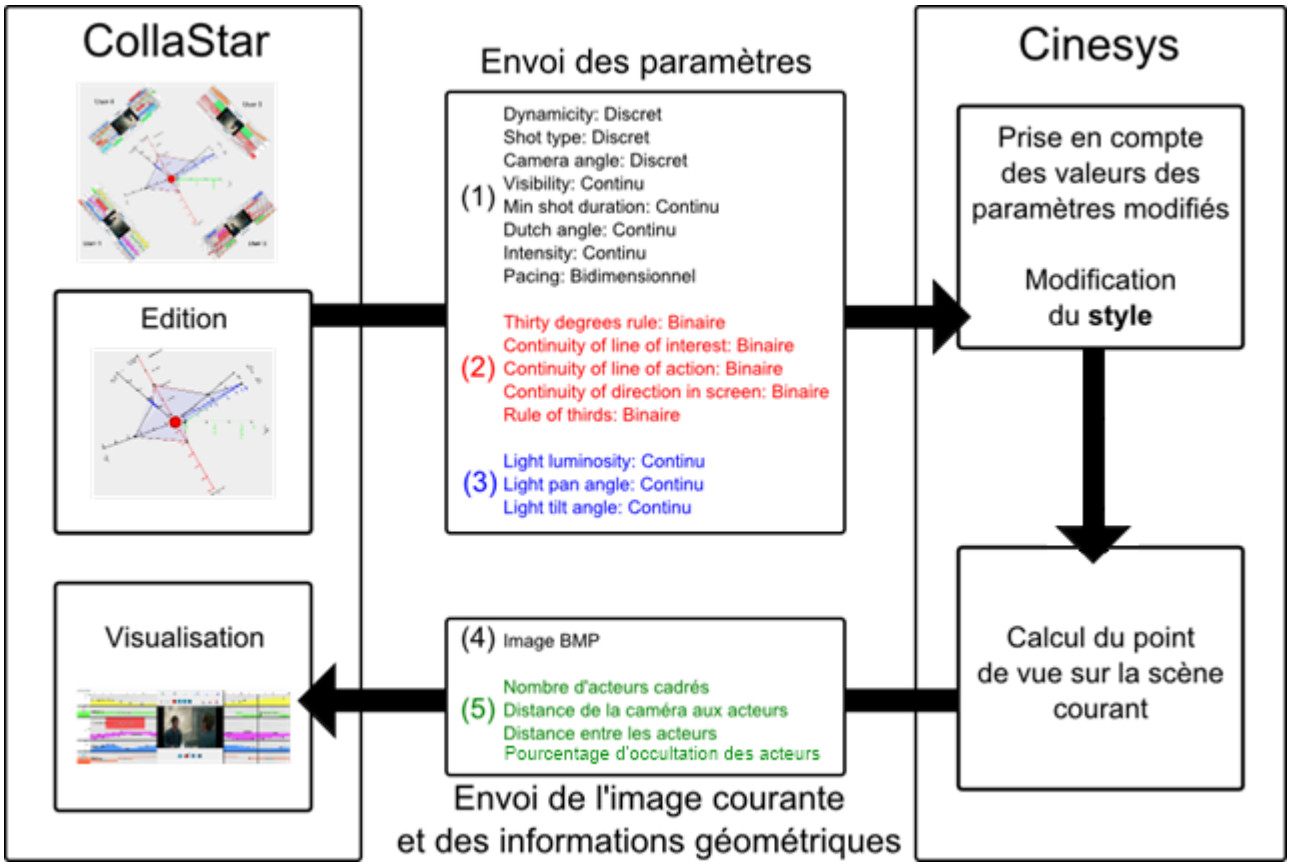


Fig. 46 : Communication bidirectionnelle entre Cinesys et Collastar effectuée à chaque pas de temps :
(1) Paramètres de haut niveau contrôlant la caméra implémentés, (2) paramètres de haut niveau contrôlant la caméra à implémenter,
(3) Paramètres de haut niveau contrôlant les lumières,
(4) Flux de bits contenant l'image courante, (5) Informations géométriques de la scène 3D non modifiables par les utilisateurs

Afin de permettre les communications entre le moteur (en C++) et notre modèle (implémenté en Java) nous avons mis en place un système de transmission des données bidirectionnel (fig. 46). Ceci est réalisé par le biais de sockets : les modifications de paramètres sont simplement codées en un message texte et décodées par le moteur. La visualisation de la scène en cours est permise en envoyant à chaque pas de temps l'image courante de la scène Ogre. Notre implémentation permet de communiquer des images de taille convenable (800*600) tous les 1/30ème de secondes avec un décalage à l'affichage de quelques centièmes de secondes. Une implémentation en JNI apporterait certainement de meilleures performances, mais par manque de temps nous avons décidé d'opter pour une transmission par sockets.

