

Titre: Automatisation de la création de scénarios pour les scènes de la visualisation scientifique
Title: visualisation scientifique

Auteur: Anis Benyoub
Author:

Date: 2015

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Benyoub, A. (2015). Automatisation de la création de scénarios pour les scènes de la visualisation scientifique [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/1751/>
Citation:

Document en libre accès dans PolyPublie Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/1751/>
PolyPublie URL:

Directeurs de recherche: Benoît Ozell
Advisors:

Programme: Génie informatique
Program:

UNIVERSITÉ DE MONTRÉAL

AUTOMATISATION DE LA CRÉATION DE SCÉNARIOS POUR LES SCÈNES DE LA
VISUALISATION SCIENTIFIQUE

ANIS BENYOUB

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
MAI 2015

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

AUTOMATISATION DE LA CRÉATION DE SCÉNARIOS POUR LES SCÈNES DE LA
VISUALISATION SCIENTIFIQUE

présenté par : BENYOUB Anis

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. GAGNON Michel, Ph. D., président

M. OZELL Benoît, Ph. D., membre et directeur de recherche

M. HURTUT Thomas, Ph. D., membre

À Chaouqui Benyoub, Smires Soad et Héloïse Bleton. Merci de votre soutien.

REMERCIEMENTS

J'aimerais remercier particulièrement le professeur Benoît Ozell de m'avoir permis de travailler sur ce sujet et m'avoir encadré tout au long de la maîtrise. Ses conseils et son aide à la rédaction de l'article et du mémoire m'ont permis de présenter un travail mieux structuré, plus cohérent et plus abouti.

L'aide financière du CRSNG qu'il m'a octroyée m'a également permis de me concentrer sur mon travail et d'être d'une plus grande productivité.

RÉSUMÉ

Bien que l'automatisation de la génération des chemins de caméra soit une pratique courante dans le cinéma et le jeu vidéo, elle fait preuve d'un retard important dans le milieu de la visualisation scientifique. La taille, la nature, la densité du maillage ainsi que l'absence de notions scénaristiques telles que les personnages et les actions font que l'importation de règles de composition issues de domaines artistiques comme le cinéma ou la photographie ainsi que l'application des règles du montage deviennent plus complexes.

Ce mémoire introduit une méthodologie qui propose une métamodélisation des attentes des scientifiques vis-à-vis de leurs données ainsi qu'une modélisation des comportements de données qui peuvent les intéresser. Ces modèles permettent de bâtir plus facilement des chemins de caméra pour des scènes numériques issues de simulations ou d'acquisitions. L'application des règles issues de la composition et du montage, dans le but de produire des déplacements de caméra qui préservent l'intention du scientifique, devient alors plus simple.

La méthode a été expérimentée sur un ensemble de scènes issues de la mécanique des fluides et du génie biomédical ; les résultats obtenus sur ces scènes nous ont permis de valider le fonctionnement de la méthodologie.

Sont également proposés par la méthode un ensemble de paramètres de contrôle afin de modifier le processus de génération pour mieux l'adapter aux besoins précis que peut avoir un scientifique vis-à-vis d'une scène.

ABSTRACT

Automatic camera path generation is a common practice in film making and video games. However, it demonstrated a significant delay in scientific visualization due to the size, the nature, the mesh density and the lack of scriptwriting notions such as characters and actions. In this case, importing cinematography composition, photography composition and match cut rules becomes a more complex operation.

This thesis presents a methodology that provides meta-models for the scientists' visualization expectations regarding their data and for the data behaviors that may interest them. These models make the camera paths generation process more intuitive. The application of composition and match cut rules, in order to produce camera moves that preserves the scientist's intention, becomes simpler.

The method was tested on a set of scenes from fluid mechanics and biomedical engineering; the obtained results showed that our approach is a simple and efficient way for producing presentation videos.

A set of control parameters are also provided by this method, in order to adapt the generation process to the specific needs that a scientist can have regarding his data.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	x
LISTE DES FIGURES	xi
LISTE DES ANNEXES	xiii
LISTE DES SIGLES ET ABRÉVIATIONS	xiv
CHAPITRE 1 INTRODUCTION	1
CHAPITRE 2 REVUE DE LITTÉRATURE	2
2.1 La synthèse d'images	2
2.2 Le storytelling	2
2.3 La visualisation scientifique	3
2.4 Les particularités	4
2.4.1 Les algorithmes de traitement	4
2.5 La détection de particularités	6
2.5.1 Le vortex (tourbillon)	7
2.5.2 L'onde de choc	8
2.5.3 Les bordures	9
2.5.4 Lignes de courant	10
2.5.5 Lignes de séparation et d'attachement	11
2.6 Détection d'évènements	11
2.6.1 Persistance	11
2.6.2 Mort et Naissance	12
2.6.3 Entrée et Sortie	12

2.6.4	Scission et Fusion	12
2.6.5	Non résolu	12
2.7	Mécaniques de Storytelling	13
2.7.1	Manipulation de la caméra	13
2.7.2	Génération procédurale de chemins de caméra	15
2.7.3	La visibilité	18
2.7.4	Les règles de la composition	18
2.7.5	Chemin de caméra en visualisation scientifique	20
2.8	Questions de recherche	21
2.9	Objectif général	21
2.10	Objectifs spécifiques	21
2.11	Hypothèses	22
CHAPITRE 3 Aperçu de la méthodologie		23
3.1	Présentation de l'article	23
3.1.1	Titre	23
3.1.2	Auteurs	23
3.1.3	Publication	23
3.1.4	Résumé	23
3.2	Travail complémentaire	24
CHAPITRE 4 Article 1 : A Practical Method For Automatic Camera Paths Generation In Scientific Visualization		25
4.1	Abstract	25
4.2	Introduction	25
4.3	Previous work	28
4.4	Camera Path	28
4.4.1	Storytelling And Features	28
4.4.2	Feature Analysis	30
4.4.3	The Cliche	31
4.4.4	Spot of interest	32
4.4.5	A Good Viewpoint	33
4.4.6	Visualization Needs	34
4.4.7	Constraints	35
4.4.8	Graph navigation	35
4.4.9	Multi-criteria Discrete Optimization	37
4.4.10	Building the paths	37

4.4.11	Interpolating the path	38
4.4.12	Photographic Compositions	38
4.5	Extensions	39
4.5.1	Multiple choices	39
4.5.2	User intervention	40
4.5.3	Time varying data	40
4.5.4	Match cut rules	40
4.6	Results	43
4.7	Discussion	43
4.8	Future work	47
4.8.1	Multiple SOI	47
4.9	Conclusion	48
 CHAPITRE 5 DISCUSSION GÉNÉRALE ET COMPLÉMENTAIRE		49
5.1	La créativité	49
5.2	Ontologie	51
5.3	Besoins de visualisation complexes	52
5.4	Critères et poids	52
5.5	Temps d'exécution	54
5.5.1	L'extraction des régions d'intérêt	55
5.5.2	Calcul de critère de qualité	58
5.5.3	Calcul d'occlusion	58
5.5.4	Transformations	58
5.6	Lignes d'action	58
5.7	Atteinte des objectifs	60
 CHAPITRE 6 CONCLUSION		65
6.1	Synthèse des travaux	65
6.2	Limitations et améliorations futures	65
 RÉFÉRENCES		66
ANNEXES		71

LISTE DES TABLEAUX

Tableau 5.1	Conversion des métriques de critères en un espace paramétrique régulier	53
Tableau 5.2	Données de complexité pour chacune des scènes et besoins présentées dans l'article	62
Tableau 5.3	Temps de traitement pour chacune des scènes et besoins présentés dans l'article	62

LISTE DES FIGURES

Figure 4.1	Generated camera for observing separation lines in the smoke scene	26
Figure 4.2	A simple story : a stick-man (character) taking a photo (action) of a plant.	29
Figure 4.3	The introduction scene from the movie <i>Lord of War</i> (Niccol, 2005).	29
Figure 4.4	The Mexican standoff scene in the movie <i>The good, the bad and the ugly</i> (Leone, 1966).	30
Figure 4.5	From left : A vortex represented as a vector-field and a shockwave as an isosurface.	31
Figure 4.6	Converting a curve to a vector-field.	32
Figure 4.7	A spherical SOI around an aneurysm.	33
Figure 4.8	Left : vector field quality heuristic. Right : volume quality heuristic.	34
Figure 4.9	Occlusion while observing smoke propagation.	35
Figure 4.10	Occlusion of the aneurysm by other components of the scene (occ. at 17.1138%).	36
Figure 4.11	Edges constraints. In red the current viewpoint, in blue previous viewpoint.	36
Figure 4.12	How to apply rule of cropping.	39
Figure 4.13	How to apply rule of thirds.	39
Figure 4.14	A simple scene illustrating match rule potential issues : three characters A, B and C facing different directions and 4 camera position observing the scene.	41
Figure 4.15	Film editing the different paths generated for each time lapse (default path in green). Each camera sub path is paired to a heuristic.	42
Figure 4.16	Up : Generated path for observing the aneurysm edges. Down : Sample viewpoint with rule of thirds metrics.	44
Figure 4.17	Up : specified SOI for streamline observation. Down : rendered image from the generated path (with particle propagation and rule of thirds metrics).	45
Figure 4.18	Up : a sample generated path for observing a SOI located at the back of the train. Down : rendered image from the generated path with rule of thirds metrics.	46

Figure 4.19	Left image : an overview over the brain simulation scene ; a pressure gradient color map is used for the representation. Middle image : a zoom on the path of a blood particle propagation (yellow curve). Right image : one set of generated camera sub paths following the particle ; streamlines were specified as the visualization need in this case.	47
Figure 4.20	Rendered image from the generated path for the smoke scene with rule of thirds metrics. The diagonal and leading line rules have produced interesting image composition.	47
Figure 5.1	Plans tirés du film «The Grand Budapest Hotel» du réalisateur <i>Wes Anderson</i> , illustrant la règle de la symétrie.	50
Figure 5.2	Deux vues des lignes de courants et de l'axe d'un vortex (face et perspective). La deuxième vue respecte mieux l'heuristique de champ de vecteurs pour chacune des représentations.	53
Figure 5.3	Poids des critères utilisés pour la génération du chemin de caméra de l'anévrisme dans l'intervalle $[0, 1.1]$	55
Figure 5.4	Images successives du chemin de caméra générée pour observer la surface de l'anévrisme	56
Figure 5.5	Vue globale du chemin générée pour observer la surface de l'anévrisme	57
Figure 5.6	Composantes principales d'un poisson et d'un personnage qui court. .	60
Figure 5.7	Images successives du chemin de caméra généré pour observer le champ de vecteurs dans la scène «Fumée»	63
Figure 5.8	Vue globale du chemin de caméra généré pour observer le champ de vecteurs dans la scène «Fumée»	64

LISTE DES ANNEXES

Annexe A	FABLE	71
----------	-------	----

LISTE DES SIGLES ET ABRÉVIATIONS

SOI *spot of interest*, région d'intérêt

CHAPITRE 1

INTRODUCTION

Les scientifiques issus de domaines comme la mécanique, la chimie ou encore le biomédical ont souvent des données numériques issues de simulations ou d'acquisitions qu'ils veulent pouvoir explorer ou présenter à un public. Produire un court métrage pour présenter ces résultats numériques est donc une tâche que les scientifiques ont très souvent à accomplir.

Cependant, étant donné la taille et la densité des maillages, beaucoup d'entre eux n'ont pas nécessairement le temps de penser la composition de leurs images et de produire des vidéos qui mettent en valeur leurs données en prenant en compte leurs intentions vis-à-vis de leur auditoire.

La génération procédurale de chemins de caméra est une pratique commune dans la cinématographie et le jeu vidéo. Ces problèmes sont souvent résolus à l'aide de systèmes de contraintes ou de techniques d'optimisation. Cependant, pour des scènes 3D volumiques et surfaciques issues de la visualisation scientifique, les notions de compositions ne peuvent pas leur être appliquées directement étant donné l'absence de notions fondamentales de Storytelling comme les personnages ou les actions.

Les attentes des scientifiques vis-à-vis de leurs données consistent souvent à chercher à mettre en évidence des comportements de données notables. En visualisation scientifique, beaucoup de travaux de recherches ont alors été menés sur la détection de ces comportements, le suivi de leurs déplacements et leurs changements d'état. Certains travaux ont également été menés dans le but de proposer des points de vue intéressants, mais la notion de chemins de caméra a été très peu approfondie dans ce domaine.

CHAPITRE 2

REVUE DE LITTÉRATURE

L'œil humain peine à percevoir les données numériques quand elles sont présentées sous forme brute. L'expérience acquise dans les procédés de communication nous a montré que le cerveau humain est conditionné pour mieux percevoir les informations sous forme de schémas, d'images ou de données sonores. L'ensemble des techniques et méthodes développées dans le but de transmettre au mieux l'information numérique en informatique constitue le domaine de l'infographie. Nous nous intéressons à certaines des familles d'algorithmes d'infographie.

2.1 La synthèse d'images

La synthèse d'images est l'ensemble des algorithmes qui permettent de transformer des données numériques d'une scène 2D ou 3D en image qui reflète à un certain point une réalité donnée. Quelques étapes notables dans ce domaine sont : l'interface graphique, les processeurs graphiques dédiés, l'évolution du pipeline de rendu. Beaucoup de travaux de recherches ont également été menés sur les modèles d'illumination afin de produire des images de qualité, et ce de manière optimisée (BRDF, tracé de chemins, harmoniques sphériques, etc.).

2.2 Le storytelling

Le « storytelling », l'art de raconter une histoire, est un procédé narratif qui permet d'accrocher le spectateur et de guider son expérience lors du récit pour lui transmettre correctement l'intention du narrateur. C'est un procédé que l'on retrouve dans tous les médias artistiques, notamment dans la littérature et au cinéma. L'ensemble des règles qui régissent le storytelling n'est pas nécessairement explicite pour le spectateur. Cependant, elles sont très importantes à étudier, car elles constituent un outil très puissant dans le processus de communication, de transmission d'informations et d'intentions (Truby, Oct 2008).

Une histoire, aussi simple soit-elle, doit comporter des personnages et des actions. Ces éléments aident le spectateur à s'identifier à l'histoire afin de mieux la percevoir. Les personnages ne sont pas nécessairement des êtres vivants et les actions n'ont pas besoin d'être complexes afin de créer une histoire captivante.

Dans l'infographie, le storytelling est un des sous-domaines qui se focalise sur les procédés de transmission d'informations. Il implique beaucoup plus que les procédés de narration ; un

ensemble de traitements supplémentaires sur les données, dont la visualisation, la détection de comportements notables dans les données, la recherche de point de vue optimal, etc.

Ce sous-domaine fait preuve d'un retard assez important comparé aux autres domaines de l'infographie (Ma *et al.*, 2012a). Ce phénomène peut être expliqué par le fait que ces mécaniques font intervenir une part de relativité vis-à-vis de l'importance des codes du média où elles sont appliquées ainsi que la subjectivité du spectateur lors de son interaction avec l'œuvre.

Elle est un goulot d'étranglement réel et freine souvent l'assimilation des informations que peuvent contenir des données. Beaucoup d'outils et de technologies ont été créés pour permettre de faciliter et accélérer l'intégration des notions du storytelling dans les mécaniques de l'infographie : environnements d'immersion (CAVE, visiocasques), accessibilité par les technologies web, etc. (Fox et Hendlar, 2011).

2.3 La visualisation scientifique

Le cadre de ce mémoire se limite à la visualisation scientifique qui est un cas spécifique des problèmes de visualisation et de narration. En effet, les simulations numériques scientifiques produisent très souvent des données de taille géante ; il est donc très difficile pour le scientifique de les explorer de manière naïve manuelle. C'est pourquoi certains outils existent pour faciliter cette exploration : Paraview, VU, etc. Cependant, ces outils permettent principalement de guider le scientifique dans la visualisation des données dans le processus d'exploration ; la partie présentation à autrui à l'aide des règles de la narration étant moins développée dans ces outils (Wohlfart, 2008) (Koop, 2008).

Les données traitées par ces algorithmes sont issues de la simulation numérique (éléments finis, Monte-Carlo, etc.) ou de l'acquisition (scanneur, IRM, microphone, etc.). La nature des données issues de ces procédés est souvent variée ; dans la majorité des cas, les données produites sont de nature volumique (Wohlfart et Hauser, 2007) ou surfacique (il peut toutefois arriver que l'on produise des données linéaires : données sonores).

Dans ce mémoire, la méthodologie est centrée autour des données 3D surfaciques et volumiques. Bien que la méthodologie soit conçue pour des données issues de tous domaines scientifiques, nous nous sommes focalisés pour nos résultats sur des simulations issues de la mécanique des fluides, car ce type de scènes en sont majoritairement issues.

La détection de comportement de données 3D est souvent plus complexe qu'en 2D ; la discréétisation induite par la résolution de systèmes à l'aide de méthode à éléments finis crée des problèmes de continuité au niveau des résultats et complique le processus de détection (Ma *et al.*, 1996).

Les simulations produisent souvent des données brutes à l'aide d'un certain nombre de primitives (toujours dans le cas des données volumiques). Les plus communes sont les données voxels, prismiques, tétraédriques, cubiques, etc. Pour plus de généricité, notre objectif est d'être capable de traiter des données indépendamment de leur forme.

Le stockage des données se fait souvent dans des formats de fichiers spécifiques au type de données qu'ils contiennent. Il n'est pas intéressant de détailler tous ces formats de données, car ils sont trop nombreux. Cependant, voici quelques-uns des plus utilisés (du plus récent au plus ancien) :

- COLLADA : très présent dans l'infographie, la manipulation et la visualisation de scènes ;
- VTK : beaucoup de sous-formats, il est très présent dans la visualisation scientifique, beaucoup de variantes, XML, binaire, etc. ;
- VRML : un format textuel, très populaire en infographie.

2.4 Les particularités

Dans un premier temps, nous nous pencherons sur des comportements de données particuliers que nous appellerons particularités (en anglais *features*) ou patrons (*patterns*). Ce sont des comportements de données qui intéressent souvent les scientifiques. Dresser une liste exhaustive des particularités serait une tâche fastidieuse ; chaque scientifique peut avoir la nécessité de détecter des comportements qui l'intéressent spécifiquement. Dans cette revue de littérature, nous nous intéressons uniquement à celles qui seront traitées dans la méthodologie de ce mémoire. En étudiant les algorithmes ou techniques de détection utilisées pour les trouver dans les données numériques, nous serons plus facilement aptes à produire une méthode qui peut s'adapter à toutes les particularités potentielles.

2.4.1 Les algorithmes de traitement

Certains types de méthodes sont communément utilisés dans la détection de particularités. On peut noter cinq grandes familles d'algorithmes de détection (classification proposée par Post *et al.* (2003)).

Traitement d'images

Comme leur nom l'indique, ce sont des algorithmes qui traitent des données 2D. Toutefois, en effectuant des projections de données 3D, par l'exemple à l'aide d'un système de caméra perspective ou orthographique, il est possible de les appliquer à des données 3D. Les algorithmes de segmentation d'image sont beaucoup utilisés pour la détection de particularités.

Analyse des caractéristiques physiques

Ce sont des algorithmes qui permettent de détecter des comportements liés à la nature des données. En effet, le fait de connaître quel type de données on manipule nous permet de déterminer plus facilement les particularités visées (Post *et al.*, 2003). En manipulant de la température, on peut être intéressé de chercher les zones où la température est très élevée ou où la variation est très grande plutôt qu'à un vortex constitué par le gradient de la température.

Analyse de la topologie des champs de vecteur

L'analyse topologique vise à déterminer les particularités par l'étude des valeurs par le biais des matrices Jacobiennes et Hessiennes.

Visualisation selective

Cette catégorie d'algorithmes est celle qui est employée dans les mécanismes de fouille de données. Ce sont généralement des opérations de présélection, de *clustering*, filtres pour mettre en évidence des informations venant des données afin de permettre au scientifique de détecter visuellement les informations qui l'intéressent.

Dans notre cas, nous ne nous basons pas sur de tels algorithmes. Cette approche entre en conflit avec notre approche qui vise l'automatisation des processus de visualisation. Néanmoins, on retrouve cette idée dans les logiciels comme Paraview qui permettent aux scientifiques d'être guidés dans l'exploration de leurs résultats numériques (Koop, 2008).

Apprentissage machine

On retrouve dans la littérature certains articles qui mettent en avant des méthodologies pour automatiser la détection de particularités au travers de mécaniques d'apprentissage machine (Zhang et Meng, 2012). Ces méthodes sont souvent utiles quand on n'est pas réellement capable de définir nos besoins ainsi que les algorithmes optimaux pour les remplir. Cependant, il peut être intéressant d'utiliser ces mécaniques afin d'améliorer la précision d'algorithmes de détection.

Évidemment, chacune de ces familles d'algorithmes qui ont présentées ne sont pas exclusives. Il arrive souvent que l'on ait besoin de combiner plusieurs algorithmes de chacune des catégories pour obtenir des résultats meilleurs (Post *et al.*, 2003).

On retrouve dans la littérature une autre classification des algorithmes de traitement des particularités. Cependant, la différentiation cette fois se fait au niveau de l'utilité et non

pas de la nature des algorithmes. Ces catégories peuvent être interprétées comme des étapes successives de la visualisation des données (classification proposée par Reinders *et al.* (2001)) :

Extraction de particularités

Ces algorithmes s'intéressent à la recherche des particularités pour un état donné des données numériques. Ils se basent généralement sur certains des attributs disponibles dans les maillages : position, vitesse, densité, etc. Les algorithmes sont spécifiques à la nature des particularités que l'on cherche.

Suivi de particularités

Cette catégorie d'algorithmes s'intéresse au suivi au fil du temps d'une particularité donnée. On possède généralement la liste des particularités existantes à des instants et le traitement consiste à chercher une cohérence entre ces différents instants.

Détection d'évènements

Il est intéressant de savoir dans quels états se trouve une particularité, les variations d'état possibles ainsi que ses interactions avec d'autres particularités constituent ce qui est nommé évènement dans la littérature. Connaître les états et les transitions possibles permet de mieux cerner cette particularité et de définir ce qu'on cherche à observer ou présenter.

