



UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

RAPPORT DE STAGE MASTER 1

Étude et évaluation de la structure de donnée SVDAG et ses variantes pour le RayTracing en visualisation scientifique

Auteur : ANTOINE ROCHE

Tuteur de stage : JÉRÔME DUBOIS

Tuteur enseignant : MICHAEL KRAJECKI

08/04/2019 — 30/08/2019

CEA, DAM, DIF, F-91297, Arpajon, France

Remerciements

Je tiens à remercier le personnel du CEA, pour son aide et son accueil. Je remercie aussi Jérôme Dubois et Jacques-Bernard Lekien pour leur aide précieuse, ainsi que l'opportunité de participer à la Journée Visu 2019 pour présenter mon travail et prendre connaissance de celui d'autres chercheurs dans le milieu de la visualisation. Je remercie également Michael Krajecki pour m'avoir donné la chance de travailler au sein du CEA.

Glossaire

CPU : Central Processing Unit

GPU : Graphics Processing Unit

CUDA : Compute Unified Device Architecture

VTK : Visualization ToolKit

SVO : Sparse Voxel Octree

DAG : Directed Acyclic Graph

SVDAG : Sparse Voxel DAG

SSVDAG : Symmetry-aware Sparse Voxel DAG

AMR : Adaptive Mesh Refinement

HTG : HyperTreeGrid

CEA : Commissariat à l'Énergie Atomique et aux Énergies Alternatives

Voxel : Volume Element

Sommaire

1	Introduction	5
2	Présentation CEA	6
3	Contexte du stage	13
4	Études menées	14
4.1	Sparse Voxel Octree	14
4.2	Sparse Voxel Directed Acyclic Graph	15
4.3	HyperTreeGrid	16
5	Travail réalisé	17
5.1	Prise de connaissances	17
5.2	Installation de l'environnement	17
5.3	Installation et fonctionnement du programme SVDAG	17
5.4	Analyse des performances du SVDAG	17
5.4.1	Résultats	18
5.4.2	Analyse	19
5.5	Présentation des résultats	19
6	Expérience personnelle et professionnelle	20
6.1	Collaboration et contribution	20
6.1.1	Université technique de Chalmers	20
6.1.2	CEA	20
6.1.3	Journée Visu 2019	20
6.2	Problèmes rencontrés	21
6.2.1	Installations	21
6.2.2	SVDAG	21
7	Travaux futurs	22
8	Conclusion	22
9	Bibliographie	23

1 Introduction

Le CEA, acteur majeur en matière de recherche et d'innovation, est considéré comme étant un expert du Calcul Haute Performance. Dans ce contexte, les équipes spécialisées en visualisation scientifique sont reconnues pour leur expertise en visualisation parallèle sur supercalculateur. Elles participent activement au développement de la bibliothèque open-source Visualization ToolKit (VTK), qui fournit un pipeline complet de visualisation associant différents types de données et algorithmes.

Pour répondre aux besoins des codes de simulation, le CEA a développé depuis quelques années des structures de données et des filtres VTK dédiés à la visualisation et la manipulation de maillages Adaptive Mesh Refinement (AMR). D'autres structures hiérarchiques utilisées dans le cadre de l'industrie cinématographique ou du jeux-videos permettent un rendu réaliste de type ray-tracing en temps réel en faisant appel à la programmation sur carte graphique (CUDA).

Ces structures sont basées sur les Sparse Voxel Octrees (SVO), une représentation d'une scène voxelisée de type octree optimisée. Des techniques de compression offrent par la suite des gains substantiels en tenant compte des motifs similaires : les Sparse Voxel Directed Acyclic Graph (SVDAG) ou encore les Symmetric SVDAG (SSVDAG). L'objectif du stage est de permettre l'évaluation de ces structures hiérarchiques pour la visualisation scientifique.

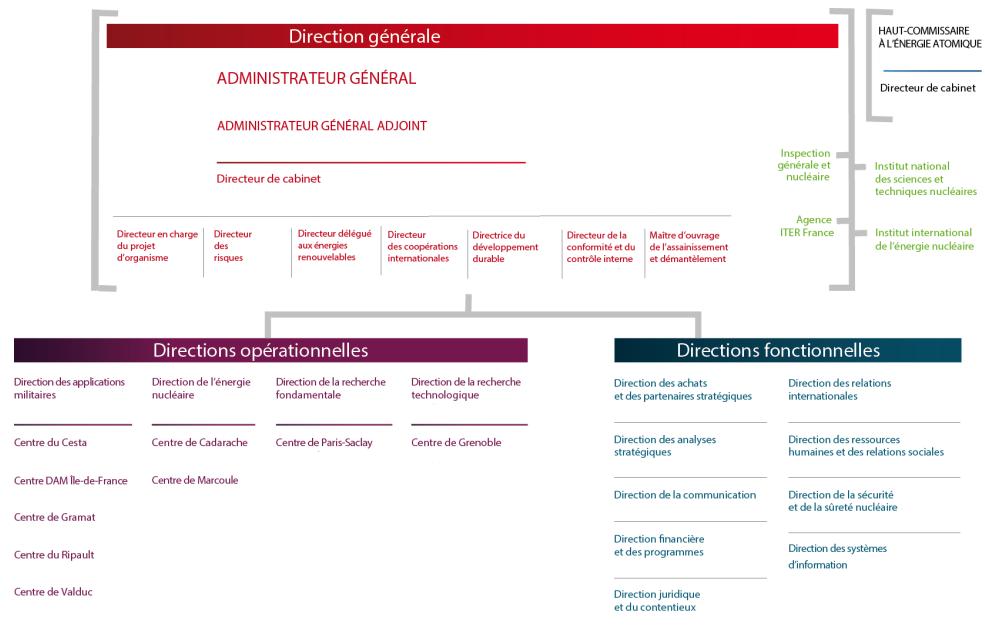
2 Présentation CEA

Acteur majeur de la recherche, du développement et de l'innovation, le Commissariat à l'énergie atomique et aux énergies alternatives intervient dans quatre domaines :

- La défense et la sécurité. ;
- Les énergies bas carbone (nucléaires et renouvelables). ;
- La recherche technologique pour l'industrie. ;
- La recherche fondamentale (sciences de la matière et sciences de la vie).