Chapitre 5. Expérimentations et évaluations prévues

L'évaluation d'une IHM (Interface Homme-Machine) est une étape essentielle de la validation des concepts et des métaphores d'interaction proposées. Les critères d'utilité classiques sont : performances, capacité fonctionnelle et assistance. Les critères d'utilisabilité sont la satisfaction, le taux d'erreurs, les facilités d'apprentissage, l'utilisation et la mémorisation de la part des utilisateurs. Dans le cadre de ce travail, les critères discriminants sont : performance, apprentissage et satisfaction.

Nous avons déjà mené des évaluations informelles qui ont permis d'identifier les bénéfices du modèle :

- Facilité de modification des données.
- Facilité de manipulation des branches.
- Apport des avertisseurs visuels de dépendances.
- Facilité de lecture dans un contexte collaboratif.
- Vue globale de la donnée et comparaison de données.

Nous devons maintenant évaluer plus précisément et individuellement ces bénéfices.

Dans ce contexte, il est nécessaire de définir à la fois les modalités d'évaluation de représentations et des interactions, et de proposer scénarii d'utilisation par rapport à un ensemble de tâches donné. Si les scénarii et les modalités sont bien identifiés et répertoriés dans les sections suivantes, nous n'avons pas eu l'opportunité de compléter et de remonter les résultats d'évaluation dans ce manuscrit. Ces évaluations sont actuellement en cours et seront présentées dans le cadre de la soutenance du travail réalisé.

Nous avons focalisé notre évaluation sur les trois aspects suivants :

5.1. Pertinence de la représentation en étoile par rapport aux coordonnées parallèles

Une alternative de choix, par rapport à la représentation en étoile, est de considérer l'utilisation de coordonnées parallèles. Ces deux représentations partagent : la multi dimensionnalité, l'interaction individuelle sur les dimensions représentées par des sliders et la possibilité d'agir sur plusieurs paramètres en même temps.

Cette évaluation, qui traite de l'utilisabilité du modèle de représentation en étoile, ne nécessite pas que les utilisateurs manipulent des données spécifiques à la cinématographie. Nous prendrons donc des données moins spécifiques afin que leur compréhension soit immédiate pour les participants. Nous utiliserons pour cette évaluation des variables simples composant un budget telles que : salaires, allocations, loyer, charges, nourriture, essence, vacances. On peut classer ces variables dans deux catégories : revenus et dépenses.

Nous nous attachons donc à comparer ces deux modèles de représentation et d'interaction dans un contexte collaboratif.

Les deux éléments de comparaison que nous avons prévus portent sur :

1. **L'orientation des composants selon les utilisateurs.**
   Hypothèse : notre modèle est mieux adapté à une interaction collaborative que les parallel coordinates. En effet, la lecture des axes, qui sont orientés selon un utilisateur unique, est difficile selon des points de vue différents. Dans la représentation en étoile, l'orientation des axes varie entre 0 et 360 degrés et permet ainsi une orientation spécifique. L'ensemble du glyphe peut ainsi être tourné pour s'orienter par rapport aux utilisateurs.

Evaluation : Les participants seront par groupes de quatre. Un seul des participants (le participant actif) modifiera les paramètres en suivant un scénario, et les trois autres (les participants passifs) devront comprendre ce scénario. Nous filmerons les séances afin d'en extraire les informations suivantes sur les participants passifs : attitude, réactions, mouvements du corps et de la tête afin de lire les données et exactitude des réponses. Un questionnaire subjectif sera ensuite rempli par chaque participant.

