



UNIVERSITÉ
LAVAL

Stage en génie informatique I

GIF-2580

Été 2020

Génie informatique

FACULTÉ DES SCIENCES ET DE GÉNIE

Stagiaire de recherches
Laboratoire de vision et systèmes numériques

Rapport adressé au
Département des stages en milieu pratique

3 septembre 2020

Christophe Bolduc
111 237 250

Professeur Denis Laurendeau
Directeur Laboratoire de vision et
systèmes numériques

Résumé

Ce document a pour but de présenter le travail effectué par Christophe Bolduc au sein du Laboratoire de vision et Systèmes numériques durant son stage au cours de l'été 2020. Le stagiaire avait la responsabilité de développer un outil de visualisation de données de scanneur 3D par la construction de champs de vecteurs et d'y ajouter le travail effectué par une étudiante au doctorat. L'étudiant a également eu la chance de proposer une méthode de fusion de courbes 3D.

Remerciements

L'étudiant remercie Denis Laurendeau pour lui avoir accordé sa confiance en lui confiant des responsabilités de taille et d'avoir fourni les ressources et le soutien pour rendre possible leur complétion.

Le stagiaire tient également à remercier Mana Eskandari pour sa disponibilité pour répondre à de nombreuses questions.

Table des matières

Résumé	1
Remerciements	2
Introduction	5
Responsabilités et tâches	7
Outils	7
Structure du champ de vecteurs	8
Visualisation du champ	10
Projet de la doctorante	14
Attentes	17
Développement et renforcement des compétences	19
Technique	19
Gestion	20
Conclusion	22
Bibliographie	23
À propos de la police	24

Liste des figures

Figure 1. Grille volumétrique (TUBIC)	8
Figure 2. Approximation de surface dans un voxel 2D (TUBIC)	8
Figure 3. Structure Octrees en 2D (TUBIC)	9
Figure 4. Lancement de rayons	10
Figure 5. Algorithme DDA (Scratchapixel 2.0)	10
Figure 6. Rendu produit par le programme	11
Figure 7. Vue decouche linéaire	12
Figure 8. Vue de couche non linéaire	12
Figure 9. Quinze configurations possibles de l'algorithme Marching Cubes (Fisher)	13
Figure 10. Géométrie de a théière produite par Marching Cubes	13
Figure 11. Interface utilisateur du programme	14
Figure 12. Calcul du type de surface (Eskandari)	15
Figure 13. Huit types de surface (Eskandari)	15
Figure 14. Types de surface de la théière	15
Figure 15. Groupes de surface "peak" de la théière	15
Figure 16. Effet du rayon d'influence de la fusion	17
Figure 17. Résultat de la théière fusionnée	17

Introduction

Christophe Bolduc, l'étudiant ayant réalisé le stage détaillé dans ce document, est étudiant en génie informatique à l'Université Laval. Son parcours, bien que quelque peu traditionnel, sciences naturelles au cégep, l'a amené à développer un intérêt profond pour la technologie du visuel. Ainsi, sa passion et son bagage académique l'ont amené à chercher un milieu dans le domaine pour son premier stage.

Il en résulte une visite au bureau du professeur et chercheur Denis Laurendeau pour s'informer sur la possibilité de faire son stage dans le Laboratoire de vision et Systèmes numériques dans le cadre du programme "Bourse de Recherche de Premier Cycle" du Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG). Un projet initial est décidé, puis une candidature pour le programme est placée.

Le Laboratoire de vision et Systèmes numériques, du Département de génie électrique et de génie informatique de l'Université Laval, représente un milieu où huit professeurs ainsi que plusieurs dizaines d'étudiants travaillent sur le thème de la vision par ordinateur et de ses applications industrielles et biomédicales. Les résultats de recherches se sont mérités des publications dans la littérature spécialisée internationale (Laboratoire de vision et Systèmes numériques).

Les semaines écoulées entre le dépôt de la candidature et la réception de la nomination du boursier entraînent avec elles des changements inimaginables dans la situation socio-économique. Heureusement, le stage a toujours lieu, mais le projet décidé tombe à l'eau. Le superviseur de stage définit ainsi un projet adapté au travail à distance. Durant les 16 semaines du stage, Christophe sera responsable du développement d'un logiciel de visualisation de données d'un scanneur 3D par la construction de champs de vecteurs dans une grille volumétrique où chaque cellule est appelée un voxel. Le projet pourra ensuite être adapté pour assister le travail de recherche d'une étudiante au doctorat : chaque voxel se verra attribuer un type de surface et un ensemble de courbes Bézières associées à chaque section de surface du même type seront fusionnées pour obtenir une géométrie finale de l'objet.

