



**Benoît Quero – ingénieur
Imagerie Numérique
2018-2019**

**Tuteur universitaire :
Pierre Maurel
Enseignant chercheur**



**IRT b<>com
1219 Avenue des Champs Blancs,
35510 Cesson-Sévigné
02 56 35 88 00**

**Tuteur entreprise :
Antonin Gilles
Ingénieur Recherche et Développement**

Implémentation d'une solution de communication client/serveur pour transmission adaptative de vidéos holographiques

Année universitaire 2018-2019



Remerciements

Je tiens à remercier toutes les personnes qui ont contribué à la réussite de mon stage chez b<>com.

Je voudrais tout d'abord remercier mon tuteur de stage Antonin Gilles, ingénieur Recherche et Développement au sein du laboratoire Nouveaux Formats Médias, pour la proposition de stage sur le thème unique de l'holographie. Il a su me donner sa confiance et une indépendance dans les tâches qu'il m'avait confiées et a su rester à l'écoute.

Je tiens également à remercier Anas El Rhammad et Patrick Gioia, membres de l'équipe holographie, qui ont su m'aider dans la réalisation de mes différents objectifs et ainsi me faire partager leur expertise.

Je remercie Jean-Yves Aubié, responsable du laboratoire Nouveaux Formats Médias, pour m'avoir accordé sa confiance pour la réalisation de ce stage. Enfin, je souhaite remercier toute l'équipe du laboratoire pour leur accueil et leur bonne humeur pendant ces quatre mois passés en leur compagnie.

Résumé en français

Ce stage s'est déroulé à b<>com sur la thématique de l'holographie. L'holographie est une technologie de visualisation 3D permettant aux spectateurs d'observer les effets de relief et de profondeur d'une scène en trois dimensions. L'holographie est un procédé optique utilisant les propriétés de diffraction d'un faisceau lumineux. Aujourd'hui, des hologrammes peuvent être calculés numériquement, ce qui permet d'afficher des scènes synthétiques ou réelles très facilement.

Ma mission lors de ce stage était d'implémenter une solution de compression et transmission de données holographiques. Cette solution a pour but de réduire la quantité de données à stocker et d'accélérer la vitesse des calculs afin d'obtenir une transmission client/serveur d'hologrammes en temps réel.

Cette solution a été développée en plusieurs étapes. La première était la reconstruction d'un hologramme à partir des données d'entrées. La deuxième était la compression de ces données grâce à un codage entropique. Enfin, la dernière étape consistait à implémenter les algorithmes permettant leur création.

English summary

I completed my internship at b<>com where I work on holography. Holography is a 3D visualization technology which gives the spectators the depth effects of a 3D scene. Optical holography uses the illuminating light beam diffraction properties. Nowadays, holograms can be generated by computer calculation. The Computer-Generated Holograms can easily display synthetic or existing scenes.

My missions during this internship was to implement a solution of compression and transmission of holographic data. The main goals of this solution were to reduce the amount of data and speed up the calculations in order to obtain a real-time client/server transmission of holograms.

This solution has been developed in several steps. The first one was to reconstruct the hologram from the data. The second one was the compression of this data with a entropy encoding. Finally, the last step was the implementation of the algorithms which create them.

Table des matières

Remerciements	1
Résumé en français	2
English summary	2
Introduction	4
I. Présentation de b<>com et contexte	5
A. Présentation de b<>com	5
B. Contexte	6
II. Solutions proposées	8
A. Méthodes et algorithmes utilisés	8
1. Décomposition de l'hologramme en ondelettes de Gabor	8
2. Matching Pursuit	9
B. Outils utilisés	10
C. Travail réalisé	10
1. Reconstruction de l'hologramme	11
2. Encodage/décodage du bitstream	12
3. Génération des atomes de Gabor d'un hologramme	13
III. Résultats	15
Conclusion	19
Bibliographie	20
Annexes	21

Introduction

Le stage que j'ai réalisé pour ma deuxième année du cycle ingénieur à l'École Supérieure d'Ingénieurs de Rennes s'est déroulé au sein du laboratoire Nouveaux Formats Médias de b<>com, du 3 juin au 20 septembre 2019.

La recherche est une voie qui m'intéresse et qui attise ma curiosité. B<>com axe ses recherches sur des sujets prometteurs et ambitieux, comme l'holographie. L'offre de stage sur l'implémentation d'une solution de transmission de vidéos holographiques était pour moi une bonne occasion d'intégrer un institut de recherche technologique pour quelques mois, et ainsi découvrir son fonctionnement.

Mes objectifs personnels sont multiples. Tout d'abord, mettre à profit les connaissances que j'ai acquises durant ma deuxième année de formation à l'ESIR et durant les années antérieures. Ensuite, développer de nouvelles compétences et connaissances en développement, sur un thème prometteur et original. Enfin, découvrir un peu plus le monde du travail et la vie en entreprise, dans un domaine qui me plaît.

Ma mission durant ces quatre mois était d'implémenter une solution de communication client/serveur pour une transmission adaptative de vidéos holographiques. La première étape était la compréhension de la thématique, du sujet de stage et de ses enjeux. Ensuite, la majeure partie du stage consistait à développer cette solution préalablement conçue par les chercheurs et doctorants du laboratoire.

Dans un premier temps, nous présenterons l'entreprise et le contexte de ce stage, puis nous étudierons le travail réalisé ainsi que les résultats obtenus.

I. Présentation de b<>com et contexte

A. Présentation de b<>com

b<>com est un Institut de Recherche Technologique qui a pour objectif d'accélérer l'innovation et créer des technologies, à travers la recherche, pour les entreprises souhaitant développer leur compétitivité grâce au numérique.

