**Impact des ROI dans le streaming vidéo**

Encadrant : **Olivier FOURMAUX**

Etudiants : **Sonia LOUNIS, Fabien MANSON et Alexandre MAZARS**

Table des matières

[1 Cahier des charges 2](#_Toc40890949)

[1.1 Présentation du projet 2](#_Toc40890950)

[1.1 Maintenant et évolution du projet 2](#_Toc40890951)

[2 Plan de développement 3](#_Toc40890952)

[3 Bibliographie 3](#_Toc40890953)

[4 Analyse 3](#_Toc40890954)

[5 Conception 3](#_Toc40890955)

[6 Compte rendu 3](#_Toc40890956)

# Cahier des charges

# 1.1 Présentation du projet

Ce projet est réalisé dans le cadre de l’UE PRES de l’année 2019, Il est proposé par Mr. Olivier FOURMAUX et porte sur l’étude de l’impact du streaming vidéo adaptatif et la mise en place d’une démonstration de cette technologie.

**1.2 Objectif du projet**

Dans un premier temps nous allons étudier le fonctionnement de l’encodage différencié des régions d’intérêts ainsi que la transmission vidéo via http. Dans un second temps nous mettrons en place un système de codage et décodage adaptatif spatial. Enfin nous devrons présenter une démonstration de l’encodage différencié au travers d’un casque de réalité virtuelle ou via une émulation sur un écran et de son impact sur le réseau.

**1.3 Contexte du projet**

Ce projet s’inscrit dans la dynamique actuelle de la diffusion massive de contenu vidéo haute résolution qui ne cesse d’augmenter. Le développement de la réalité virtuelle et du contenu vidéo en VR représente une partie du futur de la diffusion de contenu vidéo. On peut trouver quelques études qui ont déjà travaillé sur ce sujet en proposant des méthodes d’optimisation de la transmission de vidéo UHD et du découpage de celle-ci en régions d’intérêt.

**1.4 Contraintes liées au projet**

Contrainte de temps le projet est à rendre en mai 2020 en prévision d’une soutenance de ce dernier un peu plus tard le mois. Les dates ne sont pas encore précisées. Les dates ont été changées à cause de l’épidémie du Covid-19, le rapport est à rendre le 10 juin 2020 et la soutenance aura lieu le 15 ou le 16 juin 2020 probablement en visio-conférence.

**1.5 Documentation du projet**

La documentation de la partie technique du projet sera minimaliste. Elle pourra se trouver sous l’une des deux formes suivantes : directement dans le code du projet sous la forme de commentaire de code très détaillé ou dans un fichier texte annexe (on peut même envisager un fichier PDF si le projet ne prend pas trop de retard).

## Maintenant et évolution du projet

Le projet n’aura plus aucun suivi passé la date de soutenance. Il ne sera plus mis à jours ni même modifié. Cependant ou pourra retrouver l’intégralité du code source sur une page GitHub dédiée.

**- cahier des charges inchangé, c'est le même que celui du rapport 1**

**- plan du développement adapté (mis à jour) + gantt–2–Left.png + gantt–2-Right.png**

**- la bibliographie actualisée [1000-1500 mots] (dans notre cas c'est inutile car les articles que l'on a trouvés sont tous importants)**

**- l'analyse actualisée [1000-1500 mots]**

**- la conception réalisée [1000-1500 mots]**

**- un compte rendu de projet [1500-2500 mots] (détail du déroulement, de la réalisation, la validation et la livraison de ce qui est demandé)**

**- annexes (useless dans notre cas je pense)**

# Plan de développement

# Bibliographie

La bibliographie reste quasiment la même. Nous avons simplement rajouté quelques sites web tels que le site de GPAC aisni que la page wikipedia officielle de GPAC.

1] Tarek El-Ganainy and Mohamed Hefeeda. 2016. Streaming Virtual Reality Content. *arXiv:1612.08350 [cs]* (December 2016). Retrieved December 5, 2019

[2] Dan Grois and Ofer Hadar. 2011. Recent Advances in Region-of-interest Video Coding. In *Recent Advances on Video Coding*. DOI:https://doi.org/10.5772/17789

[3] Mohammad Hosseini and Viswanathan Swaminathan. 2016. Adaptive 360 VR Video Streaming: Divide and Conquer! (September 2016). Retrieved December 5, 2019 from <https://arxiv-org.accesdistant.sorbonne-universite.fr/abs/1609.08729v5>