Visualisation des particularités

Étant donné que l'on est capable de définir où se situe chaque particularité à chaque instant et que l'on est capable de savoir dans quel état elle se trouve, il est important de définir des méthodes pour la présenter. Les trois aspects principaux étant : la représentation, l'illumination et les paramètres de la caméra qui sert à l'observer.

2.5 La détection de particularités

Nous nous intéressons à la liste de particularités que nous avons choisies ainsi qu'à certains de leurs algorithmes de détections :

- Le vortex.
- L'onde de choc.
- Les bordures.
- Les lignes de séparation et d'attachement.
- Les lignes de courant.

Notre présentation des algorithmes de détection est une synthèse de plusieurs états de littératures existantes, parmi elles Post *et al.* (2003), Roth et Peikert (1996) et Kolář (2007).

2.5.1 Le vortex (tourbillon)

C'est un comportement que l'on retrouve souvent en dynamique des fluides. Il est souvent très important et fait l'objet d'une attention très particulière de la part des scientifiques en mécanique des fluides. Dans certains cas, il peut avoir des conséquences très graves si non prévues. La définition actuelle présente dans la littérature serait celle donnée par Robinson (1990) : «*A vortex exists when instantaneous streamlines mapped onto a plane normal to the vortex core exhibit a roughly circular or spiral pattern, when viewed from a reference frame moving with the center of the vortex core*».

Les écoulements tourbillonnaires peuvent faire référence à plusieurs comportements de données (Post *et al.*, 2003), pour en citer deux :

- **Le vortex de portance** : c'est un phénomène qui altère la force de portance sur l'aile d'un avion et peut donc altérer l'équilibre d'un avion en vol (Kasper, 1975).
- **Le vortex de Karman** : ce vortex qui peut engendrer un phénomène de résonance d'oscillation sur les structures comme les ponts et amener à leur destruction (Panknin *et al.*, 2002).

Il existe deux catégories d'algorithmes de détection qui s'intéressent aux tourbillons : ceux qui s'intéressent au centre (vortex-core axis) et ceux qui s'intéressent à toute la région tourbillonnaire (vortex core) (Van Gelder, 2012).

Plusieurs notions mathématiques sont souvent associées aux vortex :

- **La vorticité** : c'est un vecteur qui représente la vrille du vecteur vitesse. On le calcule en appliquant le vecteur gradient au vecteur vitesse : $\vec{\nabla} \mathbf{v}$. L'accélération serait une notion qui en est assez proche. On cherche les zones où la magnitude du vecteur de vorticité est la plus élevée, car on peut y trouver des vortex (mais de manière non déterministe, un contre exemple est une zone avec une direction constante, mais des vitesses variables) (Zabusky *et al.*, 1991).
- **L'hélicité** : elle est définie comme la projection de la vorticité sur la vitesse ($\vec{\nabla} \mathbf{v} \cdot \mathbf{v}$) dans la direction de la vitesse et permet d'éliminer la composante perpendiculaire de la vorticité par rapport à la vitesse. Les algorithmes de détection vont également se concentrer sur les régions du maillage où l'hélicité est élevée (Degani *et al.*, 1990).

D'autres méthodes de détections plus complexes ont également été développées :

Le Q-criterion : C'est une méthode définie (Hunt *et al.*, 1988) comme la zone où la vorticité est supérieure à la vitesse de déformation et où la pression est inférieure à la pression ambiante : $Q = \frac{1}{2}(\|\Omega\|^2 - \|S\|^2) > 0$ où S et Ω sont les matrices symétriques et

antisymétriques de la Jacobienne.

Le Δ -criterion : On définit les zones contenant des vortex comme étant les zones où la vorticité est complexe et où les lignes de courant forment des spirales (Dallmann (1983) ainsi que Chong *et al.* (1990)), ces conditions peuvent être formulées comme ceci : $\Delta = (\frac{Q^3}{3} + \frac{R^2}{2}) > 0$

Où Q est définie plus haut et S est le déterminant de la Jacobienne.

Le $\lambda 2$ -criterion : Les vortex-cores sont les régions où la matrice symétrique $S^2 + \Omega^2$ a des valeurs propres négatives (Jeong et Hussain, 1995). Où S et Ω sont les matrices symétriques et antisymétriques de la Jacobienne : $S = \frac{1}{2}(V + V^T)$ et $\Omega = \frac{1}{2}(V - V^T)$.

Ces méthodes restent toutefois assez peu efficaces pour des données d'une grande complexité. On retrouve dans la littérature d'autres méthodes plus complexes pour la détection de vortex :

- L'utilisation de lignes de courant du champ de vorticité en corrélation avec la pression minimale (Banks et Singer, 1995).
- Recherche d'une zone où la vorticité et la vitesse sont parallèles (semble être peu efficace) (Roth et Peikert, 1996).
- Recherche d'une zone où une des valeurs de la Jacobienne est une valeur propre réelle et où son vecteur propre est parallèle au courant (Roth et Peikert, 1996).
- L'analyse de la topologie locale des voisins pour chacune des cellules de notre maillage permet de déterminer si ce point fait partie du centre du vortex (Jiang *et al.*, 2002).

Dans le cas où cette détection ne serait pas suffisamment précise, il existe une manière de renforcer la détection introduisant de l'AdaBoost (Zhang et Meng, 2012).

2.5.2 L'onde de choc

Cette deuxième particularité est décrite (Post *et al.*, 2003) comme étant liée à une discontinuité dans un certain nombre de caractéristiques physiques au niveau des champs de flux :

- La pression.
- La densité.
- La vitesse.

Les algorithmes associés sont donc souvent les mêmes que ceux utilisés pour la détection de bordure. Nous ne nous intéresserons pas aux algorithmes de détection d'ondes de choc en données 2D, car ne sont pas extensibles aux données 3D. Cette particularité présente également beaucoup d'intérêt, car il peut être important de pouvoir la contenir afin d'éviter la détérioration de structures (un exemple simple serait la simulation du déplacement d'une plaque tectonique et l'étude des effets ondes de choc qui en naissent sur certaines structures

civiles).

Détection basée sur le nombre de Mach dans les grilles non structurées

La détection d'une onde de choc est particulièrement compliquée sur les maillages issus de résolution à l'aide de méthodes des éléments finis et ce dû à la non-continuité et non-régularité de la répartition des points dans le maillage (Ma *et al.*, 1996).

Le nombre de Mach est le rapport de la vitesse à un point donnée sur celui du son dans ce média. Le postulat de départ est que le nombre de Mach passe de supersonique (nombre de Mach > 1) à subsonique (nombre de Mach < 1) en traversant la surface de l'onde de choc.

L'idée qui peut paraître relativement évidente (étant donné le postulat de départ) est de créer une isosurface où le nombre de Mach est égal à 1. Cependant, cette méthodologie n'est pas celle qui sera retenue, car elle ne permet de détecter qu'un cas particulier qui est nommé onde de choc oblique(Ma *et al.*, 1996) . Le nombre de Mach normal est alors comme le nombre de Mach selon la direction de la normale à la surface (ce qui pose un problème, car généralement on ne connaît pas les limites de la surface quand on manipule des données volumiques). C'est pourquoi il est préconisé d'utiliser le gradient de la densité qui approxime la normale de la surface.

Si on connaît les limites de la surface, on peut directement utiliser la méthode proposée préalablement.

Nous avons alors besoin de chercher le maximum du vecteur de densité, les dérivées premières et deuxièmes sont alors calculées et le niveau échelon pour le vecteur normal de densité est défini. Il est alors utilisé pour le calcul des isosurfaces. On utilise ensuite le nombre de Mach pour différencier le cas que l'on recherche des régions lisses et des ondes d'expansion.

2.5.3 Les bordures

Les algorithmes de détection des bordures sont assimilables à ceux de la détection d'onde de choc (Post *et al.*, 2003). Ils se basent sur la variation abrupte des propriétés physiques. On entend généralement par bordure une discontinuité dans la densité de matière. Dans certains cas, nous ne disposons pas de cette information, nous devons alors utiliser des propriétés qui pourraient être interprétables comme une discontinuité physique. L'algorithme premier de détection de bordure est l'algorithme de Canny qui est souvent utilisé dans le traitement d'images, dans notre cas il suffira de traiter sur les 3 composantes (x,y,z) et sur les différentes propriétés disponibles pour chaque point ou chaque primitive du maillage existant (évidemment, l'algorithme de Canny est adapté aux données uniformément réparties dans l'espace donc pour les données volumiques de type voxel).

Un algorithme plus complexe est un algorithme de Canny avec une fonction d'évaluation des contours différente (Bähnisch *et al.*, 2009).

Ces algorithmes ne sont pas applicables aux maillages non réguliers (tétraèdres, hexaèdres, etc.). Il est toutefois possible de déterminer les bordures en convertissant les données en voxels puis revenir à la représentation initiale après avoir construit la bordure de notre volume en évaluant l'appartenance des sommets de nos tétraèdres et en utilisant un algorithme d'évaluation de surface implicite de type tétraèdre baladeur. Cependant, cette approche devient problématique si les données manipulées sont géantes.

2.5.4 Lignes de courant

La ligne de courant est un objet mathématique généralement associé aux fluides, il peut être assimilé à un chemin. Mathématiquement, c'est une courbe où la tangente est parallèle à la vitesse du fluide en ce point. L'algorithme le plus récurrent est un algorithme qui utilise un système de particules qui se déplace selon le vecteur vitesse à chaque instant (Wang *et al.*, 2009). Le critère d'arrêt de cet algorithme est souvent le pas de déplacement qui atteint un minimum.

On peut parfois tomber sur le cas de fluides qui sont des lignes de courant fermées, dans ce cas la condition de fin est complexe. Si on se contente de dire que l'on passe approximativement par le même point et la même direction, on camoufle alors le cas du vortex ainsi que tous les cas de convergence lente vers un comportement particulier (Wischgoll et Scheuermann, 2001).

En considérant que nous sommes alors capables de détecter nos particularités pour à un état donné de notre simulation scientifique, il existe également dans la littérature une notion de cohérence temporelle d'extraction des éléments (Muelder et Ma, 2009). Cette information de cohérence permet idéalement de construire une notion d'objet afin de donner une réalité au maillage manipulé (ce qui simplifiera grandement l'étape du placement de la caméra).

L'idée est d'être capable de prévoir la trajectoire pour faciliter la localisation d'un élément, ou en tout cas de guider sa détection. La solution proposée par l'article (Muelder et Ma, 2009) est l'approximation de la trajectoire de manière quadratique en se basant sur les étapes précédentes. Les premières étapes représentent donc des cas particuliers qu'il faudra traiter. Le problème dans cette méthodologie est que l'on considère que le corps garde la même forme au fur et à mesure du temps, ce qui est souvent erroné dans les résultats numériques avec une composante temporelle. L'article propose alors une méthodologie qui prend en compte les déformations potentielles ; cette méthodologie est appelée croissance et rétrécissement. L'idée est de déformer légèrement notre objet (en y ajoutant des primitives ou en supprimant à chaque endroit de la frontière) pour avoir une meilleure estimation de l'état de l'objet à l'étape temporelle t .

2.5.5 Lignes de séparation et d'attachement

Les lignes de séparation (resp. d'attachement) sont les endroits où un fluide quitte (resp. entre en contact) avec une surface de manière abrupte. Elles sont très importantes à détecter, car elles ont souvent une très grande influence sur la portance en aéronautique (Post *et al.*, 2003).

Voici deux des algorithmes de détections qui sont adaptés à nos besoins :

Lancé de particules : des particules sont lancées à partir de sources et leur cheminement correspond à des lignes de courant. Cet algorithme permet de calculer les lignes de courant sous forme de courbe. On peut ensuite trouver les points d'attachement et de séparation en trouvant les zones où ces lignes fusionnent (Kenwright, 1998).

L'algorithme du vecteur parallèle : il détecte les lignes d'attachement et de séparation en étudiant les vecteurs propres de la matrice du tenseur de gradient de la vitesse (Kenwright *et al.*, 1999). On cherche alors les points où un des vecteurs propres est parallèle au champ de vecteurs. Il est également important d'éliminer le cas de figure des zones d'infexion ; c'est le cas où la condition précédente est satisfaite, mais que la ligne n'est pas une asymptote pour les lignes de courant. La condition finale devient alors : $\Delta(\mathbf{e}_i \times \mathbf{v}) \cdot \mathbf{v} = 0$.

2.6 Détection d'évènements

Une autre notion qu'il est intéressant de mettre en avant est la notion de détection d'évènements. Dans le cas de certaines particularités, nous pouvons établir qu'elles ont un cycle de vie donné. Un exemple simple serait celui de l'onde de choc qui se propagerait seule dans un milieu infini :

- naissance ;
- propagation et perte d'énergie par frottements ;
- mort.

Évidemment, c'est une simplification du cycle de vie, car il y a beaucoup d'autres cas possibles. Cependant, si nous sommes capables de définir un cycle de vie pour chaque particularité, la manipulation postérieure sera beaucoup plus facile et la visualisation sera plus pertinente. Les sous-sections suivantes décrivent un ensemble d'états qui sont définis dans la littérature (Reinders *et al.*, 2001).

2.6.1 Persistance

La particularité reste dans l'état dans lequel elle est sans interactions avec d'autres particularités. Ce n'est pas vraiment considéré comme un évènement, mais c'est un état nécessaire pour déterminer les transitions vers d'autres états dits plus intéressants.

2.6.2 Mort et Naissance

Une particularité qui disparaît (resp. apparaît) et diminue (resp. augmente) de taille. Deux critères doivent être satisfaits pour pouvoir dire que la particularité est dans cet état :

- La croissance doit être négative (resp. positive).
- La taille de la particularité doit déjà être petite. Si on atteint une taille négative, c'est que la mort est déjà survenue. On définit un facteur de correspondance pour cet état qui permet d'évaluer l'évolution vers cet état.

Les conditions pour la naissance sont simplement l'inverse de celles de la mort.

2.6.3 Entrée et Sortie

Cet état implique l'entrée (ou la sortie) d'une particularité dans un nouveau milieu au travers d'une frontière dite ouverte (frontière qui permet le passage au moins dans un des sens entre deux milieux de densité différente). De même que pour la naissance et la mort deux critères doivent être satisfaits pour pouvoir dire que la particularité est dans cet état :

- La particularité doit se diriger vers le nouveau milieu donc vers une frontière ouverte.
- La proximité à la frontière doit déjà être petite. Un facteur de correspondance est également défini dans ce cas pour être capable d'évaluer cette condition.

Il peut arriver que les critères soient vrais pour les deux états ; il faut donc prendre celui avec le facteur de correspondance le plus haut.

2.6.4 Scission et Fusion

Ces états s'intéressent à la manière dont plusieurs particularités interagissent entre elles. Prenons l'exemple de deux ondes de choc qui entrent en contact : il y a souvent perte d'énergie et réflexion. Pour d'autres particularités on peut avoir le cas où ces deux particularités se mélangent pour ne former plus qu'une seule. L'idée proposée par cet article (Reinders *et al.*, 2001) est d'essayer de prévoir les propriétés de la nouvelle particularité (axes, position et orientation).

2.6.5 Non résolu

Pour beaucoup de particularités, nous ne sommes pas réellement capables de proposer un modèle qui définit de manière déterministe ce qui va se produire à l'interaction de plusieurs particularités. Il est alors souvent nécessaire de relancer une détection de particularités pour savoir quel sera le résultat de cette interaction.

La visualisation des différents états est très utile lors de l'élaboration d'une narration. Une méthodologie pour la représentation du graphe des états est également proposée par cet

article (Reinders *et al.*, 2001). Elle nous permet de suivre l'évolution de chaque particularité au fil du temps ainsi que tous les différents changements d'état qui peuvent survenir.

2.7 Mécaniques de Storytelling

Nous nous intéressons aux mécaniques associées à la visualisation des particularités. Si l'on considère que l'on possède une liste de particularités et que pour chacune d'elles nous pouvons définir leurs propriétés sur leur durée de vie ainsi qu'un ensemble d'états et de transitions associés à chacune des étapes, il est alors possible de construire une méthode pour visualiser des données volumiques plus facilement.

2.7.1 Manipulation de la caméra

Le placement de la caméra dans le but de mettre en évidence une particularité est la problématique qui sera étudiée dans ce mémoire. Déterminer le chemin de la caméra revient à connaître les cinq informations suivantes pour la caméra à chaque instant du court métrage :

- La position de la caméra $P(x,y,z)$;
- La direction avant($\text{LookAt } V(x,y,z)$) ;
- Le vecteur «Up» ($U(x,y,z)$) ;
- Le champ de vision ;
- La profondeur de champ.

Dans la plupart des logiciels de modélisation et de visualisation, l'utilisateur spécifie ces cinq données à certains endroits/moments de la simulation et le reste des informations nécessaires à la production d'un court métrage sont générées par interpolation linéaire ou splinaire. Ces techniques ne sont pas une approximation de la manière de faire d'un cameraman et on trouve rarement des modélisations suffisamment complexes dans les solutions logicielles existantes (Christie *et al.*, 2008).

Dans notre cas, nous ne nous intéresserons pas aux déplacements dits «libres» de la caméra de la part de l'utilisateur ni aux déplacements dits «assistés» (où l'utilisateur à un contrôle partiel sur le chemin de caméra). L'automatisation de la construction narrative pour la navigation assistée est une opération très complexe, car l'utilisateur a beaucoup plus de liberté vis-à-vis de son interaction avec la scène.

En ce qui concerne le déplacement de la caméra en cinématographie et dans le jeu vidéo, une grande partie de la recherche a été menée sur le positionnement de la caméra pendant des dialogues ou simplement lors du déplacement d'un ou plusieurs personnages (Christie *et al.* (2008), Bourne *et al.* (2008), Abdullah *et al.* (2011)). La majorité des articles sur le sujet s'intéressent d'ailleurs à ces scènes impliquant des objets ou des personnages ; on

appellera «scènes humaines» toute construction de maillage qui pourrait avoir une existence macroscopique perceptible par un humain (bâtiments, objets, êtres vivants, etc.).

L'application de certaines règles du cinéma a également été approfondie. Un exemple que l'on retrouve dans la littérature est le respect de la règle des 180 degrés (Christie *et al.*, 2008) qui stipule que quand la caméra assiste au dialogue de deux personnages, toutes les prises de vues doivent être faites d'un seul côté de la ligne imaginaire qui existe entre les deux personnages, aussi appelée ligne d'action (Arijon, 1976) (cette règle pouvant être brisée pour un effet-choc chez le spectateur). Il existe d'autres règles du cinéma auxquelles nous pourrions appliquer ces règles : les champs-contrechamp, l'effet des distances focales, etc.

Compte tenu de la nature des scènes que nous manipulons, nous ne nous intéresserons pas à la recomposition de la scène (déplacer les objets pour une disposition plus favorable). Dans notre cas, elle est, a priori, prédéterminée et ne nécessite aucune modification d'ordre structurel de notre part.

On retrouve dans la littérature une notion d'information maximale sur un objet à partir d'un point de vue (Vázquez, 2009) Vázquez *et al.* (2004) Bordoloi et Shen (2005). Ces algorithmes sont, dans la majorité des cas, construits pour traiter des maillages surfaciques ou des données sous forme de voxels. On peut les répertorier en trois catégories d'algorithmes :

- La qualité : Évaluation de la qualité d'information projetée ;
- La vraisemblance : Comparaison du point de vue à tous les autres pour définir sa pertinence ;
- La stabilité : Évaluation un point de vue par rapport à son voisinage.

Les données les plus souvent utilisées pour évaluer ces critères sont des données projetées (carte de profondeur, carte de normales) (Vázquez, 2009) ou l'entropie (Vázquez *et al.*, 2004). L'approche de ces méthodes est de chercher le point de vue qui nous permettra d'avoir une information maximale, tout en ayant d'une variation minimale de certaines propriétés en espace image. En effet, si la carte de profondeur varie peu pour de légers déplacements de la caméra, c'est un point de vue que l'œil aura tendance à choisir pour observer un objet.

L'approche est similaire pour toutes ces méthodologies ; un ensemble de points sont placés de manière uniforme sur la sphère englobante de la surface 3D. On effectue un rendu du maillage depuis chacun des points et les informations obtenues depuis chacun des points sont alors comparées avec celles d'un ensemble d'autres points de vue.

Le point ayant la variation minimale de la donnée choisie est selon ces approches le meilleur point de vue pour observer cet objet. Cette approche est conçue pour des scènes uniquement constituées d'un seul objet. Dans le cas où l'utilisateur serait intéressé par certaines parties du maillage, il est nécessaire d'introduire des critères supplémentaires afin de s'adapter à ces besoins.

La technique de rendu par rastérisation est celle qui est utilisée pour la production des informations dans la majorité des cas ; elle est conçue pour gérer un nombre de triangles très élevé étant donné que le rendu de chaque triangle est fait de manière indépendante. Cependant, dans le cas où l'on souhaite effectuer des affichages complexes de tissus ou de fluides, il est parfois nécessaire de se diriger vers des techniques mieux fondées physiquement (lancé de rayon, tracé de chemins ou de mappage de photons).

Étant donné que l'on manipule des données de très grandes tailles, il est parfois nécessaire d'envisager des techniques de simplification du maillage avant d'appliquer ces algorithmes, et ce principalement dans le cas de contraintes de temps réel. La notion de focus+contexte (définition d'une région du maillage où la résolution sera plus élevée, appelée région d'intérêt) apparaît alors dans la littérature comme une bonne piste de solution à cette problématique (Barbier et Bonneau, 2008). La méthode proposée s'intéresse au cas particulier des maillages tétraédriques où l'on connaît déjà la localisation des régions d'intérêts. La notion de racine est définie comme le centre d'une région d'intérêt. Un graphe de parcours du maillage est alors construit à partir du tétraèdre central et on découpe notre maillage en clusters en fonction de la distance à la racine.