Créé en 2012 et comptant aujourd'hui environ 300 employés répartis sur les sites de Cesson-Sévigné, Brest, Lannion et Paris, b<>com axe ses recherches sur la réalité virtuelle, la réalité augmentée, la vidéo et l'audio immersif, l'intelligence artificielle, la cybersécurité, la 5G, l'internet des objets ou encore l'e-santé.

B<>com produit des brevets dans les domaines des médias, des réseaux ultra haut débit et de l'e-santé et propose des conseils et contrats de recherche aux entreprises souhaitant innover.

Le laboratoire Nouveaux Formats Médias a pour mission d'augmenter le réalisme et l'impact des contenus son et image (spatialisation du son, Ultra Haute Définition, HDR, holographie) dans le but d'améliorer les sensations d'immersion de l'utilisateur ou du spectateur au cinéma, à la télévision, dans un jeu... Le laboratoire développe donc des outils et des solutions, comme des plug-ins, pour les professionnels du secteur afin de les aider à produire et utiliser ces nouveaux médias.



Logo de b<>com

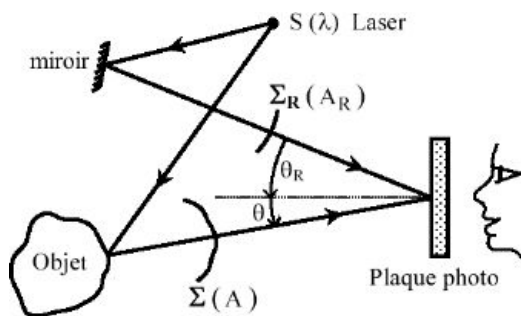


*Bâtiment de b<>com à
Cesson-Sévigné*

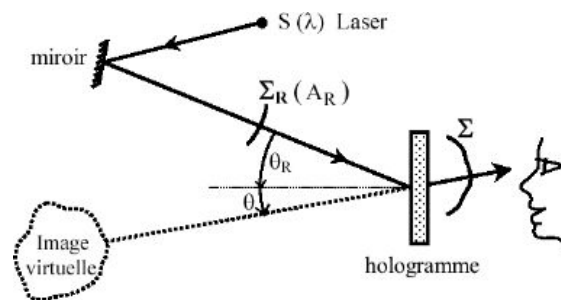
B. Contexte

L'holographie est une technologie permettant une immersion dans des scènes 3D, réelles ou synthétiques. La stéréoscopie est la méthode la plus utilisée de nos jours mais présente des inconvénients comme le fait de porter des lunettes, qu'elle ne crée pas la parallaxe de mouvement et provoque un conflit accommodation-vergence qui peut amener à des maux de tête. L'holographie permet de pallier à ces inconvénients.

Les hologrammes ont tout d'abord été acquis de manière optique en enregistrant sur une surface photosensible les franges d'interférence entre l'onde lumineuse d'un objet et une onde référence, comme un laser. Ce sont ces franges d'interférence qui constituent les hologrammes. L'image de l'objet est ensuite restituée en éclairant l'hologramme de la surface photosensible avec un laser qui, par diffraction, va recréer les reliefs de cet objet. Cette acquisition optique présente beaucoup de contraintes, comme la stabilité du système d'acquisition, le développement de la surface photosensible et l'éclairage uniquement par laser, et non par lumière naturelle.

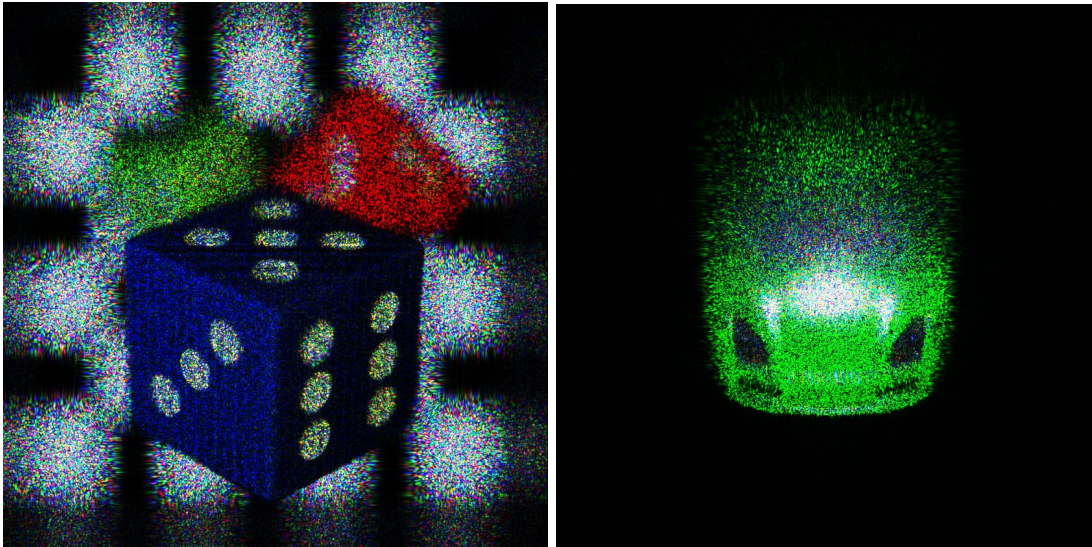


Ecriture optique d'un hologramme



Lecture optique d'un hologramme

Ces contraintes peuvent aujourd'hui être évitées grâce à la génération numérique d'hologramme. Il est donc possible de produire des hologrammes de scènes réelles, à partir de données multi-vues acquises par exemple grâce à une matrice de caméras. Des hologrammes de scènes synthétiques 3D peuvent aussi être calculés en faisant une somme des ondes lumineuses émises par les primitives de la scène. Les rendus de ces scènes peuvent être très réalistes car il est possible d'intégrer au calcul de l'hologramme l'illumination de la scène avec la technique de l'ombrage de Phong par exemple.