La nature du projet permet au stagiaire de travailler sans se déplacer et l'autonomie qui lui est accordée lui permet de gérer son horaire de 35 heures

par semaine de la manière qui lui convient. Une rencontre hebdomadaire avec le superviseur de stage, Denis Laurendeau, professeur au département de génie électrique et de génie informatique et directeur du Laboratoire de vision et Systèmes numériques, est mise à l'horaire, en plus d'une plateforme de clavardage pour une communication continue. La communication durant le projet se fera en anglais.

Responsabilités et tâches

La présente section détaille la progression du développement du logiciel de visualisation, en introduisant une base technique nécessaire, puis en présentant l'implantation dans le logiciel et des résultats pertinents.

Pour faciliter la compréhension de concepts qui peuvent paraître complexes pour certains, des sous-sections structurent le texte de manière intuitive. D'abord, une courte description du matériel pour accomplir le stage est présentée, ensuite la structure de champs de vecteurs est définie. Il en suit une explication du module de visualisation des données, puis l'addition des techniques formulées par l'étudiante au doctorat. Finalement, une discussion face aux attentes du stagiaire est développée.

Outils

Puisque le programme résultant nécessitera le calcul et l'affichage d'images en temps réel, le langage de programmation, C++ est choisi pour sa vitesse d'exécution. Pour permettre une interaction avec l'utilisateur, le module d'interface utilisateur Qt est ajouté au projet.

Pour permettre au programme d'évoluer facilement par l'ajout de modules apportant de nouvelles fonctionnalités, celui-ci sera développé par programmation de type orienté objet.

Le programme est développé entièrement localement sur l'ordinateur du stagiaire, permettant de valider l'utilisation temps réel du projet sur un ordinateur personnel haut de gamme moyen.

Finalement, pour aider au développement d'idées et à la communication avec les autres membres du laboratoire impliqués, des notes, continuellement synchronisées sur le nuage et disponibles par un lien transmis, sont rédigées sur iPad.

Structure du champ de vecteurs

La première étape au développement du projet est la définition du modèle à la base du reste du programme et l'implantation de la structure pour la stocker en mémoire.

Un champ de vecteurs et une grille uniforme dans laquelle chaque cellule près d'une surface contient la distance entre son centre et la surface et la direction de la normale à ce point. Puisque le projet étudie des modèles de 3 dimensions, la grille du champ de vecteurs est définie sur 3 dimensions (Figure 1). Ainsi, chaque cellule est appelée un voxel. Pour obtenir l'information du vecteur dans chaque voxel à partir d'une série de points, il est estimé que la surface dans chaque voxel peut être représentée par un plan. Une illustration en 2 dimensions permet de mieux comprendre le concept (Figure 2).

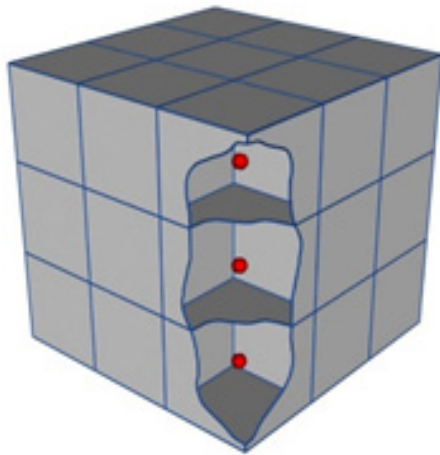


Figure 1. Grille volumétrique (TUBIC)

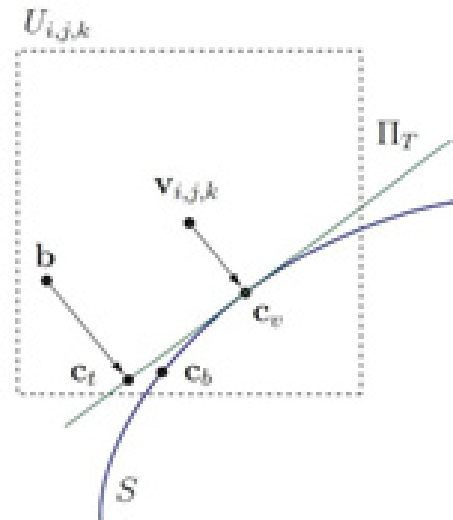


Figure 2. Approximation de surface dans un voxel 2D (TUBIC)

Étant fourni une dimension ou une taille de grille de voxels et un nuage de points comportant les coordonnées 3D relatives de la surface d'un objet, capturé par un scanner 3D par exemple, le programme peut facilement calculer la valeur du vecteur dans chaque voxel par l'algorithme suivant :

- Il est établi dans quel voxel se retrouve chaque point du nuage de points.
- Pour chaque voxel dans lequel il y a un minimum de points établis, une matrice de covariance permet d'identifier la direction vers laquelle la variation des points est moindre. Cette direction représente la normale du plan.
- La distance entre la surface et le centre du voxel est finalement calculée.