Après avoir construit nos différents clusters, on détermine les zones de transitions qui sont construites à l'aide de l'insertion de nouveaux tétraèdres. Cette construction de la zone de transition est nécessaire pour préserver la cohérence temporelle tout au long déroulement de la simulation. Cet article propose aussi l'algorithme GATOR (Barbier et Bonneau, 2008) qui permet d'effectuer une projection de tétraèdres dans un nuanceur de géométrie.

2.7.2 Génération procédurale de chemins de caméra

En ce qui concerne les problématiques de déplacement automatique de la caméra, on trouve dans la littérature une classification en quatre familles d'algorithmes (Christie *et al.*, 2008) :

Construction par optimisation

Cette catégorie d'algorithme se base principalement sur la maximisation (ou minimisation) d'une fonction d'agrégation qui fait intervenir chacune des caractéristiques de la caméra à travers une sous-fonction qui contribuent à la valeur de la fonction agrégation. On peut également jouer sur les poids pour donner plus ou moins d'importance à un critère. Il existe différentes méthodes pour optimiser la valeur de notre système :

- descente de gradient ;
- de Gauss-Seidel ;

- heuristiques ;
- probabilistes.

On retrouve des solutions à ce problème à l'aide des méthodes de discréétisation de l'espace, des directions et du champ de vue. D'autres auteurs traitent cette problématique à l'aide d'algorithmes génétiques. Ce sont des algorithmes qui fonctionnent sur le principe de la survie ; on crée un certain nombre de modèles de déplacement de caméra avec des positions initiales différentes.

La contrainte commune est d'avoir une direction définie au début et à la fin du parcours. Les trajectoires de chaque caméra sont jugées et les meilleures sont retenues en se basant sur leur évaluation de notre fonction contrainte. On va ensuite les «cloner» et les faire légèrement «muter» en modifiant aléatoirement une ou plusieurs de leurs propriétés. La sélection dite «naturelle» va faire qu'au terme de la procédure, certaines caméras auront des chemins plus intéressants que d'autres.

Malheureusement, la rapidité et la convergence de cet algorithme ne sont pas garanties. Certaines accélérations sont proposées pour accélérer la convergence (réduction de l'espace de recherche en utilisant des structures d'accélérations (BSP et Octree)). Une notion de cohérence image par image au niveau de la position, de la distance, la hauteur, l'orientation est introduite.

Certaines méthodes proposent d'autres approches se basant sur la topologie de la scène (une connaissance de la topologie permet une convergence plus rapide vers une solution optimale).

Construction par contraintes

Cette méthode diffère de la précédente par le fait que les différentes «contraintes» ne sont pas linéairement définies et le système doit être résolu et non optimisé. Il existe beaucoup d'algorithmes de résolution de systèmes de contraintes et ils ne seront pas tous détaillés dans cette revue de littérature. L'exemple que nous avons choisi de retenir pour l'état de l'art est celui proposé dans le cadre de recherche de chemin de caméra pour des scènes composées de personnages dans les jeux vidéo (Bourne *et al.*, 2008). L'idée est de pouvoir générer des chemins de caméra qui satisfont les critères suivants :

- produire un chemin en temps réel ;
- donner un point de vue clair sur les personnages ;
- être adaptable aux différentes structures des moteurs de jeu ;
- être adaptable aux différentes scènes que l'on peut trouver dans des jeux vidéo ;
- ne pas demander de précalcul ;
- proposer des trajectoires (pas de point de vue fixe).

En établissant un certain nombre de contraintes (taille de l'objet, cadrage, angle de vue, occlusion, etc.), la recherche du chemin optimal se fait à travers la résolution de ce système de contrainte. Plusieurs des éléments d'évaluation de points de vue cités dans l'article (Bourne *et al.*, 2008) sont réutilisés dans notre approche.

L'idée générale est de faire la recherche locale sous forme d'octree en prenant en compte la cohérence du chemin de la caméra. La solution pour tester l'occlusion est un tracé de quatre rayons qui permet d'évaluer l'occlusion de l'objet par rapport à la caméra.

Il est également évoqué dans cet article le cas où l'on ne souhaite pas visualiser seulement l'avatar du joueur, mais aussi des personnages non-joueurs (PNJs) ou autre. Un système de point de vue central où l'on introduit des poids sur chacun des objets à visualiser avec un poids plus élevé sur le personnage du joueur est présenté. Ce système permet de calculer dynamiquement le point de vue central.

Dans le cas d'une cinématique, il est envisageable d'introduire une machine à états pour mieux modéliser le placement la caméra. Une approche hiérarchique pour la machine à état est souvent préconisée. La gestion des différents états permet d'implémenter de manière plus évidente les codes de la composition cinématographique. Dans le cas des jeux vidéo, on peut aussi se baser sur des interrupteurs qu'actionne le joueur dans le jeu pour passer d'un état donné à un autre.

Bien que les problématiques développées dans cet article soient spécifiques à des scènes humaines, nous envisageons d'adapter ces solutions aux problématiques de la visualisation scientifique.

Il est également possible de résoudre les systèmes de contraintes par morceaux et assembler les différentes sous-solutions avec des transitions adaptées. D'autres méthodologies pour la modélisation des contraintes existent également dans la littérature. CONSTRAINTCAM, par exemple (Bares *et al.*, 1998), propose une modélisation hiérarchique des contraintes de la caméra. Cette modélisation est utile dans le cas où l'on n'arriverait pas à satisfaire l'ensemble des contraintes existantes, leur hiérarchie permet de trouver les zones où la satisfaction est meilleure pour les nœuds proches de la racine.

Construction par optimisation sous contraintes

Cette catégorie d'algorithmes est un mélange entre les deux précédentes catégories. L'idée est que l'on peut définir un niveau de satisfaction des contraintes et que l'algorithme va tenter de trouver le cas de figure le moins défavorable. Il reste toutefois très compliqué de déterminer les heuristiques d'évaluation ; entre la satisfaction d'une contrainte aux défaveurs d'une autre ou la faible satisfaction des deux contraintes, il est complexe de définir quel est le bon choix. Cette catégorie d'algorithmes donne souvent des résultats plus satisfaisants que les deux

autres catégories réunies.

Un exemple serait CAMDROID (Drucker S. M., 1992) qui propose un système de caméra qui a des contraintes à respecter et des objectifs à remplir au mieux. La seule contrainte réelle qui existe sur ce système est que les contraintes et objectifs doivent être définis continus et dérivables (ce qui n'est pas toujours le cas) ; le paquet CFSQP (*Feasible Sequential Quadratic Programming*) est utilisé pour la résolution du système.

Approches réactives

Les approches réactives sont des approches basées sur des algorithmes de traitement d'images. L'idée est de modifier le chemin de la caméra en se basant sur une ou plusieurs images. Ces algorithmes sont beaucoup utilisés en robotique, car les robots disposent généralement comme principale information de guidage les images produites par une ou plusieurs caméras. La résolution se fait également à l'aide de systèmes d'optimisations avec une fonction de coût évaluée à chaque instant étant donné le déplacement de(s) caméra(s).

2.7.3 La visibilité

Les problématiques d'occlusion sont nécessaires à traiter ; c'est un problème complexe à résoudre, surtout dans notre cas de figure. Même si on est capable de déterminer l'endroit où l'on doit se placer pour observer un objet de manière fondée, rien ne nous garantit qu'aucun objet ne s'interpose entre la caméra et cet objet, partiellement ou totalement.

Le tracé de rayon est souvent utilisé, car c'est un algorithme simple à mettre en place. Cependant, il est très coûteux d'avoir un pourcentage d'affichage de l'objet à l'écran en utilisant cette technique, car très mal adapté à des maillages très denses à moins de bénéficier de structures d'accélération.

Une autre solution existante est de construire des boîtes englobantes autour des objets et en utilisant l'algorithme des volumes d'ombre, on est aussi capable de déterminer analytiquement la proportion visible des objets (Pickering, 2002). D'autres techniques utilisant le tampon de stencil du pipeline de rendu des API communes permettent d'avoir ces informations étant donné que l'on s'intéresse principalement à l'occlusion par rapport à un *frustrum* d'observation en utilisant comme approximation la rastérisation (Halper et Olivier, 2000).

2.7.4 Les règles de la composition

On retrouve également dans la littérature des articles qui proposent des méthodologies basées sur des codes de compositions cinématographiques et photographiques (Abdullah *et al.*, 2011). L'idée est d'effectuer un rendu de la scène pour le traitement de certains des objets

rendus à l'écran appelés «*ruled objects*». L'article propose de se focaliser sur quatre règles dites «avancées» de la littérature cinématographique et photographique ainsi que quelques règles de base de la composition, en voici quelques-unes :

— **La règle des tiers :**

Cette règle dit qu'il faut placer le sujet principal d'une scène sur une des quatre intersections de deux lignes horizontales et deux lignes verticales également espacées.

— **Dominance diagonale :**

Cette règle dit que si la plupart des lignes présentes dans une scène sont plus ou moins alignées sur la diagonale la scène aura l'air plus vivante.

— **Balance visuelle :**

La répartition plus uniforme des objets sur l'image finale amènera un meilleur équilibre.

— **Profondeur de champ :**

En définissant un objet central et en appliquant un flou qui serait assimilable à une profondeur de champ, il est possible de guider l'attention du spectateur lors de son interaction avec l'image.

Les codes cinématographiques n'imposent pas que toutes les règles de l'image soient vérifiées à tout instant et à chaque plan. La méthodologie propose d'en choisir un certain nombre et de définir un pourcentage qui juge à quel point cette image respecte les codes qui la concernent. L'implémentation proposée par cet article est uniquement implémentée sur CPU et est dite conçue pour le rendu différé, toutefois l'auteur parle de la possibilité d'une accélération GPU qui pourrait faire passer cette technique en temps réel.

Nous intéressons aux méthodologies qui existent pour produire des scénarisation. On retrouve une notion de directeur (en référence au cinéma) dans la littérature (Lino *et al.*, 2010). La méthodologie est divisée en 4 étapes majeures :

Sélection d'éléments narratifs

À chaque transition du système, un processus détermine quel est l'évènement de plus grande importance qui a actuellement lieu sur la scène

Calcul des volumes directeurs

L'étape suivante est de calculer les volumes directeurs. C'est un ensemble de volumes qui permettent de cerner les caractéristiques principales de l'élément choisi à l'étape précédente. Les volumes directeurs peuvent être divisés en deux types de volumes. Les volumes sémantiques correspondent à des volumes où l'on peut appliquer certains des codes du cinéma (dont

nous avons parlé plus haut), on utilise des «*Binary Space Partitions*» (a-BSP) pour conserver leur emplacement après construction. Le deuxième type de volumes est les volumes de visibilité, ce sont les zones où l'élément narratif est plus ou moins visible (d'où l'introduction de la notion de visibilité partielle) ; on stocke également ces volumes dans un a-BSP. En se basant sur ces deux volumes, on calcule les volumes directeurs en effectuant une fusion des deux a-BSP construits précédemment.

Édition des volumes directeurs

Cette étape s'effectue en 3 passes :

Filtres de continuité : ces filtres permettent d'éliminer les zones qui pourraient représenter une discontinuité au niveau des changements de prise de vue. Le viol de la règle des 180 degrés est l'exemple donné.

Filtres de style : Cette catégorie de filtres permet de sélectionner les prises de vues en accentuant sur celles qui sont directement liées aux procédés narratifs.

Composition : Cette étape est l'association des différentes prises de vues sélectionnées. La transition se fait généralement à la fin d'un évènement donné et la recherche du chemin de transition se fait à l'aide de l'algorithme Dijkstra.

2.7.5 Chemin de caméra en visualisation scientifique

Tout ce qui a été présenté jusqu'à maintenant est pensé pour des scènes humaines. D'autres articles proposent des méthodes plus guidées qui permettent à l'utilisateur de réellement construire une visualisation de manière itérative pour les scènes de la visualisation scientifique.

Ces solutions sont celles proposées par les logiciels de visualisation scientifique. L'idée est de guider le parcours du scientifique pour permettre d'explorer les données en même temps qu'il bâtit des modèles visuels qui mettent en avant les spécificités des résultats numériques qu'il possède (Wohlfart et Hauser, 2007).

Quelques rares articles proposent également des méthodologies complètes pour générer des chemins de caméra dans le but d'observer des maillages 3D issus de visualisation scientifique en se basant sur la stabilité des points de vue échantillonnes (Chan *et al.*, 2008). Il se limitent à des maillages voxels ou surfaces et ne proposent pas l'intégration de règles de composition ainsi que de notions de storytelling.

Enfin, il existe des articles qui proposent des méthodologies pour construire des animations de scène avec des méthodologies assez similaires à celles que l'on peut retrouver dans les logiciels de montage vidéo avec la possibilité de composer des effets. On retrouve une ou plusieurs lignes temporelles avec la possibilité d'organiser la visualisation, mais aussi les transitions entre chaque étape (Koop, 2008).

Comme nous l'avons vu au travers de cette revue de littérature, même si certains scientifiques s'intéressent particulièrement aux mécaniques de Storytelling en visualisation scientifique (Wohlfart, 2008), il subsiste un réel manque pour la production de support de présentation des résultats dont un scientifique peut avoir besoin pour communiquer ses résultats à un public ; principalement pour critères de visualisation basés sur les particularités à observer. Les techniques présentées se concentrent souvent sur des sous-parties du problème et demandent une supervision constante du scientifique.

C'est alors qu'apparaît la problématique sur laquelle porte le travail décrit dans ce mémoire. La nécessité de proposer une méthodologie complète qui permet de produire, et ce de manière automatique, des chemins de caméra qui peuvent présenter les données numériques issues de simulations scientifiques et prendre en compte des particularités que désire observer le scientifique.

2.8 Questions de recherche

Les questions de recherche à l'origine de ce travail proviennent de la volonté d'appliquer des méthodes de storytelling à la visualisation scientifique. Comment modéliser les besoins d'un scientifique vis-à-vis de ses données ? Comment observer ce qu'il a retenu comme étant intéressant ? Comment appliquer les codes de composition à des scènes ne contenant aucune structure scénaristique ?

2.9 Objectif général

Proposer une méthodologie afin de permettre aux scientifiques de présenter leurs résultats, et ce indépendamment des particularités qui peuvent les intéresser.

2.10 Objectifs spécifiques

- O_1 : Établir une métamodélisation de la visualisation de particularités en se basant sur les algorithmes de détections existants ;
- O_2 : Proposer des heuristiques pour estimer la qualité des points de vue vis-à-vis du métamodèle établi ;

- O_3 : Bâtir une modélisation pour les besoins de visualisation des scientifiques ;
- O_4 : Intégrer un ensemble de critères supplémentaires pour la production de chemins de caméra ;
- O_5 : Intégrer des règles issues du montage, de la composition photographique et cinématographique ;
- O_6 : Valider le fonctionnement de notre méthode sur plusieurs ensembles de données.

2.11 Hypothèses

- H_1 : Il est possible de bâtir une métamodélisation des particularités qui puisse garder les spécificités de chacune ;
- H_2 : Il est suffisant que le processus de placement de caméra se base uniquement sur le métamodèles de comportement de données choisi afin de transmettre la bonne intention ;
- H_3 : Le placement de la caméra pour observer une particularité se base sur une représentation, mais les résultats produits seront équivalents pour d'autres représentations de cette particularité qui servent le même message dans le processus de visualisation ;
- H_4 : Les règles de compositions cinématographique et photographique peuvent être appliquées à des scènes non humaines.

CHAPITRE 3

Aperçu de la méthodologie

Ce chapitre résume les travaux effectués et présentés dans le cadre de ce mémoire. Dans un premier temps, nous synthétiserons le contenu de l'article présenté au chapitre 4 et nous entamerons dans un second temps une discussion à propos de points non développés dans le cadre de l'article au chapitre 5.

3.1 Présentation de l'article

3.1.1 Titre

A Practical Method For Automatic Camera Paths Generation In Scientific Visualization

3.1.2 Auteurs

Anis Benyoub : Étudiant à la maîtrise à l'École polytechnique de Montréal en génie informatique dans le cadre d'un double diplôme avec l'école d'ingénieur INSA de Lyon.

Benoît Ozell : Professeur au département de génie informatique et génie logiciel de l'École polytechnique de Montréal et directeur du laboratoire d'infographie et de réalité virtuelle.

3.1.3 Publication

L'article a été déposé pour publication à la conférence *IEEE SciVis 2015* le 26 mars 2015 en vue de parution ultérieure dans la revue *IEEE Transactions on Visualization and Computer Graphics*.

3.1.4 Résumé

L'article rappelle dans un premier temps le contexte présenté précédemment dans les objectifs du mémoire. Il fait ensuite un état de l'art assez succinct qui présente les travaux précédemment publiés relatifs à la détection et le suivi de comportement notable dans des données numériques issues d'acquisitions ou de simulations scientifiques ainsi que ceux liés à la génération de points de vue et de chemins de caméra dans le cinéma, le jeu vidéo et la visualisation scientifique.

Dans le but de remplir les objectifs du mémoire, l'article présente la problématique qu'il essaye de résoudre : l'importation de techniques liées aux médias artistiques à la visualisation scientifique et l'automatisation de la génération de chemins de caméra pour les scènes numériques contenant des données 3D volumiques, surfaciques ou vectorielles.

Une passerelle entre les codes du storytelling, la scénarisation et la visualisation scientifique est alors établie et permet de définir l'approche de construction de la méthodologie. Des notions pour la modélisation de scènes de résultats scientifique et des attentes des utilisateurs sont alors introduites : *le cliché, la zone d'intérêt et les besoins de visualisation*. Ces notions permettent de simplifier l'approche des données et des attentes, autant que de simplifier les traitements d'une scène et des particularités qu'elle peut contenir tout en préservant leurs spécificités.

L'article développe ensuite l'ensemble de la technique de génération de chemins de caméra au travers des contraintes utilisées (qualité, occlusions, courbure, etc.) du détail de processus d'interpolation et de règles de la composition photographique importées dans la méthode.

Une série d'extensions de la méthodologie de génération est alors mise en avant afin de proposer des chemins de caméra plus variés, plus pertinents, qui gèrent des maillages variants dans le temps ainsi que l'ensemble d'interaction que peut avoir l'utilisateur avec le processus de génération afin de l'ajuster à ces besoins spécifiques.

L'ensemble des scènes numériques et les besoins de visualisation utilisés pour tester la pertinence de la méthode sont introduits et les résultats illustrant les chemins de caméra produits, leurs temps d'exécution et des métriques d'évaluation de qualité de points de vue sont présentés. Les résultats présentés sont discutés afin d'établir la pertinence de l'ensemble des choix faits et afin de juger l'efficacité le processus de génération de chemins de caméra et les contributions faites par l'article.

Les extensions futures prévues dans le cadre de la méthodologie sont rapidement présentées au travers de pistes de recherche que nous avons développées. Finalement, un retour sur la méthodologie développée et ses apports est fait afin de résumer les points importants qui sont abordé dans le cadre de l'article.

3.2 Travail complémentaire

Dans un second temps, nous discuterons plusieurs points concernant notre approche du problème et la méthodologie développée qui n'ont pas été développés ou évoqués dans l'article. Des images et résultats complémentaires plus détaillés seront également présentés et analysés dans le but de mieux évaluer la méthode.

CHAPITRE 4

ARTICLE 1 : A Practical Method For Automatic Camera Paths Generation In Scientific Visualization

4.1 Abstract

Camera path planning is a recurrent task in computer vision (in particular for displacements in robotics) and 3D computer graphics. Solutions based on constraint system resolution and system optimization have proven to be efficient ways to build interesting camera paths either in a real environment or a virtual scene.

Even if feature detection has been deeply investigated, generating a relevant camera path for exploring or presenting scientific numerical results is still an often encountered problem. Scientific visualization scenes, in particular, are at a disadvantage when compared to video games and cinematographic scenes : characters, objects and environments are notions that do not exist and that makes the problem much more complex to solve. In this paper, we introduce a meta-modelisation of scientific scenes (*the cliche*) using storytelling notions. We are then able to import camera path generation techniques from cinema and video games and present a complete methodology to automatically create camera paths for scientific results and to easily produce video sequences.

Our results from testing the methodology on various scenes samples showed that it is a quick and efficient way for generating multiple scene-relevant camera paths. Even if the method is automatic-generation oriented, it still gives a lot of flexibility to the user in the path generation process and choice during the film editing. Producing a final cut to present results becomes an easier task for a scientist.

4.2 Introduction

Scientific visualization is the set of techniques used to represent and visualize scientific results. Scientific results are produced either by simulation (FEM, Monte Carlo, etc.) or acquisition (Scanner, FMRI, etc.). In both cases, these data can be of variable nature (discrete or continuous) and dimension (linear, surfacic or volumic).