*Exemples de reconstitution d'objets 3D à partir
d'hologrammes calculés numériquement*

Le principal problème de ces méthodes est la quantité de données à traiter et donc le temps de calcul nécessaire pour générer une scène réaliste avec une résolution suffisante. Différentes méthodes ont été conçues à b<>com par les chercheurs et doctorants pour pouvoir diminuer le temps de calcul.

II. Solutions proposées

A. Méthodes et algorithmes utilisés

1. Décomposition de l'hologramme en ondelettes de Gabor

Les hologrammes numériques contiennent beaucoup de données correspondant aux différents points de vue possibles de l'observateur. Ces données sont représentées par des nombres complexes qui contiennent les informations d'amplitude et de phase des ondes lumineuses des différents objets de la scène. En terme de compression, les méthodes de codage d'image comme HEVC se sont montrées peu efficaces car les hologrammes sont des superpositions de composants de très hautes fréquences.

La solution la plus adaptée est donc une décomposition basée sur les rayons lumineux et non basée sur les pixels. Comme elle permet d'obtenir un bon compromis temps/fréquence, c'est la décomposition en ondelettes de Gabor qui a été choisie.

Le dictionnaire de Gabor est obtenu en appliquant des translations, des rotations, et des dilatations à l'ondelette principale. Le dictionnaire est donné dans sa forme 2D par :

$$g_{(l,p)}[u, v] = \frac{1}{\sqrt{s[l]}} e^{-\frac{1}{\Delta^2 s[l]^2} (a^2 + b^2)} e^{i\pi \frac{a}{\Delta s[l]}} \quad (1)$$

avec :

$$\begin{cases} a = u \cos(\theta[p]) + v \sin(\theta[p]) \\ b = v \cos(\theta[p]) - u \sin(\theta[p]) \end{cases}$$

(u, v) sont les paramètres de translation 2D de l'ondelette, tels que $-\frac{M_x}{2} \leq u < \frac{M_x}{2}$ et $-\frac{M_y}{2} \leq v < \frac{M_y}{2}$, M_x et M_y sont les résolutions des

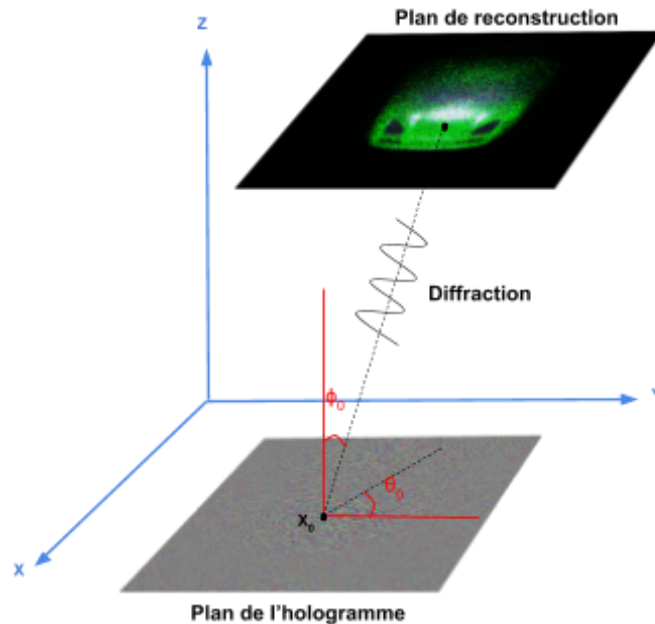
ondelettes. Le paramètre de rotation est donné par $\theta[p] = \frac{2p\pi}{P}$ avec $0 \leq p < P$. Le

paramètre de dilatation est donné par $s[l] = \frac{f_{max}}{2^{1+0.4l}}$ avec $f_{max} = \frac{1}{2\Delta}$ et $1 \leq l \leq S$, Δ la taille d'un pixel.

En calculant le produit scalaire entre l'hologramme et le dictionnaire de Gabor, on obtient un ensemble de coefficients complexes c_n définissant l'hologramme (un par pixel de l'hologramme et par indice de dilatation et rotation). Celui-ci est donc maintenant composé de N atomes, dont le $n^{ième}$ élément est donné par :

$$a_n = (c_n, l_n, p_n, t_n),$$

représentant chacun un rayon diffracté depuis une position t vers une direction $(\phi_n, \theta_n) = (\phi(s[l_n]), \theta[p_n])$.



Diffraction d'un rayon lumineux en coordonnées sphériques

2. Matching Pursuit

Le Matching Pursuit est un algorithme très utilisé en traitement du signal qui va nous permettre de diminuer le nombre d'atomes. Cet algorithme va sélectionner, à partir d'une décomposition redondante, les atomes les plus corrélés avec l'hologramme. Cela se traduit mathématiquement par calculer le produit scalaire entre l'hologramme et chaque atome et sélectionner celui donnant le plus grand produit. À chaque itération, l'hologramme est mis à jour en lui soustrayant l'atome le plus corrélé. L'algorithme s'arrête quand la norme de l'hologramme passe sous un certain seuil.

L'algorithme nous donne une approximation linéaire de l'hologramme et celui-ci peut donc être considéré comme une somme :

$$H = \sum_{k=0}^K c_k g_{l_k, p_k} \quad (2)$$

Avec K le nombre d'atomes, c_k le coefficient complexe pour l'atome k , g_{l_k, p_k} la valeur complexe de l'ondelette pour les indices de dilatation l_k et de rotation p_k . Si l'hologramme est en couleur, cette somme est à faire pour chaque composante RGB. On obtient donc un nombre d'atomes réduit composé des atomes les plus importants.