S'appuyant sur une capacité d'expertise reconnue, le CEA participe à la mise en place de projets de collaboration avec de nombreux partenaires académiques et industriels. Le CEA est implanté sur 9 centres répartis dans toute la France. Il développe de nombreux partenariats avec les autres organismes de recherche, les collectivités locales et les universités. A ce titre, le CEA est partie prenante des alliances nationales coordonnant la recherche française dans les domaines de l'énergie (ANCRE), des sciences de la vie et de la santé (AVIESAN), des sciences et technologies du numérique (ALLISTENE), des sciences de l'environnement (ALIEnvi) et des sciences humaines et sociales (ATHENA).

Reconnu comme un expert dans ses domaines de compétence, le CEA est pleinement inséré dans l'espace européen de la recherche et exerce une présence croissante au niveau international. Le CEA compte 15 942 techniciens, ingénieurs, chercheurs et collaborateurs pour un budget de 5 milliards d'euros (chiffres publiés fin 2017).



La direction des applications militaires

Une direction au service de la dissuasion

La Direction des applications militaires (DAM) du CEA, a pour mission de concevoir, fabriquer, maintenir en condition opérationnelle, puis démanteler les têtes nucléaires qui équipent les forces nucléaires aéroportée et océanique françaises.

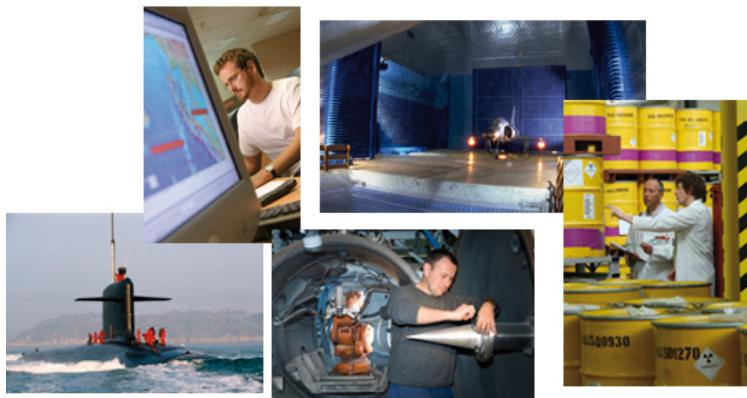
La DAM est chargée de la conception et de la réalisation des réacteurs et de coeurs nucléaires équipant les bâtiments de la Marine nationale, sous-marins et porte-avions. Elle apporte son soutien à la Marine nationale pour le suivi en service et le maintien en condition opérationnelle de ces réacteurs.

La DAM est également responsable de l'approvisionnement des matières nucléaires stratégiques pour les besoins de la dissuasion.

Dans un monde en profonde mutation, la DAM contribue aussi à la sécurité nationale et

internationale à travers l'appui technique qu'elle apporte aux autorités, pour les questions de lutte contre la prolifération nucléaire et le terrorisme et de désarmement.

Depuis le transfert du centre de Gramat en 2010 de la Direction générale de l'armement au CEA, la DAM apporte son expertise à la Défense dans le domaine de l'armement conventionnel.



Une direction ouverte à la recherche

Le partage national et international des connaissances (lorsqu'il est possible), la confrontation à l'évaluation scientifique extérieure, l'intégration à des réseaux de compétences constituent des gages de crédibilité scientifique.

Les équipes de la DAM réalisent chaque année environ 2000 publications et communications scientifiques. Cette ouverture de la DAM passe également par la mise à la disposition de la communauté des chercheurs de ses moyens expérimentaux et par la contribution de ses équipes à d'autres programmes de recherche.

Une direction actrice de la politique industrielle française

La DAM partage très largement son activité avec l'industrie française : c'est ainsi que le montant des achats, auprès de celle-ci, représente plus des deux tiers de son budget ; le dernier tiers se répartit entre les salaires des personnels (un cinquième) et les taxes.

La politique industrielle de la DAM est originale à plus d'un titre :

- d'abord parce que la DAM conserve la maîtrise d'œuvre d'ensemble de la grande majorité des systèmes dont elle a la responsabilité : elle veille ainsi au juste équilibre entre les grands groupes industriels de la Défense et les PME souvent innovantes, en contractualisant directement avec ces dernières, leur permettant ainsi de recevoir la juste rémunération de leur production ;
- ensuite, parce que la répartition de son budget est sous-tendue par une répartition des travaux : la DAM conduit la recherche dans ses laboratoires grâce à son personnel de haut niveau scientifique et technologique. Une fois la définition d'un produit acquise, la DAM transfère la définition et les procédés vers les industriels qui en réalisent le développement, puis la production.

La DAM a également pour objectif que ses centres participent à la vie économique locale par leur implication dans les pôles de compétitivité. Hors de son propre champ d'utilisation, elle valorise ses recherches par le transfert de technologies vers l'industrie et le dépôt de nombreux brevets.