2. **La visibilité des données.**
   Hypothèse : notre modèle permet de favoriser une bonne visibilité des données pour les autres utilisateurs. Les mêmes tâches de manipulation effectuées sur des parallel coordinates poseront des problèmes de visibilité de certains composants. La possibilité ou non de déplacer les branches de la représentation en étoile permet par ailleurs de minimiser les occultations.
   Evaluation : plusieurs participants se verront attribuer une tâche spécifique (modification de la valeur d'un ou plusieurs paramètres à des instants précis). En plus de manipuler leurs paramètres ils devront indiquer quelles sont les valeurs des paramètres modifiés par les autres participants.
   Nous prévoyons trois configurations et donc six ordres possibles. Les trois configurations sont les suivantes :
   1. Interaction sur les coordonnées parallèles.
   2. Interaction sur la représentation en étoile sans possibilité de déplacer les branches.
   3. Interaction sur la représentation en étoile avec possibilité de déplacer les branches.

   De la même manière que pour l'orientation des composants, nous filmerons les séances afin d'en extraire les informations pertinentes : comportement et réactions des utilisateurs, et exactitude des réponses puis réponse à un questionnaire subjectif.

5.2. Evaluation des fonctionnalités spécifiques

Nous utiliserons pour cette seconde évaluation le même jeu de paramètres relatifs à un budget.

Les deux contributions qui nous ont semblées importantes à évaluer sont la pertinence de notre métaphore de comparaison d'étoiles et la pertinence des modalités de représentation des dépendances proposées.

1. **Pertinence de notre métaphore de comparaison d'étoiles.**
   Hypothèse : la représentation en étoile permet d'avoir une vue rapide et intuitive de l'ensemble des variables qui composent une donnée grâce à son glyphe. La comparaison de deux glyphes est donc intuitive.
   Evaluation : proposer à des participants plusieurs configurations sensées de variables représentées avec notre modèle et avec les parallel coordinates. Pour chaque configuration, les participants devront répondre le plus rapidement possible à des questions précises, telles que :
   - Le budget 1 par rapport au budget 2 est-il très faible, faible, moyen, élevé, très élevé ?
   - Les budgets ont-ils des dépenses très déséquilibrées, déséquilibrées, équilibrées ?
   - Lequel des deux budgets est-il le plus positif ? le plus négatif ? le mieux réparti ?

   Cette expérience nous permettra par ailleurs de juger de la possibilité de personnaliser les branches.

2. **Pertinence des modalités de représentation des dépendances.**
   Hypothèse : Les avertisseurs visuels que nous avons proposés, associés à notre représentation en étoile, permettent une meilleure compréhension des dépendances entre variables et des réactions plus rapides.
   Evaluation : Nous prévoyons un scénario d'évaluation à deux utilisateurs illustrant une règle bloquante : un utilisateur *agissant* et un utilisateur *réagissant*. *L'agissant* aura pour objectif

d'attribuer une valeur précise impossible à atteindre dans la configuration initiale à une variable dominée. Le *réagissant* devra s'adapter, dans un contexte de collaboration, afin de permettre à *l'agissant* d'accomplir sa tâche. Il devra en particulier déterminer quel paramètre dominant parmi ceux présents dans la représentation permettra à *l'agissant* d'atteindre son objectif. Nous avons considéré pour cette évaluation trois cas d'utilisation traduits par des avertisseurs visuels différents :

1. Affichage des bornes de l'intervalle des valeurs autorisées pour l'agissant (fig. 47.a).
2. Affichage des bornes de l'intervalle des valeurs autorisées pour l'agissant et mise en surbrillance des deux branches concernées (fig. 47.b).
3. Affichage des bornes de l'intervalle des valeurs autorisées pour l'agissant, mise en surbrillance des deux branches concernées et affichage de l'arc reliant les deux valeurs en conflit (fig. 47.c).

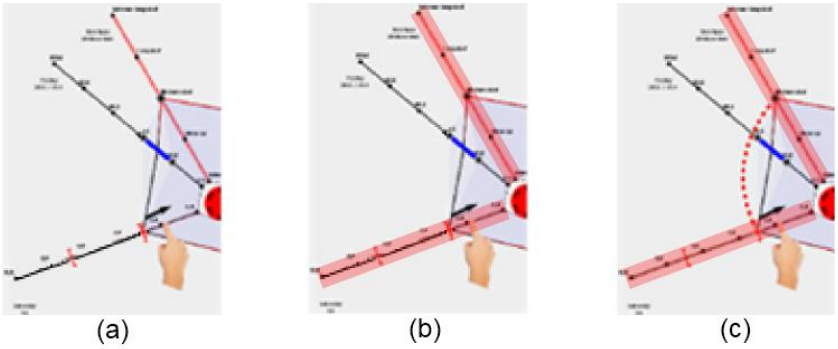


Fig. 47 : Les trois conditions expérimentales prévues afin d'évaluer les avertisseurs visuels liés aux dépendances : (a) Affichage des bornes de l'intervalle des valeurs autorisées, (b) Affichage des bornes de l'intervalle des valeurs autorisées et mise en surbrillance des deux branches concernées (c) Affichage des bornes de l'intervalle des valeurs autorisées, mise en surbrillance des deux branches concernées et affichage de l'arc reliant les deux valeurs en conflit