B. Outils utilisés

Ce projet est principalement développé en C++ et en CUDA sous Visual Studio. CUDA est une technologie développée par Nvidia permettant d'exécuter des calculs sur carte graphique (GPU). L'avantage est que les calculs peuvent être effectués en parallèle, ce qui est très pratique pour la plupart des calculs matriciels. Les *kernels* sont les fonctions appelés par le CPU permettant d'effectuer les calculs sur le GPU. Le projet Visual Studio qui m'a été fourni comprenait déjà notamment les fonctions d'initialisation et de reconstruction de l'hologramme et celles simulant la propagation de la lumière.

C. Travail réalisé

Ce rapport présente les solutions développées dans l'ordre chronologique. La première étape du développement concernait la reconstruction de l'hologramme à partir de ses atomes. La deuxième étape traitait de l'encodage et du décodage de ces données, et la troisième partie concernait la génération des atomes, en fonction des données de l'hologramme à transmettre.

1. Reconstruction de l'hologramme

La première partie consiste à reconstruire l'hologramme à partir des atomes de Gabor. Nous avons donc en entrée les données suivantes :

- Positions 2D de l'ondelette de Gabor
- Indices de rotation et de dilatation de l'ondelette
- Coefficients complexes de la composante rouge
- Coefficients complexes de la composante verte
- Coefficients complexes de la composante bleue

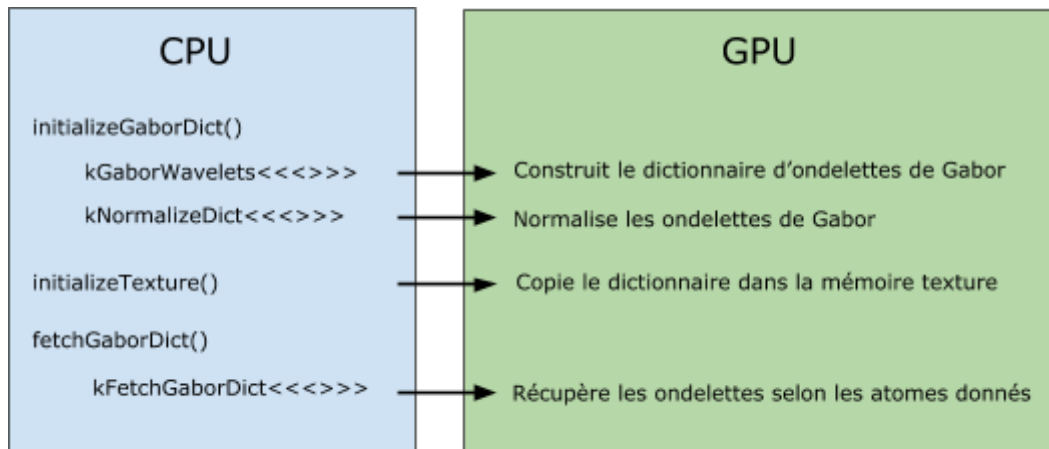
Les positions 2D et les indices de rotation et de dilatation nous permettent de retrouver les ondelettes du dictionnaire de Gabor associés à notre hologramme. Les coefficients complexes de chaque composante de couleur sont appliqués aux ondelettes lors de la reconstruction de l'hologramme (équation (2)).

Pour pouvoir utiliser les ondelettes de Gabor, il faut tout d'abord créer son dictionnaire. Celui-ci est indépendant de l'hologramme. CUDA est utile ici car il permet d'effectuer les calculs de l'équation (1), de normaliser les ondelettes et de stocker ces valeurs dans une texture. La mémoire texture est un type de mémoire GPU accessible en lecture-seule qui est très efficace quand la lecture se fait à des adresses proches les unes des autres.

Une fois la texture initialisée avec les valeurs complexes du dictionnaire de Gabor, on utilise les données de positions et d'indices pour accéder aux ondelettes voulues. Tout comme la fonction précédente, cette fonction de recherche d'ondelette (appelée *fetch*) est implémentée dans un kernel CUDA. Elle utilise la mémoire partagée, une mémoire cache cent fois plus rapide que la mémoire globale du GPU, pour accéder plus rapidement aux données de positions, indices et coefficients. En plus d'accéder aux ondelettes, ce kernel les multiplie par les coefficients, puis somme le tout afin de retrouver la formule (2).

Afin d'accélérer les calculs, la fonction de *fetch* va récupérer les atomes par blocs. Il n'est en fait pas utile de prendre en compte tous les atomes pour le rendu de chaque pixel car certains n'ont pas d'influence sur les pixels les plus éloignés. On choisit donc de diviser notre hologramme par blocs (4, 8, 16...) en ne sélectionnant que les atomes ayant un rayon d'influence inclus dans le bloc en question.

Si l'hologramme est en couleur, la fonction de *fetch* est à appeler pour chaque composante RGB. On obtient donc trois tableaux de valeurs complexes correspondant aux champs lumineux rouge, vert et bleu. La scène 3D peut maintenant être reconstruite.



Structure simplifiée de la reconstruction de l'hologramme

2. Encodage/décodage du bitstream

Le but de ce projet étant de développer des solutions réduisant la quantité de données d'un hologramme, il est donc normal que le bitstream, permettant de transmettre les données d'atomes, soit le plus compact possible. Les données de positions, les indices et les coefficients complexes sont donc stockés séparément dans des fichiers binaires.

Les indices sont encodés grâce au Context-Adaptive Binary Arithmetic Coding (CABAC). CABAC est un algorithme de codage entropique, sans perte, utilisé par les normes de compression H.264/AVC et HEVC. Il fonctionne uniquement avec des symboles binaires. Pour chaque bit, il va sélectionner un des trois modèles de probabilités proposés et va estimer sa probabilité en fonction de ses voisins. Un codage arithmétique est ensuite appliqué à ces données.

Comme CABAC ne fonctionne qu'avec des symboles binaires, il faut binariser les données en utilisant un codage unaire : chaque entier n est représenté par n occurrences de "1", suivi d'un "0" faisant office de délimiteur.