Since a scientist has precise needs from his data, visualization techniques help him satisfy his need regarding exploration and presentation. He generally expresses his expectations in one of the following ways : "I would like to observe the zones where..." or "this feature seems interesting, I would like to visualize it.". However, natural language processing complexifies

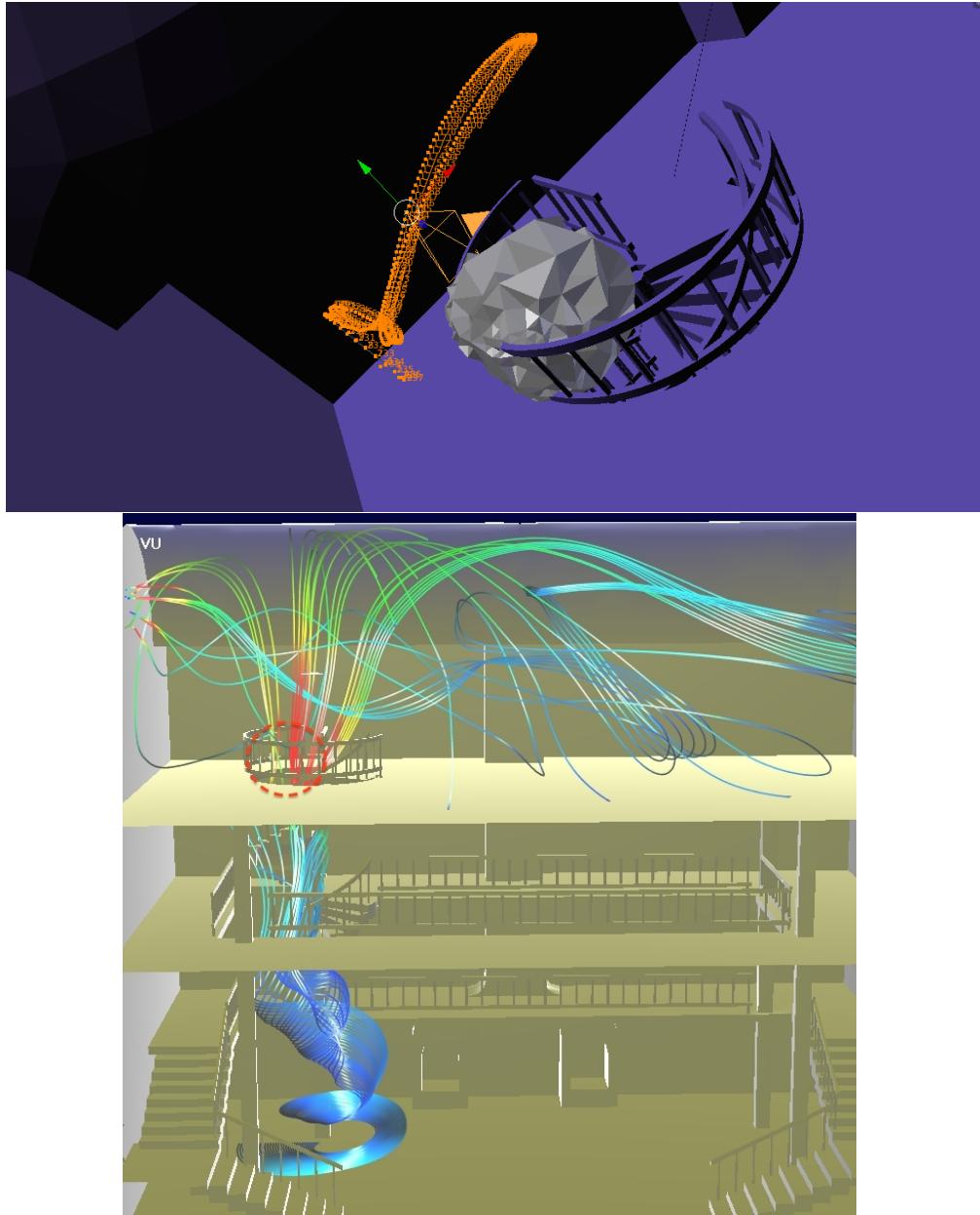


Figure 4.1 The presented scene is the numerical result of a simulation of air propagation within a building using the finite element method. The purpose of the scene is to observe the evolution of smoke concentration in the air. The visualization of streamlines within the scene (produced by the VU visualization tool) reveals interesting data behaviors such as vortexes and separation lines (bottom image). Our method has been used to generate a camera path in a way to observe features within a specific mesh zone (named spot of interest). The spot of interest, which presents interesting separation lines, is located on the red sphere represented in the bottom image. The mesh boundaries (steps, walls, floor) have been defined as complete occluders and the expressed visualization need was the fluid streamlines. The top image presents an automatically generated camera path visualized using a regular 3D animation and movie editing tool.

understanding what the scientist really expects. Our goal is to identify his needs regarding camera path planning for a better observation of the data.

Two categories of algorithms for scientific visualization have been developed : feature detection (Post *et al.*, 2003) and data visualization (Fox et Hendler, 2011) algorithms. Data from scientific visualization are often of large size and very complex. In both cases, the algorithms should then be numerically stable and scalable.

In this paper, we focus on feature detection for camera placement. Features are data behaviors that are interesting to detect in scientific results. In a single scene, a large number of features could be detected and only the scientist knows which ones he is interested in. Defining an exhaustive list of features that could interest scientific would be a long and probably useless task. However, some of them are recurrent and their techniques of detection have been deeply explored.

Vortex is an interesting behavior that is related to fluid mechanics and can be, in some cases, very damageable if unplanned and undetected. It exists when streamlines mapped onto a plane normal are seen to take a spiral or circular shape (Post *et al.*, 2003).

Shockwave is a feature which is the consequence of discontinuity in physical properties such as pressure, density and velocity. It can cause performance issues, it is therefore interesting to locate and observe ((Ma *et al.*, 1996)).

Streamlines is a feature also linked to fluid mechanics. It's a set of curves that are tangent to the velocity vector at each point. They are easy to build using particle systems (Wang *et al.*, 2009).

Separation and attachment lines are zones where a fluid leaves or reaches a surface abruptly. They are very important to detect because they influence, for instance, the lift force (Post *et al.*, 2003).

For each method, analyzing the algorithms' nature and outputs is a solid base for building our methodology ; even if the final goal of this paper is not developing a new feature detection approach.

Time varying data require more processing. State tracking (Reinders *et al.*, 2001), region (Muelder et Ma, 2009) and time coherency (Akiba *et al.*, 2010) are three important facets of feature detection. These approaches suggest models for the different behaviors of the features and the different interactions that can happen between them.

The visualization step has been investigated during the recent years. Our main interest remains camera path planning and we will not investigate the reflection models for the realistic rendering of scientific results.

4.3 Previous work

In scientific visualization, the notion of camera position seems to be linked to viewpoint quality. The main notions that are developed are stability (Vázquez, 2009) and entropy (Takahashi *et al.*, 2005; Vàquez *et al.*, 2002). Those notions are linked to viewpoint quality of a surface. We also find some papers evaluating the viewpoint quality for voxel-type volumic data (Bordoloi et Shen, 2005).

In cinematographic and video game scenes, other criteria must be taken into account for camera path planning such as occlusion (Christie *et al.*, 2008) and photography composition rules (Abdullah *et al.*, 2011). The intrinsic properties of these scenes are due to their structures (scene objects, characters, environment, etc.). These properties are not automatically available in scientific results, especially volumic data.

Camera path planning is a combinatorial problem. Combinatorial optimization and constraint systems resolution have proven to be two main techniques for quickly finding a good solution (Bourne *et al.*, 2008). Multiple techniques have been developed to automatically build relevant camera paths (Lino *et al.*, 2010) for movies and video games.

In scientific visualization, few papers (Chan *et al.*, 2008) present full methodologies for camera path planning for volumic data. However, these do not take into account complex properties of the scene.

Other techniques permit to build gradually complex visualization (Koop, 2008; Wohlfart et Hauser, 2007). Their main idea is to help the user to create a scenario of his data like a *director* would do with cinematography.

In related work, we have been investigating the application of cinematography and photography rules in scientific visualization (Benyoub, 2015).

In this paper, we introduce a new feature-based representation of scientific scenes based on storytelling codes : the cliche. Introducing this notion in scientific visualization makes building a methodology for automatically generating feature-based camera paths take into account scene intrinsic specificities an easier operation. Our method targets three-dimensional scene containing surfacic or volumic data.

4.4 Camera Path

4.4.1 Storytelling And Features

Storytelling is the art of telling stories (Ma *et al.*, 2012b). The screen writing parallel is mostly done with the cinematographic language. In artistic media, stories need to involve characters and actions (Figure 4.2). An action is a vague concept ; it can go from interaction

between the characters and the environment to something as simple as the existence of the character.

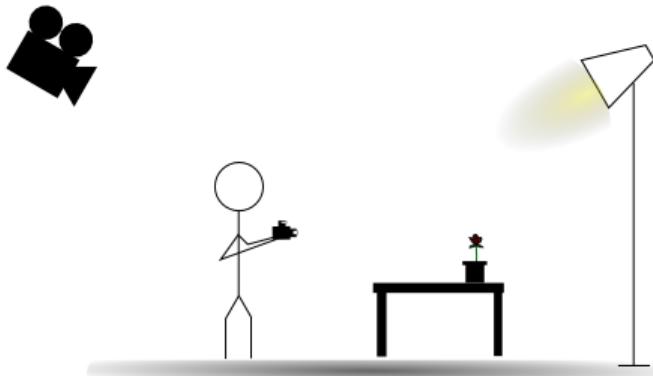


Figure 4.2 A simple story : a stick-man (character) taking a photo (action) of a plant.

In storytelling rules, the viewer needs to somehow identify or reject the character's actions (Truby, Oct 2008). The more the viewer will feel concerned by the actions, the bigger the story will have an impact on him. The characters do not have to be humans to have a huge impact on the viewer ; they do not even have to be living beings. A good example of this statement is the introduction scene of the movie *Lord of War* (Niccol, 2005). In the first scene, a fixed close-up is done on a bullet through its life : from its production to its fire. This scene, even if the main character is an object that doesn't make any action, has a really strong impact on the viewer.



Figure 4.3 The introduction scene from the movie *Lord of War* (Niccol, 2005).

Our method creates a notion of scene that is missing in a most of volumic scientific data results. The global approach is to build a parallel between these results and more conventional

scene structures (that can be found in artistic media). This goes through building a meta-model scene structure for volumic data results. The missing notion of character in scientific visualization is one huge hole in the storytelling process. Replacing this notion is the first step in our methodology.

4.4.2 Feature Analysis

For a single scene, multiple stories could be told. Each user may be interested in a different story and it is important to get him to express what he is expecting. This is what we call a visualization need.

Features could be our characters : we try to observe them and their interaction with the environment. However, defining a camera path generation pipeline that handles the specificities of every feature would be a hard task. In cinematography composition rules, focusing on the character while he is doing the action is sufficient to get the viewer understand the point of the scene. An example to make this statement clearer is the Mexican standoff scene in the movie *The good, the bad and the ugly* (Leone, 1966) where the viewer feels the tension of the scene even without having close-up shots on the characters' faces (Figure 4.4). Our approach is then to view features as actions (rather than character) and their local environment as the characters.



Figure 4.4 The Mexican standoff scene in the movie *The good, the bad and the ugly* (Leone, 1966).

The starting point is to create a meta-model of the different visualization needs and take abstraction of the data ontology (i.e. the set of scientific terms that it is related to). Our approach is to define features through the way we represent them. Each feature (a data behavior) introduced in Section 4.2 can be represented in one or multiple ways (Figure 4.5).

- **Vortex** : The vortex could be represented as a vector field (Figure 4.5), an isosurface or simply by the vertex core.
- **Shockwave** : The most common representation of a shockwave is an isosurface (Figure 4.5) ;
- **Streamlines** : Streamlines are represented either as a set of curves or as the vector-field of the fluid speed's ;
- **Attachment and detachment lines** : They usually have the same representation than streamlines ;

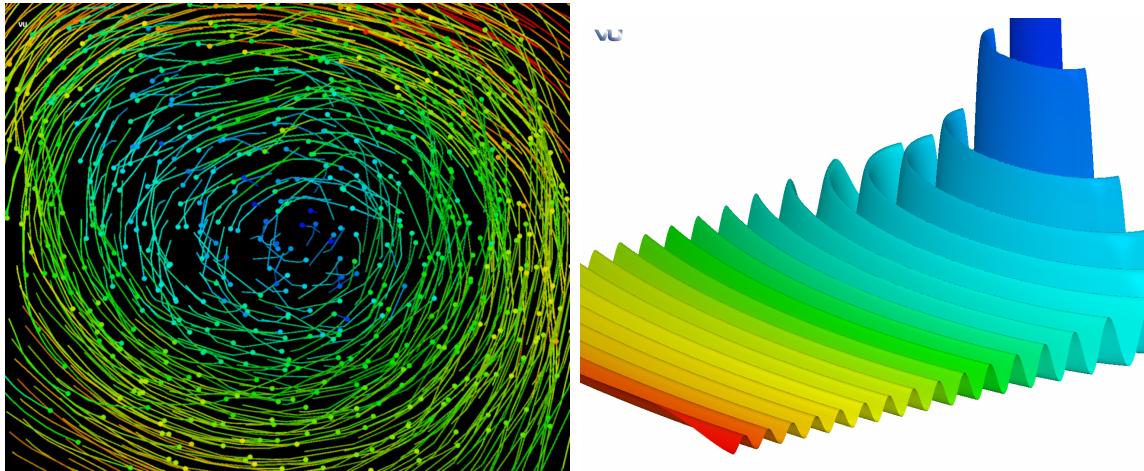


Figure 4.5 From left : A vortex represented as a vector-field and a shockwave as an isosurface.

In addition to those features, simpler data behaviors may also be interesting to observe : **scalar field**, **vector field**, **vector field variations**, **surface mesh** and **volume mesh**.

The main assumption of our method is to state that the representation of a feature constrains its visualization and especially the camera path generation. Usually, representation is defined after the path generation and the two operations and decorrelated (Koop, 2008).

4.4.3 The Cliche

Some simplifications can be made to our model, so far, in a way to reduce the set of different basic visualization needs :

- When we observe a scalar field, we are generally interested in zones where larger variations occur. Using the gradient operator on the mesh generates a vector field that preserves this information.
- A curve can be discretized as a piecewise-constant set of vectors. Handling a curve set becomes similar to a vector field.

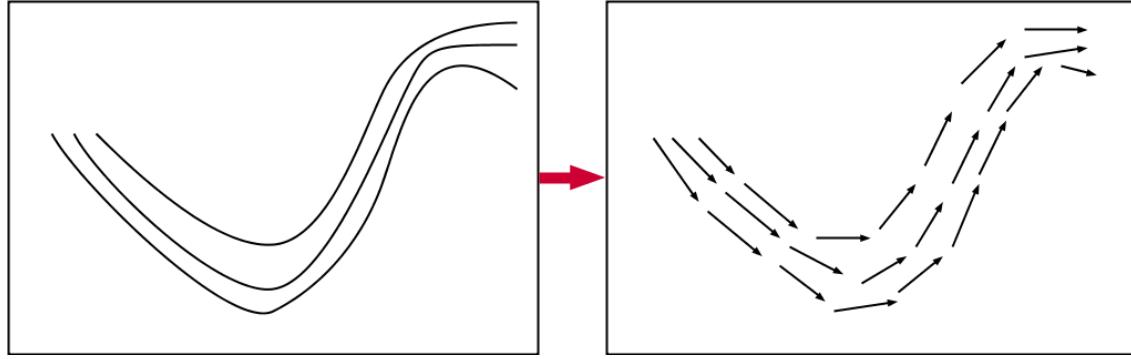


Figure 4.6 Converting a curve to a vector-field.

- Applying the Jacobian operator to a vector field, switching to the world basis and summing the components produces a good heuristic to estimate a vector field variation. For a tetrahedron, let $A = \{\vec{a}_1, \vec{a}_2, \vec{a}_3, \vec{a}_4\}$ be the vector values at each node, M the transformation matrix of the basis centered at the point p_1 and A_J the new vector-field component.

$$A_J = M^{-1} \times ((\vec{a}_4 - \vec{a}_1) + (\vec{a}_3 - \vec{a}_1) + (\vec{a}_2 - \vec{a}_1)) \quad (4.1)$$

$$= M^{-1} \times (\vec{a}_4 + \vec{a}_3 + \vec{a}_2 + \vec{a}_1 - 3\vec{a}_1) \quad (4.2)$$

Taking into account the previous simplifications, feature visualization can be meta-modeled into three primary visualization needs that we will be named as cliches :

- Visualizing a surface.
- Visualizing of a vector field.
- Visualizing of a volume.

All visualization needs that can be expressed by a user are convertible to one of these cliches.

4.4.4 Spot of interest

The next step of the methodology is to introduce a notion that is assimilable to a character in a way to use interesting camera path planning approaches from video games and cinematography : a spot of interest (SOI). A SOI is meant to define a specific mesh region that includes the targeted feature(s). In a way to have an efficient mesh extraction process, a SOI is defined as the intersection of a star-shaped surface with an implicit representation and the mesh. The latter choices make capturing the cliche easier and the inclusion test scalable.

Star shaped regions have the property of inducing less self-occlusion than a non constraint surface and offers more flexibility for defining a mesh region than a concave surface.

Camera positions are generated by doing a regular parameterization on the bounding sphere of the SOI. The *look At* position of the viewpoint being the center of the spot of interest. Each sampled camera position will be used as a potential viewpoint for the camera path generation pipeline.

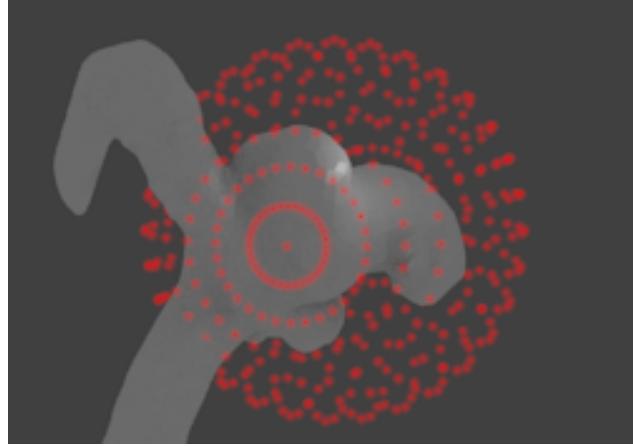


Figure 4.7 A spherical SOI around an aneurysm.

Our method requires that SOIs have been positioned before launching the generation process. They could be either placed manually by the user or automatically using features detection and tracking techniques.

4.4.5 A Good Viewpoint

Even if the notion of "quality" for a viewpoint is strongly related to camera placement, a good camera path is not simply the combination of good viewpoints. Camera paths are about building a coherent succession of camera plans and create an information continuity between them. Quality criteria are heuristics to define how good a viewpoint is. One or multiple heuristics can be used to define a notion of quality for one single viewpoint. In our methodology, we use one heuristic for each cliche :

- **Surface** : The stability of a given viewpoint is done by evaluating the similarity of its depth map compared with its neighbors. The more the depth map will be similar, the more the view point is said to be stable. The compression distance (Equation 4.3) has proven to be a good heuristic for the stability criterion.

$$C(X, Y) = \frac{C(XY) - \min(C(X), C(Y))}{\max(C(X), C(Y))} \quad (4.3)$$

X and Y being the depth maps to compare, XY their concatenation and C the com-

pression using *bzip* format.

- **Vector field** : The heuristic we are using is the ratio of projected norm into the camera space over total length. The more a vector will be perpendicular to the camera's Z axis, the better the viewpoint will be to observe it (Figure 4.8) ; $Q_v = \frac{\sum_{i=1}^N |M \times v_i|}{\sum_{i=1}^N |v_i|}$ with Q_v the vector field quality heuristic, v_i the i^{th} vector in the SOI and M the inverse view-matrix of the viewpoint.
- **Volume** : Ratio visible surface. The volume's edges are extracted and the viewpoint with the larger projected front face surface is supposed to be the best (Figure 4.8) ; $Q_s = \frac{\sum_{i=1}^N S_i^*}{\sum_{i=1}^N S_i}$ with Q_s the surface quality heuristic, S_i^* the surface of the i^{th} projected primitive and S_i the surface of the i^{th} primitive.

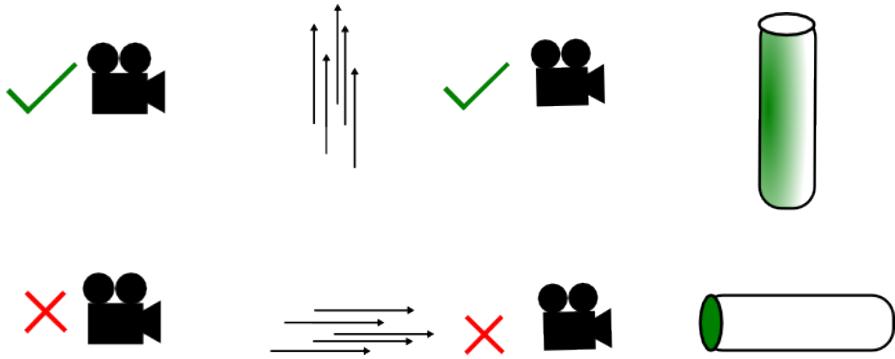


Figure 4.8 Left : vector field quality heuristic. Right : volume quality heuristic.

Other heuristics may be used for the technique (mesh density, vorticity, etc.), but the ones we use give good results for our set of scenes.

4.4.6 Visualization Needs

Cliches presented in section 4.4.3 are too basic to be directly expressed by a user. The users should have the choice between more intuitive visualization needs. In our methodology, we focus on a list of user choices that includes the majority of possible visualization needs :

- Iso-surface ;
- Zone edge ;
- Zone mesh ;
- Vector field ;
- Vector field variations ;
- Scalar field.

4.4.7 Constraints

In our method, the SOI is assimilated to a character in a scene. The constraint on the character's actions is defined through the quality criterion. However, other scene interactions must be taken into account for making the subset of camera paths richer and more relevant.

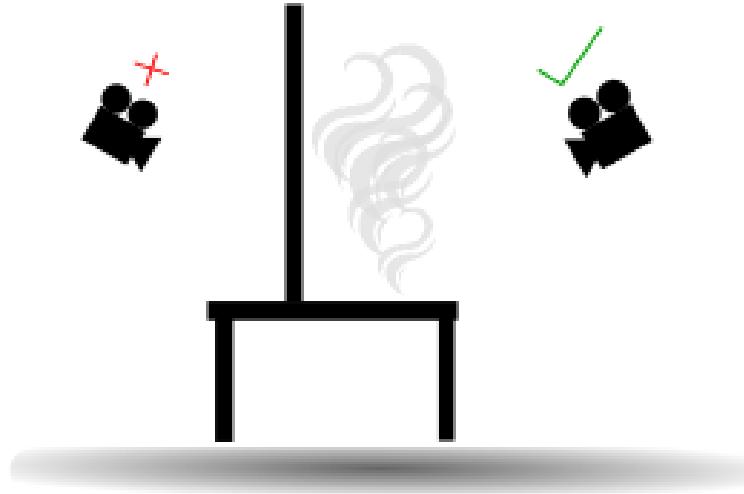


Figure 4.9 Occlusion while observing smoke propagation.