Le format

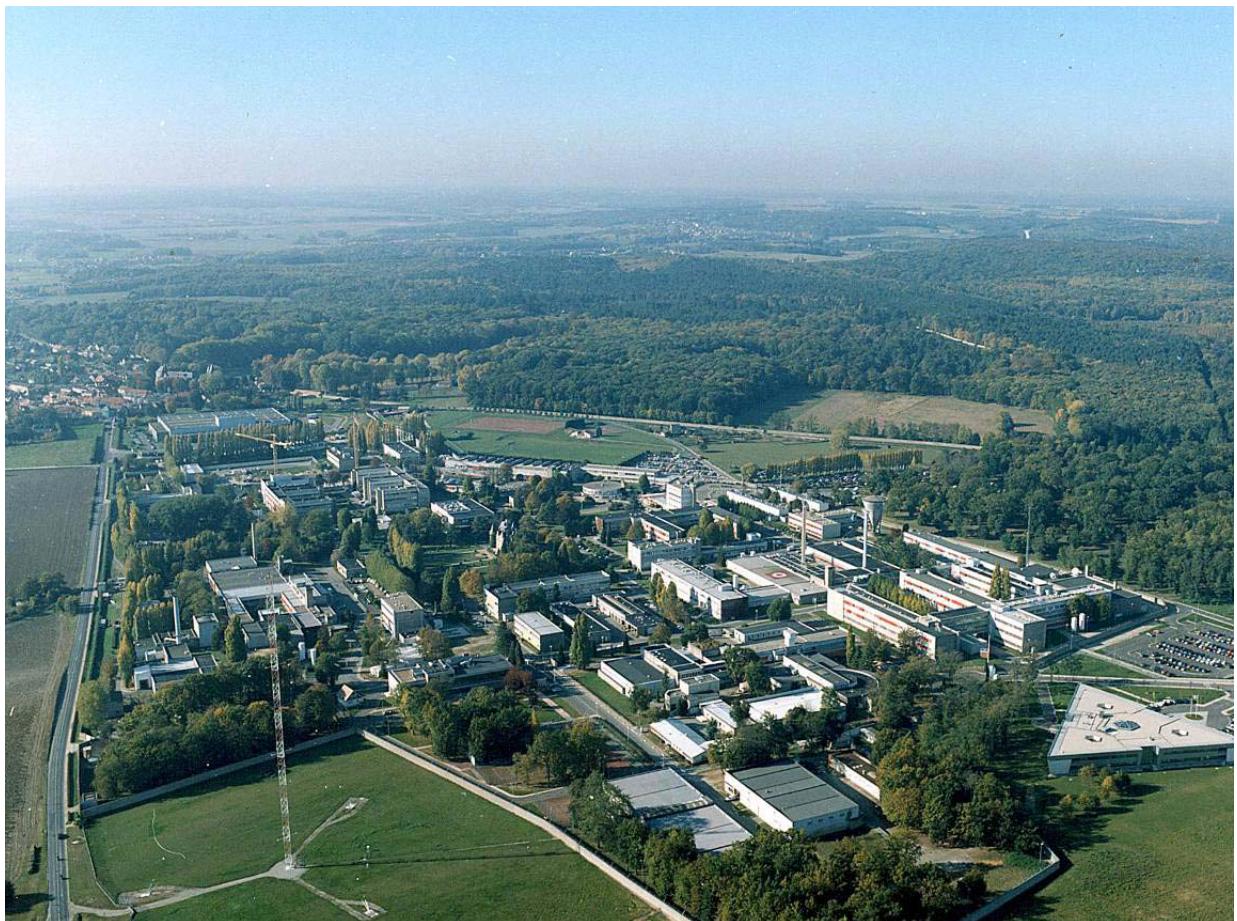
La DAM comprend cinq centres aux missions homogènes, dont les activités se répartissent entre la recherche de base, le développement et la fabrication :

- **DAM Ile-de-France (DIF)**, à Bruyères-le-Châtel, où sont menés les travaux de physique des armes, les activités de simulation numérique et de lutte contre la prolifération nucléaire ; DIF est aussi le centre responsable de l'ingénierie à la DAM ; enfin, au centre DIF est rattachée l'INBS-Propulsion Nucléaire du centre CEA/Cadarache, en région Provence Alpes-Côte d'Azur, où sont implantées les installations d'essais à terre et une partie des fabrications de la propulsion nucléaire ;



- **Le Cesta**, en Aquitaine, consacré à l'architecture des armes, aux tests de tenue à l'environnement. Il met en oeuvre le Laser Mégajoule, équipement majeur de la Simulation ;
- **Valduc**, en Bourgogne, dédié aux matériaux nucléaires et à l'installation expérimentale Epure du programme Simulation ;
- **Le Ripault**, en région Centre, dédié aux matériaux non nucléaires (explosifs chimiques...);
- **Gramat**, (ex-DGA) en Midi-Pyrénées, qui conduit au profit de la Défense des activités en vulnérabilité des systèmes et efficacité des armements. ;

Le centre DAM Ile-de-France



Centre DAM Île-de-France

Le CEA/DAM - Île de France (DIF) est l'une des directions opérationnelles de la DAM. Le site de la DIF compte environ 2000 salariés CEA et accueille quotidiennement environ 600 salariés d'entreprises extérieures. Il est situé à Bruyères-le-Châtel à environ 40 km au sud de Paris, dans l'Essonne.