A ensuite été exploré le principe de Octrees, permettant de diviser la grille de voxels là où la surface se trouve et de conserver des voxels de plus grosses dimensions ailleurs (Figure 3). La technique permet de garder moins de voxels en mémoire, limitant la quantité de données requise, mais impose le même nombre de subdivisions sur chaque axe.

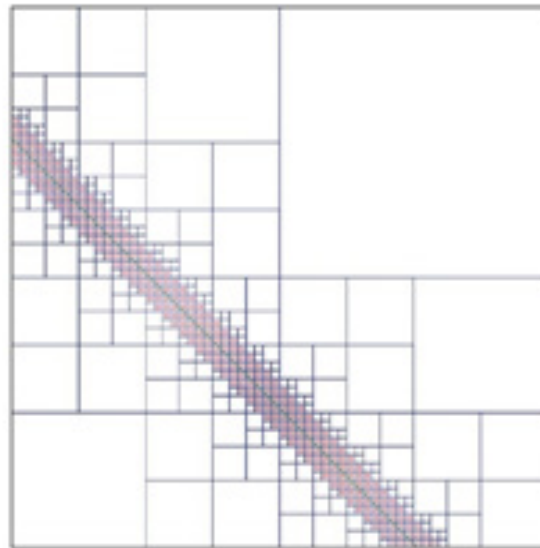


Figure 3. Structure Octrees en 2D (TUBIC)

Visualisation du champ

La visualisation principale du champ de vecteurs est effectuée de 3 manières distinctes pour obtenir l'information désirée. Un engin de rendu a été développé pour observer et tourner autour de l'objet, une vue « coupée » permet de traverser chaque couche du champ pour obtenir des données numériques dans chaque voxel et un algorithme de génération de géométrie produit des triangles pouvant être affichés dans des engins de rendu plus traditionnels. Finalement, le tout est introduit dans une interface graphique pour permettre une interaction avec l'utilisateur.

L'engin de rendu développé fonctionne par lancer de rayon. Une caméra est d'abord placée dans l'environnement 3D et pointée vers le centre de l'objet et la direction de la source de lumière est décidée. Selon des paramètres de résolution et de champ de vision modifiables, un rayon est lancé de l'origine de la caméra et traversant son pixel correspondant (Figure 4). Si le programme calcule que le rayon traverse les frontières de l'objet, chaque voxel sur son chemin est accédé pour vérifier son contenu. Si le voxel comporte une surface, le produit vectoriel entre sa normale et la direction de la lumière établit l'intensité du pixel. Après plusieurs techniques testées pour déterminer quels voxels seront traversés par le rayon et dans quel ordre, notamment l'algorithme de Bresenham, la méthode Digital Differential Analyser (DDA) a été retenue comme étant celle produisant les résultats les plus exacts. Cette dernière consiste à calculer la distance à parcourir pour traverser chacun des trois axes et de sélectionner le plus près après chaque intersection (Figure 5).

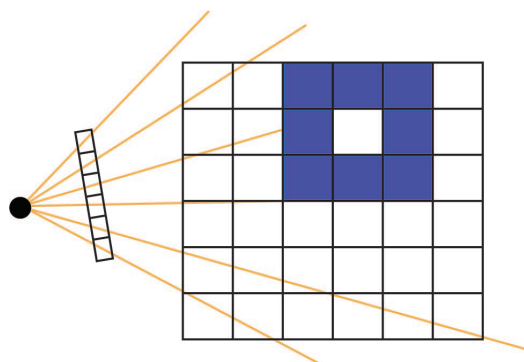


Figure 4. Lancement de rayons

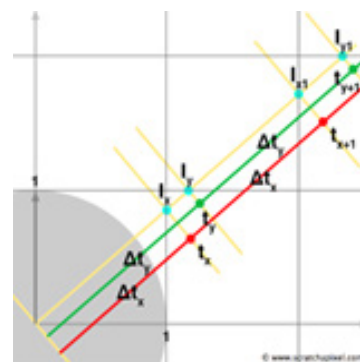


Figure 5. Algorithme DDA (Scratchapixel 2.0)

Initialement, la complexité de l'algorithme DDA limitait grandement le nombre d'images produites par seconde de l'engin de rendu, rendant son utilisation très peu agréable. Heureusement, la nature du lancer de rayon rend son exécution parallélisable, c'est-à-dire que les lancers de rayons sont indépendants les uns des autres. Pour faire usage de cette propriété, l'engin de rendu a été développé pour faire usage de plusieurs « threads » du processeur, permettant sa pleine utilisation et permettant à l'engin de produire un constant 60 images par seconde.



Figure 6. Rendu produit par le programme

Pour terminer, la mise en place d'une interaction avec la souris rend possible de tourner autour de l'objet, de s'en rapprocher ou de s'en éloigner et de se déplacer verticalement ou horizontalement. Le déplacement radial de la lumière est également possible. La figure 6 illustre une capture de l'engin de rendu.