Les positions 2D sont tout d'abord passées de deux à une dimension, puis encodées de façon différentielle, c'est-à-dire en encodant la différence entre deux positions successives. Cela permet de diminuer le nombre d'occurrences de "1" du codage unaire. Après la binarisation, les positions sont encodées grâce à CABAC.

Contrairement aux indices et aux positions qui sont des nombres entiers, les valeurs réelles et imaginaires des coefficients complexes RGB sont des flottants. Il faut donc au préalable les quantifier sur 6 bits, soit 64 niveaux, en utilisant un

quantificateur uniforme scalaire. Pour cela, on crée des vecteurs contenant les niveaux de décision de chaque composante des coefficients complexes, parties réelles et imaginaires séparées, puis on encode avec CABAC les indices de ces vecteurs correspondant aux coefficients à quantifier.

3. Génération des atomes de Gabor d'un hologramme

La génération des atomes de Gabor d'un hologramme se fait en trois étapes :

1^{ère} étape : création du dictionnaire de Gabor

Pour pouvoir obtenir les atomes correspondant à notre hologramme, il faut créer le dictionnaire des ondelettes de Gabor. Pour que la reconstruction fonctionne, il faut bien évidemment que le dictionnaire de la génération des atomes et celui de la reconstruction de l'hologramme soient identiques. On utilisera donc les mêmes fonctions et kernels CUDA que dans la partie *Reconstitution de l'hologramme*.

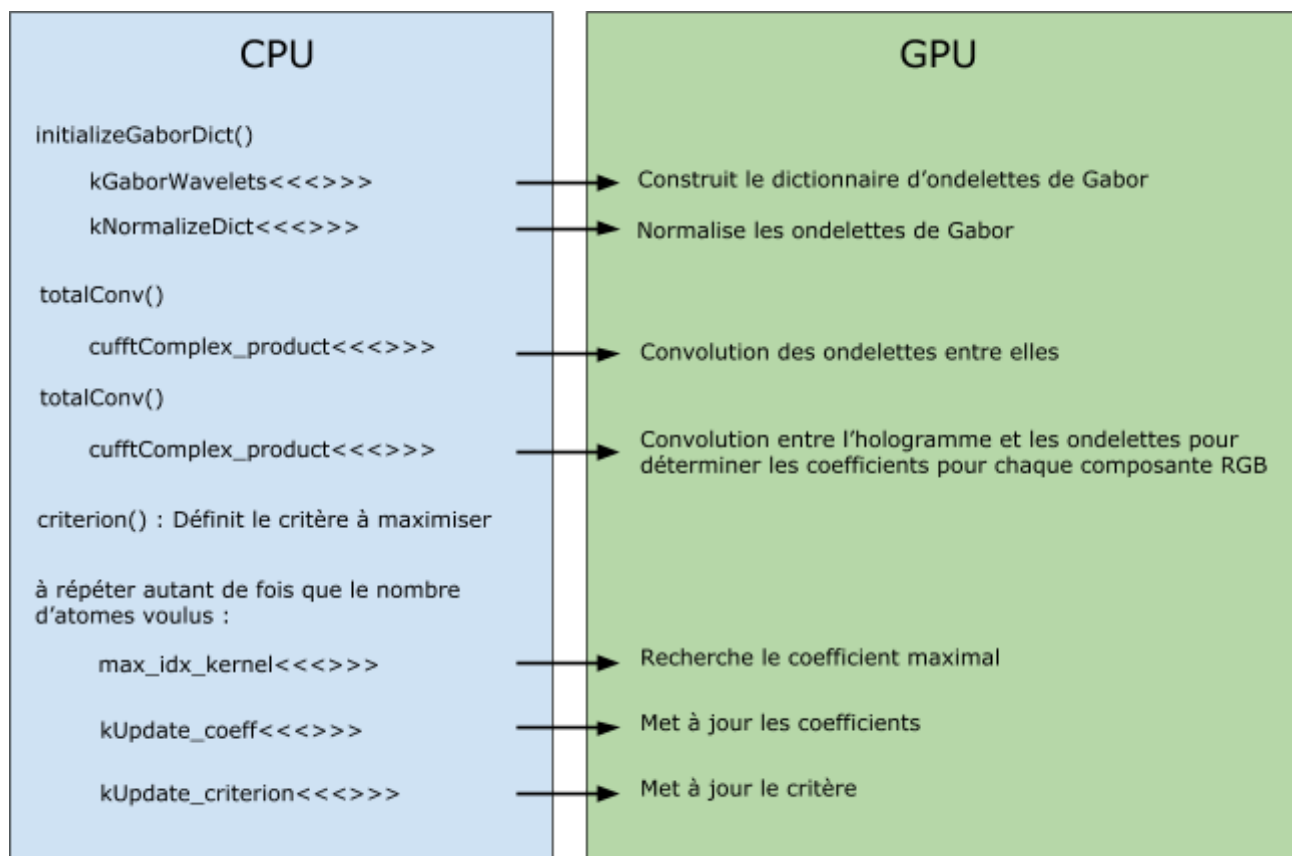
2^{ème} étape : sélection des atomes correspondant à l'hologramme

Pour obtenir les coefficients complexes RGB correspondant à l'hologramme voulu, il faut normalement calculer le produit scalaire entre cet hologramme et le dictionnaire de Gabor. Dans notre cas, il est possible d'utiliser un produit de convolution pour accélérer les calculs. Grâce à la bibliothèque CUDA cuFFT, il nous suffit d'appliquer une transformée de Fourier sur l'hologramme et le dictionnaire, de les multiplier dans un kernel, puis d'effectuer une transformée inverse pour obtenir le résultat.