Les missions de la DIF comprennent :

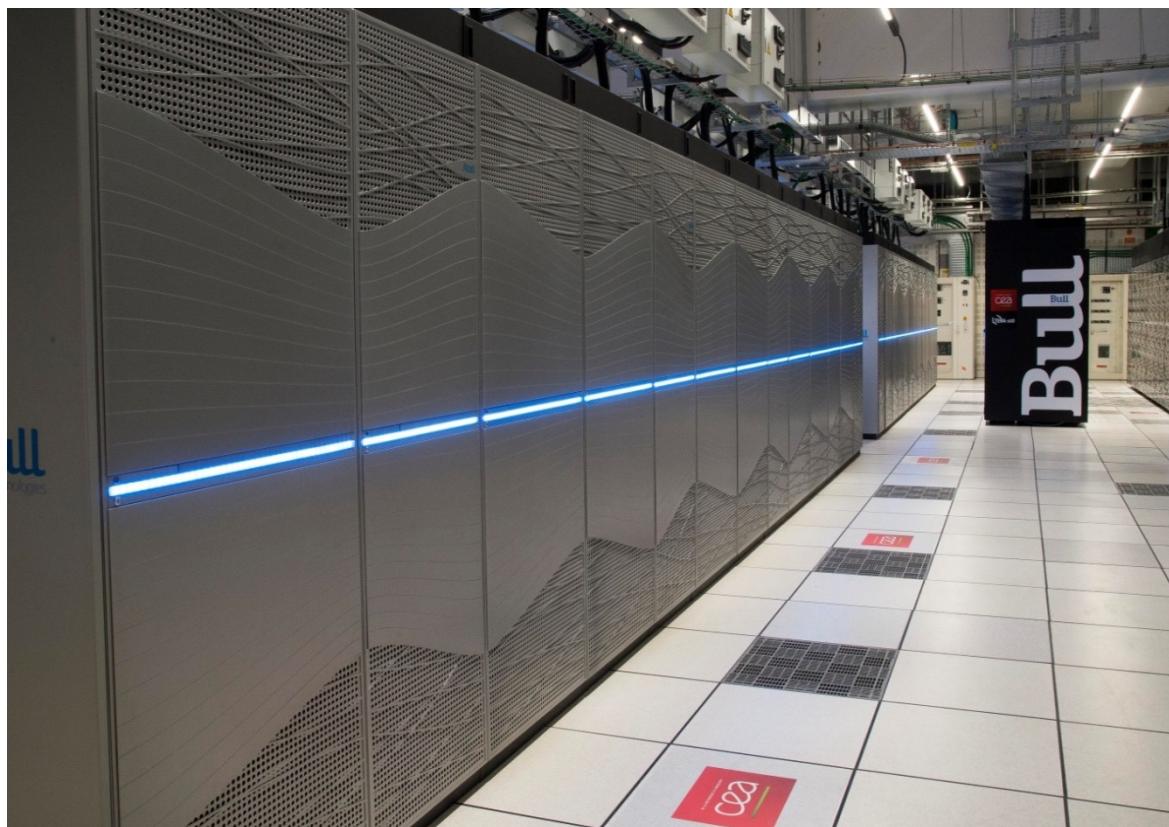
- La conception et garantie des armes nucléaires, grâce au programme Simulation. L'enjeu consiste à reproduire par le calcul les différentes phases du fonctionnement d'une arme nucléaire et à confronter ces résultats aux mesures des tirs nucléaires passés et aux résultats expérimentaux obtenus sur les installations actuelles (machine radiographique, lasers de puissance, accélérateurs de particules).;
- La lutte contre la prolifération et le terrorisme, en contribuant notamment au programme de garantie du Traité de Non-Prolifération et en assurant l'expertise tech-

nique française pour la mise en œuvre du Traité d’Interdiction Complète des Essais Nucléaires (TICE).

- L’expertise scientifique et technique, dans le cadre de la construction et du démantèlement d’ouvrages complexes ainsi que pour la surveillance de l’environnement et les sciences de la terre.
- L’alerte des autorités, mission opérationnelle assurée 24h sur 24, 365 jours par an, en cas d’essai nucléaire, de séisme en France ou à l’étranger, et de tsunami dans la zone Euro-méditerranéenne. La DIF fournit aux autorités les analyses et synthèses techniques associées.

Depuis 2003, le centre DAM-Île-de-France héberge le complexe de calcul scientifique du CEA, qui regroupe l’ensemble des supercalculateurs du CEA, et qui comprend à ce jour :

- Le supercalculateur Tera1000-1 pour les besoins du programme Simulation du CEA/-DAM, mis en service en 2016, dispose d’une puissance de calcul de 2,5 petaflops, c’est à dire capable d’effectuer 2,5 millions de milliards d’opérations par seconde. Il est complété en 2018 par Tera1000-2, autre composante du projet Tera1000, qui préfigure les architectures et technologies du futur supercalculateur qui sera installé à l’horizon 2020. Sa puissance de calcul est de 12,5 petaflops.



- Le supercalculateur Cobalt du Centre de Calcul pour la Recherche et la Technologie (CCRT), ouvert à la communauté civile de la recherche et de l'industrie, pour une puissance globale de 1,5 petaflops.



- Le supercalculateur IRENE, d'une puissance de 9 petaflops, deuxième élément d'un réseau de supercalculateurs de classe petaflopique destiné aux chercheurs de la communauté scientifique européenne. Ce supercalculateur est hébergé au TGCC (Très Grand Centre de Calcul) et exploité par les équipes du CEA, qui apporte ainsi sa contribution à la participation de la France au projet PRACE (Partnership for Advanced Computing in Europe)

3 Contexte du stage

Étant en collaboration directe avec l'équipe de visualisation du CEA, le but de mon stage est d'appliquer des technologies du jeu vidéo sur de la visualisation scientifique. Plus précisément, mon objectif est de convertir une structure HTG, technologie nouvelle dans VTK vers une structure SVDAG. Un HTG permet de réduire le volume et le temps de rendu de données de simulations massives. Le SVDAG, quant à lui, est notamment appliquée pour du raytracing sur des décors de jeux vidéo, qui permet de compresser un octree sans perte de données. Ces deux technologies peuvent par exemple permettre la visualisation de certaines scènes très massives, initialement visualisable uniquement sur supercalculateur, sur des ordinateurs portables ou des téléphones.