La motivation derrière la création d'une vue de coupe est la possibilité d'explorer des valeurs numériques non accessibles par l'engin de rendu 3D. Ainsi, sur chaque axe, il est possible de se déplacer au travers des différentes couches pour visualiser le nombre de points dans chaque voxel ou la variation de surface.

Pour permettre une compréhension intuitive des données, celles-

ci sont présentées de manière graphique par intensité ou par teinte selon leurs valeurs numériques avec une légende associée. À l'origine, une différence graphique linéairement dépendante de la valeur numérique a été implantée. Par contre, les extrêmes rendaient difficile la distinction des valeurs plus faibles. Cette limitation a été corrigée par une différence graphique non linéaire : la différence de valeur des données plus faibles engendre une différence graphique plus grande que la même différence aux données plus élevées. Les figures 7 et 8 illustrent la distinction des deux techniques pour les mêmes valeurs.

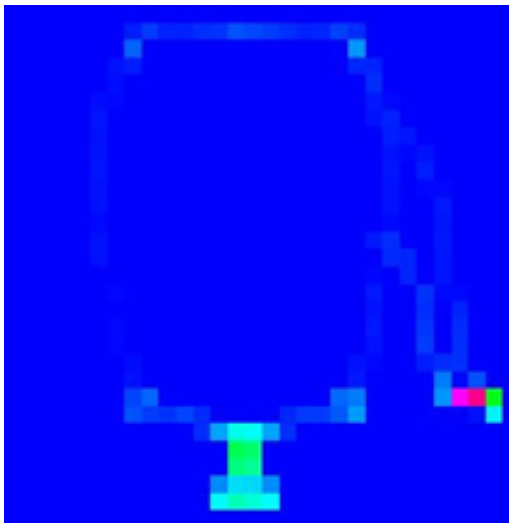


Figure 7. Vue decouche linéaire

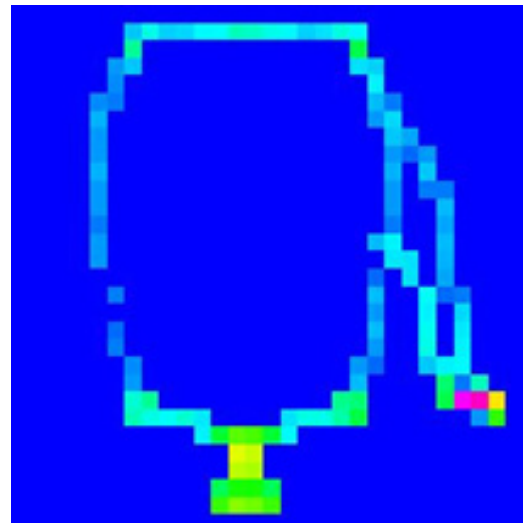


Figure 8. Vue de couche non linéaire

La génération de géométrie de triangles est produite à l'aide de l'algorithme de Marching Cubes. Ce dernier consiste à déplacer un cube dont chaque coin représente le centre d'un voxel. Il est ensuite déterminé si chacun des huit voxels est à l'intérieur ou à l'extérieur de la surface par le signe de la normale de la surface qui y est associée. Selon ces huit valeurs, la géométrie à l'intérieur du cube peut être déterminée. En fait, ces 256 combinaisons possibles représentent 15 motifs uniques d'orientations différentes (Figure 9). Une table est utilisée (Bourke) pour établir quel sera le motif et où placer les différents triangles.

Pour augmenter la fidélité du rendu, la valeur des huit coins peut inclure la distance de la surface en plus du signe. Ces valeurs sont ensuite utilisées pour interpoler la position des sommets des triangles. La géométrie est finalement affichée dans une fenêtre OpenGL dont la position de la caméra est synchronisée à la fenêtre de l'engin de rendu.

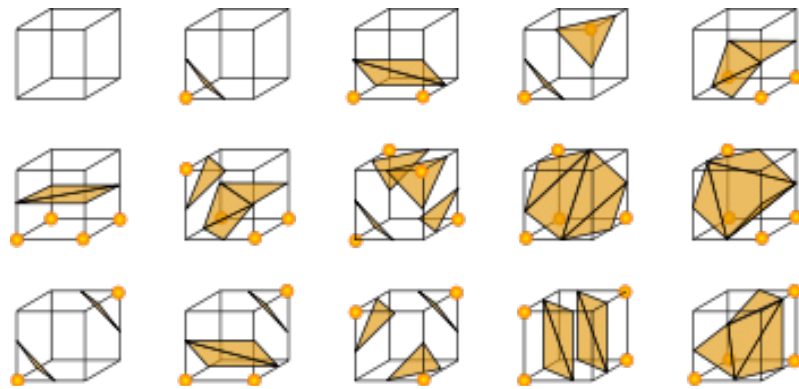


Figure 9. Quinze configurations possibles de l'algorithme Marching Cubes (Fisher)

La th  iere pr  sent  e pr  c  demment produit la g  om  trie de la figure 10. Le r  sultat n'est pas id  al, mais permet d'obtenir une approximation de l'objet.