The visibility is a primordial information in photography composition when applied conventional scenes. Occlusion and auto-occlusion are important criteria to take into account when evaluating how good a viewpoint is. In our case occlusion is created by other SOI in the scene. For instance, walls can create occlusion while observing smoke propagation (Figure 4.9) and parts of a mesh can create occlusion while observing a specific subpart, for example, in a aneurysm (Figure 4.10).

Volumic SOI can also create partial-occlusion on the main SOI; an occlusion factor may be specified by the user to add this criteria.

4.4.8 Graph navigation

Building a camera path is assimilable to a graph traversal problem ; viewpoints are nodes and transitions between viewpoints are the edges. In our case, it is processed like an iterative local optimal search.

Each graph node has a set of criteria and each criterion is paired to a weight. However, viewpoint specific criteria are not sufficient to have a continuity constraint between viewpoints. Adding constraints over the edges must be done to preserve the path coherence. For

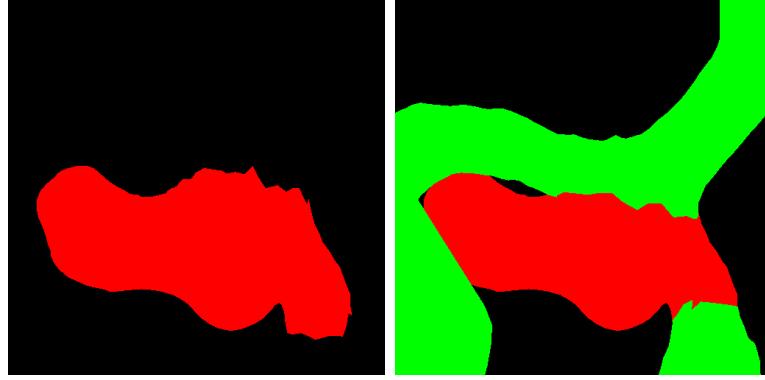


Figure 4.10 Occlusion of the aneurysm by other components of the scene (occ. at 17.1138%).

this, four addition criteria must be taken into account (Figure 4.11) :

- **Distance** : The distance (d) between the current and the next node. A closer node will have tendency to be more coherent with the current one ;
- **Orientation** : The dot product ($\cos \theta$) of the forward direction of the current (ω_o) and the next node (ω_o^*). Two nodes that look in a closer direction will give a smoother path ;
- **Curvature** : The dot value ($\cos \phi$) between the current path tangent and the current-next node vector. Abrupt direction changes in the path must be avoided ;
- **Visits** : The ratio $\frac{nbVisits}{pathLength}$. This criterion avoids the generation getting stuck in a loop.

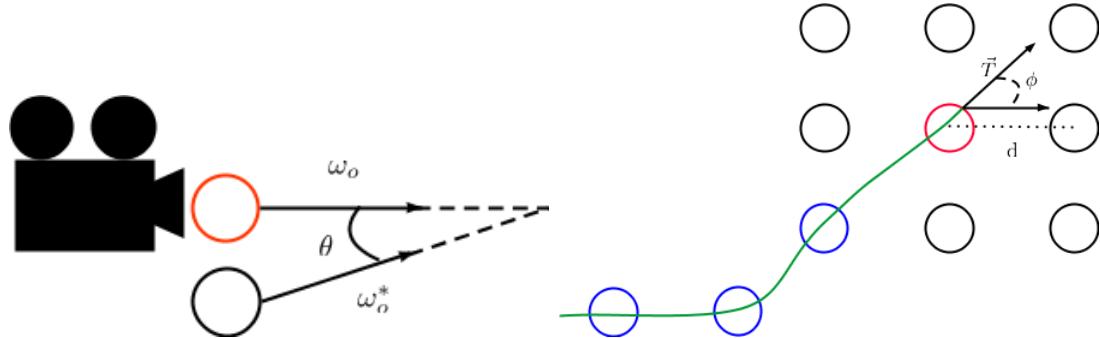


Figure 4.11 Edges constraints. In red the current viewpoint, in blue previous viewpoint.

Numerical instabilities occur if a continuity criterion is added on the up-vector between viewpoints. It is therefore processed in a different way as detailed in section 4.4.10.

4.4.9 Multi-criteria Discrete Optimization

This graph traversal is similar to a non back-tracing A^* algorithm. Defining the transition heuristic between nodes is the main issue with this approach. We use a multi-criteria discrete optimization technique to produce the final heuristic value ; a local optimum between the 8 neighbours is defined at each step, the heuristic choice if defined through the criteria using their weights.

Camera path planning being very different from viewpoint generation, it is natural that the viewpoint's quality is not a dominant criteria. Path planning is more about creating a coherent and continuous set of interesting viewpoints.

Each cliche has a specific set of criteria and weights that are used to find the next node at each transition (Figure 4.11).

In our case, we limit the subset of potential nodes at each step to the eight surrounding nodes (Figure 4.11). The other option would be to define the whole set of nodes as potential solutions of the optimization problem. The major issue of this latter choice is that the algorithm has tendency to quickly jump between viewpoints with a high quality heuristic ; our choice is built in a way to produce more numerically stable paths.

4.4.10 Building the paths

In addition to the graph traversal algorithm, the initial node of the camera path must be determined. Multiple choices can be made for the initial node and none of them is the correct answer :

- **The best node** : The quality and the occlusion criteria are used for defining the heuristic. It seems the most obvious choice. However, it implies no positive evolution during the camera path.
- **The worst node** : A good viewpoint is slightly delayed. It's a classic screen writing technique for making the viewer impatient for something. However, the generation may get stuck in a local maximum of a not interesting view-neighborhood.
- **A median node** : It's the compromise between the two previous choices.
- **A random node** : Can lead to a high variety of camera paths.

Choosing the best, the worst and the median node is made using multi-criteria discrete optimization excluding the edge-specific criteria defined in section 4.4.8. Random starting nodes reduces the cost of quality-evaluation for the first node (which can be expensive for high complexity meshes).

The generated path must be viewed as a time-dependent result. The number of output nodes must be determined to end the path-generation process. In our case, we define it

through a the maximal displacement speed of the camera. The maximum speed is related to the spot of interest's size (the bigger the SOI is, the bigger the maximum speed will be). The user simply specifies the duration of the animation he wants to produce.

Additional constraints may be added to the random node in a way to avoid non-relevant nodes. In our case, a starting node that presents total occlusion regarding the SOI is rejected and the sampling step should be repeated.

4.4.11 Interpolating the path

The subdivision level of the SOI is perhaps the bottleneck of the method ; it should remain as small as possible. For a low subdivision level, if simple linear interpolation is done between the nodes, the path discontinuities are too obvious.

It is then necessary to have a good degree interpolation. B-Spline interpolation is then applied to the camera position and the *look at* point is fixed at the SOI's center. After interpolation, the up vector is computed using the previous viewpoint for all nodes except the initial one. For the first viewpoint, its value is computed using the second node.

4.4.12 Photographic Compositions

Photographic composition in automatic camera placement has proven to have great impact on perception for video game camera path planning. In our paper, we will be focusing only on these two composition rules.

- **Rule of Thirds** : Placing the subject(s) on the power nodes (the intersection of the third lines) creates a more dynamic image composition ;
- **Cropping** : Cropping tight around the subject removes the "noise" created by other scene objects ;
- **Depth of field** : A depth of field centered on the subject helps the viewer focus on it.

The rule of thirds and the cropping rule are simple to apply in our case. In this situation, only one scene object is the center of interest. By construction, the object should initially be placed at the center of the screen. To be displaced to the right intersections (Figures 4.12 and 4.13), three constraints must be taken into account in the generation process :

- The distance between the SOI's center and the camera should be $D \cos \frac{\theta}{2}$ (Figure 4.12), D being the bounding sphere diameter and θ the FOV ; this should be taken into account when generating camera viewpoints.
- Each view matrix should be rotated around the Y axis by $\frac{\theta}{4}$ (resp. $-\frac{\theta}{4}$) in a way to be positioned on the left (resp. right) power points.

- The FOV used for rendering is $\theta^* = \arccos\left(\frac{4}{3} \cos \theta\right)$.

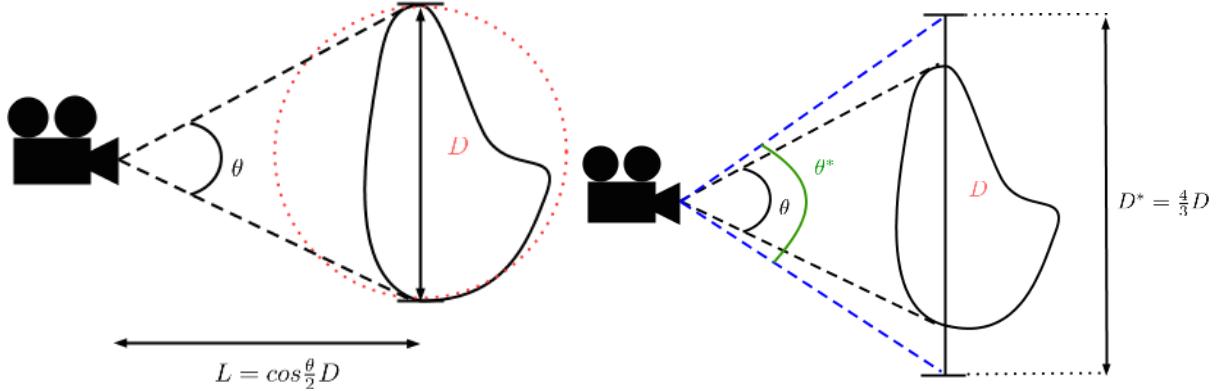


Figure 4.12 How to apply rule of cropping.

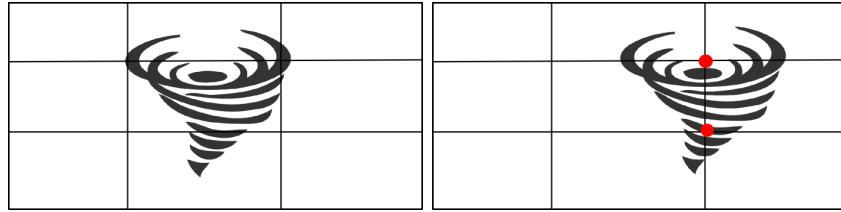


Figure 4.13 How to apply rule of thirds.

The depth of field data are pretty simple to compute ; For each camera position $D_N = L^* - D$ and $D_F = L^* + D$ matching respectively the closest and farthest points of the SOI.

4.5 Extensions

The work presented until now presents a method to generate one camera path for a fixed scene and one SOI. However, the nature of the algorithm makes extension of the properties defined in previous section possible.

4.5.1 Multiple choices

The camera paths we produce are heuristically generated. It is then interesting to build a set of camera paths instead of one single scenario. Two criteria are heuristically defined in our method : the criteria's weights and the initial position. The main idea is to generate a set of results by varying the two criteria to fit as closely to what the user would like.

A low resolution stop motion preview is computed for each camera path to help the user choose which one fits his preferences.

The user can also be interested in multiple camera sub paths. The mean value of the viewpoint's heuristic is computed for each sub path to help the user choose. It is interesting to make him combine them using film editing techniques.

4.5.2 User intervention

The main purpose of the method is to produce camera paths involving few user interactions. However, our method designed to allow the user to intervene into the generation process to redirect the generation process in a way he judges best. The interactions that we experienced relevant in the camera path generation process are :

- Defining the SOI's bounding sphere subdivision level to define the number of potential viewpoint ;
- Specifying the plan's lengths for each generation ;
- Modifying the global criteria weights ;
- Setting specific criteria/weights combinations for defined iterations ;
- Moving the selected viewpoints ;
- Adjusting the depth of field ;
- Defining the path interpolation parameters.

4.5.3 Time varying data

The time-varying mesh and node values are a direct extension of our method. However, some assumptions and choices are made :

- Camera movements will be slower than in the case of a static mesh. Quick camera movements can be confusing if the scene is varying ;
- The multi-criteria decisions is only done once per simulation step ;
- Scene duration is decomposed into multiple time intervals, each time lapse is processed independently ;
- Initial nodes will be randomly chosen.

4.5.4 Match cut rules

After producing multiple sets of camera sub paths, the user should be to able to combine them using film editing techniques for producing a data presentation movie (Figure 4.15). Within this approach, two match cut rules are integrated into the generation process for helping the user building a better movie.

The 180° rule : This rule states that to avoid the viewer's confusion, at each transition, the camera should not cross the action line. However, the action line's definition depends on

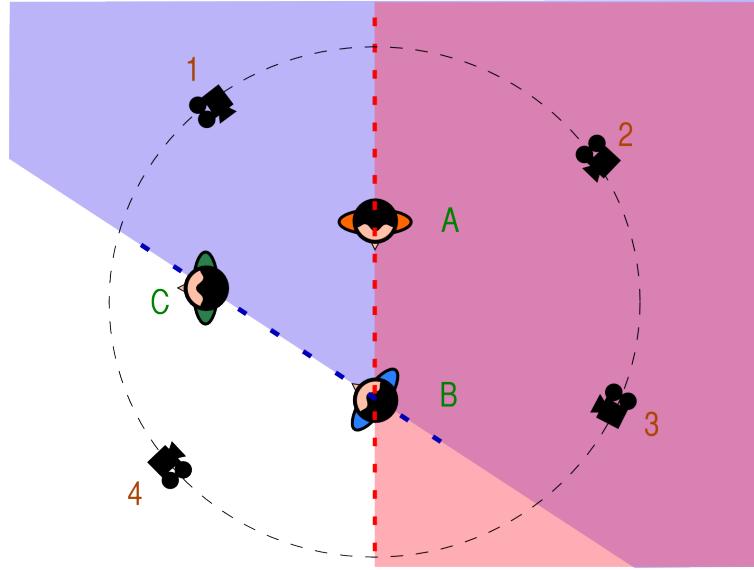


Figure 4.14 A simple scene illustrating match rule potential issues : three characters A , B and C facing different directions and 4 camera position observing the scene.

the scene composition and on the director's intentions regarding the user perception. The scene in Figure 4.14 presents three character A , B and C ; A is facing and B is looking at a C . Two main action lines may be defined in this case :

- **The red line :** The line between characters A and B is the action line. Not crossing this line states that the main action of the scene is the dialogue between both characters.
- **The blue line :** Defining the line between characters B and C as the action line states that B 's attention is the main action of the scene. Crossing the red line amplifies the perception of the user that the dialogue between the characters is auxiliary.

Automatically defining a relevant action line in complex static numerical scenes is a difficult task. In our methodology, the rule is only applied when the SOI is moving ; the action line is then defined as the local path displacement vector of the SOI. At the transition, we determine either if the SOI is moving to the *left* or the *right* in the screen space ; this can simply be done by testing the x component of the displacement vector (V^*) in the camera view space (Equation 4.4). The relative displacement vector determines in which side of the action line the camera is placed, preserving the same position between transitions fulfills the rule.

$$V^* = MV \quad (4.4)$$

M being the view matrix of the viewpoint, V the local displacement vector and V^* the local displacement vector in the view space.

The 30° rule : It states that at each transition, the viewpoint should move at least by 30 degrees with respect to the subject to create a dynamic succession of viewpoints. Random generated initial points do not guarantee that all transitions fulfill this rule. However, we can simply determine if transitions respect this rule by evaluating the displacement angle ϕ (Equation 4.5)

$$\phi = \arccos(\vec{z}_1 \cdot \vec{z}_2) \quad (4.5)$$

\vec{z}_1 and \vec{z}_2 being respectively the forward vector of the final viewpoint of the previous sub path and the forward vector of the first viewpoint of the next sub path.

In addition to the sub path heuristics, two transition indicators are then available to help the user in the montage process. The default camera path being the combination of the sub paths with the highest heuristic and fulfilled match cut rules (if possible).

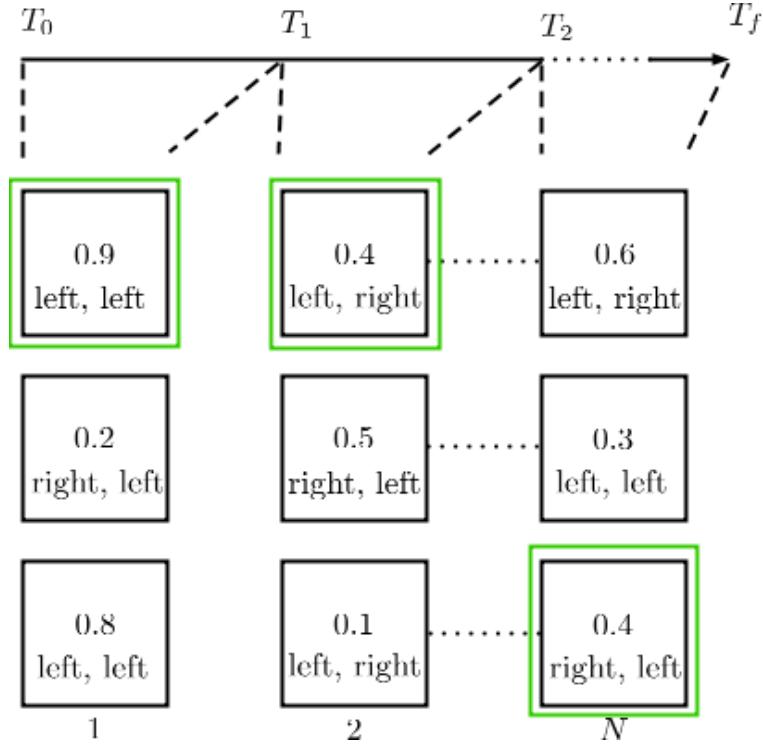


Figure 4.15 Film editing the different paths generated for each time lapse (default path in green). Each camera sub path is paired to a heuristic.

4.6 Results

An implementation of our method has been developed as a proof of concept. Numerical results have been converted to FABLE format (see Annex A for details) and output camera paths are exported the COLLADA format to be viewed in regular 3D animation and movie editing tool or a scientific visualization tool. Here are some results regarding the four simulation scenes that have been used for the paper results, with their respective processing times (P.T.) :

- **Aneurysm** : A static numerical result of blood propagation in veins presenting an aneurysm (pre-rupture). Each mesh node contains pressure and velocity information. For our tests, we focused on three visualization needs : the aneurysm surface aspect (P.T. : 10.5142s, Figure 4.16), the velocity vector field in the region (P.T. : 2.2459s, Figure 4.17) and the separation lines at its limits (P.T. : 1.7542s).
- **Train** : The simulation of the air flow around a rolling train. Multiple time varying SOIs have been placed to capture different visualization needs : A SOI moving on the top of the train to catch attachment and detachment of the air flow (P.T. : 400.2459s), a SOI at the back of the train to catch streamlines singularities (P.T. : 300.2736s, Figure 4.18) ;
- **Brain** : The result of blood pressure and velocity simulation in brain vessels. Our approach was to capture a particle propagation in the vessels while observing the surrounding streamlines (P.T. : 321.5921s, Figure 4.19) ;
- **Smoke** : Smoke propagation within a building. Some features (vortexes, separation lines) are interesting to observe (Figure 4.1) and the SOIs in the scene are positioned to observe them (P.T. : 1.5713s, Figure 4.20).

We ran the program on an *i7* processor and an average graphics card (*Ge Force GT 650M*) ; however, the processing time is scene depending. The most impacting parameters being : the mesh complexity, the extraction algorithm efficiency, the number of primitives inside the SOI and the parallelization of the quality criterion evaluation.

4.7 Discussion

For the previous set of numerical results, each visualization expectation we could think of was convertible to a visualization need. Minor modifications during the generation process (on the criteria weights ; mainly occlusion and quality) have been made in a way to adjust the generation process to specific needs regarding the paths.

For all the presented results, the automatic application of the rule of thirds has produced relevant camera path scene compositions. The viewer's attention is directed to the SOI (being

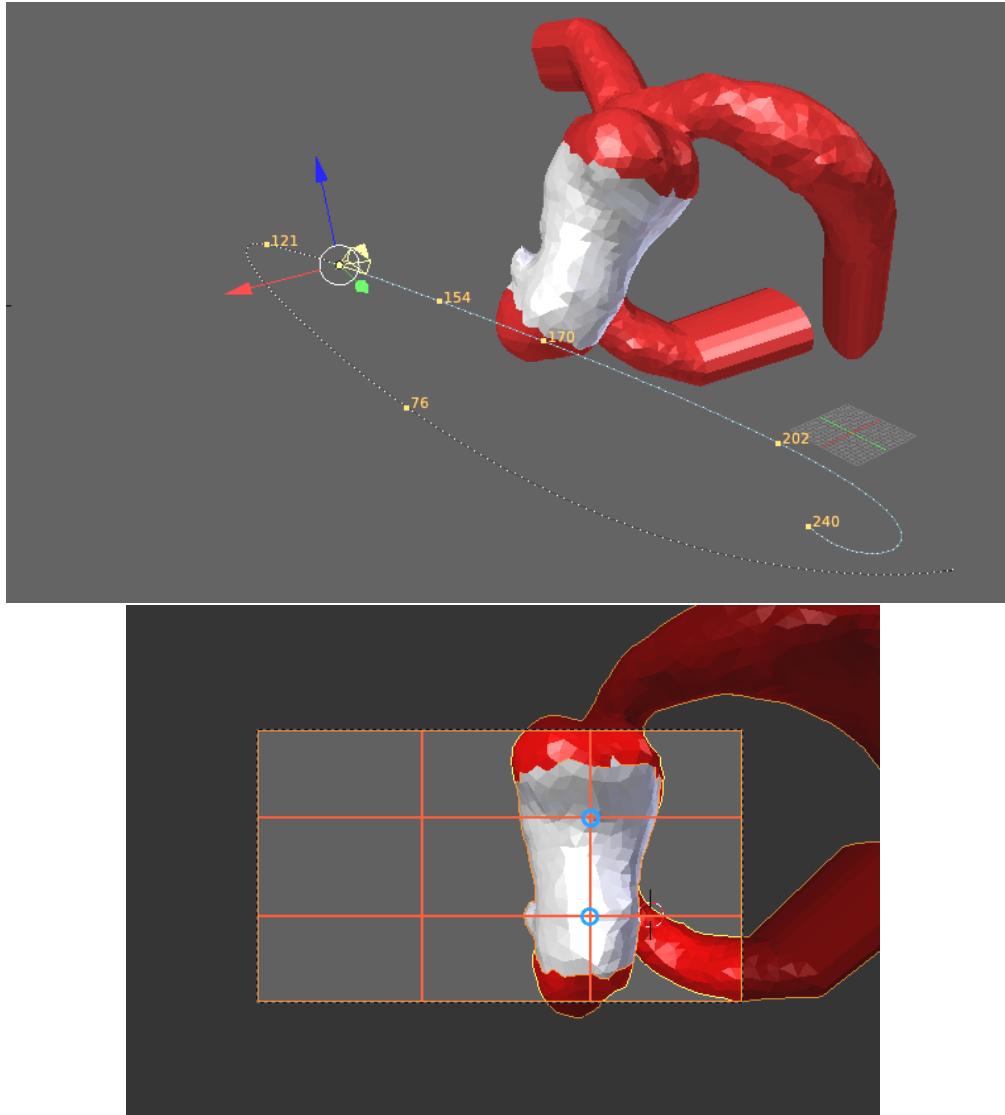


Figure 4.16 Up : Generated path for observing the aneurysm edges. Down : Sample viewpoint with rule of thirds metrics.

the image subject in composition rules) and then to its features (Figures 4.18, and 4.17, 4.20 and 4.16).