Afin de mener le projet jusqu'à son terme, je devrai tout d'abord étudier et comprendre la théorie et l'implémentation des structures SVO/SVDAG/SSVDAG pour en évaluer les performances de rendu raytracing sur des modèles connus. Pour cela, je serai en contact avec les équipes de recherche suédoise de l'Université technique de Chalmers à la base du SVDAG, avec lesquels je pourrai donc collaborer.

D'un autre côté, je devrai comprendre et maîtriser la construction de structure HTG via VTK. Je devrai notamment savoir construire un AMR à partir d'un SVO et avoir un résultat visuel identique au SVO initial.

Une fois la phase de recherche terminée, un premier convertisseur VTK-HTG vers SVDAG/SSVDAG sera à implémenter. Des tests de performances s'en suivront afin de les comparer avec celles des autres structures.

Pour finir, le travail se poursuivra par l'application de rendu réaliste de type lancé de rayons à base de SVDAG sur des données issues de VTK-HTG. Plusieurs développements seront possibles en vue de prendre en compte la notion de variables ou de compresser encore plus les informations dans le cadre de rendu de films interactifs. Les développements effectués seront évalués aussi bien sur GPU (CUDA) que sur les nombreux accélérateurs et processeurs disponibles au CEA.

4 Études menées

Dans le cadre du stage, des études préliminaires étaient nécessaires afin de comprendre les structures de données que j'allais manipuler.

4.1 Sparse Voxel Octree

Le Sparse Voxel Octree (SVO) est la structure de base à connaître afin de comprendre les autres. Pour cela, je me suis basé sur la thèse de Viktor Kämpe sur les SVDAG[4] ainsi que le papier de recherche de NVIDIA Research sur le Efficient Sparse Voxel Octree (ESVO)[6], car ils introduisent le SVO de sorte à comprendre le reste.

Le SVO est une représentation de données en arbre. Sa structure permet d'accéder à une information rapidement. Il est construit à partir d'une scène 3D non structurée, qui est ensuite découpée en grille dont les cases contenant un élément sont à leur tour découpées jusqu'à atteindre le niveau défini. Chaque feuille du dernier niveau devient alors un voxel.

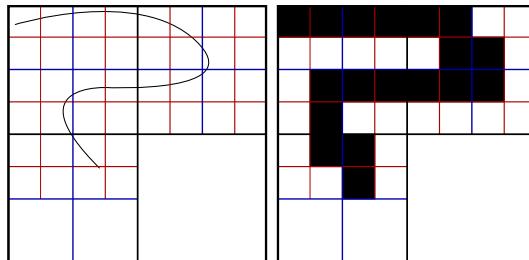


FIGURE 1: Découpage d'une scène 2D avec 3 niveaux et représentation des "voxels" obtenus

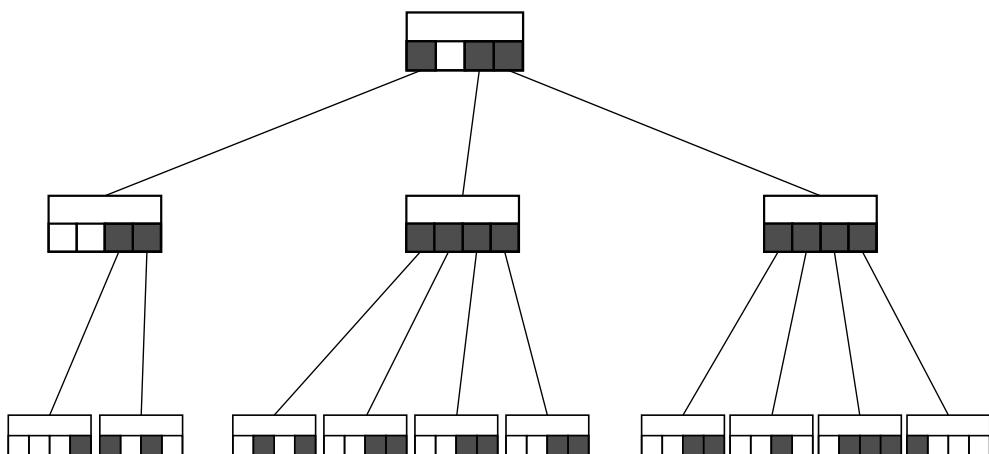


FIGURE 2: Représentation de l'arbre construit

4.2 Sparse Voxel Directed Acyclic Graph

Pour le Sparse Voxel Directed Acyclic Graph (SVDAG), j'ai de nouveau utilisé la thèse de Victor Kämpe^[4] ainsi que le papier "High Resolution Sparse Voxel Dags"^[5] écrit par Victor Kämpe, Erik Sintorn et Ulf Assarsson. J'ai également étudié le programme qu'ils ont partagé, muni d'un jeu de donnée en exemple.

Le SVDAG est un arbre de voxel compressé. Il peut être construit à partir d'un SVO. Sa particularité est qu'un noeud peut être pointé par plusieurs parents, évitant ainsi de stocker plusieurs noeuds s'ils sont identiques. Il est également acyclique, nous pouvons descendre dans l'arbre mais pas remonter, cela offre un gain de stockage supplémentaire. Tout cela permet de faire tenir en mémoire des structures trop volumineuses à la base.