L'interface d  velopp  e avec le module Qt permet de regrouper les diff  rentes composantes de construction et de visualisation du champ de vecteur dans un environnement, simplifiant l'utilisation du programme. L'interface est pr  sent  e dans la figure 11. Elle contient les   l  ments d  crits pr  c  demment en plus d'un histogramme affichant le nombre de voxels contenant un certain nombre de

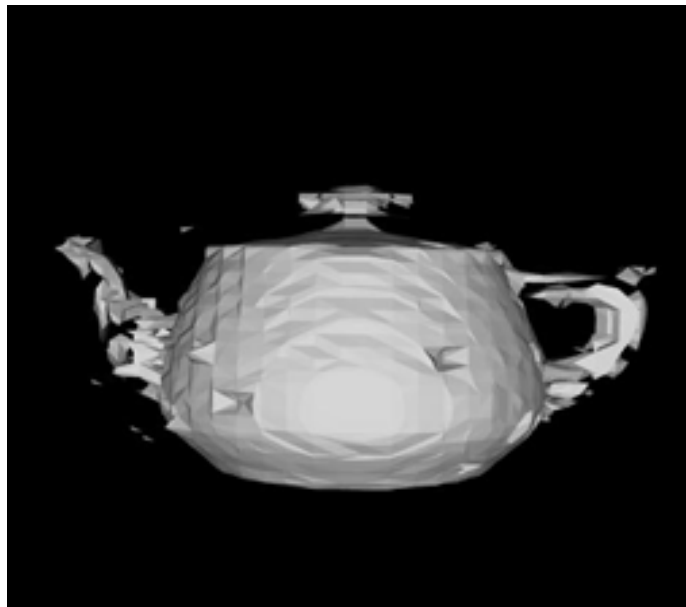


Figure 10. G  om  trie de a th  i  re produite par Marching Cubes

points. L'histogramme et l'engin de rendu peuvent également être ouverts dans de nouvelles fenêtres. La fenêtre principale peut être redimensionnée et le contenu s'ajustera automatiquement.

En plus, un système d'enregistrement et de téléversement de fichier permet d'enregistrer le contenu du champ de vecteur calculé.

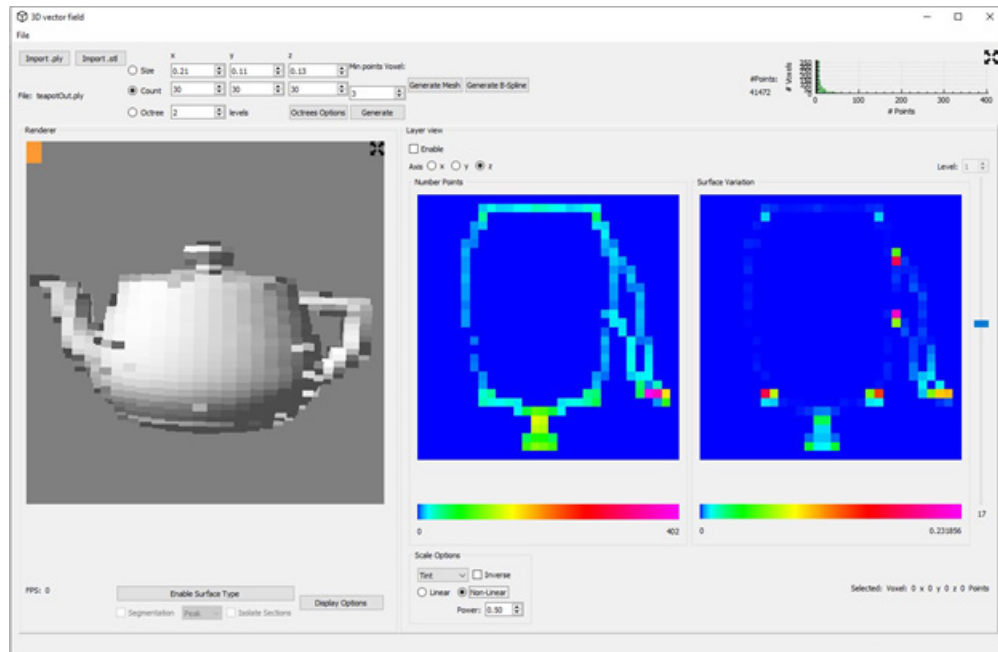










Figure 11. Interface utilisateur du programme

Projet de la doctorante

Le projet proposé par l'étudiante au doctorat s'ajoute au concept de champ de vecteurs décrit jusqu'à présent en trois parties. Premièrement, le type de surface de chaque voxel est établi. En second lieu, une courbe est générée pour chaque section de type semblable. Troisièmement, les différentes courbes sont fusionnées pour produire une surface finale représentant l'objet.