The produced camera paths for observing the aneurysm's edges (Figure 4.16) illustrates well the surface cliche visualization. Our method was able to generate a non trivial camera path that presents the surface of the aneurysm through a set of stable viewpoints with few occlusion.

Visualizing the vector field cliche is a main contributions of this paper ; we then used the whole set of scenes to test our method. In each case, the generated camera paths are non trivial and respect the quality constraint on the vector field while avoiding major occlusions.

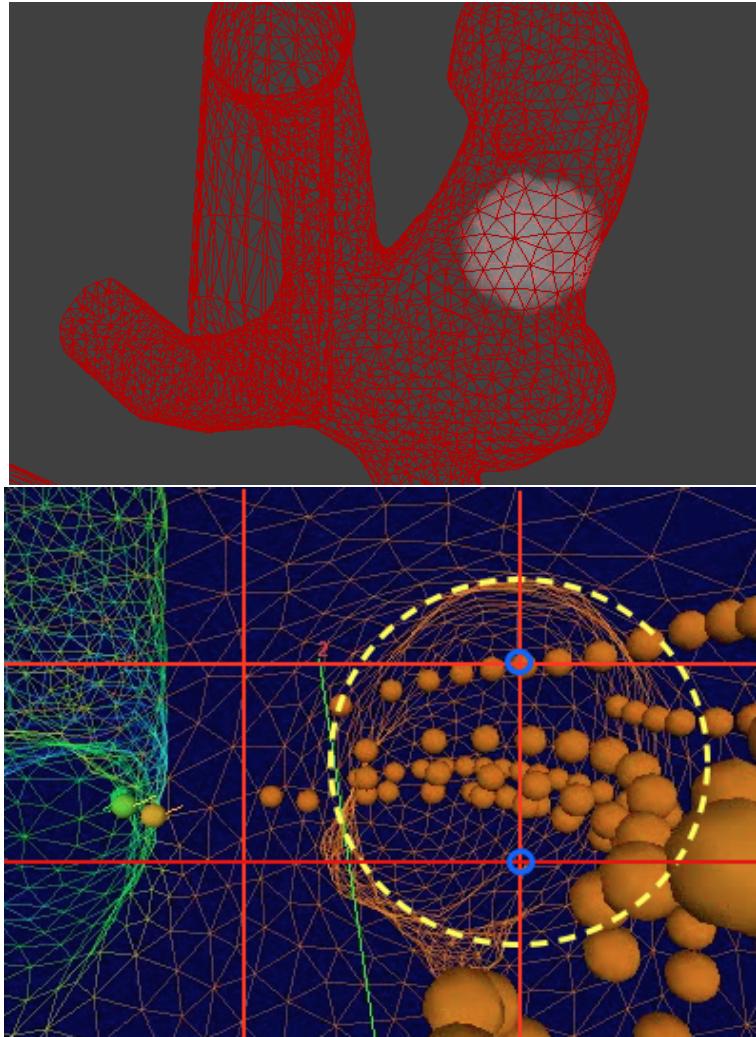


Figure 4.17 Up : specified SOI for streamline observation. Down : rendered image from the generated path (with particle propagation and rule of thirds metrics).

In addition to the previously expressed constraints, our method intrinsically produces camera paths that take into account other photographic composition rules :

- The leading line rule is respected when lines in the image lead the viewer's sight. In our case, this fulfilled thanks to the vector field quality criterion.
- The diagonal rule sets that if the major leading lines are diagonally arranged, the image will be more dynamic (Figure 4.20). It is respected in the generated paths as long the projection in the camera space of the main direction of the vector field is not colinear with the tangent vector of the path. However, other constraints (i.e. occlusion) may interfere in its application (Figure 4.17) ;

The camera path observing the blood particle propagation through the brain's veins shows

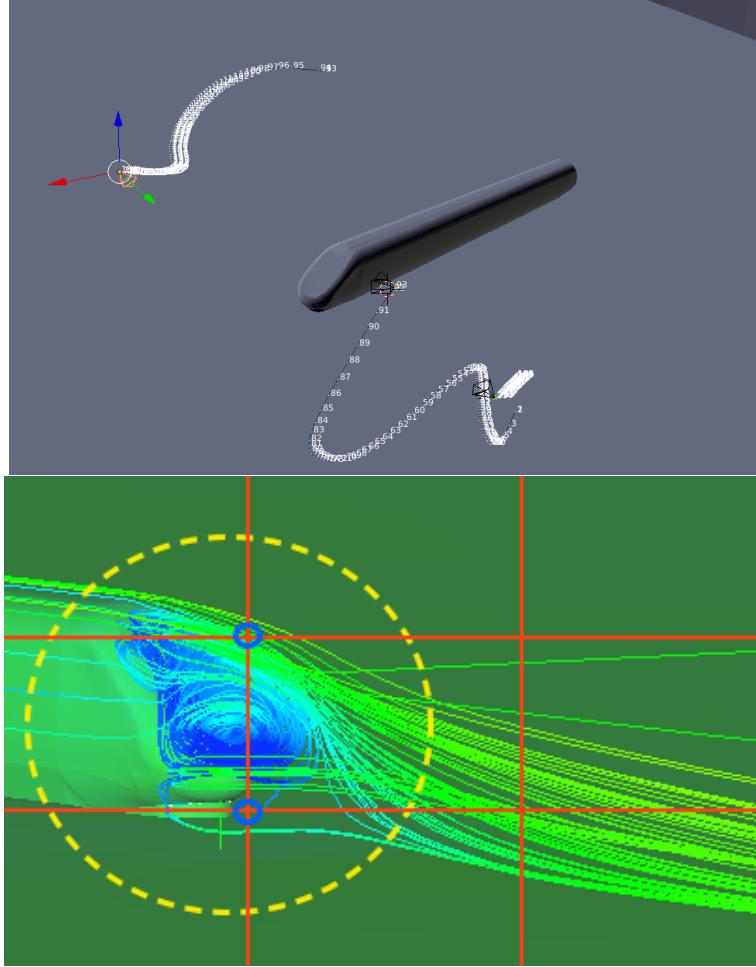
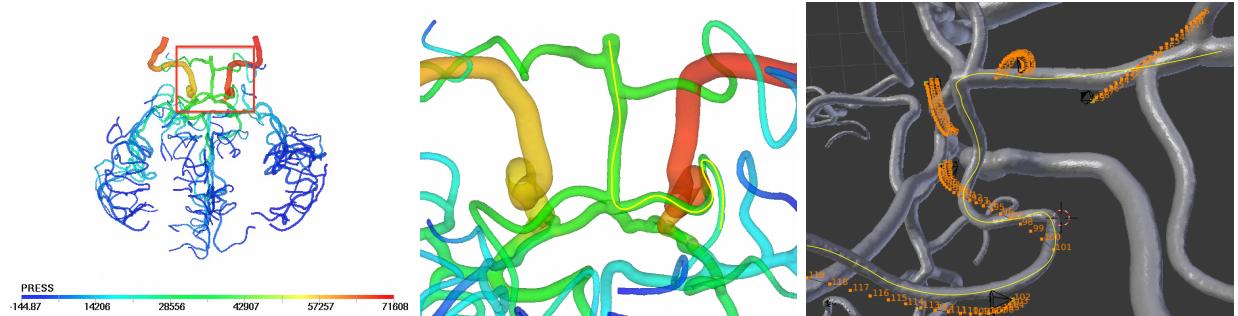


Figure 4.18 Up : a sample generated path for observing a SOI located at the back of the train. Down : rendered image from the generated path with rule of thirds metrics.

that the method is capable of producing pertinent camera paths observing a moving SOI. The variation of the streamlines during the displacement induces that the camera will have tendency to keep the local trajectory of the particle the major point of interest of the path.

Multiple camera sub paths suggestions with stop motion previews have shown to be an intuitive way to build a relevant global camera path. Dividing the generation process into multiple pipelines has also shown to be a good approach in the train and the brain scenes. The initial node being stochastically generated, the different subparts that are created are meant to capture in complementary information regarding the SOIs.

Finally, our method remains simple to implement and can be easily integrated into a regular scientific visualization tool. The camera paths generation bottlenecks (quality computation and SOI extraction) remain highly parallelization which make the method scalable.



- Unless the user specifies a specific SOI. The depth of field is set to infinity ;
- The rule of thirds, the 180° and 30° rules become more complex to apply.

4.9 Conclusion

In this paper, we presented a complete methodology to help scientists automatically build relevant camera paths for observing and presenting their 3D volumic data. Our abstraction regarding the features specificities, allows us to build a more generic generation pipeline, which still preserves each feature's interest within its local environment.

The different analogies made with cinematography and video games path generation techniques allows us generate paths that include composition, storytelling and screen writing notations. The scene modelization through *cliches* and *spots of interest* allow the user to precisely catch his needs regarding the location and the nature of the features in his numerical results. Generating camera paths that take it into account the latter constraints becomes a simpler operation.

Even if the method is automation oriented, the scientists can intervene in the generation process at each step (weight setting, initial viewpoint, building the paths, interpolation and film editing) in a way to make the final cut fit more his expectations. The scientist's interactions remain intuitive and gives him accurate control on the generated scenarios.

The method implies that features have been detected and spot of interest have been placed in the scenes. It is then a direct extension of feature detection and tracking techniques.

CHAPITRE 5

DISCUSSION GÉNÉRALE ET COMPLÉMENTAIRE

Dans cette section, nous discutons des points relatifs à la méthodologie qui ont été survolés ou n'ont pas été traités dans l'article présenté au chapitre précédent. Voici les points qui seront abordés :

- La cohérence avec l'approche créative de la prise de vue ;
- L'approche ontologique d'un scientifique vis-à-vis de ses données ainsi que les difficultés d'utilisations de la méthode qui peuvent en découler.
- Les besoins de visualisation plus complexes se basant toujours sur les clichés définis préalablement.
- La gestion des poids et critères du système d'optimisation.
- Détails du temps d'exécution et de la complexité des scènes utilisées.
- Les lignes d'action dans les scènes statiques.

5.1 La créativité

Contrairement aux méthodes préalablement existantes en visualisation scientifique, notre approche est d'automatiser le plus possible la génération de chemins et de déplacer l'intervention de l'utilisateur dans la partie synthèse d'image et montage.

Cependant, la prise de vue est avant tout une approche artistique et créative ; l'incertitude et l'apport personnel sont donc une part importante de ce qui fait la qualité d'une création. Toutes les règles de la composition ne sont pas nécessairement respectées à chaque plan ; certaines n'ont pas lieu d'être. Leur combinaison doit être pensée pour servir l'histoire visuelle qu'essaye de construire et de raconter le réalisateur.

Bien que les règles de la composition et de la narration permettent d'encadrer la création, la transgression de ces règles de manière mesurée et volontaire constitue une part importante du processus de création artistique. Chaque réalisateur ayant son approche globale par rapport à la création et spécifique à chaque plan lors du processus créatif. Un exemple illustrant bien cet énoncé est la construction symétrique dans les plans du film «The Grand Budapest Hotel» du réalisateur *Wes Anderson*. Dans ce film, la majorité des plans situés dans la partie retour en arrière du long métrage sont construits symétriquement avec une grande finesse (Figure 5.1).

Cette règle, bien que très présente dans ce film, est réputée comme étant une prison



Figure 5.1 Plans tirés du film «The Grand Budapest Hotel» du réalisateur *Wes Anderson*, illustrant la règle de la symétrie.

à la créativité; elle oblige le réalisateur à toujours construire sa composition de manière symétrique et est donc souvent non respectée dans les films. Cependant, elle sert parfaitement son message dans ce film : créer un sentiment de sécurité et de confort au spectateur.

C'est pourquoi il est très complexe de concevoir un algorithme qui produise une histoire qui choisit judicieusement quelles règles respecter et lesquelles transgresser. Notre approche est de créer cette part d'incertitude en variant les différentes heuristiques que la méthode propose et en laissant au scientifique le choix des plans qui constitueront son message.

Notre méthode a ses limitations au niveau du processus créatif : elle ne se base que sur un nombre limité de règles de compositions et elle ne sera donc pas capable de produire toutes les histoires possibles pour une même scène. Cependant, les outils inclus offrent suffisamment de marge pour produire des histoires variées et captivantes.

Deux règles de compositions (la perspective et l'éclairage) supplémentaires peuvent être intégrées simplement dans notre méthodologie pour varier la nature des histoires produites. Leur pertinence a déjà été explorée dans le milieu cinématographique et photographique. Cependant, afin de bâtir une méthodologie efficace nous n'avons pas pu les intégrer dans les résultats produits.

Le choix de l'objectif : dans le cas d'une scène avec un seul sujet principal, le choix de l'objectif (qui s'exprime au travers du champ de vision) est très important, car il définit le sentiment produit par le sujet. Un objectif à courte focale permettra de créer un sentiment d'inquiétude par rapport au sujet (ce qui peut-être judicieux dans le cas d'un comportement non voulu), des objectifs à focale plus longue permettront de mieux isoler le sujet du reste de la scène. Dans le cas de scènes virtuelles, le choix de l'objectif influe principalement sur la distance par rapport au sujet (le champ de vision étant adaptable au moment de la synthèse d'image). La profondeur de champ quant à elle a été préalablement prise en compte dans notre méthode.

Éclairage à trois points : notre approche est assez similaire à la composition d'une scène pour le portrait. La règle des trois sources de lumière permet de placer correctement les sources lumineuses afin de servir un sujet. Le positionnement de ces lumières, comme le reste des règles de placement, peut être transgressé.

- **Lumière clé (key light)** : elle doit être positionnée de manière décalée par rapport au sujet (idéalement, à 45°). Elle doit être puissante.
- **Lumière de remplissage (fill light)** : lumière plus faible que la lumière clé et qui permet de réduire les ombres créées par la lumière de remplissage. Elle est également placée de manière décalée, mais plus bas que la lumière clé.
- **Contre-jour (back light)** : cette lumière éclaire les contours du sujet et l'isole du reste de la scène.

Ces trois lumières ne doivent pas créer d'ombres dures sur le sujet. Des lumières surfaciques ou les cartes environnementales sont des bons choix. Le placement de ces sources étant toujours fixe dans l'espace caméra et on ignore les possibles occlusions de la scène par simplicité ; un rendu par illumination différée étant donc possible.

5.2 Ontologie

L'ontologie d'un domaine scientifique constitue l'ensemble des termes et des structures de phrases qui sont utilisés par des scientifiques de ce même domaine pour communiquer. Il arrive très souvent que les attentes d'un scientifique s'expriment dans l'ontologie des résultats qu'il cherche à parcourir ou à présenter. Il était donc important de se baser sur une modélisation qui s'abstrait des spécificités de chaque ontologie tout en préservant les besoins de visualisation qui peuvent leur être spécifiques. C'est pourquoi l'ontologie de nos besoins de visualisation est celle des résultats numériques scientifiques.

Si on prend l'exemple de deux scènes issues d'ontologies différentes et de deux besoins de visualisation qui leur sont associés :

- **Mécanique des fluides** : Simulation de l'écoulement d'air autour d'un avion ; visualisation de la portance.
- **Biomédical** : Simulation de la pression sanguine dans les vaisseaux d'un cerveau humain ; visualisation d'un anévrisme cérébral.

Bien que les termes utilisés pour l'expression des besoins de visualisation désignent des comportements de données ontologiquement différentes, dans notre approche le problème peut-être résolu à l'aide d'un seul cliché ; l'étude des représentations possibles permet de déterminer quel serait le choix de cliché le plus judicieux. Dans le premier cas, la visualisation du vecteur de portance équivaut à la visualisation du champ de vecteurs vitesse de l'air dans

la région des ailes ; le deuxième cas quant à lui est équivalent à la visualisation des lignes de courants du sang dans les zones où la vitesse est la plus faible. Ces deux courtes analyses nous dirigent vers la visualisation de champ de vecteur.

Cependant, une particularité ontologique peut avoir plusieurs représentations de type cliché. Dans notre méthodologie, l'interprétation des attentes en besoin de visualisation est faite par le scientifique, car il est plus apte à cerner quelle représentation répond le mieux à ces attentes.

La méthodologie prévoit néanmoins une automatisation du processus d'interprétation en développant une table de conversion d'attentes en besoins de visualisation spécifique à chaque domaine d'application.

5.3 Besoins de visualisation complexes

L'implémentation de notre méthodologie se base sur un nombre limité de besoins de visualisation. Ces besoins de visualisation ont été relevés à partir de l'étude d'un ensemble de résultats scientifiques. Bien que nos choix impliquent qu'il y ait potentiellement des attentes qui ne soient pas convertibles en besoins de visualisation, notre méthode propose un pipeline qui s'applique à la grande majorité des cas.

Toutefois, un ensemble de besoins de visualisation complémentaires sont prévus dans notre méthodologie. Une région d'intérêt peut-être représentée en utilisant plusieurs clichés, par exemple : un vortex peut-être représenté comme l'ensemble des lignes de courant qui le composent ainsi que par la courbe qui constitue son axe (Figure 5.2). Dans ce cas-ci, il est intéressant d'évaluer la qualité d'un point de vue pour chacune des représentations et d'évaluer l'heuristique générale du point de vue en prenant en compte la satisfaction du critère de qualité de chacune des représentations. Pour chaque région d'intérêt, l'utilisateur peut donc spécifier plusieurs besoins de visualisation (chaque besoin de visualisation n'étant pas nécessairement unique) pour capturer au mieux la ou les particularités incluses dans la région d'intérêt. Il est également possible de gérer les différentes occlusions entre les différentes représentations dans le cas de multiples clichés de visualisation de surface.

Cette extension permet de capturer une région d'intérêt en prenant en compte plusieurs facettes de sa représentation en rajoutant des contraintes supplémentaires sur les points de vue pendant le processus d'optimisation.

5.4 Critères et poids

La gestion des critères, des poids et l'évaluation des notes pour chacun des points de vue n'ont pas été détaillés dans l'article. Voici un rappel de la liste des critères de sélection d'un

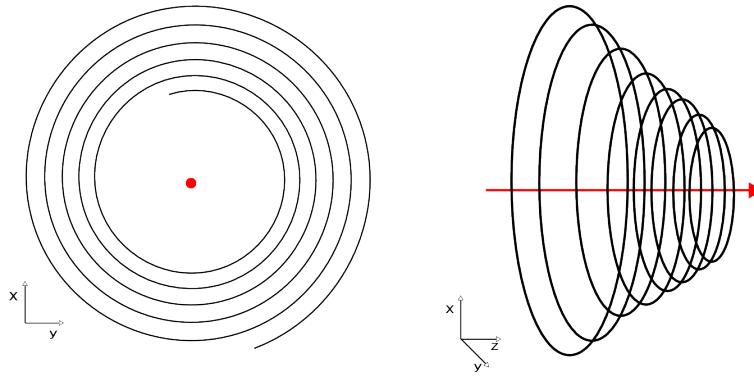


Figure 5.2 Deux vues des lignes de courants et de l'axe d'un vortex (face et perspective). La deuxième vue respecte mieux l'heuristique de champ de vecteurs pour chacune des représentations.

point de vue candidat lors du processus de génération de chemin :

- Qualité : Heuristique de qualité spécifique à un cliché ;
- Taux d'occlusion : Pourcentage d'occlusion de la surface englobante d'une région d'intérêt par les surfaces d'autres régions d'intérêt.
- Taux d'auto-occlusion : Pourcentage de la surface qui cache une autre partie de la surface plus en avant ;
- Distance : La distance entre le précédent point de vue et le point de vue candidat.
- Orientation : Variation du vecteur d'avant entre le précédent point de vue et le point de vue candidat.
- Courbure : Altération de la courbure du chemin par le point de vue candidat.
- Visites : Le ratio de visites du point de vue candidat sur le nombre total de points de vue.

Chacun des critères définis ci-dessus doit être normalisé entre $[0.0, 1.0]$ tous en proposant une paramétrisation régulière ; la valeur 0.0 correspondant à un critère non respecté. Le tableau 5.1 présente les diverses formules utilisées afin d'obtenir ces règles pour les critères non développés dans l'article qui ne les respectent pas intrinsèquement :

Critère	Formule
Distance	$Q = 1 - \frac{ p_i - p_{i+1} }{2r_i + p_i - p_{i-1} }$
Orientation	$Q = \frac{1 + \vec{z}_i \cdot \vec{z}_{i+1}}{2}$
Courbure	$Q = \frac{1 + (p_i - p_{i-1}) \cdot (p_{i+1} - p_i)}{2}$

Tableau 5.1 Conversion des métriques de critères en un espace paramétrique régulier

Où p_i est la position du point de vue courant, p_{i-1} la position du point de vue précédent, p_{i+1} la position du point de vue candidat, r_i le rayon de la sphère englobante de la région d'intérêt i , z_i le vecteur avant du point de vue courant, z_{i+1} le vecteur avant du point de vue candidat.