3^{ème} étape : réduction du nombre d'atomes avec le Matching Pursuit

Le Matching Pursuit est un algorithme qui nous permet de ne garder que les atomes les plus corrélés avec l'hologramme. L'algorithme utilisé est le Vectorial Matching Pursuit (voir annexe). Comme le calcul du produit scalaire (ici une convolution) entre l'hologramme et le dictionnaire et la recherche du plus grand produit pour trouver l'atome le plus corrélé sont des opérations très coûteuses en temps, la formule de mise à jour du Vectorial Matching Pursuit peut être modifiée en ne faisant la mise à jour de l'hologramme que si les produits scalaires des ondelettes de Gabor entre elles sont non nuls. On calcule donc un ensemble de valeurs résultantes de ces produits scalaires (ici des convolutions) qui vont permettre un coût des opérations plus faible.

Les calculs de convolutions, de recherche des maximums et de mise à jour de l'hologramme sont effectués sur le GPU. Les fonctions de convolution et de Matching Pursuit ont déjà été implémentées auparavant par le laboratoire.



Structure simplifiée de la génération des atomes

III. Résultats

Les solutions proposées ont été testées sur un hologramme représentant une voiture verte. Il est défini par deux fichiers *bitmap*, un pour l'amplitude et un pour la phase.

Résolution de l'hologramme	1024×1024
Paramètre de rotation	$\theta[p] = \frac{2p\pi}{P}$ avec $P = 8$
Paramètre de dilatation	$s[l] = \frac{f_{max}}{2^{1+0.4l}}, 1 \leq l \leq S, f_{max} = \frac{1}{2\Delta}$ avec $S = 8$
Taille des pixels	$\Delta = 0,4 \mu m$
Longueur d'onde rouge	$\lambda_R = 640 nm$
Longueur d'onde verte	$\lambda_G = 532 nm$
Longueur d'onde bleue	$\lambda_B = 473 nm$
Résolution des ondelettes de Gabor	41×41
Taille des blocs utilisés pour la reconstruction	128×128

Tableau 1 : Paramètres utilisés pour les tests

À partir de ces deux fichiers, le programme va générer les atomes, encoder les données dans des fichiers binaires, puis reconstruire l'hologramme.

Les tests suivants ont été fait sur un PC équipé d'un processeur Intel Xeon e5-2620 v3 2.40GHz, 32 Go de RAM et une carte graphique NVIDIA GeForce GTX 1070.

	C++	CUDA
Temps d'exécution du Matching Pursuit pour 50% des atomes (min)	87	10

Tableau 2 : Temps d'exécution du Matching Pursuit pour 50% des atomes

En choisissant de ne garder que 50% des atomes les plus corrélés, le Matching Pursuit met environ dix minutes à s'exécuter, soit presque neuf fois plus rapide que le même algorithme en C++ seulement.

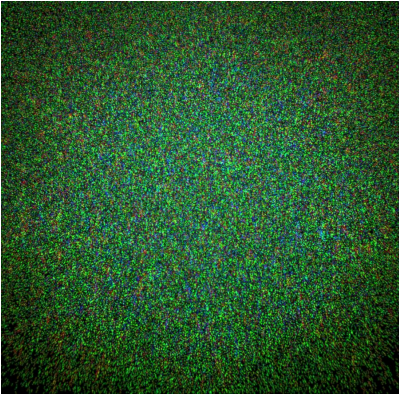
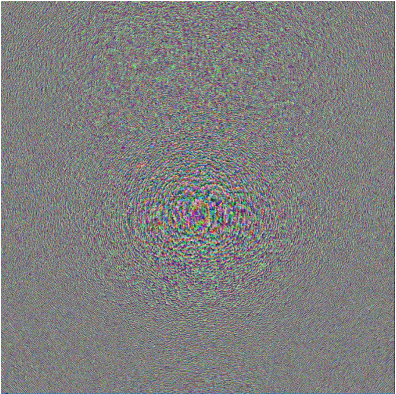
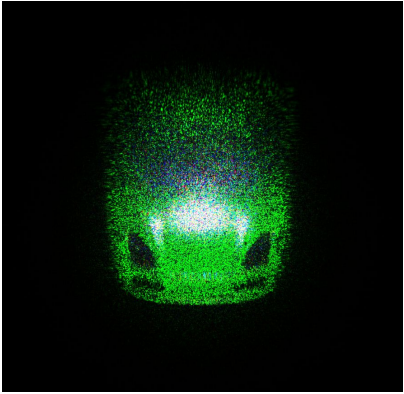
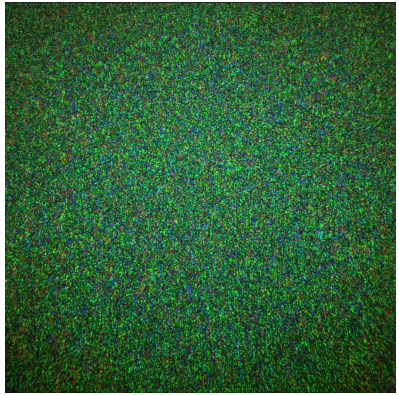
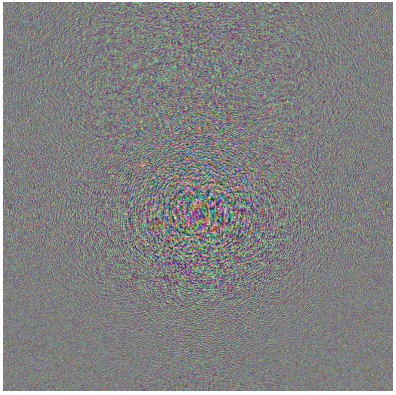
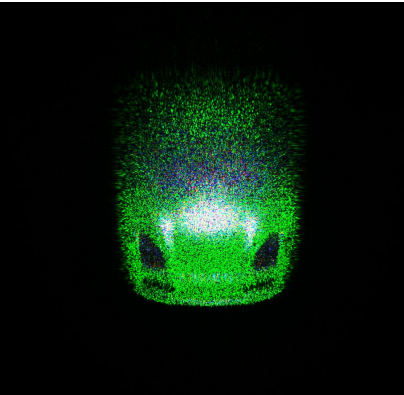
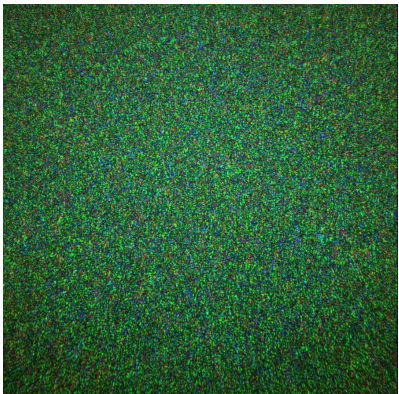
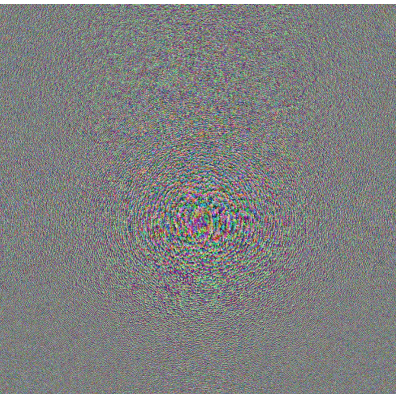
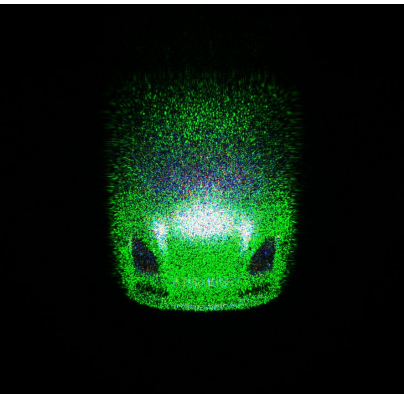
	Amplitude	Phase	Reconstruction
10% des atomes			
50% des atomes			
100% des atomes			