En faisant effectuer la même tâche sur les trois configurations et selon les neuf ordres possibles (avec une branche dominante différente pour chaque configuration), nous pourrons déterminer l'apport de chaque contribution. Pour évaluer chacune d'entre elles, nous avons prévu de mesurer le temps d'exécution de la tâche et de recueillir le ressenti des participants sous forme d'une discussion et d'un questionnaire.

5.3. Evaluation du modèle Collastar dans un contexte cinématographique

Le procédé d'évaluation du modèle Collastar appliqué à l'édition de contenus virtuels cinématographiques que nous prévoyons est le suivant :

- Définir un scénario de création : à partir d'une scène 3D, les participants devront réaliser un film court correspondant à certains critères au sens cinématographique : la première partie du film devra être lente et à caractère apaisant, alors que la seconde partie devra introduire une tension entre les acteurs (critères au sens cinématographique de très haut niveau).
- Sélectionner des participants ayant une connaissance minimale des logiciels de création de films à partir de scènes 3D (3DSMax) et leur faire prendre en main notre modèle.
- Faire réaliser aux participants le scénario défini sur 3DSMax et sur notre modèle en ordres alternés.
- Recueillir des données objectives : temps de création, correspondance au scénario, etc.
- Recueillir des données subjectives : faciliter à utiliser l'outil, facilité à collaborer.
- Analyser les données : identifier les avantages et inconvénients de notre contribution et en

déduire les éléments à modifier d'après le ressenti des utilisateurs.

En filmant chacune de ces expériences et en les annotant à l'aide de l'outil ELAN, (http://www.lat-mpi.eu/tools/elan/) nous pourrons analyser le comportement des utilisateurs lors de la réalisation des différentes tâches proposées.

Conclusion

Notre état de l'art nous a permis d'identifier un besoin de nouvelles métaphores d'interaction quant à la création de films à partir de scènes virtuelles 3D. En effet, les logiciels dédiés à cette tâche ne proposent d'autre alternative que de manipuler directement les degrés de liberté de la caméra et des lumières afin de produire une composition à l'écran satisfaisante. Bien que dans la littérature on trouve de nombreuses métaphores d'interaction permettant des approches de plus haut niveau, celles-ci ne sont pas adaptées au contrôle de paramètres cinématographiques.

Nous avons proposé le modèle CollaStar qui comprend une interface de visualisation (*Linear Wall & Focus+Context*) et un outil d'édition collaborative de paramètres qu'est la représentation en étoile. Cet outil est basé sur une extension interactive de la représentation multidimensionnelle en étoile.
Notre contribution permet :

- La visualisation de données multidimensionnelles et temporelles grâce à l'interface de visualisation présentée dans le chapitre 2. Cette interface permet de manipuler et observer des données de types variés tout en permettant de personnaliser l'affichage et la disposition des données. Des données de différents niveaux (éditables ou non) peuvent être affichées, ainsi que des comparaisons de données. Nous avons de plus optimisé l'affichage des données dans un espace restreint afin de faciliter la lecture de celles-ci (Horizon Curves, inverse-video, utilisation de la dimension couleur).
- L'interaction collaborative avec les paramètres de haut niveau contrôlant la caméra et les lumières grâce à la représentation en étoile collaborative présentée dans le chapitre 3. Notre proposition permet de spécifier des dépendances entre ces paramètres et de signaler celles-ci à l'utilisateur par le biais d'avertisseurs visuels, de contrôler des paramètres de types différents ou d'effectuer des opérations d'ajout, suppression et modification des paramètres affichés dans la représentation (partage des branches).
- La comparaison de données : *Linear Wall & Focus+Context* permet de comparer des dimensions une à une (grâce entre autres à notre contribution apportée aux Horizon Curves) alors que la représentation en étoile permet de comparer par superposition deux données à des instants différents en superposant leurs glyphes respectifs.