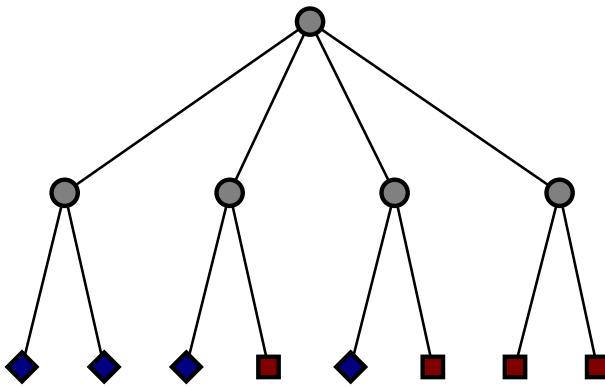


FIGURE 3: Représentation d'un arbre de données

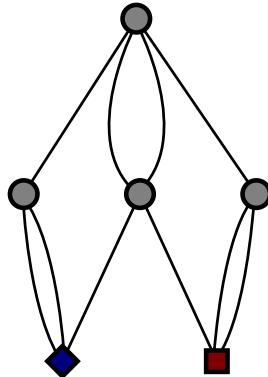


FIGURE 4: Représentation de l'arbre après compression via SVDAG

4.3 HyperTreeGrid

Prendre connaissance de l'HyperTreeGrid (HTG) fut plus basé sur des explications venant de Jérôme Dubois et Jacques-Bernard Lekien qui m'ont également fourni des programmes exemples. Je me suis tout de même appuyé sur une présentation de Massimiliano Guarasi[3], introduisant l'AMR en général et les outils associés.

Le HTG est une grille d'arbres semblables au SVO, à la grande différence que les voxels peuvent se situer à des niveaux différents dans l'arbre. Ceci implique qu'une scène peut contenir des voxels plus grossiers que d'autres si un affinage n'est pas nécessaire. En contrepartie, des zones peuvent être très détaillées sans alourdir la mémoire de façon excessive.

De plus, il est aisément de diminuer le niveau maximal des voxels pour obtenir une scène moins détaillée mais beaucoup plus fluide (voire en temps réel) afin de configurer rapidement les différents paramètres pour le rendu final.

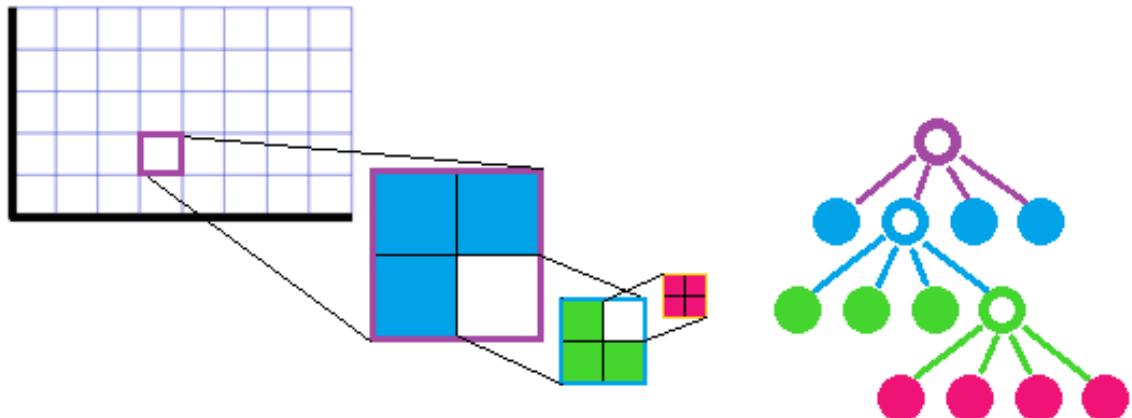


FIGURE 5: Grille de type AMR de niveau 3 avec son arbre associé

5 Travail réalisé

5.1 Prise de connaissances

La première semaine du stage fut dédiée à la connaissance et à la compréhension des structures de données citées plus haut. Pour cela, j'ai lu plusieurs papiers de recherche, en me focalisant sur les parties qui m'étaient nécessaires.

5.2 Installation de l'environnement

Plusieurs semaines ont ensuite été dédiées à l'installation de l'environnement nécessaire. N'ayant pas les droits d'administrateur, j'ai installé Spack, un gestionnaire de paquets pour supercalculateurs sur Linux et MacOS. Il permet d'installer des paquets en une commande. Il est aussi possible de produire un paquet avec plusieurs versions, configurations, plateformes et compilateurs, et tous ces productions peuvent coexister sur la même machine. (voir www.spack.io)

J'ai ensuite installé VTK, contenant des modules pour l'AMR, et le programme SVDAG fourni, ainsi que toutes les dépendances. J'ai également créé une recette Spack pour le SVDAG, permettant ainsi de l'installer avec toutes ses dépendances facilement, et également de partager mes travaux avec l'équipe de visualisation.

5.3 Installation et fonctionnement du programme SVDAG

Contrairement à tous les autres code sources, celui du SVDAG a été développé sur Windows. J'ai donc dû résoudre les problèmes de compatibilité, notamment causés par un paquet dépendant au système.

Le programme maintenant fonctionnel sous Unix, j'ai pu commencer à le tester avec les fichiers d'entrée donnés en exemple. Afin d'utiliser d'autres fichiers en entrée, j'ai du chercher et installer un constructeur de SVO, qui voxelise une scène à partir d'un maillage non structuré. Je l'ai également modifié de sorte à sortir les fichiers aux bons formats, pour qu'ils soient compatibles avec le SVDAG. De l'autre côté, j'ai ajusté l'entrée du SVDAG. J'ai également ajouté une option permettant d'écrire le SVDAG sur le disque pour ne pas avoir à le construire à chaque fois que l'on veut le visualiser.