Tableau 3 : Amplitude, phase et reconstruction de l'hologramme en fonction du nombre d'atomes

Nom du fichier binaire	Taille (ko)		
	10 % des atomes	50 % des atomes	100 % des atomes
coeffRReal.bin	94	472	943
coeffRImag.bin	96	480	960
coeffGReal.bin	95	479	960
coeffGImag.bin	96	480	960
coeffBReal.bin	98	492	983
coeffBImag.bin	96	480	960
rotationIndices.bin	47	237	477
scaleIndices.bin	17	57	85
coords1D.bin	67	181	259
Temps d'écriture total	360 ms	1621 ms	3257 ms
Temps de lecture total	198 ms	958 ms	1935 ms

Tableau 4 : Taille des fichiers binaires et temps d'écriture/lecture en fonction du nombre d'atomes

Pourcentage d'atomes sélectionnés	Temps de reconstruction par le kernel (seconde)
10%	0,15
30%	0,5
50%	1,1

Tableau 5 : Temps de reconstruction de l'hologramme par le kernel en fonction du nombre d'atomes

Comme on peut le remarquer dans le *Tableau 3*, 10% des atomes suffisent pour avoir une bonne représentation de la scène. Cela est dû au fait que, grâce au Matching Pursuit, seuls les atomes les plus représentatifs de l'hologramme ont été gardés.

Dans le *Tableau 4*, on peut voir que sélectionner 10% des atomes permet d'obtenir des tailles de fichier et des vitesses d'écriture/lecture bien plus faibles (10 fois inférieures) qu'en prenant tous les atomes. On remarque aussi dans le *Tableau 5* que 10% des atomes permet de reconstruire l'hologramme environ 10 fois plus vite qu'en prenant 50%. L'écriture pouvant se faire *offline*, la vitesse de lecture est importante pour permettre un affichage temps-réel. Le fait d'utiliser seulement 10% des atomes est donc un bon compromis entre taille des fichiers, temps d'écriture, de lecture et qualité de reconstruction.

Conclusion

Lors de ce stage ingénieur au sein des équipes de b<>com, j'ai pu travailler sur une technologie prometteuse qui m'a intéressée dans son ensemble. Durant ces quatre mois, j'ai donc eu la chance de participer au développement d'une solution de transmission et de compression de vidéos holographiques. Ce sujet m'a permis de mettre en application de nombreuses connaissances et compétences que j'ai pu acquérir, notamment dans ma spécialité Imagerie Numérique de ma deuxième année du cycle ingénieur à l'ESIR. Il m'a également permis de solidifier certaines de ces compétences, comme le C++ qui est le langage de programmation que j'ai principalement utilisé, et d'en découvrir d'autres, comme la technologie CUDA dont j'ai pu en appréhender les bases.

Ce stage était aussi ma première expérience professionnelle dans une entreprise du numérique et sur un sujet de développement. Le fait d'être focalisé à plein temps sur des missions bien précises d'un même sujet était donc une nouvelle expérience qui m'a permis de mettre en place des méthodes de travail, d'organisation et d'appliquer une certaine rigueur pour que mes missions soient accomplies au mieux.

Bien que l'implémentation des méthodes de transmission d'hologramme ne soit pas un sujet de recherche en soit, intégrer un Institut de Recherche Technologique m'a fait découvrir le monde de la recherche, que ce soit en discutant avec les chercheurs et doctorants du laboratoire ou à travers leurs articles de recherche. C'est donc un domaine qui attire toujours mon attention et que je vais continuer de découvrir lors de la dernière année de mon cursus d'ingénieur en faisant un contrat de professionnalisation dans un laboratoire de recherche et développement.