Afin que notre modèle permette l'interaction multi-utilisateurs collaborative, qui est au cœur de notre sujet de recherche, nous avons exploré les possibilités qu'apportent les tables tactiles multipoints. Leurs propriétés nous ont permis d'autoriser et contrôler des actions simultanées de plusieurs utilisateurs sur des éléments interdépendants mais aussi de spécifier des métaphores d'interaction que ne permettent pas les périphériques standards (dessin à deux doigts d'une courbe bidimensionnelle par exemple).

Nous avons porté un intérêt particulier à l'intégration de la dimension temporelle :

- Dimension temporelle commune à tous les utilisateurs.
- Affichage de la donnée courante sous forme d'image (*Focus*).
- Affichage des valeurs passées et futures (*Context*).
- Affichage des valeurs passées dans la représentation en étoile.

Nous avons aussi étudié différentes modalités d'évaluation et scénarii d'utilisation de ces contributions et les expérimentations sont en cours. Le système CollaStar couplé au moteur Cinesys fait l'objet d'une soumission (conférence IUI 2012) et les contributions particulières que sont la comparaison d'Horizon Curves et la représentation en étoile pour une interaction collaborative vont être soumises sous forme de posters (à INFOVIS et Interactive Tabletops and Surfaces, respectivement).

Nous avons identifié certaines limitations à notre modèle (spécificité de la dimension temporelle, traitement des données dynamiques, traitement des données binaires, besoin de haute résolution) et proposé des pistes à explorer afin de les prendre en compte. Plusieurs axes de recherche restent donc ouverts et sont à approfondir.

De façon plus générale, il est essentiel de continuer à explorer les couplages fins entre des modèles calculatoires expressifs et des modèles d'interaction nouveaux. Ainsi les utilisateurs ne sont pas contraints par le système, mais peuvent dialoguer avec lui par l'intermédiaire de métaphores pertinemment choisies et développer leur créativité par des interactions de haut niveau.

Références

[All91] F.J. Allen : Time and time again: The many ways to represent time. In *International Journal of Intelligent Systems*, 6(4), pp. 341-355, July 1991.

[Ari76] Daniel Arijon : Grammar of the film language. *Communication Arts Books, Hasting House, Publishers,* New York (1976).

[AS99] C. Ahlberg, B. Shneiderman : Readings in Information Visualization: Using Vision to Think, Morgan Kaufmann Publishers (1999).

[BiSlider] http://www.limsi.fr/Individu/vernier/BiSlider/

[BKFBK02] N. Burtnyk, A. Khan, G. Fitzmaurice, R. Balakrishnan, G. Kurtenbach : StyleCam: Interactive stylized 3D navigation using integrated spatial & temporal controls. In Proceedings of ACM UIST 2002 (2002), pp. 101-110.

[CAHSWC96] David B. Christianson, Sean E. Anderson, Li-wei He, David H. Salesin, Daniel S. Weld, Michael F. Cohen : Declarative camera control for automatic cinematography. In *Proceedings of the AAAI-96*, August 1996.

[CPDD03] S. Chick, P.J. Sanchez, D. Ferrin, D.J. Morrice : Visualization methods for time-dependant data – An overview. In *Proceedings of the 2003 Winter Simulation Conference* (2003).

[Chi00] Ed H Chi : A taxonomy of visualization techniques using the data state reference model. In *Proceedings of InfoVis 2000* , Salt Lake City UT (October 2000).

[CM01] N. Courty, E. Marchand : Computer animation: A new application for image-based visual servoing. In *Proceedings of International Conference on Robotics and Automation* (2001).

[CON08] M. Christie, P. Olivier, J-M Normand : Camera control in computer graphics. *Computer Graphics Forum* 27, 8 (2008).

[Daa00] C. Daassi : Techniques d'interaction avec un espace de données temporelles. *PhD thesis in Computer Science from Joseph-Fourier University of Grenoble*, France (2003).

[DNF04] C. Daassi, L. Nigay, M-C. Fauvet : Vizualisation process of temporal data. In *Proceedings of DEXA*, pp. 914-924 (2004).

[EDF08] N. Elmqvist, P. Dragicevic, J.D. Fekete : Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1141-1148 (Nov/Dec 2008).