Deux choix s'offrent à nous pour la gestion des poids :

- **Décomposition discrète** : chaque intervalle associé à un critère est alors décomposé en un ensemble sous-intervalles ; le choix de la décomposition est lié à la granularité des préférences ou la nécessité d'éviter une plage de valeur particulière, par exemple : le sous-intervalle $[0.0, 0.2]$ est une plage de valeurs que l'on voudra éviter pour le facteur d'occlusion ou de courbure. Un poids réel est ensuite attribué à chaque sous-intervalle.
- **Profil continu** : l'intervalle $[0.0, 1.0]$ d'un critère est appliqué sur une plage de poids $[A, B]$ à l'aide d'une fonction continue. Cette méthode à l'avantage de proposer une répartition intuitive des poids sans paliers qui peuvent créer des instabilités numériques au niveau des chemins. Cette injection n'étant pas nécessairement bijective, les profils qui peuvent être attribués peuvent être très variés ; des exigences très particulières peuvent être rapidement satisfaites. L'approche étant de proposer une liste de profils par critère à l'utilisateur afin qu'il puisse composer sa génération de chemins.

Chacune des deux méthodes de gestion des critères et leurs poids ont leurs avantages et inconvénients, c'est pourquoi il est également possible de les combiner afin de saisir précisément l'intention de l'utilisateur.

La figure 5.3 présente un exemple de critères ainsi que les poids qui leur ont été affectés ; cette composition a été utilisée pour la génération du chemin observant la surface extérieure de l'anévrisme.

5.5 Temps d'exécution

Notre approche était de développer une méthodologie complète plutôt qu'une solution logicielle prête à l'emploi ; la solution logicielle a simplement servi à démontrer le fonctionnement de la méthodologie.

Bien que la méthode ait été conçue pour être hautement parallélisable étant donné la taille des résultats numériques des simulations scientifiques visées, nous n'avons pas pu expérimenter et intégrer l'effet de l'accélération par la parallélisation des routines de traitement dans nos résultats. Il est cependant important de donner plus de détails et d'explications concernant les temps d'exécution évoqués dans l'article afin de comprendre les critères qui ont un grand impact sur la durée du processus de génération.

Comme expliqué précédemment, les temps d'exécution de notre méthode dépendent de

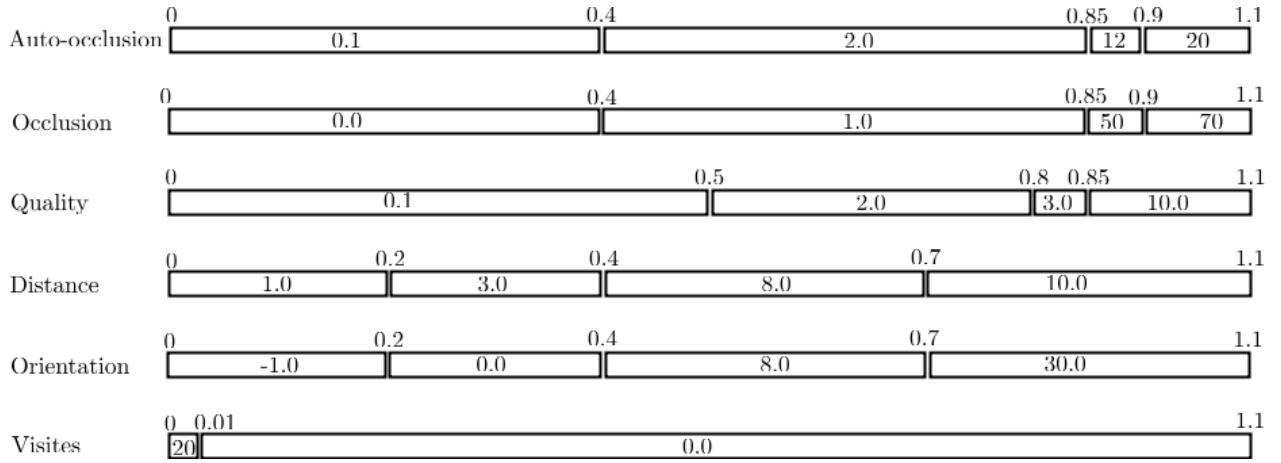


Figure 5.3 Poids des critères utilisés pour la génération du chemin de caméra de l'anévrisme dans l'intervalle [0, 1.1].

beaucoup de différents critères. Certaines ont été citées dans l'article, mais un ensemble plus grand d'opérations peut être un goulot d'étranglement (Tableaux 5.2 et 5.3).

5.5.1 L'extraction des régions d'intérêt

Notre méthodologie prévoit une opération très peu coûteuse pour évaluer la position relative d'un sommet par rapport à la région d'intérêt : l'utilisation d'une équation implicite. Cependant, étant donné la taille des données à traiter, plusieurs algorithmes d'extraction sont à envisager :

- **Traitements itératifs parallélisés** : Un parcours parallélisé (CPU ou GPU) de la liste des primitives pour déterminer quelles primitives sont à l'intérieur ou à l'extérieur de la zone d'intérêt.
- **Parcours en largeur du graphe** : Cet algorithme suppose que le sous-graphe que constituent les primitives dans la région d'intérêt est connexe. Dans un premier temps, l'algorithme trouvera à l'aide d'un algorithme A^* utilisant la distance par rapport à la surface implicite comme heuristique une primitive à l'intérieur de la région d'intérêt. Dans un deuxième temps, un parcours en largeur permettra de dresser la liste des primitives incluses à l'intérieur de la région d'intérêt.
- **Extraction préalablement faite** : Il arrive très souvent que les zones du maillage qui présentent des comportements notables qui intéressent les scientifiques soient déjà isolées du maillage lors de la simulation. L'ensemble des primitives à l'intérieur de la région d'intérêt est alors déjà séparé du reste du maillage.

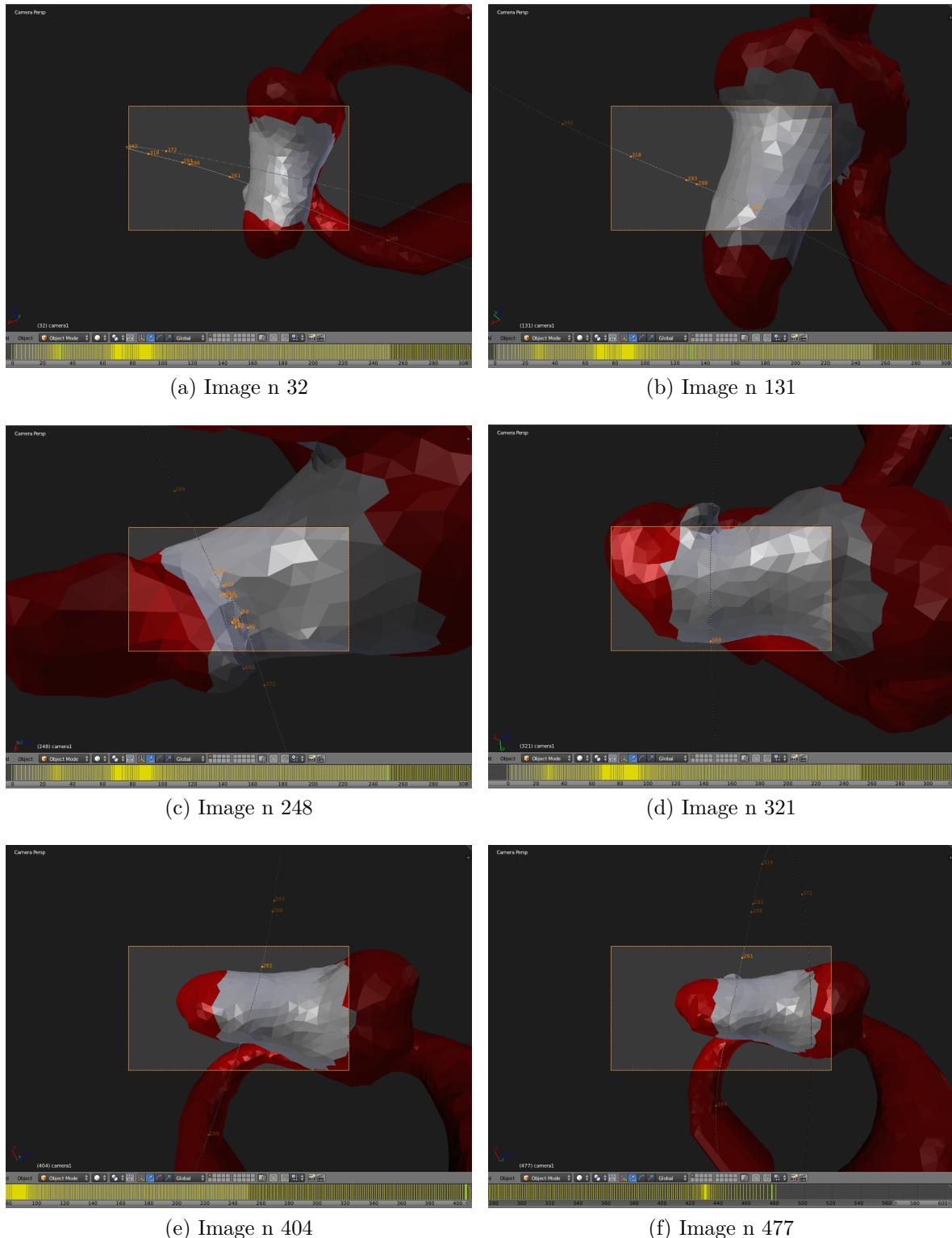


Figure 5.4 Images successives du chemin de caméra générée pour observer la surface de l'anévrisme

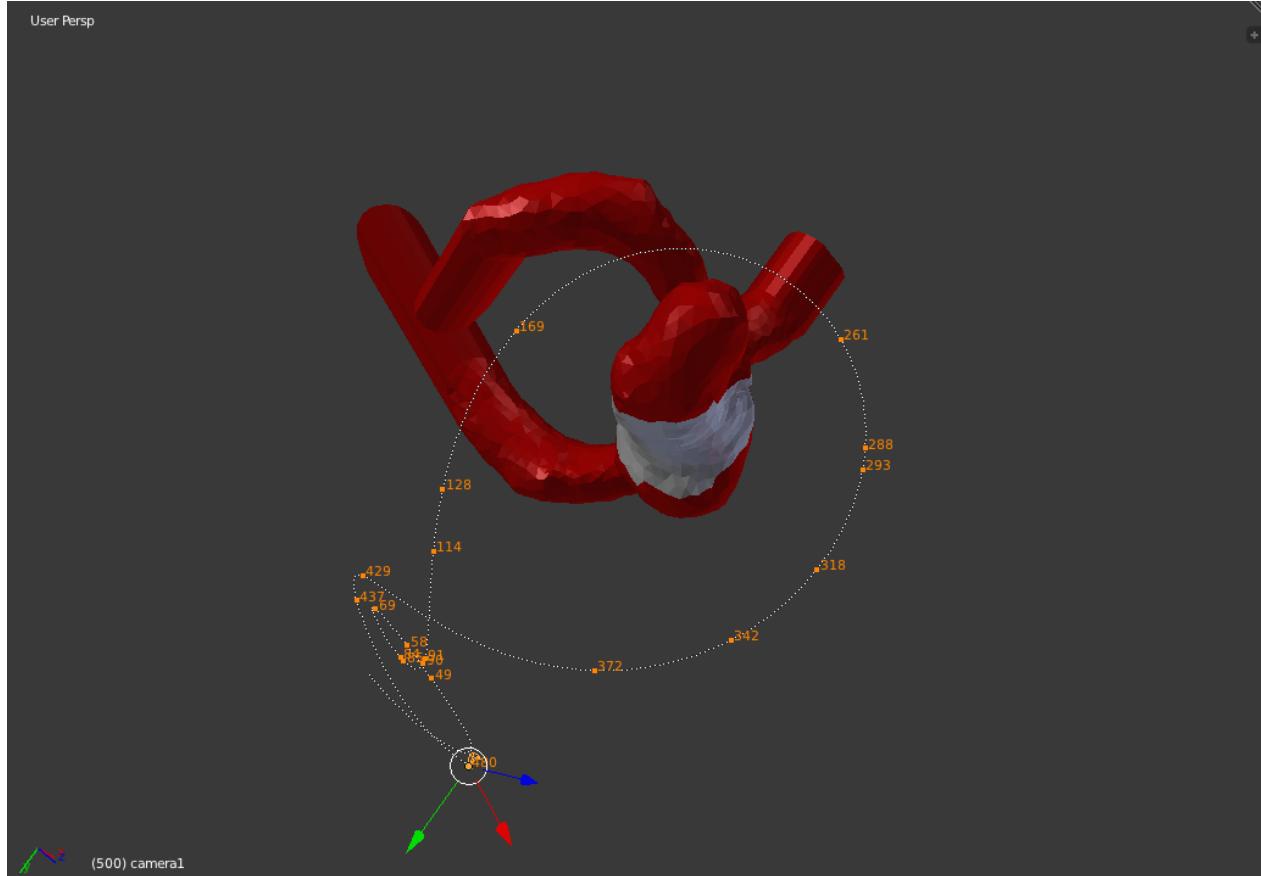


Figure 5.5 Vue globale du chemin générée pour observer la surface de l'anévrisme

5.5.2 Calcul de critère de qualité

La charge de calcul que peut représenter l'évaluation du critère de qualité dépend du nombre de primitives à l'intérieur de la région d'intérêt. Pour les clichés de visualisation de champ de vecteurs et de volume, les heuristiques proposées dans la méthodologie ont été prévues pour une implémentation GPU simple. La contribution de chaque primitive est évaluée indépendamment dans un premier temps, puis elles sont sommées par réduction.

Dans le cas du cliché de visualisation de surface, les contributions des différentes primitives sont liées ; l'évaluation des critères est donc plus coûteuse en nombre d'opérations. La première partie du processus de calcul du critère de qualité, l'évaluation des différentes cartes de profondeur de la région d'intérêt, est accélérée en utilisant le pipeline graphique OpenGL. Le coût de la deuxième partie du processus à un coût fixe vis-à-vis de la complexité du maillage ; les opérations de compression et de décompression sont effectuées à l'aide de la librairie *boost*.

5.5.3 Calcul d'occlusion

L'occlusion, comme l'évaluation de la stabilité, est un critère qui s'évalue en deux temps. La première partie consiste en la production de deux cartes de visibilités : l'une avec la région d'intérêt et l'autre avec les bloqueurs potentiels. La deuxième partie de l'algorithme, étant également à coût fixe vis-à-vis de la complexité du maillage, consiste à évaluer la proportion de pixels qui sont passés de visibles à occlus. Une image de faible résolution (256x256) est suffisante pour obtenir une information suffisante sur la scène. Cette opération pouvant être simplement parallélisée sur GPU.

5.5.4 Transformations

Pour les besoins de visualisation tels que la variation du champ de vecteurs ou le champ de scalaire, avant de pouvoir calculer le critère de qualité, des opérateurs (gradient, Jacobien) doivent être appliqués aux données des primitives dans la région d'intérêt afin d'extraire les données recherchées et pouvoir appliquer un cliché.

Pour ces critères également, les opérations de calcul sont organisées de manière vectorielle, la mise à l'échelle se fait assez aisément.

5.6 Lignes d'action

Comme il a été expliqué dans l'article, dans des scènes «humaines», les lignes d'action sont déterminées par la construction de la scène, mais également par l'histoire qu'essaye de raconter le réalisateur. Dans notre méthodologie, une approche a été proposée pour les scènes

qui présentent un déplacement de la région d'intérêt, mais aucune approche n'a été présentée pour les scènes statiques. Notre démarche est de déterminer une «direction principale» pour la région d'intérêt en fonction du cliché.

- **Champ de vecteurs** : l'heuristique que nous avons décidé de choisir pour le champ de vecteurs est la direction moyenne. Ce choix est le plus évident ; il associe la direction générale des données à une notion d'action produite. Dans le cas d'un vortex, qui se propage vers le bas. Cette règle indiquera à l'utilisateur qu'il est préférable de ne pas enchaîner deux plans qui le montre inversé.
- **Surface** : nous proposons d'utiliser l'analyse en composantes principales afin de déterminer un ensemble de directions qui peuvent être intéressantes comme lignes d'action pour l'utilisateur. Un ensemble de primitives sont aléatoirement échantillonnées, les coordonnées des points de chacune de ces primitives sont alors moyennées et leurs surfaces sont alors calculées. Pour un nombre d'échantillons n , les matrices d'échantillons M et de poids D s'expriment comme il suit :

$$M_{n,3} = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{pmatrix} D_{n,n} = \begin{pmatrix} \frac{s_1}{S} & 0 & \cdots & 0 \\ 0 & \frac{s_2}{S} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{s_n}{S} \end{pmatrix} \quad (5.1)$$

avec (x_i, y_i, z_i) les composantes moyennées de la primitive i , s_i la surface de la primitive i et $S = \sum_{i=1}^n s_i$.

Cette approche propose 3 directions principales, avec 3 valeurs propres associées. Par défaut, c'est la direction associée à la plus grande valeur propre qui est retenue pour calculer le critère de transition. L'utilisateur peut cependant en choisir une des deux autres pour l'aider dans le processus de montage.

- **Volume** : nous utilisons la même approche que pour la surface. La variante est que nous nous servons du volume des primitives pour évaluer les poids respectifs des échantillons.

La composante principale n'est pas nécessairement celle qu'intéresse le réalisateur. Dans la figure 5.6 les composantes principales de deux dessins sont présentées : celles d'un homme qui court et celle d'un poisson qui nage. Dans le cas du poisson, la direction principale serait un bon candidat comme ligne d'action : elle marque la synergie du poisson ainsi que son regard. Cependant, pour l'homme qui court, c'est la composante $n^{\circ} 3$ qui serait un bon candidat, car elle marque le regard du personnage.

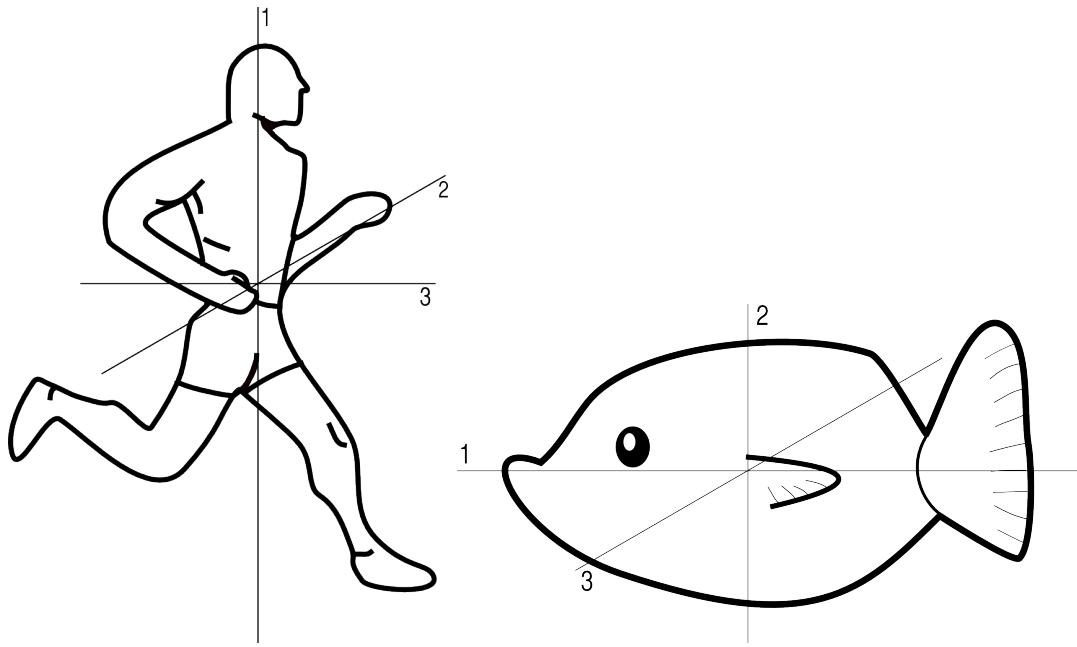


Figure 5.6 Composantes principales d'un poisson et d'un personnage qui court.

Ces métriques ne garantissent toutefois pas la production d'une histoire structurée, elles proposent simplement des indicateurs pour aider le scientifique à construire son court métrage en respectant certaines règles de la composition cinématographique.