Pour débiter, pour chaque voxel comportant une surface, un type de surface y est associé. Pour ce faire, chacun des 26 voxels voisins en 3 dimensions est observé. La différence entre le point le plus près du centre du voxel concerné et celui du voxel observé est multipliée par la normale du voxel concerné. Cette valeur est considérée soit négative,

soit nulle (en dedans d'un seuil), soit positive (Figure 12). Après avoir effectué l'opération sur chacun des voisins, un type de surface est attribué selon le signe des voisins (Figure 13).

		K		
		+	0	-
H	-	Peak 	Ridge 	Saddle Ridge 
	0	none	Flat 	Minimal surface 
	+	Pit 	Valley 	Saddle Valley 

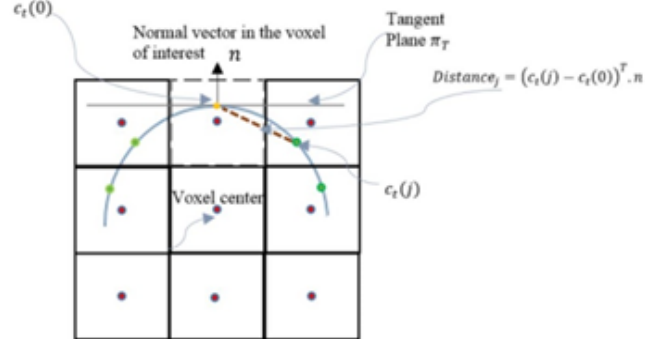


Figure 13. Huit types de surface (Eskandari)

Figure 12. Calcul du type de surface (Eskandari)

Une fois chacun des voxels non vides parcourus et assignés à un type de surface, les voxels connectés ayant le même type sont regroupés.

L'affichage des données reliées aux types de surface est visible dans l'engin de rendu par l'activation d'une option. Les types de surface pour chaque voxel (Figure 14) et les groupes de même type (Figure 15)

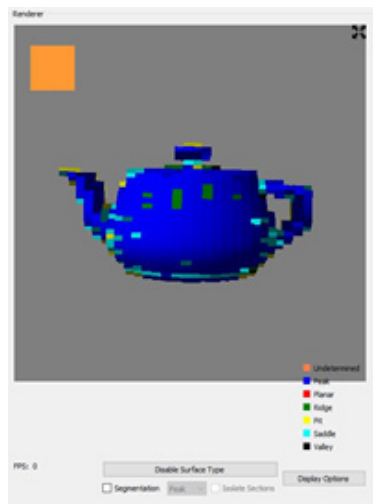


Figure 14. Types de surface de la théière



Figure 15. Groupes de surface "peak" de la théière

peuvent être visualisés.

Une fois les groupes de même type identifiés, une courbe est produite pour chacun d'entre eux. Pour ce faire, une fonction est définie sur le volume de l'objet pour représenter une surface implicite. La technique est basée sur le travail de Mohammad Rouhani, Angel D. Sappa et Edmond Boyer et consiste à optimiser des points de contrôle par moindre carrée pour obtenir une surface où le croisement du zéro de la fonction représente une courbe lisse se rapprochant des points d'entrée, c'est-à-dire une surface implicite.

La contribution du stagiaire consiste à la définition d'un algorithme de fusion des différentes surfaces implicites dans le but d'obtenir une courbe continue représentant l'objet complet. La technique développée par l'étudiant est la suivante:

Un rayon d'influence des points est d'abord défini par l'utilisateur dans l'interface. Pour chaque surface implicite, la valeur à chaque intervalle d'affichage est multipliée par la fonction gaussienne dépendante du rayon défini et de la distance au point définissant la surface la plus près.

$$e^{-\frac{x^2}{0.25*r^2}}$$

La figure 16 illustre l'effet des points sur la fonction de la surface implicite. Puis, les différentes surfaces sont additionnées et la surface finale est produite avec l'algorithme de Marching Cubes présenté à la sous-section précédente. La figure 17 illustre le résultat de la technique développée. Une amélioration est sans aucun doute nécessaire, mais la méthode proposée représente un point de départ possible pour une solution plus robuste.