5.4 Analyse des performances du SVDAG

Avec tout en place, j'ai pu tester le programme afin de comprendre comment il se comporte selon différents paramètres. J'ai également analysé ses performances avec quelques jeux de données connus.

5.4.1 Résultats

Les résultats ci dessous ont été obtenus avec un Dell Precision 7920 Tower, doté de 2 x Intel Xeon Gold 5122 CPU 3.60GHz (16 coeurs au total), d'une Nvidia Quadro P4000 et 64Go de mémoire vive.

La version du driver Nvidia est 390.59, avec CUDA 9.0, compilé avec GCC 6.4.0 sous CentOS 7.



FIGURE 6: Rendu de modèles avec du raycasting sur SVDAG
De gauche à droite : Bunny (Stanford), Dragon (Stanford), Hairball (NVIDIA Research), Lucy (Stanford)

	Nombre Triangles	Taille non-structuré	Temps construction SVO	Temps construction SVDAG
Bunny	69 451	3.0 Mo	3.9s	4.3s
Dragon	871 414	33.8 Mo	5.7s	3.0s
Hairball	2 880 000	236.1 Mo	58.7s	47.7s
Lucy	28 055 742	533.1 Mo	77.5s	1.7s

	Nombre voxels	Taille SVO Structure + grandeurs	Taille SVDAG Structure + grandeurs	Taux compression
Bunny	3 591 666	14.4 + 57.5 Mo	1.3 + 8.5 Mo	86.3%
Dragon	2 688 970	10.8 + 43.0 Mo	1.0 + 6.4 Mo	86.2%
Hairball	41 521 450	166.1 + 664.3 Mo	13.3 + 94.6 Mo	87.0%
Lucy	1 540 004	6.2 + 24.3 Mo	0.6 + 3.6 Mo	86.3%

TABLE 1: Résultats avec des scènes voxélisées de résolution 1K³

5.4.2 Analyse

Tout d'abord, les modèles sont ordonnés selon le nombre de triangles qui les composent. Nous pouvons en premier lieu constater que la taille des modèles non-structurés dépend du nombre de triangles. Cependant, le nombre de voxels obtenus, quant à lui, dépend du volume du modèle dans le cube. Par exemple, la boule de poil est sphérique et "pleine" tandis que Lucy est un modèle creux et haut, rendant une grosse partie du cube vide.

Concernant le SVO, sa taille dépend du nombre de voxels, tandis que son temps de construction dépend du nombre de triangles.

La taille et le temps de construction du SVDAG sont liés au nombre de voxels. De plus, sa taille dépend également de la régularité des modèles.

La structure du SVDAG est environ 10 fois moins volumineuse que le SVO, et les grandeurs 85% plus légères. Ce qui donne un taux de compression total d'environ 86% ;

5.5 Présentation des résultats

Dans le cadre de plusieurs conférences, telles que la journée Visu 2019 et Eurovis 2019, il m'a été demandé de faire une slide afin de montrer mon travail dans le domaine de la visualisation avec les premiers résultats obtenus.

cea FUTURE WORK (1 / 2) - ANTOINE ROCHE

Apply video games technologies to scientific visualization

SVDAG : compressed octree used to raycast rasterized scenes

Compare SVDAG compression to VTK HyperTreeGrid

GPU-raycast HyperTreeGrid with SVDAG data structure

Data set	Triangles	Voxels 1024 ³	Octree	SVDAG
Hairball	2 880 000	41 521 450	166 + 664 MB	13 + 95 MB
Bunny	69 451	3 591 666	14 + 56 MB	1.3 + 8.5 MB

17th May 2020 7

6 Expérience personnelle et professionnelle

6.1 Collaboration et contribution

6.1.1 Université technique de Chalmers

Afin de comprendre certaines choses sur le SVDAG, j'ai sollicité Dan Dolonius et Ulf Assarsson, deux chercheurs de l'Université technique de Chalmers à l'origine du SVDAG. La collaboration n'a pas beaucoup abouti, j'ai cependant eu l'occasion des les renseigner sur la compatibilité de leur code vers Unix.

De plus, le programme fourni est une ancienne version de ce qu'ils possèdent actuellement, et en avoir une plus récente nous offrirait plus de possibilités.

6.1.2 CEA

Quant à VTK et à au HTG, Jérôme Dubois et Jacques-Bernard Lekien, contributeurs à VTK, m'ont beaucoup expliqué le fonctionnement de la structure HTG et sa construction, me donnant de bonnes bases pour la suite.

6.1.3 Journée Visu 2019

A l'occasion de la journée de la visualisation 2019 se déroulant à Paris le 17 mai, j'ai pu présenter mon travail aux personnes présentes en m'appuyant sur la slide produite. Cela a suscité l'intérêt de certains chercheurs, demandant par exemple si mon travail pourrait être intégré à VTK, ou encore si de la compression temporelle serait possible.

6.2 Problèmes rencontrés

6.2.1 Installations

Tout d'abord, l'accès aux machines internes a pris du temps, ce qui a retardé les installations de quelques jours. De plus, les restrictions de l'entreprise causés par le proxy nous empêchant d'utiliser curl et wget librement, il a donc fallu créer des miroirs Spack via Roméo, le supercalculateur de l'Université de Reims, pour ensuite les transférer en SSH. Cependant, cela nous exposait à des incidents externes, auquelcas nous devions télécharger les paquets un par un via un navigateur. Au final, une solution pour le proxy nous a été proposée, nous permettant de télécharger via curl depuis la plupart des sources.