[4] JongBeom Jeong, Dongmin Jang, Jangwoo Son, and Eun-Seok Ryu. 2018. 3DoF+ 360 Video Location-Based Asymmetric Down-Sampling for View Synthesis to Immersive VR Video Streaming. *Sensors* 18, 9 (September 2018), 3148. DOI:https://doi.org/10.3390/s18093148

[5] Christopher Müller and Christian Timmerer. 2011. A VLC Media Player Plugin Enabling Dynamic Adaptive Streaming over HTTP. In *Proceedings of the 19th ACM International Conference on Multimedia* (MM ’11), 723–726. DOI:https://doi.org/10.1145/2072298.2072429

[6] D. V. Nguyen, Huyen T. T. Tran, and Truong Cong Thang. 2019. A client-based adaptation framework for 360-degree video streaming. *Journal of Visual Communication and Image Representation* 59, (February 2019), 231–243. DOI:https://doi.org/10.1016/j.jvcir.2019.01.012

[7] Duc V. Nguyen, Huyen T. T. Tran, and Truong Cong Thang. 2019. Adaptive Tiling Selection for Viewport Adaptive Streaming of 360-degree Video. *IEICE Transactions on Information and Systems* E102.D, 1 (2019), 48–51. DOI:https://doi.org/10.1587/transinf.2018MUL0001

[8] Cagri Ozcinar, Ana De Abreu, and Aljosa Smolic. 2017. Viewport-aware adaptive 360° video streaming using tiles for virtual reality. *arXiv:1711.02386 [cs]* (November 2017). Retrieved December 5, 2019 from <http://arxiv.org/abs/1711.02386>

[9] Georgios Papaioannou and Iordanis Koutsopoulos. 2019. Tile-based Caching Optimization for 360° Videos. *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing* 171--180 (July 2019). Retrieved December 5, 2019 from <http://graphics.cs.aueb.gr/graphics/docs/papers/MOBIHOC-2019.pdf>

[10] Y. Sánchez de la Fuente, R. Skupin, and T. Schierl. 2017. Video processing for panoramic streaming using HEVC and its scalable extensions. *Multimed Tools Appl* 76, 4 (February 2017), 5631–5659. DOI:https://doi.org/10.1007/s11042-016-4097-4

[11] Thomas Stockhammer. 2011. Dynamic Adaptive Streaming over HTTP –: Standards and Design Principles. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems* (MMSys ’11), 133–144. DOI:https://doi.org/10.1145/1943552.1943572

[12] Alireza Zare, Alireza Aminlou, Miska M. Hannuksela, and Moncef Gabbouj. 2016. HEVC-compliant Tile-based Streaming of Panoramic Video for Virtual Reality Applications. In *Proceedings of the 24th ACM International Conference on Multimedia* (MM ’16), 601–605. DOI:https://doi.org/10.1145/2964284.2967292

[13] Alireza Zare, Maryam Homayouni, Alireza Aminlou, Miska M. Hannuksela, and Moncef Gabbouj. 2019. 6K and 8K Effective Resolution with 4K HEVC Decoding Capability for 360 Video Streaming. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 2s (July 2019), 68:1–68:22. DOI:https://doi.org/10.1145/3335053

[14] GPAC Multimedia Open Source Project, Telecom Paris, [*https://gpac.wp.imt.fr*](https://gpac.wp.imt.fr)*,*  (dernière mise à jour en Octobre 2019), site consulté entre Mars et Juin 2020

[15] GPAC Project on Advanced Content, Wikipedia The Free Encyclopedia, [*https://en.wikipedia.org/wiki/GPAC\_Project\_on\_Advanced\_Content*](https://en.wikipedia.org/wiki/GPAC_Project_on_Advanced_Content), page créée en 2008, dernière mise à jour de la page en mars 2020, site consulté entre Mars et Juin 2020

[16] GPAC Code source, Github.com, [*https://github.com/gpac/gpac*](https://github.com/gpac/gpac), Dernière mise à jour du code source le 3 Juin 2020, site consulté régulièrement entre Mars et Juin 2020