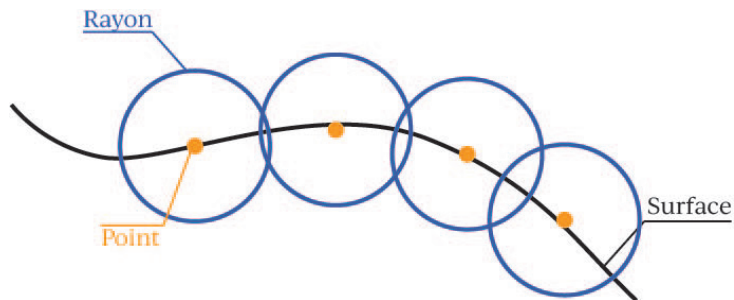


Figure 16. Effet du rayon d'influence de la fusion

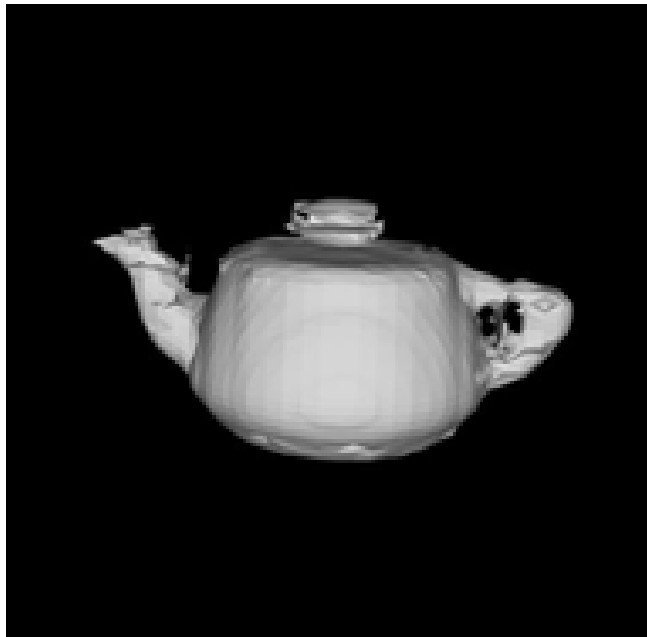


Figure 17. Résultat de la théière fusionnée

Attentes

Malgré le fait que Christophe soit une personne de nature ambitieuse, le travail effectué durant le stage semble colossal en rétrospective. Le stagiaire se lançait dans le projet en prévoyant contribuer légèrement et apprendre faiblement. Ces attentes furent fracassées dès la première semaine par l'immersion immédiate dans un domaine dans lequel un apprentissage considérable est nécessaire. Les responsabilités qui lui furent confiées étaient également largement supérieures à celles attendues à priori, une surprise

bienvenue.

À force d'avancer le projet, les défis devinrent plus grands et les attentes de productivité basées sur la progression des premières semaines ne semblaient plus à portée de main. Elles l'étaient. Les défis présentaient pour l'étudiant une motivation pour maîtriser les concepts plus complexes, notamment les courbes.

En somme, l'environnement, quoique virtuel, et le projet ont dépassé largement les attentes initiales du stagiaire.

Développement et renforcement des compétences

À travers le stage, l'étudiant a pu renforcer des compétences acquises en classe et parfois même aller bien au-delà de ce qui est présenté en cours. Les habiletés cultivées se distinguent en deux majeures catégories : les habiletés techniques et les habiletés de gestion.

Technique

Il est inévitable d'acquérir de nouvelles compétences techniques durant un stage de développement, mais l'apprentissage technique du stagiaire tout au long est considérable, autant d'un côté de programmation que d'un côté de recherche.

Contrairement à ce qui est réalisable dans le cadre d'un cours, le projet accompli est d'une ampleur grandement supérieure. Ainsi, en bâtissant le code de plusieurs milliers de lignes, des techniques d'organisation et de structuration ont été acquises pour faciliter la lisibilité, l'efficacité et la possibilité d'évolution du projet. Un second aspect que le stagiaire n'aurait pu étudier en classe est l'optimisation de l'engin de rendu pour atteindre une fréquence d'images temps réel agréable. De ce fait, plusieurs principes ont été explorés pour accélérer l'exécution du programme.

La possibilité de continuer leurs études en recherche n'est pas une option considérée par beaucoup d'étudiants en génie informatique. Par conséquent, les techniques utiles à la recherche ne sont malheureusement pas présentées en cours. Le stage représentait donc pour l'étudiant une opportunité de développer ses habiletés en recherche académique. Notamment, des méthodes de recherche et de lecture efficace d'articles scientifiques ont été développées. Cet apprentissage a déclenché le désir d'effectuer des études graduées.

Finalement, le sujet du projet, la visualisation de données 3D, a convaincu le stagiaire à s'inscrire au cours de Vision numérique donnée par le superviseur de stage à l'automne.

Gestion

La nature et le contexte du projet ont encouragé une croissance dans le modèle de gestion de l'étudiant. D'une part, des outils ont permis de faciliter l'assimilation de concepts théoriques, ont facilité la communication et ont encouragé la planification à la programmation. D'autre part, le contexte de travail a encouragé une gestion du temps réfléchie en raison de l'autonomie accordée.