Concernant VTK, des données de référence de test étaient téléchargées durant la compilation, faisant échouer une cible de la compilation à cause du proxy. Une solution envisagée était de les télécharger sur Roméo puis de les transférer en SSH, mais nous avons décidé de désactiver le téléchargement des tests.

De plus, l'installation de VTK pour python est en option, ce que je ne savais pas, et les programmes d'exemples qui m'étaient donnés, issus de TPs faits à Reims pour les Master 2 CHPS, étaient en python. Après plusieurs heures de recherches sur l'importation de VTK (car je pensais toujours que VTK python était installé), j'ai fini par retranscrire les exemples en C++, ce qui m'a permis au passage de comprendre les différentes étapes.

6.2.2 SVDAG

Le problème de compatibilité Windows - Unix expliqué plus haut m'a également pris beaucoup de temps. En effet, en plus du paquet dépendant au système, l'appel d'une fonction template externe ne passait pas à la compilation. Les solutions sur internet liées à l'erreur retournée ne parlant que de la fonction en elle-même et non de son appel, j'ai dû en partie comprendre les types des paramètres donnés (qui étaient des vecteurs très complexes) afin de trouver la source du problème.

De plus, le code est fourni sans documentation ni commentaires, ce qui m'a donné quelques difficultés pour comprendre son fonctionnement.

7 Travaux futurs

Durant le reste du stage, mon travail consistera dans un premier temps à prendre connaissance du SSVDAG, pour le comparer avec le SVDAG et voir quel est le plus intéressant. Ensuite, je devrai maîtriser la structure VTK-HTG en créant un HTG à partir d'un SVO en obtenant le même résultat, pour par la suite faire un convertisseur AMR vers SVDAG/SSVDAG. Ensuite, des rendus seront faits à partir de données plus riches que celles utilisées jusqu'à présent afin d'évaluer les performances et les améliorations possibles.

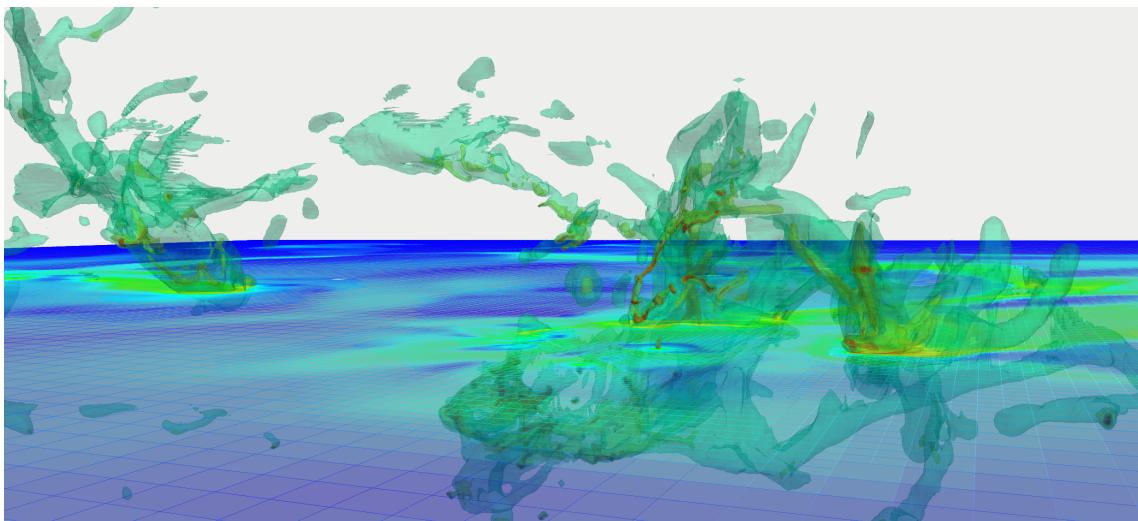


FIGURE 7: Deep water impact à un temps donné, 72 milliards éléments

8 Conclusion

Manipuler des structures de données complexes demande beaucoup de recherches préliminaires. De plus, certains problèmes m'ont fait perdre beaucoup de temps. Cependant, les résultats obtenus sont encourageants pour la suite. La familiarisation avec l'environnement de travail me permet désormais de travailler plus efficacement qu'au début du stage.

Le domaine de la recherche en visualisation scientifique est nouveau pour moi, et j'en tire toujours plus d'expérience chaque jour, m'encourageant d'avantage à mener ce projet à son terme.

9 Bibliographie

Références

- [1] Jeroen Baert. Out-of-core svo builder. https://github.com/Forceflow/ooc_svo_builder.
- [2] Dan Dolonius. Dag-example. <https://github.com/gogoggigog/DAG-example>.
- [3] Massimiliano Guarrasi. An introduction to adaptive mesh refinement (amr) : Numerical methods and tools, 2015.
- [4] Viktor Kämpe. *Sparse Voxel DAGs*. PhD thesis, Chalmers University of Technology, 2016.
- [5] Viktor Kämpe, Erik Sintorn, and Ulf Assarsson. High resolution sparse voxel dags. *ACM Transactions on Graphics*, 32(4), 2013. SIGGRAPH 2013.
- [6] Samuli Laine and Tero Karras. Efficient sparse voxel octrees. In *Proceedings of ACM SIGGRAPH 2010 Symposium on Interactive 3D Graphics and Games*, 2010.
- [7] Samuli Laine and Tero Karras. Efficient sparse voxel octrees – analysis, extensions, and implementation. NVIDIA Technical Report NVR-2010-001, NVIDIA Corporation, February 2010.