# Analyse [1000 – 1500 mots]

Le concept de région d’intérêt a été mis en place dans le but de réduire la charge réseau tout en conservant un maximum de qualité vidéo. Dans ce projet on s’intéresse particulièrement à son impact sur les vidéos en 360 degré car c’est là où elle a le plus d’impact. En effet, lorsqu’on visionne une vidéo en 360 degré, notre champ de vision est réduit à une petite partie de la vidéo car cela évite de voir l’effet de déformation lié à la projection cylindrique équidistante (type de projection le plus courant pour passer d’une sphère a un plan). Sans utiliser les régions d’intérêt, toute la partie de la vidéo qui se trouve en dehors du champ de vision est quand même téléchargée depuis le serveur vers le client ce qui pose en problème de bande passante. La région d’intérêt permet de définir une région particulière (généralement la zone que regarde l’utilisateur) qui sera prioritaire par rapport aux autres zones. Un débit élevé et donc une meilleur qualité d’image sera attribuée aux régions ayant une forte priorité. Le reste des régions moins prioritaires auront un débit attribué plus faible permettant de réduire grandement la charge de la bande passante. Ce principe de région d’intérêt est très important car il ne faut pas oublier qu’une vidéo en 360° est au minimum en 4k 30ips (ultra haute définition). La plupart du temps les vidéos ont une résolution bien supérieur à de la simple 4K, on trouve notamment de la 6K et même dans certains cas de la 8K (images de respectivement 6000 et 8000 pixels de large).

La méthodologie pour mettre en place un système de stream de vidéo utilisant le principe de région d’intérêt est toujours la même. Dans un premier temps, il faut trouver un média source. De préférence libre de droit est ayant une résolution d’image au moins égale à de la 4K avec une fréquence d’image de 30ips (images par secondes) au minimum. On peut Notamment utiliser le site Mettle.com qui propose quelques vidéos d’une trentaine de secondes en VR filmées depuis un drone ou une caméra statique. Une fois que l’on a téléchargé le média source, il faut appliquer un découpage en tuiles selon une grille de taille choisie, généralement une grille carrée de 5x5 tuiles. Ce découpage en tuiles peut se faire avec des encodeurs comme ffmpeg et kvazaar. Il faut aussi produire des clones du média source avec une qualité d’image volontairement dégradée. On prend par exemple la vidéo source en pleine résolution, et on crée d’autres clones, ayant chacun des résolutions de plus en plus faibles. Par exemple avec un média source ayant 4096p (pixels) de large, on peut créer des clones ayant les résolutions suivantes 2560p, 1920p, 720p, 360p. Une fois le découpage effectué, on obtient pour chaque tuile plusieurs segments de différentes résolutions. Il faut ensuite créer un fichier descripteur de flux mpeg-dash (.mpd) qui permettra de localiser les différentes tuiles et les différentes résolutions disponibles pour chaque tuile. Ensuite il ne reste plus qu’à trouver un lecteur qui permette d’ouvrir un flux mpeg-dash ce qui n’est pas chose aisée car la plupart des lecteurs sont des technologies d’entreprises privées comme YouTube ou Facebook. Il existe cependant un projet open source de lecteur dash en javascript et GPAC qui est un lecteur multimédia capable de lire ce type de flux. Le rôle du lecteur n’est pas si simple que cela car il doit non seulement afficher la vidéo de manière fluide (ce qui n’est pas trivial avec des très hautes résolutions) mais aussi télécharger les bonnes tuiles correspondant à la bande passante disponible.

Si on connaît la région d’intérêt de l’utilisateur, c’est-à-dire l’endroit où il regarde et que l’on arrive à prédire ces différents mouvements, on peut minimiser la qualité vidéo de toute la partie qu’il ne regarde pas et maximiser celle dont il regarde, ce qui permet de maximiser l’utilisation de la bande passante.

Pour les ROI, il existe de nombreux type d’encodage qui fonctionne chacun plus ou moins différemment.

La plupart des encodeurs vidéo cherchent à maximiser l’utilisation de la bande passante donc ils n’utilisent pas que le concept de ROI, mais aussi celui découpage en tuile. La vidéo est coupée en plusieurs carreaux, l’encodeur génère plusieurs fois la vidéo découpée avec des qualités vidéo différentes. Lors du visionnage de la vidéo, le décodeur va faire un mélange entre les carreaux de mauvaise qualités et bonne qualités pour à la fin reformer la vidéo suivant les besoin de l’utilisateur.

Comme pour les ROI, il existe de nombreuse technique de découpage.

A la fin le décodeur fait un mixe entre découpage en carreaux et les ROI pour que la région d’intérêt correspondent aux carreaux de bonne qualités et celle en dehors aux carreaux de mauvaise qualité

# Conception [1000 – 1500 mots]

# Compte rendu de projet [1500 – 2000 mots]