Les notes rédigées par l'étudiant au cours du projet se sont révélées être un outil grandement utile. La raison évidente est la clarification des idées, la formulation d'idées sur papier engendrant une meilleure assimilation de concepts techniques complexes que la simple lecture. Cette tendance à étaler ses idées sur papier a convaincu l'étudiant à passer plus de temps à écrire qu'à programmer, transformant ainsi le travail de programmation en une simple tâche d'implantation basée sur une structure clairement définie. Cette méthode a sans aucun doute évité énormément de temps de « débogage » causé par l'oubli d'ajouter une fonction liée à une autre par exemple. De plus, les notes ont représenté une référence précieuse de définitions arbitraires utilisées durant le projet. Finalement, celles-ci étant disponibles sur le nuage à tout moment à l'aide d'un lien, le superviseur de stage et la doctorante ont pu accéder aux réflexions et à la progression du stagiaire à tout moment, encourageant sans aucun doute la confiance accordée à ce dernier.

Cette confiance est possiblement en partie source de l'autonomie qui a été allouée à l'étudiant, contexte qu'il considère à la base de son efficacité. Étant responsable de sa propre gestion de temps, il a privilégié la déconnexion du projet lorsque sa concentration n'était pas optimale, évitant de longues périodes assises devant l'ordinateur sans savoir par où commencer.

Tout compte fait, il est difficile de ne pas comparer avec la série de cours de Design, visant à développer des compétences techniques et de gestion. La série de cours propose une liste d'outils de travail précise sans encourager leur adaptation aux personnes qui les utilisent, une lacune flagrante après l'expérience du stagiaire.

En somme, le stagiaire estime que la liberté qui lui a été accordée est source de l'apprentissage technique énorme qu'il considère avoir acquis et du développement de son modèle de gestion de projet. La responsabilité de développer un outil ayant une utilisation réelle est une immense motivation à apprendre et s'appliquer dans le travail. En revanche, le contexte de travail à distance a limité les interactions sociales avec les autres membres du laboratoire, seul point négatif du stage identifié par l'étudiant. Un avenir en recherche est envisagé par le stagiaire.

Conclusion

En conclusion, le projet développé se divise en trois parties. D'une part, une structure de calcul de champ de vecteur à partir de nuages de points capturés à l'aide d'un scanneur 3D est construite. Puis, des fonctions de visualisation permettent d'observer et de se déplacer autour de l'objet représenté par le champ de vecteur et rendre possible d'accéder aux valeurs numériques contenues dans chaque voxel. Finalement, le projet proposé par l'étudiante au doctorat, consistant à la séparation en types de surface et le calcul de courbes pour chaque groupe connecté du même type, est implanté. Ces surfaces sont ensuite fusionnées par une fonction proposée par le stagiaire.

À la fin du stage, l'étudiant remet un logiciel utilisable pour la visualisation de données de champs de vecteurs. Du travail reste à être accompli en lien avec le projet de la doctorante, notamment dans l'amélioration de la technique de fusion.

L'étudiant quitte le projet avec une compréhension beaucoup plus solide de l'infographie, un modèle de gestion de projet robuste et une expérience de recherche inestimable. Si bien qu'il considère grandement un avenir d'études en recherche dans le domaine du visuel.

Bibliographie

Bourke, Paul. Polygonising a Scalar Field. Mai 1994, paulbourke.net/geometry/polygonise/. Accédé le 24 Aout 2020.

Eskandari, Mana, and Denis Laurendeau. "Covariance Based Differential Geometry Segmentation Techniques for Surface Representation Using Vector Field Framework." Université Laval.

Fisher, Matthew. "Marching Cubes." Matt's Webcorner, 2014, graphics.stanford.edu/~mdfisher/MarchingCubes.html. Accédé le 24 Aout 2020.

"Introduction to Acceleration Structures." Scratchapixel 2.0, 8 Oct. 2015, www.scratchapixel.com/lessons/advanced-rendering/introduction-acceleration-structure/grid. Accédé le 24 Aout 2020.

Mohammad Rouhani, Angel D. Sappa, Edmond Boyer. "Implicit B-Spline Surface Reconstruction. " IEEE Transactions on Image Processing, Institute of Electrical and Electronics Engineers, 2015, 24 (1), pp.22 - 32.

"Publications." Laboratoire De Vision Et Systèmes numériques, vision.gel.ulaval.ca/fr/publications/index.php. Accédé le 24 Aout 2020.

TUBIC, DRAGAN. "ON SURFACE REPRESENTATION IN 3D MODELLING." UNIVERSITE LAVAL, 2006.

Type Network Staff. "Ivy Crops up at Type Network." Type Network, 24 Jan. 2018, ivy.typenetwork.com/news/article/ivy-crops-up-at-type-network. Accédé le 24 Aout 2020.

À propos de la police

Ivy Journal est une police conçue par le designer Jan Maack originaire du Danemark. La police avec sérif est inspirée des polices classique Bembo, Baskerville et Times New Roman pour produire des lettres avec une grande x-hauteur, un contraste modéré et un axe légèrement incliné. Ivy Journal est construite pour permettre une lecture facile et résulte en une police multifonctionnelle et élégante, suggérant une alternative aux polices sur-utilisées. (Type Network Staff)