[ER07] D. Elson, M. Riedl : A lightweight intelligent cirtual cinematography system for machinima production. In *Proceedings of the 3rd Conf. on AI Interactive Entertainment* (2007)

[FCI05] E. Fanea, S. Carpendale, T. Isenberg : An interactive 3D Integration of Parallel Coordinates and Star Glyphs. In *Proceedings of the 2005 IEEE Symposium on Information Visualization* (2005)

[FP03] B. Francis, J. Pritchard : Visualization of historical events using Lexis Pencils. Disponible à http://www.agocg.ac.uk/reports/visual/casestud/francis/compar_1.htm (2003).

[GW92] M. Gleicher, A. P. Witkin : Through-the-lens camera control. In *Proceedings of the Siggraph Conference* (1992),

ACM Computer Graphics, pp. 331-340.

[HCS96] Li W. He, Michael F. Cohen, David H. Salesin : The virtual cinematographer : A paradigm for automatic real-time camera control and directing. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996).

[HHN00] S. Havre, B. Hetzler, L. Nowell : ThemeRiver : Visualizing theme changes over time. *Proceedings of the IEEE Symposium on Information Visualization* (2000).

[HKA09] J. Heer, N. Jeffrey, M. Agrawala : Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the 27th international conference on Human factors in computing systems*, New York (2009).

[HMKBH97] L. Hong, S. Muraki, A. Kaufman, D. Bartz, T. He : Virtual Voyage: Interactive navigation in the human colon. In *Proceedings of the 1997 ACM SIGGRAPH* (1997).

[Infovis] http://www.infovis-wiki.net

[Jha09] A. Jhala : Cinematic Discourse Generation. *PhD thesis, Faculty of North Carolina State University* (2009)

[Ker00] I.V. Kerlow : The art of 3-D computer animation and imaging, 2nd edition, John Wiley & Sons, Inc. (2000).

[KHLW09] A. Klippel, F. Hardisty, R. Li, C. Weaver : Color enhanced star plot glyphs – Can salient shape characteristics be overcome?. In *Cartographica*, 44:3, pp. 217-231 (2009).

[KKSFK05] A. Khan, B. Komalo, J. Stam, G. Fitzmaurice, G. Kurtenbach : HoverCam: Interactive 3D navigation for proximal object inspection, In *Proceedings of the 2005 Symposium on Interactive 3D graphics and games*, ACM Press (2005), pp. 73-80.

[Lam09] F. Lamarche : Topoplan: a topological path planner for real time human navigation under floor and ceiling constraints. *Computer Graphics Forum 28*, 2 (2009).

[LCLSO10] C. Lino, M. Christie, F. Lamarche, G. Schofield, P. Olivier : A Real-time cinematography system for interactive 3D environments. In *Proceedings of the 2010 ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2010), Madrid, Spain.

[MB98] E. MacKay, M. Beaudouin-Lafon : DIVA: Exploratory data analysis with multimedia streams. In *Proceedings of the ACM CHI'98 Conference on Human Factors in Computing Systems* (1998).

[MRC91] J. D. Mackinlay, G. Robertson, S. Card : The perspective wall : Detail and context smoothly integrated. In *Proceedings of the CHI`91*, ACM Press (1991).

[OHPL99] P. Olivier, N. Halper, J. Pickering, P. Luna : Visual composition as optimisation. In *AISB Symposium on AI and Creativity in Entertainment and Visual Art*, pp. 22-30 (1999).

[PBG92] C.B. Phillips, N.I. Badler, J. Granier : Automatic viewing control for 3d direct manipulation. In *Proceedings of the symposium on Interactive 3D graphics (SI3D 92)* (1992), ACM Press New York, USA, pp. 71-74.

[RC94] R. Rao, S.K. Card : The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, Boston, MA (April 1994).

[Sho92] K. Shoemake : Arcball : A user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of Graphics Interface (GI 92)* (May 1992), pp. 151-156.

[Sny92] J. Snyder : Interval analysis for computer graphics. In *Proceedings of the Siggraph Conference*, vol. 22-4, ACM Computer Graphics, pp. 121-130 (1992).

[SVFR04] C. Shen, F.D. Vernier, C. Forlines, M. Ringel : DiamondSpin: an extensible toolkit for around-the-table interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2004).

[TB09a] R. Thomson, C. J. Bowen : Grammar of the edit, second edition, Focal Press (2009).

[TB09b] R. Thomson, C. J. Bowen : Grammar of the shot, second edition, Focal Press (2009).

[Ver00] F. Vernier : La multimodalité en sortie et son application à la visualisation de grandes quantités d'information. *PhD thesis in Computer Science from Joseph-Fourier University of Grenoble*, France (2001).