5.7 Atteinte des objectifs

À ce stade de la discussion, il est important de s'assurer que notre méthode a rempli les objectifs et vérifié les hypothèses que nous avons définies au début du mémoire. Voici les réalisations apportées par notre travail :

- La méthodologie que nous avons produite et présentée dans ce mémoire propose une modélisation des particularités et attentes des scientifiques au travers de métamodèles (cliché, besoins de visualisation) (O_1 et H_1) ;
- La modélisation des particularités ontologiques en besoins de visualisation propose un ensemble d'outils qui permettent de répondre simplement à un grand nombre des attentes que les scientifiques peuvent avoir vis-à-vis de leurs données (O_3) ;
- Les critères de parcours de graphe (orientation, distance, etc.) ainsi que de gestion de visibilité (occlusion, auto-occlusion) orientent le processus de génération vers des chemins de caméra pertinents (O_4) ;
- Des règles de la composition (règle de tiers, profondeur de champ, dominance diagonale, etc.) ainsi que du montage (180° et 30°) ont été intégrées dans notre méthodologie

- et produisent des chemins de meilleure qualité et esthétique (H_4 et O_5) ;
- Les heuristiques proposées pour visualiser les champs de vecteurs et les volumes montrent des résultats qui mettent bien en évidence les particularités voulues (O_2) ;
 - L'ensemble des scènes retenues pour évaluer le fonctionnement de notre méthode met en évidence l'efficacité de notre méthode pour produire des chemins de caméra qui mettent en évidence les particularités visées (O_6) ;
 - Bien que la méthode se base sur une représentation pour évaluer la pertinence d'un point de vue, les chemins générés mettent bien en évidence les comportements visés en utilisant une autre représentation pour la visualisation (H_2 et H_3) ;

Scène	Taille (o)	Primitives	Images	N scène	Cliche	Besoin	Subdiv.	Noeud initial	Prim/SOI	Images de sortie
Anévrisme	3e6	42208	1	1	surface	bordures	10	meilleur	8980	4800
				2	vecteur	lignes de courant	10	meilleur	570	4800
				3	vecteur	jacobien vecteur vitesse	10	meilleur	821	4800
Fumée	68e6	261642	1	4	vecteur	lignes de courant	10	meilleur	3395	12000
				5	vecteur	lignes de courant	10	meilleur	4467	12000
Cerveau	18e9	76150560	20	6	vecteur	ligne de courant	10	aléatoire	~ 15332	120
Train	17e9	400637400	200	7	vecteur	ligne de courant	20	aléatoire	~ 217371	13632

Tableau 5.2 Données de complexité pour chacune des scènes et besoins présentées dans l'article

Scène	N scène	Extraction (s)	Critères (s)	Génération (s)	Total (s)
Anévrisme	1	0.110634	10.671403	0.007397	10.786
	2	0.095062	1.958957	0.090783	2.15049
	3	0.096641	2.15692	0.08963	2.34646
Fumée	4	0.618298	2.93865	1.30882	4.87766
	5	0.682814	3.1143	1.29533	5.09654
Cerveau	6	540.62	87.574	0.1091	628.3031
Train	7	670.984	250.762	1.342	923.088

Tableau 5.3 Temps de traitement pour chacune des scènes et besoins présentés dans l'article

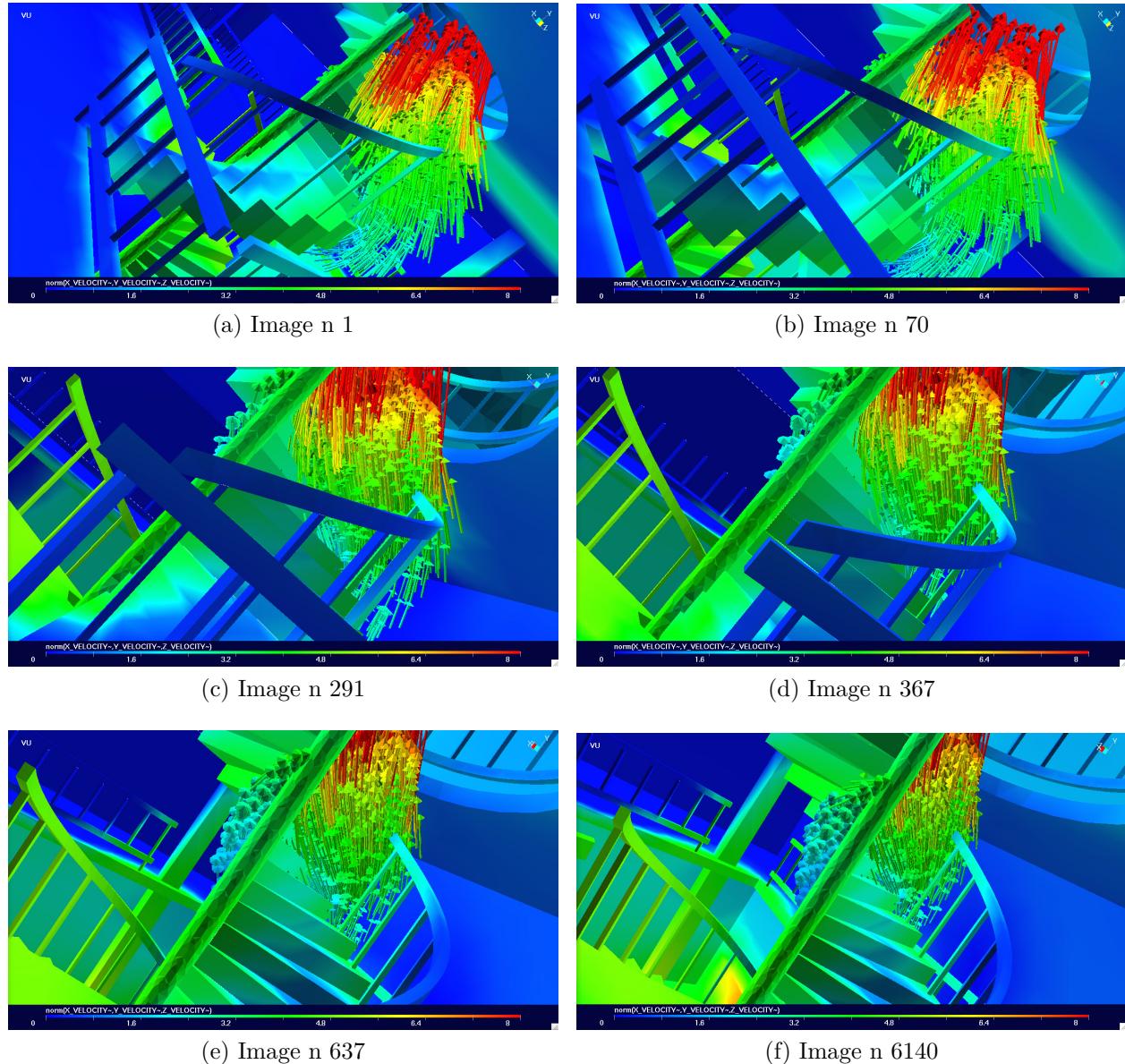


Figure 5.7 Images successives du chemin de caméra généré pour observer le champ de vecteurs dans la scène «Fumée»

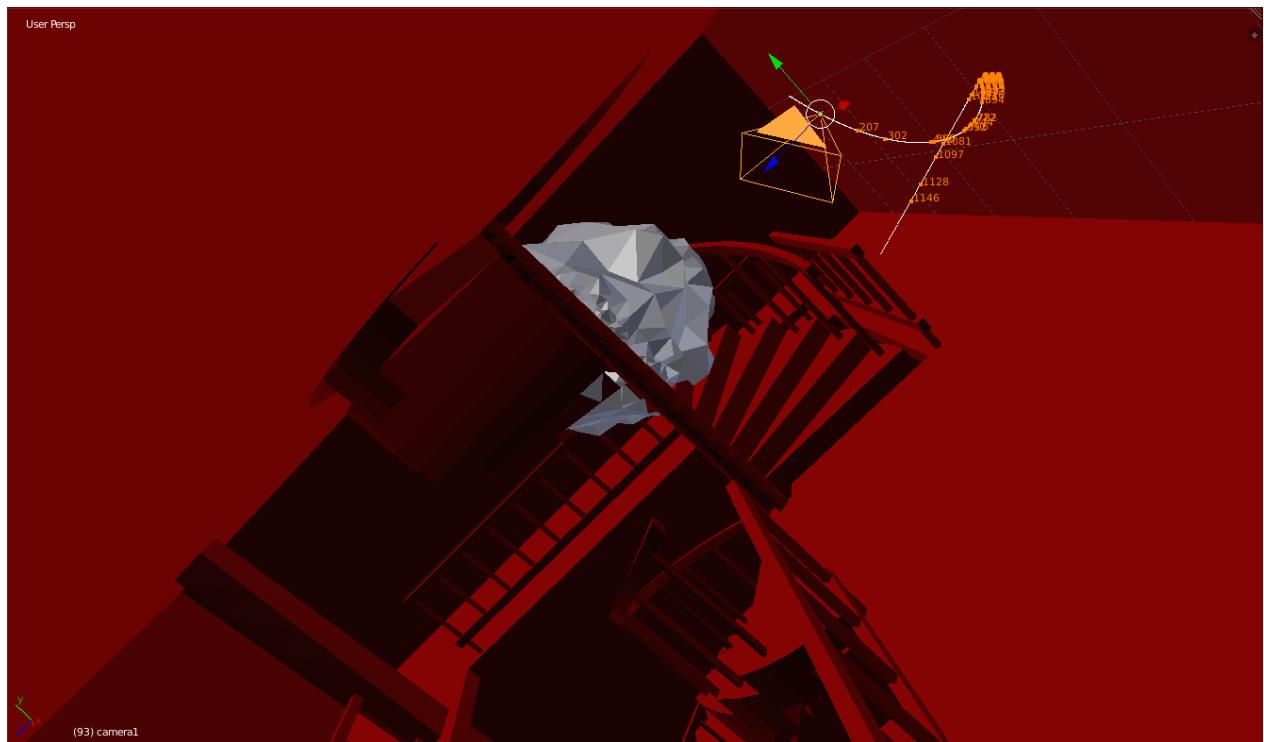


Figure 5.8 Vue globale du chemin de caméra généré pour observer le champ de vecteurs dans la scène «Fumée»

CHAPITRE 6

CONCLUSION

6.1 Synthèse des travaux

La méthode décrite dans cet article permet de générer automatiquement un ensemble de chemins de caméra en mettant en avant des comportements de données et en préservant le message qu'essaye de transmettre le réalisateur. Le processus de montage est quant à lui guidé afin de permettre à l'utilisateur de construire facilement un court métrage de qualité.

Des règles de compositions issues du cinéma et de la photographie permettent de construire des images plus esthétiques afin de mieux servir l'histoire que cherche à mettre l'utilisateur en avant. Un ensemble de paramètres de contrôle sont également disponibles afin d'ajuster le processus de générations aux besoins très spécifiques que peut avoir l'utilisateur pour une scène.

La méthode reste simple d'implémentation et d'intégration dans les outils de visualisation scientifiques utilisés par la communauté de scientifiques (Paraview, VU, etc.) et est conçue pour une mise à l'échelle simple au travers de la parallélisation CPU, GPU ou en grappe de calcul.

6.2 Limitations et améliorations futures

La majeure extension prévue par la méthodologie est la composition de scène pour de multiples régions d'intérêt. Pour l'instant, notre méthodologie est capable de construire des chemins de caméra pour présenter une seule région d'intérêt dans un environnement plus complexe. L'utilisateur ne peut pas encore mettre en avant un ensemble de comportements ainsi que les différentes interactions entre eux.

Même si les régions d'intérêt sont traitées comme des personnages, l'approche devient beaucoup plus complexe, car elle doit prendre en compte l'importance relative de chaque région d'intérêt tout en préservant les comportements à l'intérieur de ces zones du maillage. L'application des règles de composition est altérée ; on passe du portrait à un modèle de scène plus complexe.

RÉFÉRENCES

- ABDULLAH, R., CHRISTIE, M., SCHOFIELD, G., LINO, C. et OLIVIER, P. (2011). Advanced composition in virtual camera control. L. Dickmann, G. Volkmann, R. Malaka, S. Boll, A. Krüger et P. Olivier, éditeurs, *Smart Graphics*, Springer Berlin / Heidelberg, vol. 6815 de *Lecture Notes in Computer Science*. 13–24.
- AKIBA, H., WANG, C. et MA, K.-L. (2010). Aniviz : A template-based animation tool for volume visualization. *IEEE Computer Graphics and Applications*, 30, 61 –71.
- ARIJON, D. (1976). *Grammar of the Film Language*. Hastings House Publishers.
- BANKS, D. C. et SINGER, B. A. (1995). A predictor-corrector technique for visualizing unsteady flow. *IEEE Transactions on Visualization and Computer Graphics*, 1, 151–163.
- BARBIER, S. et BONNEAU, G.-P. (2008). Visualisation focus+contexte pour l'exploration interactive de maillages tétraèdriques. *Revue Electronique Francophone d'Informatique Graphique*, 2, 61–71.
- BARES, W. H., GRÉGOIRE, J. P. et LESTER, J. C. (1998). Realtime constraint-based cinematography for complex interactive 3d worlds. *In Proceedings of fifteenth national conference on artificial intelligence and tenth innovative applications of artificial intelligence conference (AAAI/IAAI)*. Madison, WI, USA : AAAI Press, 1101–1106.
- BENYOUB, A. (2015). *Automatisation de la création de scénarios pour les scènes de la visualisation scientifique*. Mémoire de maîtrise, École polytechnique de Montréal.
- BORDOLOI, U. et SHEN, H.-W. (2005). View selection for volume rendering. *Visualization, 2005. VIS 05. IEEE*. 487–494.
- BOURNE, O., SATTAR, A. et GOODWIN, S. (2008). A constraint-based autonomous 3d camera system. *Constraints*, 13, 180–205.
- BÄHNISCH, C., STELLDINGER, P. et KÖTHE, U. (2009). Fast and accurate 3d edge detection for surface reconstruction. J. Denzler, G. Notni et H. Süße, éditeurs, *Pattern Recognition*, Springer Berlin Heidelberg, vol. 5748 de *Lecture Notes in Computer Science*. 111–120.
- CHAN, M.-Y., MAK, W.-H. et QU, H. (2008). An efficient quality-based camera path planning method for volume exploration. G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, F. Porikli, J. Peters, J. Klosowski, L. Arns, Y. Chun, T.-M. Rhyne et L. Monroe, éditeurs, *Advances in Visual Computing*, Springer Berlin Heidelberg, vol. 5359 de *Lecture Notes in Computer Science*. 12–21.

- CHONG, M. S., PERRY, A. E. et CANTWELL, B. J. (1990). A general classification of three dimensional flow fields. *Physics of Fluids A : Fluid Dynamics (1989-1993)*, 2, 765–777.
- CHRISTIE, M., OLIVIER, P. et NORMAND, J.-M. (2008). Camera control in computer graphics. *Computer Graphics Forum*, 27, 2197–2218.
- DALLMANN, U. (1983). Topological structures of three-dimensional vortex flow separation (AIAA-83-1735). *AIAA 16th Fluid and Plasmadynamics Conference*. AIAA, American Institute of Aeronautics and Astronautics.
- DEGANI, D., SEGINER, A. et LEVY, Y. (1990). Graphical visualization of vortical flows by means of helicity. *AIAA Journal*, 28, 1347–1352.
- DRUCKER S. M., GALYEAN T. A., Z. D. (1992). Cinema : A system for procedural camera movements. *In Proceedings of the 1992 symposium on Interactive 3D graphics (SI3D 92)*. ACM Press, 67–70.
- FOX, P. et HENDLER, J. (2011). Changing the equation on scientific data visualization. *Science*, 331, 705–708.
- HALPER, N. et OLIVIER, P. (2000). A camera planning agent. in smart graphics. *AAAI Spring Symposium*. 92–100.
- HUNT, J., WRAY, A. et MOIN, P. (1988). Eddies, stream, and convergence zones in turbulent flows. *Summer Program 1988*. Center for Turbulence Research Report, vol. CTR-S88, 193–208.
- JEONG, J. et HUSSAIN, F. (1995). On the identification of a vortex. *Journal of Fluid Mechanics*, 285, 69–94.
- JIANG, M., MACHIRAJU, R. et THOMPSON, D. (2002). A novel approach to vortex core region detection. *Proceedings of the Symposium on Data Visualisation 2002*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, VISSYM '02, 217–ff.
- KASPER, W. A. (1975). Some ideas of vortex lift. *SAE Preprints. VORTEX LIFT*.
- KENWRIGHT, D. N. (1998). Automatic detection of open and closed separation and attachment lines. *In IEEE Computer Society Press, editor, IEEE Visualization Proceedings*. 151–158.
- KENWRIGHT, D. N., HENZE, C. et LEVIT, C. (1999). Feature extraction of separation and attachment lines. *IEEE Transactions on Visualization and Computer Graphics*, 5, 135–144.
- KOLÁŘ, V. (2007). Vortex identification : New requirements and limitations. *International Journal of Heat and Fluid Flow*, 28, 638 – 652. Including Special Issue of Conference on

Modelling Fluid Flow (CMFF'06), Budapest 13th event of the international conference series in fluid flow technologies : conference on modelling fluid flow.

KOOP, D. (2008). Viscomplete : Automating suggestions for visualization pipelines. *IEEE Transactions on Visualization and Computer Graphics*, 14, 1691–1698.

LEONE, S. (1966). The good, the bad and the ugly. *Produzioni Europee Associati*.

LINO, C., CHRISTIE, M., LAMARCHE, F., SCHOFIELD, G. et OLIVIER, P. (2010). A real-time cinematography system for interactive 3d environments. *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, Aire-la-Ville, Switzerland, SCA'10, 139–148.

MA, K.-L., LIAO, I., FRAZIER, J., HAUSER, H. et KOSTIS, H.-N. (2012a). Scientific storytelling using visualization. *IEEE Computer Graphics and Applications*, 32, 12–19.

MA, K.-L., LIAO, I., FRAZIER, J., HAUSER, H. et KOSTIS, H.-N. (2012b). Scientific storytelling using visualization. *Computer Graphics and Applications, IEEE*, 32, 12–19.

MA, K.-L., VAN ROSENDALE, J. et VERMEER, W. (1996). 3d shock wave visualization on unstructured grids. *Volume Visualization, 1996. Proceedings., 1996 Symposium on*. 87–94, 104.

MUELDER, C. et MA, K.-L. (2009). Interactive feature extraction and tracking by utilizing region coherency. *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific*. 17–24.

NICCOL, A. (2005). Lord of war. Lions Gate Films.

PANKANIN, G. L., KULINCZAK, A. et BERLINSKI, J. (2002). Karman vortex street parametrization with image processing application. *Proceedings of SPIE - The International Society for Optical Engineering*. Rzeszow, Poland, vol. 5124, 186 – 192. Flowmeter design and optimization ;Karman vortex street ;Vortex flowmeters ;.

PICKERING, J. H. (2002). *Intelligent Camera Planning for Computer Graphics, PhD thesis*. Thèse de doctorat, Department of Computer Science, University of York.

POST, F. H., VROLIJK, B., HAUSER, H., LARAMEE, R. S. et DOLEISCH, H. (2003). The state of the art in flow visualisation : Feature extraction and tracking. *Computer Graphics Forum*, 22, 775–792.

REINDERS, F., POST, F. H., SPOELDER, H. J. W. et TIME-DEPENDENT, K. V. (2001). Visualization of time-dependent data using feature tracking and event detection. *The Visual Computer*, 17, 55–71.

ROBINSON, S. (1990). A review of vortex structures and associated coherent motions in turbulent boundary layers. A. Gyr, éditeur, *Structure of Turbulence and Drag Reduction*.

Springer Berlin Heidelberg, International Union of Theoretical and Applied Mechanics, 23–50.

ROTH, M. et PEIKERT, R. (1996). Flow visualization for turbomachinery design. *Visualization '96. Proceedings*. 381–384.

TAKAHASHI, S., FUJISHIRO, I., TAKESHIMA, Y. et NISHITA, T. (2005). A feature-driven approach to locating optimal viewpoints for volume visualization. *Visualization, 2005. VIS 05. IEEE*. 495–502.

TRUBY, J. (Oct 2008). *The Anatomy of Story*. Faber and Faber; Reprint edition.

VAN GELDER, A. (2012). Vortex core detection : back to basics. *Proc. SPIE*, 8294, 829413–829413–8.

VÀQUEZ, P.-P., FEIXAST, M., SBERT, M. et HEIDRICH, W. (2002). Image-based modeling using viewpoint entropy. J. Vince et R. Earnshaw, éditeurs, *Advances in Modelling, Animation and Rendering*, Springer London. 267–279.

VÁZQUEZ, P.-P., FEIXAS, M., SBERT, M. et HEIDRICH, W. (2004). Automatic view selection using viewpoint entropy and its application to image-based modelling. *Computer Graphics Forum*, 22, 689–700.

VÁZQUEZ, P.-P. (2009). Automatic view selection through depth-based view stability analysis. *The Visual Computer*, 25, 441–449.

WANG, Y., LI, H., CAI, Q. et LI, W. (2009). Constructing streamlines in steady flow on unstructured grids. *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*. 20–24.

WISCHGOLL, T. et SCHEUERMANN, G. (2001). Detection and visualization of closed streamlines in planar flows. *Visualization and Computer Graphics, IEEE Transactions on*, 7, 165–172.

WOHLFART, M. (2008). Story telling aspects in medical applications. *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games*. ACM, New York, NY, USA, I3D '08, 35–43.

WOHLFART, M. et HAUSER, H. (2007). Story telling for presentation in volume visualization. K. Museth, T. Möller et A. Ynnerman, éditeurs, *EuroVis '07*. Eurographics Association, 91–98.

ZABUSKY, N. J., BORATAV, O. N., PELZ, R. B., GAO, M., SILVER, D. et COOPER, S. P. (1991). Emergence of coherent patterns of vortex stretching during reconnection : A scattering paradigm. *Phys. Rev. Lett.*, 67, 2469–2472.

ZHANG, L. et MENG, X. (2012). Enhanced robust vortex detection. *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. vol. 2, 222–225.

ANNEXE A

FABLE

The description of the fable format is :

```

FRAME 0 [frameID]
{
  FRAMETIME 0 [frameTime]
  ZONELIST_VERTEX
  {
    [ZONE {zoneName} {typeData} {fileName} {stride} {nbPoints}]
    ...
  }
  ZONELIST_INDEX
  {
    [ZONE {idZone} {typeData} {fileName} {stride} {nbPrimitive}]
    ...
  }
  ZONELIST_NEIGHBOUR
  {
    [ZONE {idZone} {typeData} {fileName} {stride} {nbPrimitive}]
    ...
  }
  VARIABLES
  {
    ZONE zone0
    {
      [SCALAR {dataLabel} {dataType} {fileName} {stride}]
      [VECTOR<N> {dataLabel} {dataType} {fileName} {stride}]
    }
  }
  SURACELIST_VERTEX
  {
    [SURFACE {surfaceName} {typeData} {fileName} {stride} {nbPoints}]
    ...
  }
  SURACELIST_INDEX
  {
    [SURFACE {surfaceName} {typeData} {fileName} {stride} {nbPrimitive}]
  }
  AOI
  {
    [AOI {aoiname} {equation}]
    ...
  }
}

```