Bibliographie

Article scientifique et thèse

Anas El Rhammad, Patrick Gioia, Antonin Gilles, Marco Cagnazzo, and Béatrice Pesquet-Popescu, "*Color digital hologram compression based on matching pursuit*," Applied Optics, vol. 57, no. 17, pp. 4930–4942, 2018.

Antonin Gilles, "*Fast hologram synthesis methods for realistic 3D visualization*". Thèse de doctorat : Traitement du Signal et de l'Image. Rennes: Institut National des Sciences Appliquées, 2016.

Sites internet

<https://b-com.com>

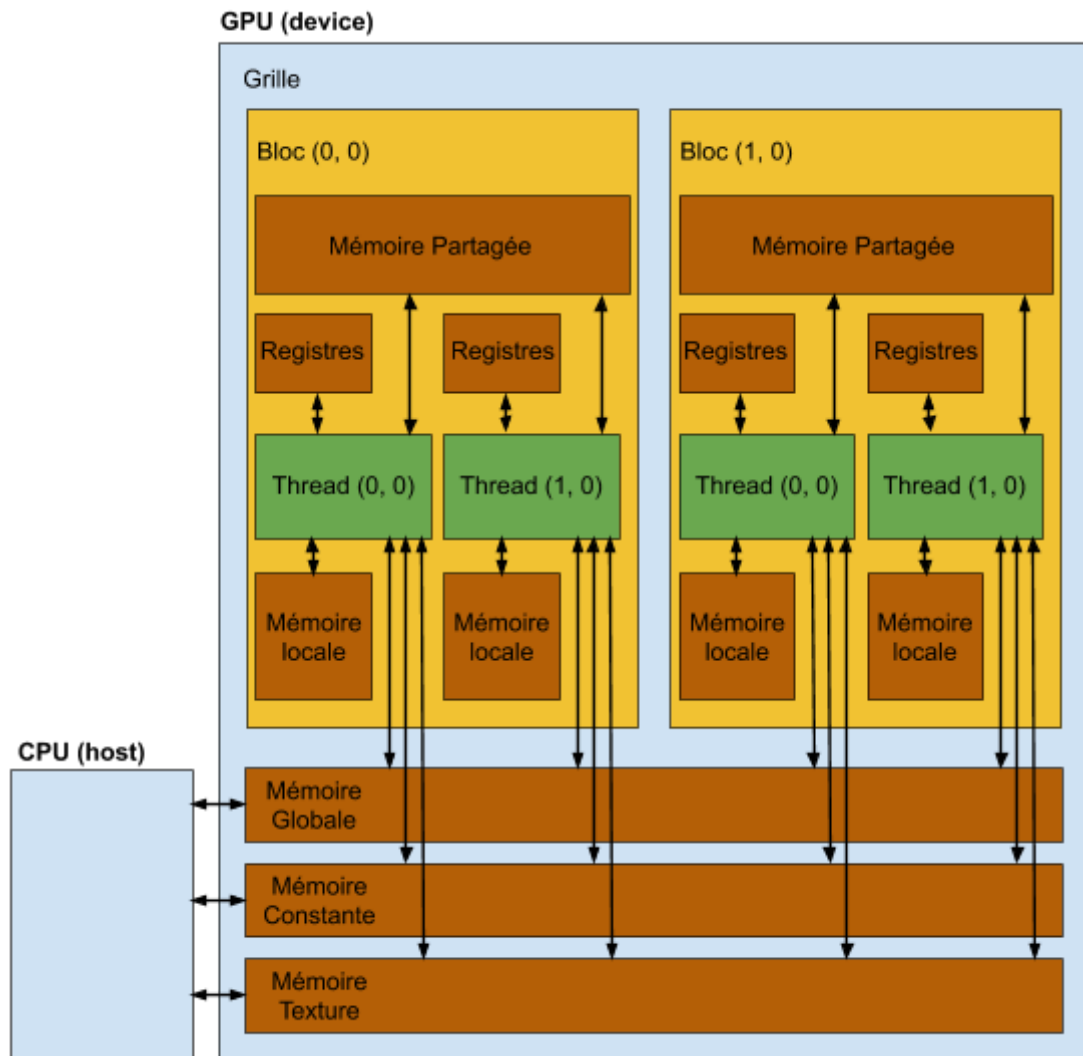
<https://developer.nvidia.com/cuda-zone>

https://en.wikipedia.org/wiki/Matching_pursuit

<https://fr.wikipedia.org/wiki/Holographie>

Annexes

Annexe 1 : Architecture CUDA



Annexe 2 : Vectorial Matching Pursuit

Input : Hologram \mathbf{H} , Dictionary D
Output: Coefficients and atom indices $\{(c_k^{(i)}, \gamma_k)\}$
for $i \in \{r, g, b\}$ **do**
 | Initialization: $R_0(H^{(i)}) = H^{(i)}$
end
for $k \in \{1, \dots, K\}$ **do**
 $\gamma_k = \arg \max_{\gamma \in \Gamma} \|\langle R_k(\mathbf{H}), g_\gamma \rangle\|_p$
 for $i \in \{r, g, b\}$ **do**
 | $c_k^{(i)} = \langle R_k(H^{(i)}), g_{\gamma_k} \rangle$
 | Update: $R_{k+1}(H^{(i)}) = R_k(H^{(i)}) - c_k^{(i)} g_{\gamma_k}$
 end
end

Pour accélérer les calculs, la ligne *Update* peut être remplacée par :

$$\langle R_{k+1}H^{(i)}, g_{\gamma^{(i)}} \rangle = \langle R_kH^{(i)}, g_{\gamma^{(i)}} \rangle - \langle R_kH^{(i)}, g_{\gamma_k} \rangle \langle g_{\gamma_k}, g_{\gamma^{(i)}} \rangle$$